MultiDocFusion: Hierarchical and Multimodal Chunking Pipeline for Enhanced RAG on Long Industrial Documents

Joongmin Shin[1], Chanjun Park[3], Jeongbae Park[1], Jaehyung Seo[1,2‡], Heuiseok Lim[1,2‡]
[1]Human-inspired AI Research, Korea University
[2]Department of Computer Science and Engineering, Korea University
[3]School of Software, Soongsil University
{tlswndals13, insmile, seojae777, limhseok}@korea.ac.kr
bcj1210nlp@ssu.ac.kr

Abstract

RAG-based QA has emerged as a powerful method for processing long industrial documents. However, conventional text chunking approaches often neglect the complex structures of long industrial documents, causing information loss and reduced answer quality. To address this, we introduce **MultiDocFusion**, a multimodal chunking pipeline that integrates: (i) detection of document regions using vision-based document parsing, (ii) text extraction from these regions via OCR, (iii) reconstruction of document structure into a hierarchical tree using large language model (LLM)-based document section hierarchical parsing (DSHP-LLM), and (iv) construction of hierarchical chunks through DFS-based Grouping. Extensive experiments across industrial benchmarks demonstrate that **MultiDocFusion** improves retrieval precision by 8–15% and ANLS QA scores by 2–3% compared to baselines, emphasizing the critical role of explicitly leveraging document hierarchy for multimodal document-based QA. These significant performance gains underscore the necessity of structure-aware chunking in enhancing the fidelity of RAG-based QA systems.

1 Introduction

The emergence of retrieval-augmented generation (RAG) has significantly advanced the capabilities of large language models (LLMs) in handling long and information-dense documents (Lewis et al., 2021; Jeong, 2023; Ge et al., 2023). Central to the success of RAG pipelines is the document chunking strategy, which segments source documents into manageable and semantically coherent units. Despite its importance, existing chunking methods remain predominantly text-centric, relying on fixed-length splits or shallow semantic cues, and fail to account for the rich visual and structural attributes inherent in real-world documents (Gong et al., 2020; Gao et al., 2024).

This limitation becomes especially problematic in industrial and academic domains where documents often take the form of scanned images, multi-page PDFs, or reports with intricate visual and hierarchical layouts. For instance, visual elements such as tables, figures, and section headers may span multiple pages, while hierarchical section structures encode critical semantic relationships that are lost under naive chunking. Optical Character Recognition (OCR) artifacts further exacerbate this issue by introducing noise and misalignments in the extracted text, thereby degrading both retrieval and QA performance (Tito et al., 2023; Hong et al., 2024). As a result, general RAG systems frequently fail to preserve the documents' semantic continuity, leading to information fragmentation and suboptimal generation quality.

While recent advances in vision-based document parsing (DP) and OCR techniques enable the extraction of visually coherent regions such as tables and text blocks (Dosovitskiy et al., 2021; Pfitzmann et al., 2022), these approaches lack an explicit representation of logical structure, particularly the parent-child relationships embedded in hierarchical sectioning (Xing et al., 2024). This structural gap limits their effectiveness in tasks that depend on accurate context reconstruction and long-range reasoning.

To bridge this gap, we introduce **MultiDocFusion**, a multimodal chunking pipeline that explicitly incorporates both visual layout and a document's structural hierarchy into the chunking process. Our framework integrates four key components: (i) detection of document regions and layout structure using vision-based DP, (ii) text extraction from these regions via OCR, (iii) section hierarchical parsing with large language models (DSHP-LLM), and (iv) depth-first search (DFS)-based chunk assembly. By reconstructing a document's

‡ Co-corresponding authors

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 20985–21004
November 4-9, 2025

semantic hierarchy and aligning it with visual segmentation, **MultiDocFusion** produces structurally faithful and semantically coherent chunks that are better suited for downstream RAG-based QA.

We evaluate our approach across diverse document types, such as financial statements, scientific reports, scanned forms, and visually intricate multi-page documents, consistently demonstrating improvements in both retrieval precision and answer accuracy. Our results highlight that explicitly modeling the document's hierarchical structure is essential for robust and context-aware question answering. Our main contributions are summarized as follows:

- **MultiDocFusion**: A novel pipeline that systematically integrates DP, OCR, DSHP-LLM, and DFS-based Grouping, effectively handling the structural complexities unique to industrial documents that conventional approaches typically overlook.
- **DSHP-LLM**: We introduce DSHP-LLM, an instruction-tuned LLM that robustly reconstructs hierarchical section structures from diverse and complex documents, enabling precise context preservation for downstream retrieval and QA.
- **Comprehensive Experiments**: We conduct extensive validation across various industrial and academic domains, including financial reports, technical documents, scanned images, and documents with complex layouts, and demonstrate consistent improvements in both retrieval and QA performance (retrieval precision by 8–15% and ANLS QA scores by 2–3%).

2 Related Work

Chunking for QA on Long Industrial Documents Chunking has emerged as an essential strategy for effectively handling long, multi-page documents (Gao et al., 2024). Traditionally, documents have been segmented using Length chunking (Gong et al., 2020) or Semantic chunking (Qu et al., 2025). However, these methods often fail to adequately reflect hierarchical relationships among sections or incorporate visual layout elements such as tables and figures. Recent approaches leveraging LLMs, such as LumberChunker (Duarte et al., 2024) and Perplexity chunking (Zhao et al., 2024), still suffer from contextual fragmentation because they lack explicit modeling of document hierarchies (Hong et al., 2024). StyleDFS, which constructs hierarchical trees using font size and style, struggles with scanned documents lacking text layers or irregular layouts (Hong et al., 2024). While end-to-end multimodal models combining textual and visual information have been proposed to mitigate these limitations (Hu et al., 2024; Wang et al., 2024a; Fujitake, 2024), most methods face challenges due to limited context lengths, making it difficult to process entire multi-page documents at a time. Consequently, there is a growing need for chunking methods that comprehensively capture document structure and context (Saad-Falcon et al., 2023; Kang et al., 2024).

Document Parsing and Hierarchical Parsing Recent studies in Visual Question Answering (VQA) have increasingly focused on document parsing (DP), aiming to segment PDFs and image-based documents into visual components such as tables, figures, and text blocks (Dosovitskiy et al., 2021; Pfitzmann et al., 2022). However, these object-detection-centric approaches inherently lack the capability to fully reconstruct semantic hierarchical relationships, such as the relationship between sections "1.2" and "1.2.1" (Rausch et al., 2023; Wang et al., 2024a). Document hierarchical parsing (DHP) methods have been proposed to address these issues (Rausch et al., 2021, 2023; Zhang et al., 2024c), but their applicability remains limited to structured templates or certain document types, struggling with scanned or irregularly formatted documents (Wang et al., 2024b; Xing et al., 2024). However, LLMs have emerged as promising candidates for DHP tasks due to their advanced text understanding and long-context handling capabilities (Fujitake, 2024). LLMs still face challenges in inferring hierarchical semantic connections between sections, necessitating additional fine-tuning or specialized instruction-tuning methods (Zhang et al., 2024b; Wang et al., 2024a; Tabatabaei et al., 2025). While existing approaches may capture either visual layout or textual content, they fail to unify structural and semantic hierarchies. The **MultiDocFusion** pipeline addresses this gap by systematically combining visual region detection, OCR, and LLM-based hierarchical parsing to enable accurate, context-aware chunking of long industrial documents.

