Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3394~3401
e-ISSN: 2548-6861

Improving Retrieval-Augmented Generation Grounding Using Document Hierarchy Chunk Graph

Putri Cristin [1]*, Hilmil Pradana [2]*
* Teknik Informatika, Institut Sepuluh Nopember
putri.cristin@gmail.com [1], hilmi@its.ac.id [2]

Article Info

Article history:

Received 2026-06-17
Revised 2026-07-24
Accepted 2026-08-08

Keyword:

Retrieval-Augmented Generation (RAG), Document Hierarchy, Information Retrieval, Grounding, Large Language Models.

ABSTRACT

Retrieval-Augmented Generation (RAG) is widely used to enhance question-answering systems across various domains. However, while real-world source documents are inherently structured, conventional RAG approaches primarily rely on semantic similarity between isolated text chunks, which can overlook document hierarchy and limit retrieval effectiveness. To address this issue, this study introduces a document hierarchy-based Chunk Graph approach to improve retrieval grounding in RAG systems. The proposed framework preserves document hierarchy during chunking and models inter-chunk relationships using a weighted graph that combines structural proximity and semantic similarity. The approach was evaluated using the StructuredQA and CUAD benchmark datasets, with performance measured via Precision, Recall, and F1-Score. Experimental results demonstrate that the effectiveness of the Chunk Graph depends heavily on the source document format. On the highly structured StructuredQA dataset, the proposed method successfully connects fragmented information, improving the F1-Score from 47.62% to 50.23%. Conversely, on the CUAD dataset which consists of raw text with implicit hierarchy and no nested structure, the model becomes redundant and does not yield performance gains. These findings conclude that integrating structural-semantic relationships significantly improves context selection, specifically for documents with explicit hierarchical structures.

I. INTRODUCTION

Retrieval-Augmented Generation (RAG) has emerged as a definitive framework to integrate external retrieval mechanisms with generative modelling, effectively addressing the inherent limitations of Large Language Models (LLMs) [1]. While LLMs have achieved significant milestones in Natural Language Processing — demonstrating excellence in few-shot learning, cross-task generalization, and linguistic understanding — they remain fundamentally constrained by the static nature of their training data. Extensive surveys confirm that while LLMs perform well in complex reasoning [2], [3], context-based summarization [4] and question answering [5], they generate information based solely on knowledge encoded within their pre-trained parameters. Consequently, these models lack direct access to real-time facts or external information outside their initial training corpus [6]. This limitation introduces significant risks, such as hallucinations [7] — convincing outputs that lack factual accuracy — particularly in specialized domains [8] or knowledge-intensive tasks where up-to-date information is critical [9].

By incorporating relevant external knowledge into the generative process, RAG has proven capable of improving factual grounding [10] and reducing inference errors [11]. Generally, RAG functions by segmenting documents into text chunks, performing semantic similarity searches [12] and appending the retrieved results as additional context [13] for the LLMs. However, retrieval approaches based solely on semantic similarity have inherent limitations. Studies indicate that LLMs are highly sensitive to irrelevant noise [14], unrelated or incomplete information can act as a distraction, negatively affecting the model's interpretation of the original query.

In real-world documents, information is often organized within hierarchical structures consisting of sections, subsections, and nested content. Conventional chunking methods based on fixed token counts or paragraphs frequently disregard these structural boundaries, potentially separating semantically related content and reducing retrieval effectiveness [15]. Existing research shows that chunk quality directly affects RAG performance, especially when semantic independence and information preservation are not maintained [16]. Consequently, document segmentation strategies are recognized as a crucial factor influencing retrieval effectiveness and answer quality [17], [18].

Although hierarchy-aware chunking strategies have recently gained traction, most conventional approaches still treat chunks as isolated text fragments. As a result, contextual relationships between structurally related sections may not be fully utilized during retrieval, potentially limiting the ability of RAG systems to access complementary information distributed across multiple document segments. Structural documents are authored sequentially to build a coherent logical narrative, meaning that related sections are designed to be interpreted together to provide a comprehensive and holistic perspective. When chunks from the same segment are scattered and treated as independent text fragments, this narrative continuity is broken, thereby stripping the retrieved evidence of the vital surrounding context that ought to be interpreted as a unified whole.

