Structural Chunking: A Semantic–Structural Integrated Method for Retrieval-Augmented Generation

Sangyong Lee*
AI Research Center
OKESTRO Co., Ltd.
Seoul, Republic of Korea
sy.lee@okestro.com

NaHun Kim
AI Research Center
OKESTRO Co., Ltd.
Seoul, Republic of Korea
nh.kim2@okestro.com

Junseok Lee
AI Research Center
OKESTRO Co., Ltd.
Seoul, Republic of Korea
js.lee6@okestro.com

***Abstract*—Retrieval-Augmented Generation (RAG) systems rely heavily on the quality of document chunking, which determines the granularity and contextual continuity of retrievable units. Existing chunking methods inevitably trade off between semantic coherence and computational efficiency. To overcome this limitation, we propose Structural Chunking, a novel paradigm that integrates semantic cohesion with structural consistency by leveraging both surface- and physical-level document features. Unlike purely semantic methods, Structural Chunking quantifies structural patterns and fuses them with semantic embeddings to compute a semantic–structural cohesion score. Chunk boundaries are then detected where this composite distance sharply increases. Experiments on five BEIR benchmark datasets demonstrate that Structural Chunking achieves the most consistent recall and average precision across domains, outperforming semantic and statistical baselines while maintaining stable chunk-size distributions. The results indicate that incorporating structural hierarchy into the RAG preprocessing pipeline substantially enhances retrieval efficiency and contextual fidelity, suggesting a new direction for structure-aware information retrieval.**

***Index Terms*—Retrieval-Augmented Generation (RAG), Structural Chunking, Semantic Cohesion, Structural Consistency**

I. Introduction

Retrieval-Augmented Generation (RAG) systems enable large language models (LLMs) to retrieve and utilize external documents, thereby generating more accurate and contextually grounded responses. Among the key preprocessing steps in this pipeline, chunking—dividing long documents into manageable units—plays a crucial role in balancing retrieval efficiency and contextual preservation. The quality of chunking directly determines RAG performance [1]. Poorly designed chunking may fragment context and distort semantics, while effective chunking enhances query-document alignment and coherence in generated responses. Existing chunking methods are generally categorized as fixed-size, language-unit-based, semantic, clustering, agentic, and hybrid. Each method exhibits domain-dependent strengths and weaknesses, but no single approach handles all document types effectively.

Fixed-size chunking divides text by token or character count, offering computational simplicity and consistent indexing [2]. It is effective in short-answer domains such as FAQs (Frequently Asked Questions) or news summaries but fails to preserve logical continuity in structured texts like legal or technical documents. Language-based chunking follows grammatical boundaries to maintain linguistic completeness, performing well in discourse-rich data such as interviews or educational materials. However, it produces uneven chunk lengths, which complicates tokenization and scaling. Semantic chunking relies on embedding similarity to segment semantically coherent regions [2], [3]. While effective for topic-centric corpora, it incurs high computational cost and ignores physical layout cues such as paragraph breaks or section headers, often resulting in inaccurate segmentation. Clustering-based chunking groups non-contiguous sentences by global embedding similarity [4]. It enhances topic diversity but disrupts positional continuity, making it unsuitable for documents with hierarchical order. Agentic chunking employs LLM reasoning to infer semantic boundaries autonomously [5], [6], providing adaptability to unstructured text but with prohibitive time and cost overhead. Finally, hybrid chunking integrates multiple signals—such as semantic embeddings and sliding windows—to improve continuity [7]. These methods enhance recall but require domain-specific tuning and add system complexity.

Overall, current chunking strategies face a trade-off between semantic coherence and computational efficiency [8]–[10]. Fixed-size chunking suffices for simple retrieval tasks, yet structured domains—legal, policy, or technical—demand awareness of document hierarchy and formatting cues. Thus, an approach that unifies structural hierarchy with contextual cohesion is needed.

To address this limitation, we propose **Structural Chunking**, a method integrating both semantic and structural information. Unlike prior approaches based solely on embedding similarity, Structural Chunking explicitly models document hierarchy (titles, sections, paragraphs), *semantic cohesion*, and *positional continuity*. This unified, structure-aware segmentation strategy aims to preserve both logical flow and contextual integrity across diverse document domains.