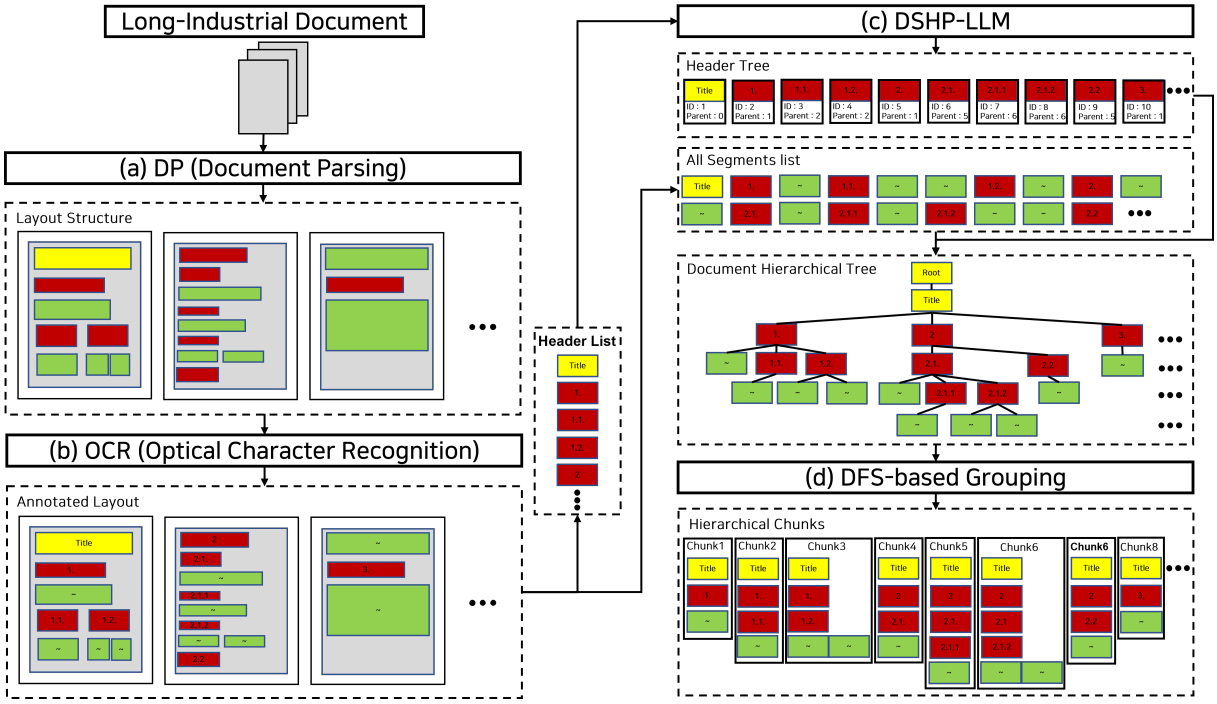


Figure 1: The pipeline for **MultiDocFusion**. The figure illustrates the step-by-step process for handling a long industrial document. (a) DP extracts layout structures; (b) OCR recognizes and annotates text; (c) DSHP-LLM constructs a hierarchical tree from identified section headers and general nodes; (d) DFS-based Grouping constructs coherent hierarchical chunks for retrieval tasks. The color-coded blocks represent document elements: yellow for Root and Title, red for Section Headers, and green for general nodes (tables, figures, and text blocks).

3 MultiDocFusion

MultiDocFusion (*Multimodal Document Structure Fusion*) is a pipeline designed to effectively integrate visual layouts and hierarchical semantic structures of long industrial documents, enhancing chunking and retrieval performance. The term "MultiDoc" emphasizes the pipeline's capability to handle diverse document formats frequently encountered in industrial settings such as PDFs, scanned images, and documents with complex layouts, and to support corpus-level multi-document RAG scenarios, enabling retrieval-augmented generation across large collections of documents. Meanwhile, "Fusion" highlights the integration of visual information, textual content, and hierarchical document analysis to produce refined and contextually accurate chunks. The pipeline consists of four stages: (a) DP (Document Parsing), (b) OCR (Optical Character Recognition), (c) DSHP-LLM (Document Section Hierarchical Parsing with LLM), and (d) DFS-based Grouping. Figure 1 provides an overview of this four-stage process and the resulting hierarchical chunks. Below, we detail each stage using the components and terminology shown in the figure.

3.1 DP (Document Parsing)

As shown in (a), DP examines each page of a long industrial document to identify and extract its *Layout Structure*.

Process Advanced vision models detect Titles, Section Headers, text blocks, tables, figures, etc. Each detected segment is assigned bounding-box coordinates and segment type. The pipeline constructs a page-by-page *Layout Structure* that captures the spatial arrangement of all segments.

Output For each page, DP generates metadata including page numbers, segment IDs, segment types, and bounding box coordinates. This *Layout Structure* is passed to the OCR stage.

3.2 OCR (Optical Character Recognition)

As described in (b), OCR processes the *Layout Structure* from DP to extract text from each bounding box, resulting in an *Annotated Layout*.

Input The page-by-page *Layout Structure* with bounding boxes and segment information from DP.

Process Each segment image is sent to OCR engines tailored to the document's languages and fonts. The recognized text is then linked back to the corresponding bounding box.

Output The *Annotated Layout* merges bounding boxes, segment types, and recognized text into structured metadata, preparing the necessary inputs for the subsequent processing stage.

3.3 DSHP-LLM

As depicted in (c), DSHP-LLM constructs a *Document Hierarchical Tree* by identifying, ordering, and attaching section headers along with other nodes based on *Parent–Child* relationships.

Model Setup DSHP-LLM is built upon an LLM backbone and is instruction-tuned on public datasets of document hierarchies (Zhang et al., 2024a). To improve training efficiency, we employ LoRA-based parameter-efficient fine-tuning (PEFT) (Hu et al., 2021; Han et al., 2024). Hyperparameters and further details are provided in Appendix A.2.

Input The DSHP-LLM receives a *Header List*, which consists of candidate section headers extracted during the DP and OCR stages from the *Annotated Layout*.

Process The DSHP-LLM initially performs *Header Tree* construction by analyzing the *Header List* and assigning each header a unique identifier and parent reference (e.g., `ID:3 Parent:2`), resulting in an initial hierarchical structure (e.g., `Root` → `Title` → `Section 1` → `Section 1.1`). Next, it proceeds to link general nodes, utilizing the *All Segments list* maintained from the DP stage. This list includes tables, figures, text blocks, and other document elements sorted by spatial coordinates, such as page number and bounding box position. As the DSHP-LLM traverses the Header Tree, it sequentially scans through the *All Segments list*. General segments encountered before reaching the next header from the *Header List* are attached as child nodes of the current header node. This ensures accurate grouping of tables, figures, and text blocks, preserving both logical and spatial document structures.

Output The output is a fully *Document Hierarchical Tree* explicitly detailing the hierarchical placement of section headers and associated general nodes (e.g., `Root` → `Title` → `Section 1` → `Section 1.1` → `Text`). By integrating LLM-identified headers with spatially sorted child nodes, the pipeline maintains coherent logical and visual relationships. For example prompts and outputs, refer to Table 10.

3.4 DFS-based Grouping

As illustrated in (d), DFS-based Grouping performs a depth-first traversal of the *Document Hierarchical Tree* to construct coherent *Hierarchical Chunks* (e.g., Chunk1, Chunk2, Chunk3, . . .). During this stage, the hierarchical structure is explicitly reflected within each chunk using Markdown headers, where each chunk's depth corresponds directly to the heading level. Detailed algorithms are provided in Appendix A.5.

Input The *Document Hierarchical Tree* from DSHP-LLM and text corresponding to each node in the tree.

Process In this process, a virtual node called FAKE_ROOT is created, which points directly to the actual root node. The algorithm performs a recursive traversal of the nodes following a depth-first approach, aggregating the text content from parent nodes along with their child nodes to preserve the contextual information. When the aggregated text length surpasses a predefined threshold (max_len), the algorithm splits the chunk at that specific point.

Output A list of *Hierarchical Chunks* that encapsulate entire sections or sub-sections, thereby minimizing token waste. The resulting chunks explicitly represent the document's hierarchical structure via Markdown headers corresponding to each node's depth. For example, if "1" is a parent of "1.1," both might be combined into "Chunk4" to preserve continuity in retrieval/QA tasks. An illustrative example is shown below:

```
# Document Title
## Section 1 {name}
### Section 1.1 {name}
Section 1.1 {Text Content...}
```

By combining *Layout Structure* (from DP) with recognized text (from OCR) into an *Annotated Layout*, and then applying the DSHP-LLM model to build a *Header Tree*, **MultiDocFusion** captures both spatial and semantic relationships in long industrial documents. The final DFS-based Grouping stage yields *Hierarchical Chunks* that main-

tain these relationships, clearly marked by Markdown headers, outperforming traditional text-only chunking in real-world retrieval and QA scenarios.

4 Experimental Settings

This section briefly describes the experimental settings for training and evaluating the DSHP-LLM model and the RAG-based VQA system for multi-page documents. We utilize various datasets and model configurations, with additional details (e.g., dataset statistics, hyperparameters, and model setups) provided in the Appendix A.

Datasets For DSHP-LLM training and testing, we combine documents from DocHieNet (Xing et al., 2024) and HRDH (Ma et al., 2023). These datasets include diverse domains and complex layouts, making them suitable for evaluating the generalization of hierarchical parsing models. For multi-page RAG-based VQA performance evaluation, we use four datasets: DUDE (Landeghem et al., 2023), MPVQA (Tito et al., 2023), CUAD (Hendrycks et al., 2021), and MOAMOB (Hong et al., 2024). These datasets encompass financial reports, contracts, scanned documents, and various structures, allowing comprehensive evaluation of chunking and retrieval performance. For each dataset, we index all test documents jointly and retrieve top-k chunks from the entire corpus (not restricted to a gold document). Unless stated otherwise, $k = 4$. This corpus-level setup reflects realistic deployment and stresses cross-document disambiguation.

Models DP is performed with object detection models such as DETR (Carion et al., 2020) and VGT (Da et al., 2023), while OCR text extraction uses Tesseract (Smith, 2007), EasyOCR (Vedhaviyassh et al., 2022), and TrOCR (Li et al., 2022). The DSHP-LLM, which infers hierarchical parent-child relationships among document section headers, is trained via instruction tuning on LLMs such as Llama-3.2-3B (Grattafiori et al., 2024), Qwen-2.5-3B (Yang et al., 2024), and Mistral-8B (AI, 2024) to predict JSON-structured hierarchies. In the retrieval stage, chunk embeddings are generated using BGE (Chen et al., 2024), E5 (Wang et al., 2024c), and BM25 (Robertson and Zaragoza, 2009). Top-k retrieved chunks are then fed into LLMs (e.g., Llama-based models) for final answer generation.