Several recent studies have explored more advanced representations of document data to address these retrieval challenges. For instance, frameworks like Structure-RAG[19] utilize structured document layouts to guide the retrieval process and improve answer synthesis. Other approaches leverage graph architectures to bridge the interconnected relationships between different chunks, as exemplified by entity-centric graph RAG framework [20], which captures global and local contexts through entity-relation networks [20]. While these methodologies significantly enhance retrieval capabilities, they have not yet fully explored the impact of preserving the complete structural integrity of the document during the initial chunking process. Consequently, a gap remains for an approach that explicitly maintains hierarchical continuity to prevent context fragmentation.

Based on these findings, this study proposes a document hierarchy-based Chunk Graph approach for RAG. This method focuses on preserving document hierarchy and builds structural and semantic relationships between chunks and is represented by a graph. The resulting Chunk Graph enables a graph-based retrieval mechanism that explores inter-chunk relationships more thoroughly than conventional similarity-based retrieval. This approach is better able to preserve structural context that could be lost in the conventional chunking process. This design also aims to improve retrieval grounding and context relevance, especially for documents that have a more complex structural structure. The main contributions of this study are as follows:

- Document Hierarchy-Based Chunk: A chunking strategy that preserves document hierarchy.
- Structural-Semantic Chunk Graph: A weighted graph integrating structural and semantic relationships between chunks.
- Comprehensive Performance Evaluation: An evaluation against a conventional RAG baseline.

II. METHODOLOGY

In this section, we present the proposed method to pre-processing data, build knowledge construction, integration into RAG pipeline, and evaluate the performance. The system workflow presented in Figure 1.

A. Dataset

To evaluate the effectiveness of the proposed structural-semantic Chunk Graph retrieval approach, we conducted experiments using two public benchmark datasets: StructuredQA [21] and CUAD (Contract Understanding Atticus Dataset)[22]. These datasets were selected to evaluate the effectiveness of hierarchy-aware chunk organization and structural-semantic retrieval under different document conditions.

StructuredQA consists of short-answer pairs based on scientific papers where the source documents are original, highly structured PDF files. These documents possess explicit hierarchical levels and nested sections, characterized by diverse visual formatting, font styles, and clear typographic hierarchies that render heading boundaries explicit. In contrast, CUAD comprises long-form legal contracts presented in a pre-extracted JSON format as flat raw text. Unlike StructuredQA, the document hierarchy in CUAD is predominantly single-leveled and inherently flat. The structural markers and section boundaries are entirely relying on textual patterns such as capitalized strings, line breaks, or section numbers rather than explicit layout metadata. Consequently, these two datasets represent fundamentally different structural challenges—ranging from extracting explicit multi-level hierarchies in PDFs to inferring implicit, flat text boundaries—making them suitable for testing the boundary limitations of our structure-aware approach.

B. Pre-processing

In the initial stage, PDF documents are converted into Markdown format to facilitate the identification of document hierarchies and logical document flow. During this conversion, the original dataset is iterated row by row to identify whether the row is a chapter or subchapter heading. This identification is based on several factors, such as the presence of chapter numbering in the row, different font styles, such as bold, different font sizes, or capitalization. Subsequently, the extracted text then cleaned to remove noise, including anomalous characters, formatting artifacts, misaligned line breaks, and page-level extraction errors. This

process aims to retain relevant textual context for further analysis.

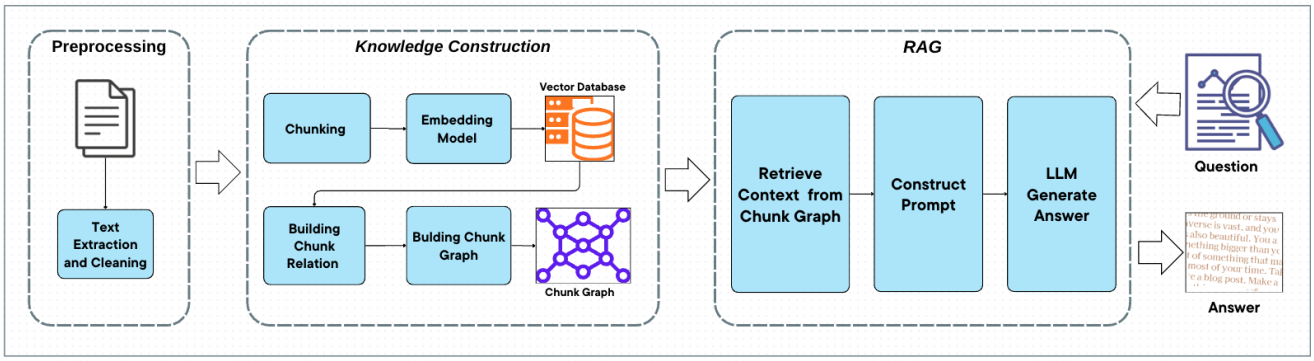

Figure 1 Workflow of the proposed Chunk Graph-based RAG framework

C. *Knowledge Construction*

This stage transforms the pre-processed data into a structured knowledge source that supports the retrieval process in the RAG framework.