* Corresponding author

II. Related Works

Semantic and agentic chunking approaches determine segmentation boundaries primarily based on inter-sentence semantic cohesion. While this improves contextual consistency, such methods overlook the inherent structural composition of real-world documents. In practice, documents are not mere semantic sequences but complex systems combining both surface structures (e.g., sentence length, spacing, punctuation, line breaks) and physical structures (e.g., paragraphs, hierarchical titles, sections, or clauses). For instance, legal, policy, research, and technical documents all follow hierarchical layouts such as "Article–Clause–Subclause" or "Chapter–Section–Paragraph," which serve as essential devices for maintaining logical flow. Hence, a purely semantic approach fails to fully represent such formal organization.

Existing chunking methods that neglect structural context often produce misaligned paragraph boundaries, fragmented logical units, or redundant information within the same topic. Consequently, retrieval efficiency deteriorates, contextual reconstruction cost increases, and LLM responses become unstable. Recent studies [11]–[13] have incorporated structural knowledge such as grammatical boundaries, parse trees, or Abstract Syntax Trees (AST) into pre-training to enhance model understanding. However, parser- or OCR-based approaches (e.g., LayoutLM) are computationally expensive and difficult to apply to multilingual or unstructured text.

Therefore, this study redefines structure-aware chunking in a lightweight, language-agnostic manner by directly computing feature-level structural patterns instead of relying on syntactic parsing. Specifically, we extract and normalize two types of features: (1) **Surface structure**: sentence length, punctuation distribution, and spacing patterns. (2) **Physical structure**: paragraph count, line breaks, capitalization, and inter-paragraph spacing. These structural features can be efficiently combined with semantic embeddings without a parser, allowing quantitative fusion of *semantic cohesion* and *structural consistency*.

As illustrated in Fig. 1, even when two sentences convey the same meaning, differences in their physical or typographical layout—such as line breaks, spacing, or indentation—can significantly affect embedding-based similarity computations. This subtle variation reflects how structural cues contribute to perceived coherence beyond lexical meaning. Hence, purely semantic embeddings fail to capture the influence of structural form on contextual continuity and local discourse flow.

To address this, we propose **Structural Chunking**, a method that integrates hierarchical document formats and structural indicators into the chunking process. The approach preserves external boundaries (e.g., paragraphs and sections) while maintaining semantic continuity, ensuring both contextual and formal consistency. By incorporating these features, Structural Chunking can distinguish genuine contextual transitions from superficial formatting variations, leading to more coherent and structurally consistent chunk boundaries.

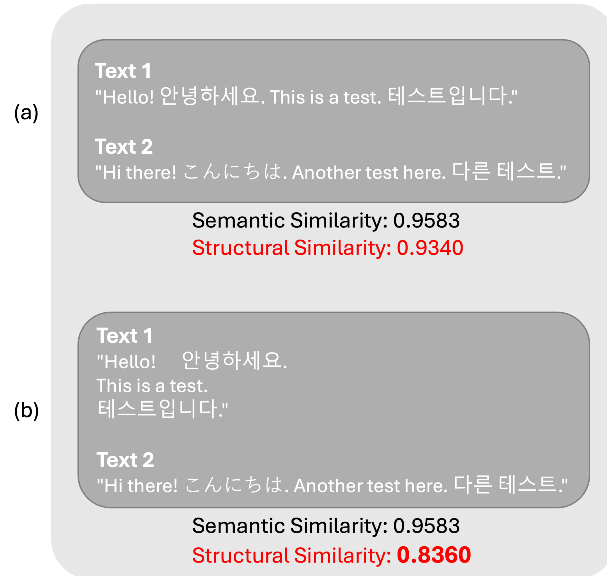

Fig. 1. Effect of structural variation on semantic similarity. Despite identical content, structural differences (e.g., line breaks, spacing) lead to reduced structural similarity (0.9340→0.8360) while semantic similarity remains constant (0.9583).

III. Proposed Method

This study introduces **Structural Chunking**, a method that integrates surface- and physical-level structural features with semantic embeddings to enhance document segmentation. Unlike conventional semantic-based approaches that detect contextual transitions but ignore structural cues such as hierarchy, punctuation, or spacing, the proposed method computes language-agnostic structural indicators and derives a semantic–structural cohesion score to identify high-variation boundaries.