Model	DocHieNet		HRDH	
	F1	TEDS	F1	TEDS
GPT-4	0.5139	0.6961	0.2594	0.3342
Llama-3.2-3B	0.2558	0.5464	0.4389	0.4904
↪**DSHP-LLM**	**0.4894**	**0.7549**	**0.8664**	**0.8459**
Qwen-2.5-3B	0.4122	0.6995	0.3299	0.3734
↪**DSHP-LLM**	**0.4808**	0.6957	**0.8856**	**0.8658**
Mistral-8B	0.3907	0.6559	0.3445	0.3974
↪**DSHP-LLM**	**0.6291**	**0.8230**	**0.9321**	**0.9199**
Qwen-2.5-7B	0.5230	0.7356	0.2962	0.3807
↪**DSHP-LLM**	**0.5565**	**0.8104**	**0.6330**	**0.6381**

Table 1: Performance on DHP datasets (DocHieNet + HRDH) for DSHP-LLM (section headers). ↪: DSHP-LLM applied. **Bold**: improvement over baseline.

Evaluation We evaluate the DSHP-LLM performance using accuracy, F1, and TEDS (Zhong et al., 2020) metrics. Retrieval quality is measured using Precision, Recall, and nDCG (Järvelin and Kekäläinen, 2002), while generated VQA answers are quantitatively assessed via ANLS (Biten et al., 2019), ROUGE-L (Lin, 2004), and METEOR (Banerjee and Lavie, 2005).

5 Experimental Results

In this section, we comprehensively evaluate the performance of the proposed **MultiDocFusion** pipeline using the experimental setup. The evaluation consists of: (1) DSHP-LLM performance comparison across different fine-tuned LLMs, (2) retrieval performance comparison among different chunking methods, (3) QA performance analysis, and (4) retrieval robustness analysis under various DP, OCR, and embedding model combinations. To provide objective comparative benchmarks, we include several baseline chunking methodologies, such as Length chunking (Gong et al., 2020), Semantic Chunking (Qu et al., 2025), LumberChunker (Duarte et al., 2024), Perplexity chunking (Zhao et al., 2024), and Structure-based Chunking (W/O DSHP-LLM). Detailed chunking methods are explained in Appendix A.3.

5.1 DSHP-LLM Performance

Table 1 summarizes the performance results of section hierarchy parsing on the DocHieNet and HRDH datasets. Each dataset has distinct characteristics: DocHieNet comprises documents from diverse domains, including reports, academic papers, and industrial documents, with complex

Chunking Method	DUDE			MPVQA			CUAD			MOAMOB		
	Recall	Precision	nDCG	Recall	Precision	nDCG	Recall	Precision	nDCG	Recall	Precision	nDCG
Length chunking	0.2628	0.1686	0.2166	0.2523	0.1587	0.1933	0.9011	0.8537	0.8776	0.6462	0.5676	0.6209
Semantic chunking	0.0956	0.0549	0.0775	0.0939	0.0524	0.0680	0.7684	0.6719	0.7181	0.2737	0.1950	0.2453
LumberChunker	0.2395	0.1533	0.1986	0.2152	0.1298	0.1609	**0.9031**	0.8576	0.8800	0.6130	0.5205	0.5692
Perplexity chunking	0.2428	0.1559	0.2020	0.2159	0.1318	0.1629	0.8869	0.8395	0.8603	0.6173	0.5241	0.5785
Structure-based chunking	0.2219	0.1450	0.1862	0.2036	0.1230	0.1524	0.8844	0.8311	0.8581	0.5544	0.4662	0.5149
MultiDocFusion	**0.2927**	**0.2001**	**0.2505**	**0.2705**	**0.1759**	**0.2131**	0.9021	**0.8651**	**0.8819**	**0.6758**	**0.6184**	**0.6554**

Table 2: Retrieval performance by Chunking Method (Average Recall, Precision, nDCG for top-$k = 1 \sim 4$), Best scores are in **bold**.

scanned images, while HRDH focuses on academic papers characterized by intricate layouts. The experimental results show that GPT-4, used without any fine-tuning, demonstrated limited performance on both DocHieNet (TEDS 0.6961) and HRDH (TEDS 0.3342), indicating similar deficiencies across other general-purpose LLMs. This suggests that general pre-training alone is insufficient for effective section hierarchy parsing. Conversely, applying our proposed DSHP-LLM approach significantly improved performance (measured by TEDS) across both datasets, with varying degrees of improvement depending on model and dataset characteristics. Specifically, for the diverse domains and layout complexities in DocHieNet, Mistral-8B +16.71% and Llama-3.2-3B +20.85% showed substantial improvements. For HRDH, characterized by complex yet relatively regular academic document structures, Mistral-8B +52.25% and Qwen-2.5-3B +49.24% achieved the most significant enhancements. These results clearly indicate that general-purpose LLMs have inherent limitations when performing section hierarchy parsing tasks, underscoring the necessity for dataset-specific fine-tuning. Furthermore, the results emphasize the importance of selecting appropriate models and training strategies tailored to the unique characteristics of each dataset. Based on these findings, we selected the fine-tuned Mistral-8B model as the backbone of our DSHP-LLM for integration into the **MultiDocFusion** chunking pipeline, and subsequently evaluated its performance in various multi-page VQA scenarios against other chunking methods.

5.2 Retrieval Performance in Different Chunking Methods

Table 2 presents the average Recall, Precision, and nDCG values for top-$k = 1 \sim 4$ retrieval results, comparing various chunking methods across four multi-page VQA datasets: DUDE, MPVQA, CUAD, and MOAMOB.

MultiDocFusion consistently achieved the best overall retrieval performance across most datasets. In particular, **MultiDocFusion** demonstrated significant advantages in retrieval accuracy on DUDE (Recall 0.2927, Precision 0.2001, nDCG 0.2505) and MPVQA (Recall 0.2705, Precision 0.1759, nDCG 0.2131), clearly outperforming other methods. These results underscore **MultiDocFusion**'s effectiveness even under challenging conditions such as the diverse domains and complex document structures inherent in the DUDE dataset, and the varied layouts characteristic of the MPVQA dataset. On CUAD, while LumberChunker attained the highest Recall (0.9031), **MultiDocFusion** showed superior Precision (0.8651) and nDCG (0.8819), confirming its capability for precise retrieval in specialized legal documents. Moreover, in MOAMOB, an extreme scenario characterized by highly intricate document structures and challenging questions within a specialized nuclear domain, **MultiDocFusion** (Recall 0.6758, Precision 0.6184, nDCG 0.6554) markedly outperformed other approaches across all evaluation metrics, demonstrating robust and superior performance even with a limited dataset.

Additionally, compared to methods solely dependent on LLM-based chunking (e.g., LumberChunker, Perplexity chunking), **MultiDocFusion** significantly enhanced retrieval performance by explicitly capturing and utilizing the hierarchical structure of documents. Specifically, in datasets with complex document structures such as DUDE and MPVQA, simple LLM-based chunking methods failed to sufficiently incorporate structural relationships or context between sections, thus limiting retrieval performance. Conversely, **MultiDoc-**

	DUDE			MPVQA			CUAD			MOAMOB		
Chunking Method	ANLS	ROUGE-L	METEOR	ANLS	ROUGE-L	METEOR	ANLS	ROUGE-L	METEOR	ANLS	ROUGE-L	METEOR
Length chunking	0.1611	0.1444	0.1988	0.1398	0.0966	0.1408	0.2585	0.1677	**0.1662**	0.2497	0.0823	0.1115
Semantic chunking	0.1548	0.1261	0.1657	0.1332	0.0805	0.0978	0.2593	0.1491	0.1468	0.2455	0.0846	0.1043
LumberChunker	0.1531	0.1284	0.1752	0.1307	0.0769	0.0993	0.2657	0.1630	0.1650	0.2536	0.0848	0.1167
Perplexity chunking	0.1653	0.1390	0.1855	0.1344	0.0751	0.0950	0.2641	0.1646	0.1524	0.2532	0.0894	0.1190
Structure-based chunking	0.1751	0.1489	0.1921	0.1537	0.0980	0.1278	0.2498	0.1556	0.1591	0.2501	**0.0979**	0.1114
MultiDocFusion	**0.1859**	**0.1692**	**0.2285**	**0.1615**	**0.1316**	**0.1850**	**0.2738**	**0.1762**	0.1650	**0.2596**	0.0916	**0.1257**

Table 3: Average QA performance (ANLS, ROUGE-L, METEOR) of six chunking strategies on DUDE, MPVQA, CUAD, and MOAMOB datasets, for top-$k \in \{1, 4\}$. Results are averaged over Llama-3.2-3B, Mistral-8B, and Qwen-2.5-7B models. Best scores are in **bold**.

Fusion effectively captured hierarchical and semantic relationships among sections, significantly improving chunking quality and retrieval performance.

Furthermore, in comparison to Structure-based Chunking, **MultiDocFusion** continuously demonstrated superior performance by incorporating DSHP-LLM, enhancing Recall by 7.08% and Precision by 5.51% on the DUDE dataset. This clearly indicates that explicitly recognizing hierarchical structures and semantic contexts of sections provides more robust and accurate retrieval performance than approaches based solely on physical document structures. Overall, these results affirm the efficacy and practical utility of **MultiDocFusion**'s chunking strategy across various document types (e.g., financial reports, legal contracts, multi-page documents) within multi-page VQA scenarios.