1) Chunking: The Markdown file is then converted into shorter pieces of text, called chunks. The chunking process is carried out by recognizing the heading markers that were assigned during pre-processing. The result of this process is that each chunk represents a complete part of a subchapter or chapter. From the generated chunks, a second stage of the chunking process is performed to further divide chunks whose size exceeds the token limit. The length of each chunk is limited to a maximum of 150 tokens with no token overlap. Token overlap is avoided because it may bias the weight of the semantic relationships between chunks, which are later used to construct the Chunk Graph. During this chunking process, if missing structures, fragmented text, or unnumbered chapters are encountered, the system detects the nearest preceding chapter title to logically assign the orphaned segment to that chapter. This mechanism is applied to maintain structural completeness, thereby enabling the structural weights to be adequately captured.
2) Embedding Model: Each chunk is converted into an embedding vector using a Transformer-based model. These embeddings, along with their metadata, are then stored in a vector database.
3) Building Chunk Relation: To preserve context between document segments, two types of relationships are created between chunks: semantic relations and structural relations. Semantic relations are quantified using cosine similarity between chunk embedding vectors, as defined in Equation (1).

$$W_{semantic(C_i,C_j)} = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| \times |\vec{B}|} \quad (1)$$

where $W_{semantic(C_i,C_j)}$ represents the semantic similarity between chunks C_i and C_j, while $\vec{A}$ and $\vec{A}$ denote their corresponding embedding vectors. Higher similarity values indicate stronger conceptual relationships between chunks, even when they originate from different sections of a document.

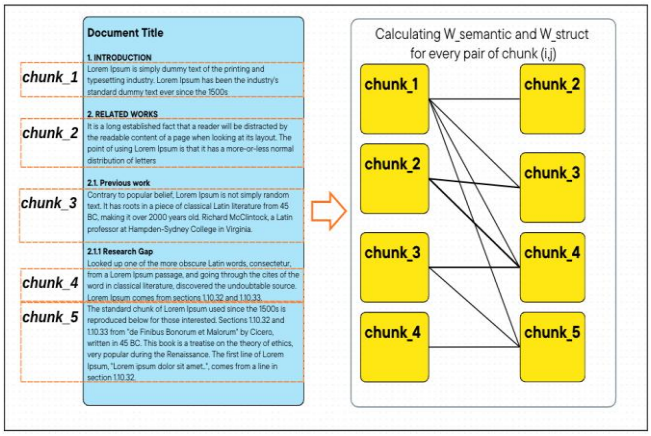

Figure 2 Process of building semantic and structural relation between chunks

Structural relations are computed based on the hierarchical positions of chunks within the same document, as defined in Equation (2). The structural weight $W_{struct(C_i,C_j)}$ represents the structural proximity between chunks C_i and C_j.

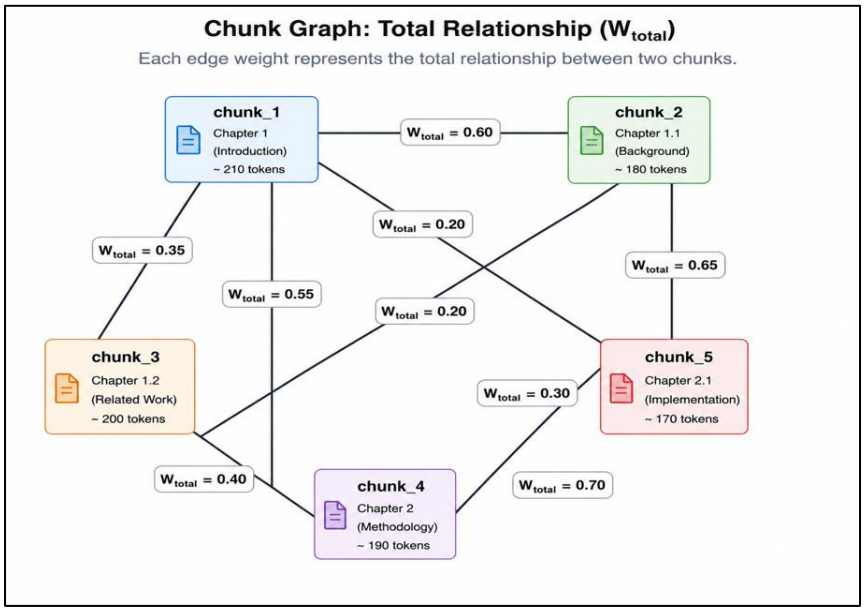

Figure 3 Illustration of the proposed Chunk Graph with total relationship weights between chunks