As shown in Fig. 2, the input document is processed by both semantic and structural encoders. The semantic encoder captures contextual coherence, while the structural encoder models hierarchical and typographical patterns. Their similarity scores are fused through a weighted combination to yield the final semantic–structural cohesion score, which determines chunk boundaries for segmentation.

A. Structural Feature Extraction

Without relying on parsers or grammar models, Structural Chunking quantifies two categories of features:

- **Surface structure:** average sentence length, punctuation distribution and density, word length distribution, token-level patterns (case, digits, symbols), and average word length.
- **Physical structure:** paragraph count and variance, spacing and line-break patterns, character-type ratios, and length distributions (mean, variance, std).

Each feature is normalized and vectorized. Structural similarity between consecutive sentences is measured via cosine similarity.

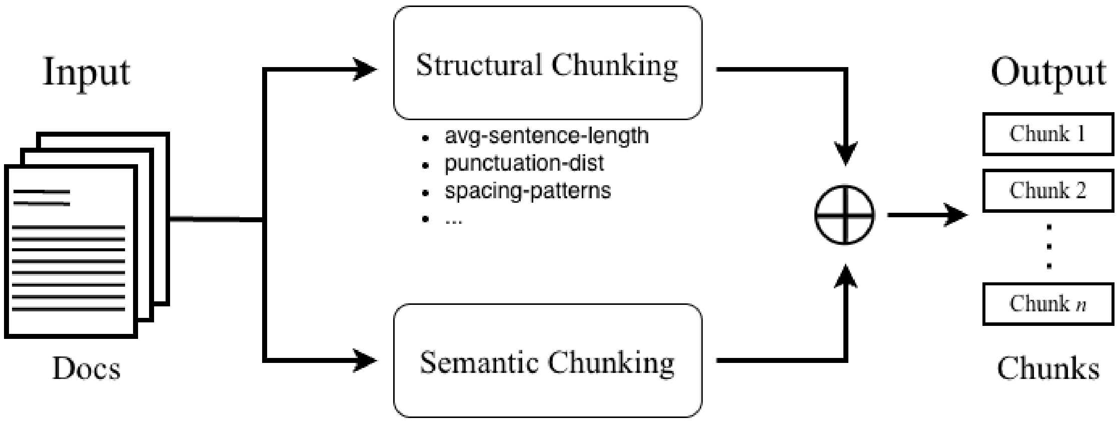

Fig. 2. Overall architecture of the proposed Structural Chunking method.

B. Structural Consistency Computation

To model the consistency of document formatting, **Structural Chunking** quantifies how adjacent sentences share similar surface and physical characteristics. For each consecutive index pair $(i, i+1)$, structural similarity is computed as the weighted sum of feature-level cosine similarities between the corresponding structural feature vectors $\mathbf{f}_i$ and $\mathbf{f}_{i+1}$. $\mathbf{f}_i = [f_i^1, \ldots, f_i^K]$ represents the structural feature vector of sentence i, and each f_i^k denotes its k-th feature value. The overall structural similarity S_{stru} is then derived as:

$$S_{\text{stru}}(i, i+1) = \sum_k w_k \cdot \text{sim}(f_i^k, f_{i+1}^k), \tag{1}$$

where w_k is a predefined weight assigned to the k-th feature dimension. A corresponding structural distance D_{stru} is then derived to represent the degree of discontinuity between consecutive sentences.

$$D_{\text{stru}}(i, i+1) = 1 - S_{\text{stru}}(i, i+1), \tag{2}$$

A higher D_{stru} indicates abrupt structural transitions, such as paragraph breaks or formatting shifts, which formatting coincide with natural chunk boundaries.

C. Semantic Cohesion Scoring

In parallel, semantic cohesion is measured by computing the cosine distance between consecutive sentence embeddings. Each sentence index i is represented as an embedding vector $\mathbf{E}_i$, and semantic distance D_{sem} is defined as:

$$D_{\text{sem}}(i, i+1) = 1 - \text{sim}(\mathbf{E}_i, \mathbf{E}_{i+1}), \tag{3}$$

D. Fusion of Semantic and Structural Information

To jointly capture both semantic and structural discontinuities, a fused distance D_{fusion} integrates the two components through a weighted combination.

$$\begin{aligned} D_{\text{fusion}}(i, i+1) &= \alpha D_{\text{sem}}(i, i+1) \\ &\quad + (1-\alpha) D_{\text{stru}}(i, i+1). \end{aligned} \tag{4}$$