5.3 Impact on QA Performance in Chunking Methods

Table 3 presents a comparative analysis of QA performance (ANLS, ROUGE-L, METEOR) across four multi-page VQA datasets, DUDE, MPVQA, CUAD, and MOAMOB, using various chunking methods.

MultiDocFusion consistently achieved the best QA performance across most datasets. Particularly, it demonstrated significant advantages on MPVQA (ANLS 0.1615, ROUGE-L 0.1316, METEOR 0.1850) and DUDE (ANLS 0.1859, ROUGE-L 0.1692, METEOR 0.2285), clearly outperforming other chunking approaches. These results indicate **MultiDocFusion**'s robustness and effectiveness even under challenging conditions such as the diverse document layouts in MPVQA and the broad range of domains and complex document types characteristic of DUDE. On the CUAD dataset, **MultiDocFusion** achieved the highest ANLS (0.2738) and ROUGE-L (0.1762), though its METEOR score was slightly lower than that of Length chunking (0.1662). This outcome highlights the positive impact of hierarchical structure information in enhancing the coherence and consistency of QA responses. Furthermore, **MultiDocFusion** also recorded the highest scores on MOAMOB in terms of ANLS (0.2596) and METEOR (0.1257), confirming its capability to effectively improve QA quality even under limited and highly complex document scenarios.

Compared to existing LLM-based chunking methods (LumberChunker, Perplexity chunking) as well as simple Length chunking and Semantic chunking (Length chunking, Semantic chunking), **MultiDocFusion** significantly improved retrieval precision by more comprehensively capturing the structural context of documents, thereby enhancing both the accuracy and consistency of QA responses. Particularly notable is that compared to Structure-based Chunking, the additional integration of DSHP-LLM within **MultiDocFusion** substantially elevated RAG-based QA performance. Overall, these findings confirm that **MultiDocFusion**, by explicitly utilizing hierarchical document structures, consistently provides superior QA performance across diverse document scenarios.

5.4 Robustness to Pipeline Components (DP/OCR/Embeddings)

This section provides an in-depth analysis of the robustness of retrieval performance across different chunking methods when varying DP, OCR, and embedding models. All results are compared based on the average nDCG for top-$k = 1 \sim 4$ retrieval outcomes.

(1) Comparison across DP Models Table 4 shows the average nDCG performance of differ-

Chunking Method	DETR	DiT	VGT	Avg
Length chunking	0.4952	0.4863	0.4497	0.4771
Semantic chunking	0.3247	0.3263	0.2297	0.2936
LumberChunker	0.4620	0.4533	0.4412	0.4522
Perplexity chunking	0.4636	0.4460	0.4436	0.4511
Structure-based chunking	0.4396	0.4171	0.4269	0.4279
MultiDocFusion	**0.5014**	**0.4976**	**0.5061**	**0.5017**

Table 4: Average performance of Chunking Methods by DP model (top-$k = 1 \sim 4$ nDCG).

Chunking Method	EasyOCR	Tesseract	TrOCR	Avg
Length chunking	0.5369	0.4799	**0.4144**	0.4771
Semantic chunking	0.3213	0.2757	0.2423	0.2798
LumberChunker	0.5057	0.4546	0.3963	0.4522
Perplexity chunking	0.5115	0.4674	0.3739	0.4509
Structure-based chunking	0.5194	0.4650	0.2993	0.4279
MultiDocFusion	**0.5681**	**0.5068**	0.4097	**0.4949**

Table 5: Average performance of Chunking Methods by OCR model (top-$k = 1 \sim 4$ nDCG).

ent chunking methods across three DP model environments: DETR, DiT, and VGT. Overall, **MultiDocFusion** achieved the highest average performance (0.5017) and consistently delivered superior results across all individual DP models. Particularly noteworthy is its substantial improvement of up to +27.64% over the lowest-performing Semantic chunking method in the VGT environment. This clearly demonstrates that **MultiDocFusion**, by explicitly incorporating hierarchical document structures, consistently maintains robust and superior performance regardless of variations in DP models.

(2) Comparison across OCR Models Table 5 compares the average nDCG scores of various chunking methods across different OCR models, namely EasyOCR, Tesseract, and TrOCR. **MultiDocFusion** consistently achieved the highest performance (avg 0.4949) and notably outperformed other methods, particularly in EasyOCR (avg 0.5681) and Tesseract (avg 0.5068) settings. Even with TrOCR, where the overall performance was lower, **MultiDocFusion** maintained a relatively high score (avg 0.4097), demonstrating that hierarchical structure-based chunking remains robust and provides stable retrieval performance despite variations in OCR quality.

(3) Comparison across Embedding Models Table 6 shows the average nDCG performance across various embedding models (BGE, E5, and BM25) for each chunking method. On average, **MultiDocFusion** achieved the highest overall

Chunking Method	BGE	E5	BM25	Avg
Length chunking	0.4834	0.4715	0.4764	0.4771
Semantic chunking	0.3114	0.3378	0.1825	0.2772
LumberChunker	0.4708	0.4319	0.4539	0.4522
Perplexity chunking	0.4715	0.4318	0.4495	0.4509
Structure-based chunking	0.4679	0.4040	0.4118	0.4279
MultiDocFusion	**0.5213**	**0.4884**	**0.5085**	**0.5061**

Table 6: Average performance of Chunking Methods by embedding model (top-$k = 1 \sim 4$ nDCG).

performance (0.5061), consistently outperforming other chunking methods across all embedding environments. In particular, **MultiDocFusion** recorded the best performance (0.5213) in the BGE embedding environment. These results demonstrate that chunking methods leveraging hierarchical document structure are robust and effective in enhancing retrieval accuracy, irrespective of the embedding model utilized.

6 Conclusion

This work targets a core bottleneck in RAG over long industrial documents: context fragmentation caused by text-only chunking that ignores visual layout and explicit section hierarchy. We formalize this problem and introduce **MultiDocFusion**, a structured multimodal pipeline that (i) parses page-level layout regions (DP), (ii) extracts text with OCR, (iii) reconstructs an explicit section hierarchy with DSHP-LLM, and (iv) assembles hierarchical chunks via DFS to preserve both spatial and semantic context.

Evaluated under a corpus-level setting on four multi-page VQA benchmarks, MultiDocFusion consistently outperforms baseline chunking methods in retrieval and QA. DSHP-LLM, fine-tuned for hierarchical parsing, accurately reconstructs complex section structures and surpasses general-purpose LLMs (e.g., GPT-4) on DHP datasets. The gains hold across diverse domains and layouts and remain stable under different DP, OCR, and embedding choices, underscoring the pipeline's practical reliability.

Taken together, these results support a clear conclusion: *Hierarchy-aware, visually grounded chunking should be a first-class design principle for RAG on long, complex, and often scanned industrial documents*. By aligning visual segmentation with an explicit document tree and reflecting it in chunk boundaries, MultiDocFusion reduces contextual breakage and yields more faithful re-

trieval and answers.

Limitations

Although the **MultiDocFusion** chunking pipeline effectively incorporates document hierarchy to improve retrieval and QA, several limitations remain.

Limited visual grounding of DSHP-LLM Our DSHP-LLM was trained on DHP datasets, which do not provide fine-grained layout signals such as font size/style, color, whitespace, alignment/ruling lines, or column structure. As the model is inherently LLM-centric and primarily conditioned on OCR text with coarse bounding boxes, it underutilizes these visual cues that are often decisive for reliable hierarchy reconstruction in scanned or visually complex pages. Future work should incorporate detailed layout features and, more broadly, visually ground DSHP-LLM via multimodal document encoders or VLM backbones to enable more accurate structural analysis.

Graph-structured retrieval not evaluated Because **MultiDocFusion** induces an explicit hierarchical *document graph* (headers $\rightarrow$ subheaders $\rightarrow$ content blocks) with typed relations (e.g., parent–child, reading order), it can naturally be instantiated as a *GraphRAG* pipeline that retrieves and reasons over nodes and paths. This formulation is likely to better support multi-hop and other reasoning-intensive tasks. However, we did not systematically validate this direction, as the present work focuses on verifying multimodal hierarchical chunking under standard RAG settings. Future research should rigorously evaluate graph-augmented retrieval and reasoning on benchmarks requiring multi-hop, compositional reasoning, and cross-page evidence aggregation.

Error propagation and end-to-end alternatives While a serial, multi-module pipeline is pragmatic and familiar in industrial settings, such a design is inherently susceptible to error propagation: mistakes in earlier-stage components (DP/OCR) can cascade into DSHP-LLM, DFS-based chunking, retrieval, and ultimately QA. To mitigate this risk, future work should investigate the substitutability of VLM-based end-to-end models as drop-in replacements or hybrid components that jointly optimize visual parsing, hierarchical structuring, chunking, and retrieval/answering. Such end-to-end formulations may reduce the accumulation of earlier-stage noise and provide stronger cross-modal consistency, albeit with trade-offs in controllability and interpretability that warrant careful study.

Computational overhead Hierarchical chunking duplicates parent context across multiple children to preserve coherence, which can increase index size, retrieval latency, and storage costs. Budget-aware chunking, graph pruning, and node-level caching/deduplication are practical mitigations to explore.