Chunks originating from different documents are assigned a structural weight of zero. For chunks within the same document, structural proximity is determined according to their relative positions within the document hierarchy.

$$W_{struct(C_i,C_j)} = \begin{cases} 0 \ (if\ doc_i \neq doc_j) \\ 1 - d_1 \cdot level_{gap} - d_2 \cdot f(lcp) \end{cases} \quad (2)$$

Parameter d_1 controls the sensitivity to differences in hierarchical depth, while parameter d_2 regulates the influence of positional proximity within the document structure. The variable $level_{gap}$ represents the difference in hierarchical depth between two chunks, calculated from the lengths of their hierarchical numbering sequences. For example, a chunk labelled "2.1" (level 2) and another labelled "2.1.1" (level 3) yield a $level_{gap}$ value of 1.

Furthermore, the function $f(lcp)$ incorporates hierarchical proximity based on the Longest Common Prefix (lcp) between chunk identifiers. This function normalizes structural similarity within the range [0,1] relative to the maximum document depth, as defined in Equation (3).

$$f(lcp) = 1 - \left(\frac{lcp}{max\_level}\right) \quad (3)$$

The lcp value is calculated as the number of identical prefix elements shared by two hierarchical identifiers. For example, chapter "2" and subchapter "2.1" have an lcp value of 1, while subchapters "2.1" and "2.1.3" have an lcp value of 2. Higher lcp values indicate stronger structural relatedness. The parameter max_level denotes the maximum hierarchical depth present in the document. The process of building chunk relation is illustrated in Figure 2.

4) Chunk Graph: constructed by representing chunks as nodes and inter-chunk relationships as weighted edges. Mathematically, a Chunk Graph is represented as a directed and weighted graph defined in Equation (4) and illustrated in Figure 3.

$$G = (V, E). \quad (4)$$

V represents the set of nodes, where each node corresponds to a generated text chunk. E denotes the set of edges, representing the total relationship weights between the chunks. Total relationship weight is formulated as $W_{total(C_i,C_j)}$, which combines semantic and structural connectivity between chunks. The total edge weight $W_{total(C_i,C_j)}$ is computed by integrating $W_{semantic(C_i,C_j)}$ and $W_{struct(C_i,C_j)}$. Parameter α, where $0 < \alpha \leq 1$, acts as a balancing coefficient between semantic and structural contributions. The optimal value of α is determined experimentally.

$$W_{total(C_i,C_j)} = \alpha \cdot W_{struct(C_i,C_j)} + (1 - \alpha) \cdot W_{semantic(C_i,C_j)} \quad (5)$$

D. Retrieval-Augmented Generation

The RAG process consists of three stages: context retrieval, prompt construction, and response generation.
Given an input query Q, the query is first transformed into an embedding vector $\vec{Q}$. The first retrieval phase involves identifying chunks that exhibit semantic proximity to vector $\vec{Q}$. These search results are called Initial Chunks. The initial chunks then serve as the foundation for the second retrieval phase, which identifies neighbouring chunks based on their cumulative relationship weights through a Chunk Graph traversal. This second process is designed to prevent the inclusion of duplicate chunks in the list, thereby mitigating context redundancy. Next, all selected chunks are organized into contexts and integrated into prompts. The prompts instruct the LLM to answer questions based solely on the provided context and to return a 'Not Found' response if the relevant information is absent from the context.

III. RESULT AND DISCUSSION

This section presents the experimental setup, the evaluation results, and a discussion of the findings.

A. Experimental Setup

The experiments were conducted in a Python environment utilizing a *Google Colab T4 GPU* instance. During the pre-processing stage, the PyMuPDF4LLM and Regex libraries were employed as parsers to convert the documents into Markdown format and to identify chapter and subchapter headings.