$\alpha = 0.05$, giving higher priority to structural discontinuities while maintaining semantic sensitivity. Chunk boundaries are assigned where D_{fusion} exceeds the 95th percentile threshold.

$$\text{Boundary}(i) = \begin{cases} 1, & \text{if } D_{\text{fusion}}(i, i+1) > P_{95}(D_{\text{fusion}}) \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

This percentile-based criterion adaptively detects significant segmentation points while filtering out minor fluctuations, yielding stable chunk boundaries across diverse document types. Algorithm 1 summarizes the proposed **Structural Chunking** process. Semantic embeddings and structural features are extracted for each sentence, inter-sentence distances are computed and fused by weighted combination, and major discontinuities are selected as chunk boundaries.

Algorithm 1 Structural Chunking

Require: Document T, semantic weight α
Ensure: Chunk list C
Split T into sentences $S = [s_1, \ldots, s_n]$
for each sentence index i **do**
  Compute embedding $\mathbf{E}_i \leftarrow \text{Embed}(s_i)$
  Extract structural features $\mathbf{f}_i \leftarrow [f_i^1, \ldots, f_i^K]$
for each consecutive pair $(i, i{+}1)$ **do**
  $D_{\text{sem}}(i, i{+}1) \leftarrow 1 - \text{sim}(\mathbf{E}_i, \mathbf{E}_{i+1})$
  $D_{\text{stru}}(i, i{+}1) \leftarrow 1 - \sum_k w_k \cdot \text{sim}(f_i^k, f_{i+1}^k)$
  $D_{\text{fusion}}(i, i{+}1) \leftarrow \alpha\, D_{\text{sem}}(i, i{+}1) + (1 - \alpha)\, D_{\text{stru}}(i, i{+}1)$
$\tau \leftarrow P_{95}\left(\{D_{\text{fusion}}(i, i{+}1)\}_{i=1}^{n-1}\right)$
Mark boundaries where $D_{\text{fusion}}(i, i{+}1) > \tau$
Merge sentences between boundaries to form C
return C

The Structural feature extractor scans the text once and therefore runs in linear time $O(n)$, with only constant-time updates at sentence boundaries.

TABLE I
RETRIEVAL PERFORMANCE COMPARISON ACROSS FIVE BEIR DATASETS

Method	NFCorpus		SciFact		ArguAna		SCIDOCS		FiQA	
	Recall	**AP**	**Recall**	**AP**	**Recall**	**AP**	**Recall**	**AP**	**Recall**	**AP**
Percentile	0.5975	0.3028	0.7000	0.4257	**0.4804**	0.1278	0.5810	0.2866	0.4151	0.2265
StdDeviation	0.4211	0.1889	0.5033	0.3803	0.3512	0.1099	0.5980	0.2926	0.3488	0.2118
Interquartile	0.6037	0.2929	0.6267	0.4113	0.3983	0.1174	0.5830	0.2846	0.4059	0.2316
Fixedlen	0.5418	0.2801	0.5467	0.2896	0.0107	0.0026	0.1990	0.0753	0.3410	0.1936
Gradient	0.6130	0.3393	**0.7067**	0.4457	0.4490	0.1233	0.5880	0.2903	**0.4583**	**0.2481**
Structural	**0.6471**	**0.3677**	0.7033	**0.4562**	0.4611	**0.1308**	**0.5990**	**0.3005**	0.4336	0.2344

IV. EXPERIMENTAL RESULTS

A. Datasets

Experiments were conducted on five representative Benchmarking Information Retrieval (BEIR) datasets [14]—NFCorpus [15], SciFact [16], ArguAna [17], SCIDOCS [18], and FiQA [19]—to evaluate the generalization of chunking strategies across diverse document structures and query types.

NFCorpus [15] contains medical web documents and FAQs with descriptive and advisory responses, where contextual flow across paragraphs is crucial. **SciFact** [16] focuses on scientific abstracts for claim verification tasks, where splitting evidence sentences degrades precision. **ArguAna** [17] consists of argument–counterargument pairs from online debates, requiring the preservation of logical discourse structure. **SCIDOCS** [18] models inter-paper relations in scientific literature, favoring chunkers that retain section and title hierarchies. **FiQA-2018** [19] includes finance-related Q&A and analyses, characterized by stylistic noise and weak structural cues.