Ethical Considerations

The primary objective of this research is to enhance multimodal document parsing and question-answering capabilities; however, ethical considerations must be carefully addressed when applying this technology. First, documents processed by the pipeline may contain sensitive information such as personal data, copyrighted materials, or proprietary business content. Thus, meticulous care must be exercised in data collection, processing, and usage to ensure strict adherence to privacy regulations and data security standards.

Second, despite aiming to provide accurate information, the proposed system could inadvertently generate incorrect or biased responses, potentially misleading users. When deploying the system in practical settings, clear guidelines for accountability and measures against misuse should be implemented.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI). This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2021R1A6A1A03045425). This work was supported by the Commercialization Promotion Agency for R&D Outcomes(COMPA) grant funded by the Korea government(Ministry of Science and ICT)(2710086166)

References

Mistral AI. 2024. Ministral-8b-instruct-2410. https://huggingface.co/mistralai/Ministral-8B-Instruct-2410. Accessed: 2024-05-16.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marcal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. 2019. Scene text visual question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4291–4301.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *Computer Vision – ECCV 2020*, pages 213–229.

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

Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. 2023. Vision grid transformer for document layout analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, and et al. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*.

André V Duarte, João Marques, Miguel Graça, Miguel Freire, Lei Li, and Arlindo L Oliveira. 2024. Lumberchunker: Long-form narrative document segmentation. *arXiv preprint arXiv:2406.17526*.

Masato Fujitake. 2024. LayoutLLM: Large language model instruction tuning for visually rich document understanding. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10219–10224, Torino, Italia. ELRA and ICCL.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2024. Retrieval-augmented generation for large language models: A survey. *Preprint*, arXiv:2312.10997.

J. Ge, Steve Sun, Joseph Owens, Victor Galvez, O. Gologorskaya, Jennifer C Lai, Mark J Pletcher, and Ki Lai. 2023. Development of a liver disease-specific large language model chat interface using retrieval augmented generation. *medRxiv*.

Hongyu Gong, Yelong Shen, Dian Yu, Jianshu Chen, and Dong Yu. 2020. Recurrent chunking mechanisms for long-text machine reading comprehension. pages 6751–6761.

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

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Preprint*, arXiv:2403.14608.

Dan Hendrycks, Collin Burns, Anya Chen, and Spencer Ball. 2021. Cuad: An expert-annotated nlp dataset for legal contract review. *NeurIPS*.

Seongtae Hong, Joong Min Shin, Jaehyung Seo, Taemin Lee, Jeongbae Park, Cho Man Young, Byeongho Choi, and Heuiseok Lim. 2024. Intelligent predictive maintenance RAG framework for power plants: Enhancing QA with StyleDFS and domain specific instruction tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 805–820, Miami, Florida, US. Association for Computational Linguistics.

Anwen Hu, Haiyang Xu, Jiabo Ye, Ming Yan, Liang Zhang, Bo Zhang, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mPLUG-DocOwl 1.5: Unified structure learning for OCR-free document understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3096–3120, Miami, Florida, USA. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

CheonSu Jeong. 2023. A study on the implementation of generative ai services using an enterprise data-based llm application architecture. *Adv. Artif. Intell. Mach. Learn.*, 3:1588–1618.

Lei Kang, Rubèn Tito, Ernest Valveny, and Dimosthenis Karatzas. 2024. Multi-page document visual question answering using self-attention scoring mechanism. In *Document Analysis and Recognition - ICDAR 2024: 18th International Conference, Athens, Greece, August 30–September 4, 2024, Proceedings, Part VI*, page 219–232, Berlin, Heidelberg. Springer-Verlag.

Jordy Van Landeghem, Rafał Powalski, Rubèn Tito, Dawid Jurkiewicz, Matthew Blaschko, Łukasz Borchmann, Mickaël Coustaty, Sien Moens, Michał Pietruszka, Bertrand Ackaert, Tomasz Stanisławek, Paweł Józiak, and Ernest Valveny. 2023. Document understanding dataset and evaluation (dude). In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19471–19483.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Preprint*, arXiv:2005.11401.

Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. 2022. Trocr: Transformer-based optical character recognition with pre-trained models. *Preprint*, arXiv:2109.10282.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Jiefeng Ma, Jun Du, Pengfei Hu, Zhenrong Zhang, Jianshu Zhang, Huihui Zhu, and Cong Liu. 2023. Hrdoc: dataset and baseline method toward hierarchical reconstruction of document structures. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3743–3751, New York, NY, USA. Association for Computing Machinery.

Renyi Qu, Ruixuan Tu, and Forrest Sheng Bao. 2025. Is semantic chunking worth the computational cost? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2155–2177, Albuquerque, New Mexico. Association for Computational Linguistics.

Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. 2021. Docparser: Hierarchical document structure parsing from renderings. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35:4328–4338.

Johannes Rausch, Gentiana Rashiti, Maxim Gusev, Ce Zhang, and Stefan Feuerriegel. 2023. Dsg: An end-to-end document structure generator.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Jon Saad-Falcon, Joe Barrow, Alexa Siu, Ani Nenkova, David Seunghyun Yoon, Ryan A. Rossi, and Franck Dernoncourt. 2023. Pdftriage: Question answering over long, structured documents. *Preprint*, arXiv:2309.08872.

R. Smith. 2007. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633.

Seyed Amin Tabatabaei, Sarah Fancher, Michael Parsons, and Arian Askari. 2025. Can large language models serve as effective classifiers for hierarchical multi-label classification of scientific documents at industrial scale? In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 163–174, Abu Dhabi, UAE. Association for Computational Linguistics.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2023. Hierarchical multimodal transformers for multi-page docvqa. *Preprint*, arXiv:2212.05935.

D.R. Vedhaviyassh, R. Sudhan, G. Saranya, M. Safa, and D. Arun. 2022. Comparative analysis of easyocr and tesseractocr for automatic license plate recognition using deep learning algorithm. In *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pages 966–971.

Prashant Verma. 2025. S2 chunking: A hybrid framework for document segmentation through integrated spatial and semantic analysis. *Preprint*, arXiv:2501.05485.

Dongsheng Wang, Natraj Raman, Mathieu Sibue, Zhiqiang Ma, Petr Babkin, Simerjot Kaur, Yulong Pei, Armineh Nourbakhsh, and Xiaomo Liu. 2024a. DocLLM: A layout-aware generative language model for multimodal document understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8529–8548, Bangkok, Thailand. Association for Computational Linguistics.

Jiawei Wang, Kai Hu, Zhuoyao Zhong, Lei Sun, and Qiang Huo. 2024b. Detect-order-construct: A tree construction based approach for hierarchical document structure analysis. *Pattern Recognition*, 156:110836.

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

Hangdi Xing, Changxu Cheng, Feiyu Gao, Zirui Shao, Zhi Yu, Jiajun Bu, Qi Zheng, and Cong Yao. 2024. Dochienet: A large and diverse dataset for document hierarchy parsing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, and Zhihao Fan. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Antonio Jimeno Yepes, Yao You, Jan Milczek, Sebastian Laverde, and Renyu Li. 2024. Financial report chunking for effective retrieval augmented generation. *Preprint*, arXiv:2402.05131.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024a. Instruction tuning for large language models: A survey. *Preprint*, arXiv:2308.10792.

Yizhuo Zhang, Heng Wang, Shangbin Feng, Zhaoxuan Tan, Xiaochuang Han, Tianxing He, and Yulia Tsvetkov. 2024b. Can LLM graph reasoning generalize beyond pattern memorization? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2289–2305, Miami, Florida, USA. Association for Computational Linguistics.

Yue Zhang, Zhihao Zhang, Wenbin Lai, Chong Zhang, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024c. PDF-to-tree: Parsing PDF text blocks into a tree. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10704–10714, Miami, Florida, USA. Association for Computational Linguistics.

Jihao Zhao, Zhiyuan Ji, Pengnian Qi, Simin Niu, Bo Tang, Feiyu Xiong, and Zhiyu li. 2024. Meta-chunking: Learning efficient text segmentation via logical perception.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation. In *European Conference on Computer Vision (ECCV)*, pages 564–580. Springer.

A Appendix

This appendix complements the main text and provides a concise roadmap for reproduction and inspection of results.

- **Datasets:** scope, splits, and language coverage (Sec. A.1; Table 7).
- **Pipeline & Hyperparameters:** DP, OCR, DSHP-LLM, embeddings, and QA LLMs with default settings (Sec. A.2).
- **Compared Chunkers:** definitions and assumptions (Sec. A.3).
- **Detailed Results:** retrieval by k, OCR/DP/Embedding ablations, and QA metrics (Tables 12–15).
- **Server Evaluations:** DUDE/MPVQA ANLS on official test servers (Table 9).
- **Algorithm & Prompts:** DFS-based chunking algorithm and DSHP-LLM prompts/examples (Sec. A.5; Sec. B.1; Figures 2, 3).

A.1 Dataset Details

Dataset	Type/Domain	#Documents	Avg. Pages	#QA pairs
DocHieNet	Mixed (Reports/Papers/Industrial)	1,673	5.3	-
HRDH	Academic Papers (arXiv)	1,500	7.1	-
MPVQA	General Documents (Multi-page)	17,000	3.4	48,000+
CUAD	Legal Documents (Contracts)	510	6.2	13,000+
DUDE	Mixed (Financial Reports/Manuals)	3,000+	4.9	7,000+
MOAMOB	Industrial Technical Documents	2	35.5	71

Table 7: Summary of key datasets used in MultiDocFusion.