This study utilized the *BAAI/bge-small-en-v1.5* model [23] due to its high efficiency and strong performance in capturing semantic text representations. Subsequently, the generated embedding vectors were stored in *ChromaDB* as the vector database. This study utilized the is *Qwen2.5-7B-Instruct* model [24] as the LLMs due to its advanced instruction-following capabilities and strong performance in complex reasoning tasks within a computationally efficient 7-billion-parameter architecture.

To establish a baseline for comparison, this study implemented a vanilla RAG pipeline. The baseline approach utilized the same dataset, but the text was partitioned into fixed-size chunks without considering the chapter or subchapter structures.

These baseline chunks were then embedded using the same embedding model. During the retrieval phase, text chunks were retrieved based solely on semantic similarity, and the retrieved chunks were directly composed into a context to be integrated with the prompt for the LLM. Conversely, for the Chunk Graph configuration, the expansion was limited to a single traversal step, retrieving up to the *top-n* neighbouring chunks. The retrieved context is then fed into the same LLMs for final response generation.

To evaluate the performance of the proposed chunk relations and graph-based retrieval, the Precision, Recall, F1 Score are utilized as the primary indicator to measure the balance between Precision and Recall relative to the established ground truth.

B. Result and Discussion

The experimental results on the StructuredQA dataset shows that the proposed Chunk-Graph RAG consistently outperforms the baseline across all evaluation metrics. As detailed in the Table I, when retrieving a smaller number of chunks $top_k = 3$ the Chunk-Graph RAG with an alpha parameter α = 0.3 achieves a significant improvement in F1-Score, rising from 40.83% to 47.58%. This performance gain is further sustained at $top_k = 5$ and α = 0.3 where the Chunk-Graph RAG reaches the highest overall F1-Score of 50.23%, compared to the baseline's 47.62%.

The sensitivity analysis of the α parameter reveals an insightful trend regarding the interaction between structural and semantic weights. At $top_k = 5$, configuring α = 0.3 yields the highest enhancement, achieving an F1-Score of 50.23% (a +2.61% improvement over the baseline). This indicates that integrating a 30% structural weight is already effective in injecting layout-aware context into the retrieval pipeline, allowing the system to bridge fragmented chunks without losing focus on the user's explicit query intent. Interestingly, when the structural weight is further increased to α = 0.5 and α = 0.7, the performance remains perfectly stable at an F1-Score of 49.38%. This plateau effect demonstrates a structural information saturation. It indicate that the proposed Chunk-Graph does not require high parameter tuning to function optimally; a minor structural constraint α = 0.3 is sufficient to resolve the contextual boundary issues inherent in traditional RAG. Therefore, the structural component acts as a highly efficient contextual anchor rather than a volatile variable, ensuring robust and stable performance gains across different hyperparameter boundaries.

However, this accuracy improvement introduces a direct trade-off with computational duration. As shown in Table I, at $top_k = 5$ and α = 0.3 increases the runtime duration to 254.17 seconds compared to the baseline's 152.50 seconds. This latency overhead is caused by the execution of graph traversal algorithms required to reconstruct fragmented text contexts. Despite the additional processing time, this computational trade-off is highly acceptable because the gain in F1-Score (+2.61%) significantly reduces context loss, which is much more critical for the reliability of the RAG system. It is important to note that this duration represents the end-to-end RAG pipeline—measured from the moment a query is submitted until the LLM generates the final response—rather than the retrieval phase alone. The pre-processing and knowledge construction stage are not contributed to this runtime.

TABLE I
EVALUATION RESULT OF EXPERIMENT IN STRUCTUREDQA DATASET

Method	top_k	α	Precision (%)	Recall (%)	F1_Score (%)	Exact Match (%)	Rouge-L (%)	Duration (seconds)
Fixed size RAG (baseline)	3	-	42.13	42.33	40.83	31.07	40.25	**129.43**
	5	-	**48.33**	49.54	**47.62**	**37.86**	**47.09**	152.50
Chunk-Graph RAG (ours)	3	0.3	47.27	48.01	47.58	37.62	46.13	**172.90**
	3	0.5	48.14	48.18	46.73	36.89	46.20	174.86
	3	0.7	48.14	48.18	46.73	36.89	46.20	176.89
	5	0.3	50.00	50.60	**50.23**	**39.60**	48.69	254.17
	5	0.5	**50.84**	**50.77**	49.38	38.83	**48.76**	291.75
	5	0.7	**50.84**	**50.77**	49.38	38.83	**48.76**	261.97