B. Evaluation Metrics

Two standard retrieval metrics were used: Recall and Average Precision (AP). Recall measures the ratio of retrieved relevant documents, while AP reflects ranking quality by considering both precision and retrieval order. Scores were computed per query and averaged across all queries.

C. Experimental Setup

We compared six chunking methods under identical BEIR settings: (1) Percentile Chunking, which partitions text based on percentile thresholds of length distribution; (2) Standard Deviation (StdDeviation) Chunking, using statistical dispersion to determine boundary points; (3) Interquartile Chunking, which segments text according to the interquartile range of sentence lengths; (4) Fixed-Length (Fixedlen) Chunking, dividing text by a uniform token window; (5) Gradient-Based Chunking, detecting boundaries at steep semantic gradient changes; and (6) the proposed Structural Chunking, integrating both semantic and structural cohesion.

For each method, raw texts were segmented, embedded, and indexed using id entical tokenization and embedding parameters. Retrieval was performed via cosine similarity between query and chunk embeddings, selecting the top-K ($K = 10$) candidates. Ground-truth labels from BEIR were used to compute mean Recall and Average Precision (AP) across all queries.

TABLE II
STRUCTURAL FEATURES USED IN THE STRUCTURAL CHUNKING

Feature	**Description**	**Type**
avg_sentence_length	Average number of words per sentence, reflecting text density and complexity.	Surface
punctuation_distribution	Distribution of punctuation marks (.,!?;:) indicating rhythm and boundary patterns.	Surface
word_length_distribution	Word length frequency capturing lexical diversity and stylistic variation.	Surface
paragraph_structure	Paragraph organization and mean length representing physical hierarchy.	Physical
character_level	Ratio of alphabets, numerals, and symbols showing character composition.	Surface
avg_word_length	Mean word length indicating lexical compactness and writing density.	Surface
punctuation_density	Ratio of punctuation to total text length measuring descriptive density.	Surface
token_level	Proportion of token types (word, number, symbol) reflecting syntactic patterns.	Surface
spacing_patterns	Distribution of spaces, tabs, and newlines quantifying spatial structure.	Physical
length_distributions	Sentence and paragraph length stats (mean, variance) assessing consistency.	Physical

Table II lists the structural features employed in Structural Chunking, which quantify surface- and physical-level regularities for computing inter-sentence structural similarity.

TABLE III
MEAN CHUNK SIZE AND TOTAL NUMBER OF CHUNKS ACROSS BEIR DATASETS

Method	**NFCorpus**		**SciFact**		**ArguAna**		**SCIDOCS**		**FiQA**	
	Mean	**Total**	**Mean**	**Total**	**Mean**	**Total**	**Mean**	**Total**	**Mean**	**Total**
Percentile	116.5	7,289	106.9	10,408	84.4	17,176	89.0	50,957	68.9	111,244
StdDeviation	221.7	3,831	212.0	5,248	165.4	8,769	174.4	26,007	131.2	58,418
Interquartile	133.1	6,380	140.0	7,946	115.6	12,544	124.0	36,567	99.1	77,385
Gradient	116.8	7,269	107.5	10,344	86.4	16,781	90.0	50,365	71.1	107,855
Structural	116.2	7,288	106.7	10,408	83.9	17,162	88.9	50,952	67.9	111,216