(A) Datasets for DSHP-LLM Training and Evaluation DocHieNet and HRDH include annotations of hierarchical section structures (parent-child relationships) within documents, making them suitable for training the DSHP-LLM model.

- **DocHieNet** (Xing et al., 2024): Comprising 1,673 PDF documents (average 5.3 pages/document), this dataset covers diverse domains, including reports, academic papers, and industrial documents, with many scanned images. Each document is annotated with hierarchical JSON structures (parent-child relationships among titles, paragraphs, tables, figures, etc.), making it suitable for training and evaluating models on diverse structural layouts and domains.
- **HRDH** (Ma et al., 2023): Consisting of approximately 1,500 PDF academic papers sourced from arXiv (average 7.1 pages/document), HRDH is a carefully selected subset of the HRDoc dataset featuring particularly complex layouts (HRDoc-Hard). It includes more than 30 types of complicated layouts ranging from single-column to specialized templates. Each line is labeled with its corresponding parent section, making it ideal for training and evaluating hierarchical parsing models.

(B) Multi-page VQA Datasets The datasets *MPVQA*, *CUAD*, *DUDE*, and *MOAMOB* were utilized for practical RAG-based Question Answering (QA) experiments. These datasets include various document formats and layouts, such as indus-

trial reports, legal contracts, and financial documents.

- **DUDE** (Landeghem et al., 2023): Over 3,000 documents spanning various domains such as financial reports and user manuals, with more than 7,000 annotated QA pairs. This dataset enables broad semantic understanding and structural evaluation across diverse document types. Because the official server-evaluated test set does not release ground-truth answers, retrieval metrics cannot be computed, and we cannot directly assess how test-set QA gains are driven by retrieval improvements. Accordingly, for our joint retrieval+QA analysis we report results on the validation split in Table 2 and Table 3, while the official test results[1] are provided in Table 9(approximately 700 documents and 1,500 QA pairs).

- **MPVQA** (Tito et al., 2023): Contains approximately 17,000 documents with more than 48,000 questions (average 2.8 questions per document). As with DUDE, the official server-evaluated test set does not release ground-truth answers, which precludes computing retrieval metrics. Therefore, we use the validation split for the retrieval+QA results reported in Table 2 and Table 3, and include the official test results[2] in Table 9 (around 2,000 documents and 6,000 questions). Its multi-page documents with varied layouts enable assessment of stability and robustness in chunking and retrieval processes.

- **CUAD** (Hendrycks et al., 2021): Comprises 510 legal contracts annotated with over 13,000 QA pairs, primarily targeting specific contractual clauses and legal details. This study uses only the test set (approximately 50 documents and 1,200 QA pairs), making it suitable for verifying the effectiveness of RAG approaches in specialized domains such as law.

- **MOAMOB** (Hong et al., 2024): A small dataset containing just two documents with 71 challenging QA pairs. This study utilizes the entire dataset. Despite its limited size, the dataset's complex document structures and challenging questions provide a rigorous evaluation under constrained conditions.

Language Information: DocHieNet consists of English and Chinese documents, MOAMOB contains Korean documents, and all other datasets are in English.

A.2 Model and Implementation Settings

In our experiments, we cross-applied multiple models for each pipeline component to verify the robustness of the proposed **MultiDocFusion** pipeline across realistic scenarios with varied performance.

(1) Document Parsing (DP) Models To identify layout components (e.g., tables, figures, text blocks) from PDF or scanned images, we utilized several object detection-based models, including DETR, DiT, and VGT, each fine-tuned on the DocLayNet dataset (Pfitzmann et al., 2022). These models generated page-level segment information (`segment ID`, `segment type`, `bounding box`) used for subsequent steps.

(2) OCR Models For text extraction within identified segments, we employed multiple OCR models such as EasyOCR, Tesseract, and TrOCR. The accuracy varied significantly depending on document quality, font types, and languages.

(3) DSHP-LLM (Document Hierarchical Parsing) Models We fine-tuned various LLMs—Llama-3.2-3B, Qwen-2.5-3B, Mistral-8B, and Qwen-2.5-7B—using instruction tuning on hierarchical section structures (represented in `JSON`) derived from the DocHieNet and HRDH datasets. To enhance parameter efficiency, we combined LoRA (Hu et al., 2021) and 4-bit quantization (QLoRA).

(4) Embedding Models For chunk embedding in the retrieval phase, we compared multiple methods including BGE, E5, and traditional BM25. For *top-k* retrieval, we used $k = 4$, selecting the top-ranked chunks as input context for the LLM to generate the final answers.

(5) QA Generation (LLM) Models Answer generation was performed using various LLMs such as Llama-3.2-3B, Mistral-8B, and Qwen-2.5-7B. These models produced responses based on the top-ranked chunks retrieved in the previous step, following a RAG-based approach.

[1] https://rrc.cvc.uab.es/?ch=23
[2] https://rrc.cvc.uab.es/?ch=17

Hardware and Software Environment Experiments were conducted on a single `NVIDIA A100 40GB GPU`, with an `Intel Xeon 32-core CPU` and 256GB RAM. Model training and inference utilized `PyTorch 2.0` and the `Transformers` library. Embedding inference was batch-processed in a CPU/GPU hybrid environment.

Hyperparameters For DSHP-LLM training, the baseline hyperparameters were set as epochs=5, batch size=16, learning rate=1×10^{-5}, with further tuning via grid search. Retrieval utilized a default *top-k* of 4, with BM25 parameters k_1=1.2, b=0.75. To maintain experimental consistency and adhere to the embedding model's context length constraints, the maximum chunk length (`max_len`) was fixed at 550 tokens, following prior studies (Duarte et al., 2024; Hong et al., 2024; Yepes et al., 2024).

A.3 Detailed Descriptions of Compared Chunking Methods

This section provides comprehensive descriptions of the chunking methodologies compared against our proposed **MultiDocFusion** pipeline.

Length chunking (Gong et al., 2020) This method divides documents into chunks based on a fixed token length limit. Each chunk is created uniformly, without considering semantic or structural boundaries. While simple and computationally efficient, it risks splitting important contexts, leading to potential information loss and degraded performance in retrieval and QA tasks.

Semantic chunking (Qu et al., 2025) Semantic chunking leverages encoder-based language models to maintain semantic consistency. Chunks are formed by grouping sentences based on semantic similarity scores derived from language models (e.g., E5 embeddings). Although effective in maintaining semantic coherence, it tends to produce shorter, numerous chunks, potentially impacting retrieval efficiency. Following prior work (Hong et al., 2024), we employed the E5 model for consistency in our experiments.

LumberChunker (Duarte et al., 2024) LumberChunker employs Large Language Models (LLMs) to dynamically partition documents by identifying topical shifts between sentences or paragraphs. It effectively captures the semantic independence of textual segments, resulting in chunks of variable sizes optimized for dense retrieval tasks. For experimental consistency across LLM-based methods, we employed the Mistral-8B model as the base model.

Perplexity chunking (Zhao et al., 2024) Based on the concept of Meta-Chunking, Perplexity chunking identifies optimal chunk boundaries by analyzing the perplexity distribution of sentences and paragraphs. It dynamically merges or splits textual segments at a fine-grained level, effectively balancing granularity and computational efficiency. To ensure fairness among LLM-based methods, we also used the Mistral-8B model for these experiments.

Structure-based Chunking (W/O DSHP-LLM) This approach partitions documents solely based on their structural layouts, such as section headers, tables, and figures. Similar methodologies have been explored in recent works (Yepes et al., 2024; Verma, 2025). In our experiments, Structure-based Chunking served as a baseline to clearly isolate and demonstrate the impact of the proposed DSHP-LLM. Specifically, chunks were created by ordering structural elements obtained via DP (Document Parsing), without explicitly considering hierarchical parent-child relationships identified by DSHP-LLM. Segment types were included in the resulting chunks.

MultiDocFusion Our proposed multimodal chunking pipeline integrates hierarchical document structure into the chunking process. It utilizes the best-performing DSHP-LLM model (fine-tuned Mistral-8B) identified from our previous experiments to explicitly reconstruct section hierarchies, significantly enhancing the semantic and structural coherence of document chunks and thus improving retrieval and QA outcomes.

A.4 Detailed experimental results

A.4.1 Chunking Statistics and Examples

Table 8 summarizes the chunking statistics for the six evaluated chunking methods. Length chunking consistently generates chunks close to the predefined maximum token length. Semantic chunking tends to produce the shortest and highest number of chunks. LumberChunker and Perplexity methods yield intermediate chunk sizes and counts, whereas Structure-based chunking produces relatively longer chunks by explicitly including segment types.