TABLE II
EVALUATION RESULT OF EXPERIMENT IN CUAD DATASET

Method	top_k	α	Precision (%)	Recall (%)	F1_Score (%)	Exact Match (%)	Rouge-L (%)	Duration (seconds)
Fixed size RAG (baseline)	3	-	0.10%	0.09%	0.04%	0.00%	0.04%	97.78
	5	-	**0.18%**	**0.09%**	**0.04%**	0.00%	**0.04%**	**94.19**
Chunk-Graph RAG (ours)	3	0.3	**0.09%**	**0.09%**	**0.04%**	0.00%	**0.04%**	101.65
	3	0.5	**0.09%**	**0.09%**	**0.04%**	0.00%	**0.04%**	**101.20**
	3	0.7	**0.09%**	**0.09%**	**0.04%**	0.00%	**0.04%**	106.47
	5	0.3	0.08%	0.01%	0.02%	0.00%	0.02%	107.24
	5	0.5	0.08%	0.01%	0.02%	0.00%	0.02%	102.27
	5	0.7	0.08%	0.01%	0.02%	0.00%	0.02%	106.96

In stark contrast to the structured environments, the experimental evaluations on the CUAD dataset show a complete flattening of retrieval performance across all hyperparameter configurations. As presented in TABLE II, both the baseline and the proposed Chunk-Graph RAG yield near-zero accuracy scores, with the F1-Score saturating strictly at 0.04% for $top_k = 3$, and slightly shifting to 0.02% for $top_k = 5$. Furthermore, varying the structural relational weight $\alpha = 0.3, 0.5, 0.7$ results in a plateau, failing to alter the precision, recall, or F1-Score. This extreme alignment and lack of variance sensitivity demonstrate a critical boundary limitation that aligned with the theoretical foundation of our approach. The CUAD dataset consists of legal contracts delivered in a pre-extracted, raw JSON text format.

Since it lacks explicit physical layout metadata—such as multi-level font variations, nested document structured, or clear hierarchical tags typically found in parsed PDFs, the structural graph topology becomes redundant.

When α is varied, the system attempts to infer structural relationships from a text that is inherently single-levelled and linear. Since the structural boundaries are entirely implicit, the framework generates an identical, flat chunk network regardless of the hyperparameter weight. This explains the perfect replication of scores across different α levels (e.g., 0.04% F1-Score for all α at $top_k = 3$). This performance ceiling suggests that forcing a structure-aware graph routing on non-hierarchical raw text no measurable performance gain on the CUAD dataset, while still incurring a baseline runtime latency overhead, increasing from 94.19 seconds to an average of 105.4 seconds at $top_k = 5$. Therefore, these results indicate that the proposed Chunk-Graph RAG operates as a specialized framework optimized is particularly effective for documents with explicit hierarchical structure, rather than a generalized text retriever

The findings of this study are consistent with previous research emphasizing the importance of document segmentation and structure preservation in RAG systems [25]. Brådland et al. [16] demonstrated that chunk quality has a direct impact on retrieval effectiveness, particularly when contextual information is fragmented during chunk construction. Similarly, recent studies have shown that hierarchy-aware chunking strategies can improve retrieval

quality by preserving document structure during segmentation [10], [26].

Our results show that the benefits of preserving structure depend heavily on the format of the document. The significant accuracy increase in the StructuredQA dataset shows that document hierarchy is important. When a document has a clear physical layout, our graph method successfully connects fragmented information. Conversely, the results on the CUAD dataset show a clear limitation. If the source data is just text without explicit structure, forcing a graph method becomes redundant and does not improve performance.

```
Dataset          : StructuredQA
Doc Title        : "Regulation"
Question         : "Can LLMs be used as an alternative to visiting a doctor?"
Ground Truth     : "No"
Answer Section   : "Limitations of generative AI and LLMs"
Retrieved Context :
LLMs are not true
domain experts. On their own, they are not a substitute for professional
advice, especially in legal, medical, or other critical areas where precise and
contextually relevant information is essential.
```

Figure 4 Example of case where context retrieved but LLM not answer

An additional qualitative analysis of the "Not Found" responses revealed a case in which the Chunk Graph successfully retrieved relevant evidence, yet the final answer remained unanswered ("Not found"). This occurred when the required information was distributed across multiple chunks or required implicit reasoning beyond explicit textual statements. For example, as illustrated in Figure 2, the retrieval process successfully identified a section stating that LLMs are not substitutes for professional medical advice. However, because the document did not explicitly answer the question "Can LLMs be used as an alternative to visiting a doctor?", the LLM abstained from generating a conclusion under the imposed grounding constraints. This observation suggests that retrieval effectiveness and answer generation are distinct challenges. While the proposed Chunk Graph improves retrieval grounding by providing structurally and semantically relevant context, the final performance remains influenced by the reasoning behaviour of the downstream language model [27], [28].