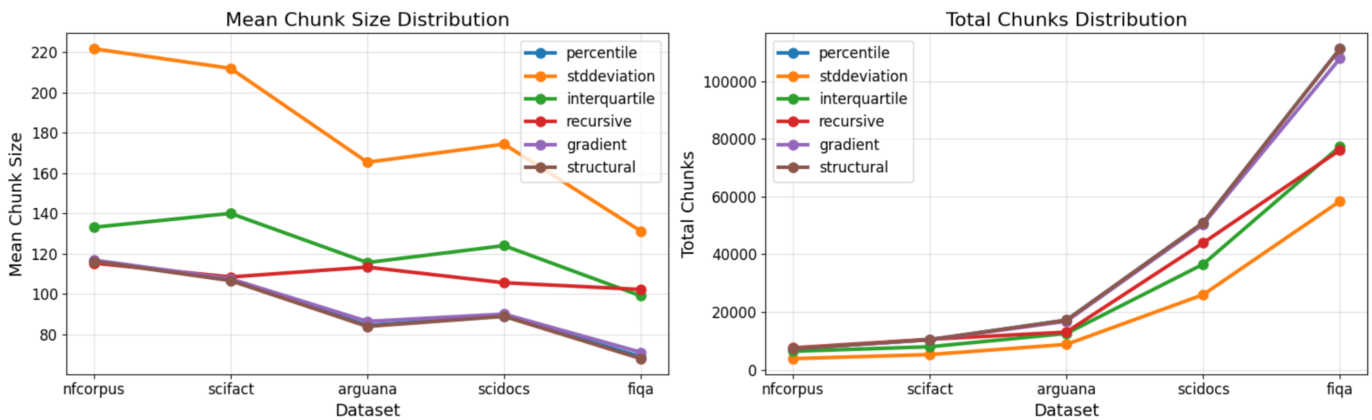

Fig. 3. Average chunk size and total chunk distribution across datasets

D. Results and Analysis

Table I presents the retrieval performance across five BEIR domains. The proposed Structural Chunking consistently achieved robust and high performance, recording the best Recall and AP on NFCorpus (Recall=0.6471, AP=0.3677), SciFact (Recall=0.7033, AP=0.4562), ArguAna (Recall=0.4611, AP=0.1308), and SCIDOCS (Recall=0.5980, AP=0.3005). These results highlight its ability to preserve both *structural consistency* and *contextual cohesion*, particularly in hierarchically organized texts such as scientific or legal documents.

Gradient Chunking yielded the highest Recall on SciFact (0.7067) and the best overall score on FiQA (Recall=0.4583, AP=0.2481), demonstrating superior responsiveness to subtle semantic transitions and style variations in less structured data. Meanwhile, Percentile Chunking showed localized advantages on ArguAna (Recall=0.4804), suggesting that uniform ratio-based segmentation can effectively preserve short argumentative or debate-style units.

Table III and Fig. 3 further compare the average chunk size and total number of chunks across datasets. StdDeviation Chunking produced the largest chunks and smallest counts, indicating coarse and sparse segmentation that often overlooks fine-grained contextual shifts. In contrast, both Gradient and Structural Chunking generated finer-grained and denser chunk distributions, capturing nuanced transitions between discourse units. Interquartile and Percentile Chunking yielded moderately balanced chunk sizes, while a recursive baseline (not shown) maintained relatively stable but length-dependent patterns.

Overall, the comparative analysis confirms that chunking effectiveness strongly depends on domain-specific document structures. Gradient Chunking excels in unstructured or narrative-heavy domains (e.g., FiQA) where meaning shifts frequently, yet it suffers from unstable chunk-length variance. In contrast, Structural Chunking maintains consistent segmentation and superior retrieval accuracy by jointly modeling surface-level and physical document features. Its balanced performance across all benchmarks—especially in structurally explicit corpora such as NFCorpus, SciFact, and SCIDOCS—demonstrates strong generalization and adaptability. These results substantiate the proposed method as a domain-agnostic, semantically and structurally integrated chunking paradigm that effectively bridges computational efficiency and contextual fidelity in Retrieval-Augmented Generation (RAG) systems.

V. Conclusion

This study proposed **Structural Chunking**, a novel document segmentation paradigm that jointly models semantic cohesion and structural consistency. Unlike conventional chunking methods that rely solely on semantic similarity, the proposed approach quantifies surface- and physical-level structures and fuses them with sentence embeddings to detect both contextual and formal discontinuities.

Experimental results on five BEIR benchmark datasets—NFCorpus, SciFact, ArguAna, SCIDOCS, and FiQA—demonstrated that Structural Chunking consistently achieved superior and stable performance, particularly in structurally organized domains such as scientific, medical, and argumentative texts. These findings empirically confirm that incorporating structural information not only preserves hierarchical document organization but also enhances retrieval efficiency within Retrieval-Augmented Generation (RAG) pipelines.

Nevertheless, the current evaluation primarily covers datasets with relatively short or moderately structured documents. Further validation on long-form and high-variance texts (e.g., policy reports, technical white papers, full-length research articles) remains necessary. Additionally, while the fusion weight parameters ($\alpha = 0.05$) were empirically fixed, their sensitivity across domains has not been systematically explored. Future work will investigate adaptive parameter optimization and extend the framework to large-scale, multimodal, and multilingual document environments, enabling automated and domain-aware structural chunking.