Method	Metric	Avg. Length (Characters / Tokens)	Number of Chunks
Length chunking	Characters Tokens	789.25 548.30	6,807
Semantic chunking	Characters Tokens	289.10 201.54	18,498
LumberChunker	Characters Tokens	702.45 483.82	10,650
Perplexity	Characters Tokens	503.12 350.90	10,615
Structure-based	Characters Tokens	719.50 498.30	9,478
MultiDocFusion	Characters Tokens	766.85 521.65	20,773

Table 8: Chunk statistics (average length and total number) for Length chunking, Semantic chunking, LumberChunker, Perplexity, Structure-based, and **MultiDocFusion** chunking methods (`max_len=550` tokens)

The proposed **MultiDocFusion** generates the highest number of chunks (20,773), each of which tends to approach the maximum token length (averaging 766.85 characters and 521.65 tokens). This increase results from the hierarchical approach where chunks with identical parent headers include duplicated content. Despite generating more chunks, **MultiDocFusion** consistently achieves superior retrieval performance, demonstrating the effectiveness of fine-grained, hierarchical chunking in retrieving relevant context.

A.4.2 Detailed Retrieval and QA Performance Comparisons

Tables 2, 12, 13, 14, 15, 16 extend the summarized results presented in the main text, providing comprehensive comparisons across top-k $= 1 \sim 4$, DP models, OCR models, and embedding models. Consistent with the summarized experiments, these detailed tables further confirm that the **MultiDocFusion** pipeline consistently outperforms other chunking methods across diverse datasets and industrial document scenarios, highlighting its robust chunking performance.

A.4.3 Official Test-Server Results for DUDE and MPVQA

Official test-server result on Table 9

A.5 DFS-based Grouping Algorithm

The DFS-based algorithm traverses the parsed `hierarchy_tree` in a *Depth-First Search* manner, accumulating text from parent to child sections. When the accumulated text exceeds `max_len`, it splits appropriately to create new chunks. This method efficiently maintains the hierarchical document structure while managing the chunk length constraint.

Chunking Method	DUDE	MPVQA
Length chunking	0.1592	0.1348
Semantic chunking	0.1537	0.1294
LumberChunker	0.1573	0.1351
Perplexity chunking	0.1668	0.1299
Structure based Chunking	0.1683	0.1488
MultiDocFusion	**0.1793**	**0.1544**

Table 9: Official test-server ANLS on DUDE and MPVQA. Ground-truth is hidden on the server, so retrieval metrics cannot be computed; main paper reports corpus-level retrieval+QA on validation splits.

Algorithm 1 DFS-based Hierarchical Chunking Algorithm (Conceptual Summary)

Require: `hierarchy_tree`, `max_len`
function DFS_CHUNKING(*node*, *context*)
  $currentText \leftarrow$ node.text
  $temp \leftarrow context + currentText$
  if length($temp$) $>$ max_len **then**
    Split $temp$ into multiple chunks
  else
    Append $temp$ to chunk list
  end if
  for child $\in$ node.children **do**
    DFS_CHUNKING(child, $temp$)
  end for
end function

DFS_CHUNKING(root, "")

B Examples

B.1 Prompt Examples for DSHP-LLM

Table 10 provides condensed examples of the **system prompts**, **user inputs**, and **output examples** used to instruct the DSHP-LLM model to infer the hierarchical structure of document headers. In training, hundreds or thousands of header lists paired with corresponding JSON ground truths are employed.

B.2 Results of Document Chunking Using Different Methods

Table 11 presents the results of applying each chunking method to the document shown in Figure 2. Conventional text-based chunking approaches (Length, Semantic, LumberChunker, Perplexity) often lack clear segmentation criteria between chunks and frequently fail to maintain contextual continuity. In contrast, our proposed method includes higher-level hierarchical nodes within each chunk, thereby preserving contextual coherence and enabling the generation of well-structured, hierarchically organized chunks.

System Prompt (Common)

You are an expert in analyzing section headers of documents and creating a hierarchical structure.
The following is a list of 'section header' texts extracted from a document.

For each item, determine its relationship with the parent section (parent-child relationship).

If possible, follow standard document numbering rules, such as treating '3.1' as a child of '3' and '3.1.1' as a child of '3.1'.

Even if there is no numeric pattern, infer hierarchy based on textual context.

If an item is a top-level heading (i.e., the root node is its parent), set 'parent' to null.

Output format:

json only.

DO NOT include any other explanations or text.

[

{"id": "<id from the original header_list>", "parent": "<id of the parent node or null if root>"}

]

Example 1 (HRDH)

User:

```
{ "id": "2", "text": "1. Introduction",
"page_number": 0, "top": 395, "left":
257 },
{ "id": "3", "text": "2. The puzzle",
"page_number": 1, "top": 659, "left":
263 },
{ "id": "4", "text": "3. Next section",
...},
{ "id": "5", "text": "4. Another
topic", ...},
{ "id": "6", "text": "4.1 Subtopic of
4", ...},
{ "id": "7", "text": "5. Additional
info", ...},
...
```

Assistant:

```
{ "id": "2", "parent": null },
{ "id": "3", "parent": "2" },
{ "id": "4", "parent": "3" },
{ "id": "5", "parent": "4" },
{ "id": "6", "parent": "5" },
{ "id": "7", "parent": "5" },
...
```

Example 2 (DocHieNet)

User:

```
{ "id": "2", "text": "CONGRESSIONAL
BUDGET OFFICE COST ESTIMATE",
"page_number": 1, "top": 74, "left":
171 },
{ "id": "5", "text": "SUMMARY",
"page_number": 1, "top": 282, "left":
71 },
{ "id": "23", "text": "MAJOR PROVISIONS
OF THE LEGISLATION", ...},
{ "id": "37", "text": "ESTIMATED COST
TO THE FEDERAL GOVERNMENT", ...},
{ "id": "39", "text": "BASIS OF
ESTIMATE", ...},
{ "id": "44", "text": "Use of the March
2016 Baseline", ...},
{ "id": "46", "text": "Methodology for
Estimating Effects of ...", ...},
...
```

Assistant:

```
{ "id": "2", "parent": null },
{ "id": "5", "parent": "2" },
{ "id": "23", "parent": "2" },
{ "id": "37", "parent": "2" },
{ "id": "39", "parent": "37" },
{ "id": "44", "parent": "37" },
{ "id": "46", "parent": "37" },
...
```

Table 10: Prompt examples for DSHP-LLM model training. The common prompt (top) is used for both Example 1 (easier, with numbered sections) and Example 2 (harder, no section numbers). Lines expanded so that both examples reach a similar height. The parent value *null* denotes the root node. The symbol `...` indicates omitted content for brevity.

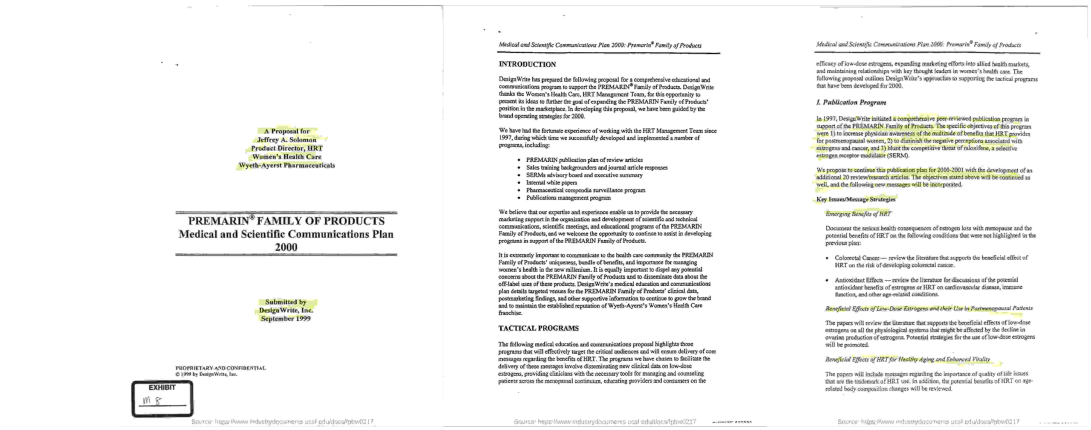
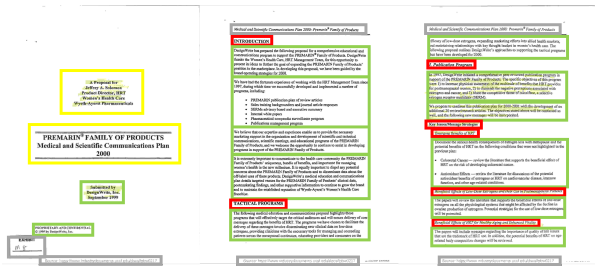
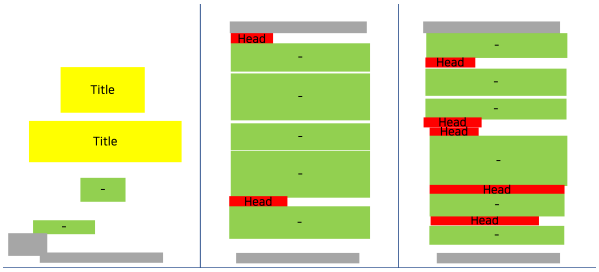


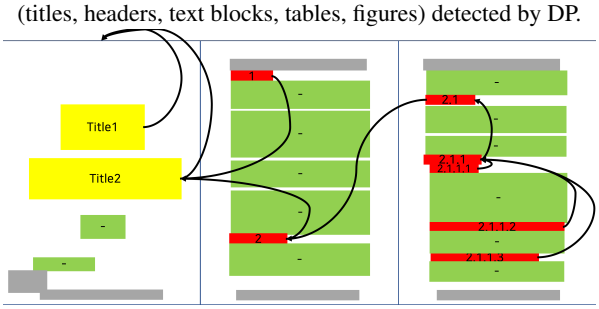
Figure 2: An example of a long industrial document (MPVQA) illustrating the content structure and formatting used for guidelines and requirements in nuclear power plant operations. The document contains various sections, such as general information, application scope, and specific criteria, serving as a representative case for evaluating document chunking methods.