IV. Conclusion

This study proposes a document hierarchy-based Chunk Graph approach to enhance retrieval grounding in RAG. By integrating structural hierarchy and semantic similarity into a graph representation, this method offers an alternative extension to conventional retrieval strategies. Experimental evaluations demonstrate that the effectiveness of this approach depends heavily on the format of the source document. The most significant advancement was observed on the StructuredQA dataset, where the proposed method effectively connects fragmented information in highly structured PDFs, achieving a substantial accuracy increase over the baseline RAG. Conversely, on the CUAD dataset which consists of raw text with implicit hierarchy and no nested structure, the structural graph routing becomes redundant and does not yield performance gains.

These findings show the value of document structure in retrieval and demonstrate that structural information is highly impactful when explicit structural section and layout metadata is available. Within its optimal scope, the Chunk Graph approach effectively improves retrieval relevance across related document sections. However, further exploration is required to benchmark this approach against other state-of-the-art frameworks or more complex hierarchical retrieval systems.

Future research may further investigate mechanisms for reasoning over retrieved evidence across broader benchmarks. Potential directions include adaptive weighting strategies, multi-hop graph traversal, Chain-of-Thought prompting, evaluation using larger-scale language models, and the integration of multimodal document elements such as tables and images. These extensions may further improve the ability of RAG systems to utilize structured information in complex document environments.

References

[1] P. Lewis and others, "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems*, 2020, pp. 9459–9474. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf

[2] W. X. Zhao *et al.*, "A Survey of Large Language Models," *Front. Comput. Sci.*, vol. 20, no. 12, p. 2012627, Dec. 2026, doi: 10.1007/s11704-026-60308-3.

[3] Y. Chang and others, "A Survey on Evaluation of Large Language Models," *ACM Trans. Intell. Syst. Technol.*, vol. 15, no. 3, pp. 1–45, 2024, doi: 10.1145/3641289.

[4] T. Zhang, F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto, "Benchmarking Large Language Models for News Summarization," *Trans. Assoc. Comput. Linguist.*, vol. 12, pp. 39–57, Jan. 2024, doi: 10.1162/tacl_a_00632.

[5] A. Mansurova, A. Tleubayeva, A. Nugumanova, A. Shomanov, and S. E. Seker, "A Systematic Evaluation of Large Language Models and Retrieval-Augmented Generation for the Task of Kazakh Question Answering," *Information*, vol. 16, no. 11, p. 943, Oct. 2025, doi: 10.3390/info16110943.

[6] H. Xiong *et al.*, "When Search Engine Services Meet Large Language Models: Visions and Challenges," *IEEE Trans. Serv. Comput.*, vol. 17, no. 6, pp. 4558–4577, Nov. 2024, doi: 10.1109/TSC.2024.3451185.

[7] L. Huang and others, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *ACM Trans. Inf. Syst.*, vol. 43, no. 2, pp. 1–55, 2025, doi: 10.1145/3703155.

[8] L. M. Amugongo, P. Mascheroni, S. Brooks, S. Doering, and J. Seidel, "Retrieval augmented generation for large language models in healthcare: A systematic review," *PLOS Digital Health*, vol. 4, no. 6, p. e0000877, Jun. 2025, doi: 10.1371/journal.pdig.0000877.

[9] Y. Pu *et al.*, "Customized Retrieval Augmented Generation and Benchmarking for EDA Tool Documentation QA," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 12, pp. 4615–4628, Dec. 2025, doi: 10.1109/TCAD.2025.3568776.

[10] P. Zhao *et al.*, "Retrieval-Augmented Generation for AI-Generated Content: A Survey," *Data Sci. Eng.*, vol. 11, no. 1, pp. 1–29, Mar. 2026, doi: 10.1007/s41019-025-00335-5.

[11] S. Liu, A. B. McCoy, and A. Wright, "Improving large language model applications in biomedicine with retrieval-augmented generation: a systematic review, meta-analysis, and clinical development guidelines," *J. Am. Med. Inform. Assoc.*, vol. 32, pp. 605–615, 2025, doi: doi:10.1093/jamia/ocaf008.