Acknowledgment

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2710088929, AI Assistant-Based Automation Technologies for Operations and Management in Cloud Failure Recovery).

References

[1] S. Ganatra and P. Bhattacharyya, "Retrieval augmented generation: A comprehensive survey," 2025, arXiv preprint.

[2] R. Qu, R. Tu, and F. Bao, "Is semantic chunking worth the computational cost?" *arXiv preprint arXiv:2410.13070*, 2024.

[3] S. Latif, H. Ameer, and M. H. Akram, "The chunking paradigm: Recursive semantic for rag optimization," in *Proceedings of the International Conference on Natural Language and Speech Processing (ICNLSP)*, 2025.

[4] C. Merola and J. Singh, "Reconstructing context: Evaluating advanced chunking strategies for rag," in *Knowledge-Enhanced NLP Workshop*, 2025.

[5] R. D. Pesl, J. G. Mathew, M. Mecella, and M. Aiello, "Retrieval-augmented generation for service discovery: Chunking strategies and benchmarking," *arXiv preprint arXiv:2505.19310*, 2025.

[6] R. Khanda, "Agentic ai-driven technical troubleshooting with weighted rag," *arXiv preprint arXiv:2412.12006*, 2024.

[7] J. Zhao, Z. Ji, S. Niu, H. Wang, F. Xiong, and Z. Li, "Mom: Mixtures of scenario-aware document memories for rag systems," *arXiv preprint arXiv:2510.14252*, 2025.

[8] A. Brown, M. Roman, and B. Devereux, "A systematic literature review of rag: Techniques, metrics, and challenges," *arXiv preprint arXiv:2508.06401*, 2025.

[9] H. T. Nguyen, T. D. Nguyen, and V. H. Nguyen, "Enhancing rag with hierarchical text segmentation chunking," in *Springer Proceedings on Information Systems and Applications*. Springer, 2024.

[10] M. Stäbler, S. Turnbull, and T. Müller, "The impact of chunking strategies on domain-specific ir in rag systems," in *Proceedings of the IEEE OmniConf*. IEEE, 2025.

[11] W. Wang, B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng, and L. Si, "Structbert: Incorporating language structures into pre-training for deep language understanding," *arXiv preprint arXiv:1908.04577*, 2019.

[12] P. Qian, T. Naseem, R. Levy, and R. F. Astudillo, "Structural guidance for transformer language models," *arXiv preprint arXiv:2108.00104*, 2021.

[13] Y. Zhang, X. Zhao, Z. Z. Wang, C. Yang, J. Wei, and T. Wu, "cast: Enhancing code retrieval-augmented generation with structural chunking via abstract syntax tree," *arXiv preprint arXiv:2506.15655*, 2025.

[14] N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych, "Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models," *arXiv preprint arXiv:2104.08663*, 2021.

[15] V. Boteva, D. Gholipour, A. Sokolov, and S. Riezler, "A full-text learning to rank dataset for medical information retrieval," in *European Conference on Information Retrieval*. Cham: Springer International Publishing, 2016, pp. 716–722.

[16] D. Wadden, S. Lin, K. Lo, L. L. Wang, M. van Zuylen, A. Cohan, and H. Hajishirzi, "Fact or fiction: Verifying scientific claims," *arXiv preprint arXiv:2004.14974*, 2020.

[17] H. Wachsmuth, M. Potthast, K. Al Khatib, Y. Ajjour, J. Puschmann, J. Qu, and B. Stein, "Building an argument search engine for the web," in *Proceedings of the 4th Workshop on Argument Mining*, 2017, pp. 49–59.

[18] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. S. Weld, "Specter: Document-level representation learning using citation-informed transformers," *arXiv preprint arXiv:2004.07180*, 2020.

[19] M. Maia, S. Handschuh, A. Freitas, B. Davis, R. McDermott, M. Zarrouk, and A. Balahur, "Www'18 open challenge: Financial opinion mining and question answering," in *Companion Proceedings of The Web Conference 2018*, 2018, pp. 1941–1942.

[20] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: ACM, 2020, pp. 1192–1200.

[21] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press, 2008, vol. 39.

[22] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Kuettler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.