[12] V. Karpukhin and others, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, 2020, pp. 6769–6781. doi: 10.18653/v1/2020.emnlp-main.550.

[13] S. Borgeaud *et al.*, "Improving Language Models by Retrieving from Trillions of Tokens," in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., in Proceedings of Machine Learning Research, vol. 162. PMLR, Jun. 2022, pp. 2206–2240. [Online]. Available: https://proceedings.mlr.press/v162/borgeaud22a.html

[14] F. Shi *et al.*, "Large language models can be easily distracted by irrelevant context," in *Proceedings of the 40th International Conference on Machine Learning*, in ICML'23. JMLR.org, 2023.

[15] C. A. Gomez-Cabello *et al.*, "Comparative Evaluation of Advanced Chunking for Retrieval-Augmented Generation in Large Language Models for Clinical Decision Support," *Bioengineering*, vol. 12, no. 11, p. 1194, Nov. 2025, doi: 10.3390/bioengineering12111194.

[16] H. Brådland, M. Goodwin, P.-A. Andersen, A. S. Nossum, and A. Gupta, "A New HOPE: Domain-agnostic Automatic Evaluation of Text Chunking," in *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2025, pp. 170–179. doi: 10.1145/3726302.3729882.

[17] Z. Wang *et al.*, "Document Segmentation Matters for Retrieval-Augmented Generation," in *Findings of the Association for Computational Linguistics: ACL 2025*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2025, pp. 8063–8075. doi: 10.18653/v1/2025.findings-acl.422.

[18] M. R. Hajar, E. Utami, and A. Hendi Muhammad, "A Systematic Literature Review of Retrieval-Augmented Generation: Methods, Applications, and Future Research Directions," *Journal of Applied Computer Science and Technology*, vol. 6, no. 2, pp. 115–128, Dec. 2025, doi: 10.52158/jacost.v6i2.1170.

[19] X. Zhu, X. Guo, S. Cao, S. Li, and J. Gong, "StructuGraphRAG: Structured Document-Informed Knowledge Graphs for Retrieval-Augmented Generation," *Proceedings of the AAAI Symposium Series*, vol. 4, no. 1, pp. 242–251, 2024, doi: 10.1609/aaaiss.v4i1.31798.

[20] D. Edge and others, "From Local to Global: A Graph RAG Approach to Query-Focused Summarization," *arXiv preprint arXiv:2404.16130*, 2025, doi: 10.48550/arXiv.2404.16130.

[21] Mozilla AI, "Structured QA," 2025, *Mozilla AI Blog*. Accessed: Jan. 14, 2026. [Online]. Available: https://blog.mozilla.ai/structured-question-answering-2/

[22] D. Hendrycks, C. Burns, A. Chen, and S. Ball, "CUAD: An Expert-Annotated NLP Dataset for Legal Contract Review," *arXiv preprint arXiv:2103.06268*, 2021, doi: 10.48550/ARXIV.2103.06268.

[23] J. Chen, S. Xiao, P. Zhang, K. Luo, D. Lian, and Z. Liu, "M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation," in *Findings of the Association for Computational Linguistics ACL 2024*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2024, pp. 2318–2335. doi: 10.18653/v1/2024.findings-acl.137.

[24] Q. A. Yang *et al.*, "Qwen2.5 Technical Report," *ArXiv*, vol. abs/2412.15115, 2024, [Online]. Available: https://api.semanticscholar.org/CorpusID:274859421

[25] S. Toro *et al.*, "Dynamic Retrieval Augmented Generation of Ontologies using Artificial Intelligence (DRAGON-AI)," *J. Biomed. Semantics*, vol. 15, no. 1, p. 19, Oct. 2024, doi: 10.1186/s13326-024-00320-3.

[26] 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.

[27] K. Rangan and Y. Yin, "A fine-tuning enhanced RAG system with quantized influence measure as AI judge," *Sci. Rep.*, vol. 14, no. 1, p. 27446, Nov. 2024, doi: 10.1038/s41598-024-79110-x.

[28] M. Abo El-Enen, S. Saad, and T. Nazmy, "A survey on retrieval-augmentation generation (RAG) models for healthcare applications," *Neural Comput. Appl.*, vol. 37, no. 33, pp. 28191–28267, Nov. 2025, doi: 10.1007/s00521-025-11666-9.