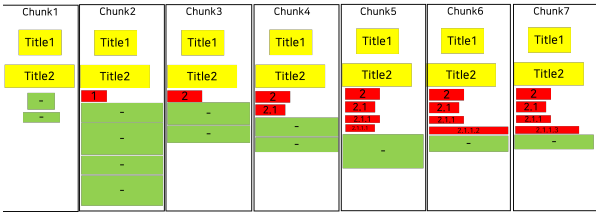
(a) **Document Parsing example** — page-level layout regions (titles, headers, text blocks, tables, figures) detected by DP.



(b) **OCR example** — recognized text linked to bounding boxes, forming an annotated layout.



(c) **DSHP-LLM example** — section headers parsed into a document hierarchical tree with parent–child links; general nodes attached by spatial order.



(d) **Chunking example** — DFS-based Grouping assembles hierarchy-aware chunks that preserve both spatial and semantic context.

Figure 3: Step-by-step illustration aligned with the **MultiDocFusion** pipeline: (a) Document Parsing (DP), (b) OCR, (c) DSHP-LLM for section hierarchy reconstruction and node attachment, and (d) DFS-based Grouping for hierarchy-aware chunking.

Method	Chunk	Example Content
Length chunking	Chunk 1	A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals REMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 tember 1999 PROPRIETARY AND CONFIDENTIAL …
	Chunk 2	INTRODUCTION Design Write has prepared the following proposal for a comprehensive educational and communications program …
	Chunk 3	we have been guided by the brand operating strategies for 2000. We have had the fortunate experience of working with the HRT Management Team since 1997 …
Semantic Chunking	Chunk 1	A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals PREMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 tember 1999 PROPRIETARY AND CONFIDENTIAL © 1999 by Design Write, Ine. EXHIBIT W_§ Source: hops vAwaaw indusirvvdagumens cst eduideesipbwila |?
	Chunk 2	Medical and Scientific Communications Plan. 2000: Premarin® Family of Products
	Chunk 3	INTRODUCTION Design Write has prepared the following proposal for a comprehensive educational and communications program to support the PREMARIN® Family of Products. Design Write thanks the Women's Health Care, HRT Management Team, for this opportunity to present its ideas to further the goal of expanding the PREMARIN Family of Products' position in the marketplace, In developing this proposal, we have been guided by the brand operating strategies for 2000.
LumberChunker	Chunk 1	A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals PREMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 tember 1999 PROPRIETARY AND CONFIDENTIAL © 1999 by Design Write, Ine. EXHIBIT W_§ Source: hops vAwaaw indusirvvdagumens cst eduideesipbwila |? Medical and Scientific Communications Plan. 2000: Premarin® Family of Products
	Chunk 2	Medical and Scientific Communications Plan 2000: Premarin® Fi amily of Products INTRODUCTION Design Write has prepared the following proposal for a comprehensive educational and communications program to support the PREMARIN® Family of Products.
	Chunk 3	Design Write thanks the Women's Health Care, HRT Management Team, for this opportunity to present its ideas to further the goal of expanding the PREMARIN Family of Products' position in the marketplace, In developing this proposal, we have been guided by the brand operating strategies for 2000. We have had the fortunate experience of working with the HRT Management Team since 1997, during which time we successfully developed and impleme
Perplexity chunking	Chunk 1	A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals PREMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 tember 1999 PROPRIETARY AND CONFIDENTIAL © 1999 by Design Write, Ine.EXHIBIT W_§ Source: hops vAwaaw indusirvvdagumens cst eduideesipbwila |?Medical and Scientific Communications Plan.
	Chunk 2	"2000: Premarin® Family of Products Medical and Scientific Communications Plan 2000: Premarin® Fi amily of Products INTRODUCTION Design Write has prepared the following proposal for a comprehensive educational and communications program to support the PREMARIN® Family of Products.Design Write thanks the Women's Health Care, HRT Management Team, for this opportunity to present its ideas to further the goal of expanding the PR
	Chunk 3	EMARIN Family of Products' position in the marketplace, In developing this proposal, we have been guided by the brand operating strategies for 2000.
Structure-based Chunking	Chunk 1	[Title] | A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals
	Chunk 2	[Title] PREMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 - [Text] " - [Text] Medical and Scientific Communications Plan 2000: Premarin® Fi amily of Products
	Chunk 3	[Section] INTRODUCTION - [Text] Design Write has prepared the following proposal for a comprehensive educational and communications program to support the PREMARIN® Family of Products. Design Write thanks the Women's Health Care, HRT Management Team, for this opportunity to present its ideas to further the goal of expanding the PREMARIN Family of Products' position in the marketplace, In developing this proposal, we have been guided by the bra
MultiDocFusion	Chunk 1	# | A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals + PREMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 ## INTRODUCTION text_split_1 : Design Write has prepared the following proposal for a comprehensive educational and communications program to support the PREMARIN® Family of Products. Design Write thanks the Women's Health Care, HRT Management Team, for this opportunity to present its ideas to further the goal of expanding the PREMARIN Family of Products' position in
	Chunk 2	# | A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals + PREMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 ## INTRODUCTION text_split_2 : the marketplace, In developing this proposal, we have been guided by the brand operating strategies for 2000. We have had the fortunate experience of working with the HRT Management Team since 1997, during which time we successfully developed and implemented a number of programs, including: PREMARIN publication plan of review articl
	Chunk 3	# | A Proposal for — Jeffrey A. Solomon _ Product Director, HRT _ Women's Health Care "Wyeth-Ayerst Pharmaceuticals + PREMARIN" FAMILY OF PRODUCTS Medical and Scientific Communications Plan 2000 ## INTRODUCTION text_split_3 : es Sales training backgrounders:and journal article responses SERMs advisory board and executive summary Internal white papers Pharmaceutical compendia surveillance program Publications management program We believe that our expertise and experience enable us:to provide the necessary marketing support in the organization and development of sci

Table 11: Qualitative comparison of chunking methods applied to the document in Figure 2. Each method shows three chunks (1 to 3) for six approaches: Length chunking, Semantic chunking, LumberChunker, Perplexity chunking, Structure-based chunking, and **MultiDocFusion**.

Chunking Method	k	DUDE			MPVQA			CUAD			MOAMOB		
		R	P	nDCG	R	P	nDCG	R	P	nDCG	R	P	nDCG
Length chunking	1	0.1642	0.1642	0.1642	0.1556	0.1556	0.1556	0.8607	0.8607	0.8607	0.5847	0.5847	0.5847
	2	0.2454	0.1684	0.2105	0.2367	0.1591	0.1815	0.9001	0.8513	0.8770	0.6435	0.5711	0.6218
	3	0.3001	0.1700	0.2368	0.2889	0.1599	0.2084	0.9157	0.8514	0.8842	0.6696	0.5605	0.6349
	4	0.3416	0.1717	0.2546	0.3278	0.1601	0.2278	0.9278	0.8513	0.8883	0.6869	0.5542	0.6423
Semantic chunking	1	0.0586	0.0586	0.0586	0.0562	0.0562	0.0562	0.6810	0.6810	0.6810	0.2079	0.2079	0.2079
	2	0.0883	0.0559	0.0750	0.0856	0.0530	0.0629	0.7560	0.6657	0.7100	0.2598	0.1943	0.2406
	3	0.1095	0.0534	0.0848	0.1079	0.0508	0.0724	0.8063	0.6726	0.7353	0.2963	0.1903	0.2589
	4	0.1258	0.0516	0.0916	0.1258	0.0495	0.0803	0.8304	0.6681	0.7463	0.3309	0.1874	0.2738
LumberChunker	1	0.1538	0.1538	0.1538	0.1282	0.1282	0.1282	0.8611	0.8611	0.8611	0.5022	0.5022	0.5022
	2	0.2222	0.1526	0.1922	0.1979	0.1288	0.1489	**0.9049**	0.8577	0.8820	0.6242	0.5304	0.5792
	3	0.2715	0.1530	0.2157	0.2484	0.1308	0.1742	0.9179	0.8568	0.8867	0.6524	0.5272	0.5932
	4	0.3107	0.1538	0.2325	0.2862	0.1315	0.1924	**0.9287**	0.8549	0.8902	0.6731	0.5221	0.6022
Perplexity chunking	1	0.1566	0.1566	0.1566	0.1313	0.1313	0.1313	0.8355	0.8355	0.8355	0.5195	0.5195	0.5195
	2	0.2293	0.1572	0.1975	0.1991	0.1312	0.1511	0.8859	0.8389	0.8602	0.6262	0.5336	0.5868
	3	0.2753	0.1553	0.2196	0.2481	0.1322	0.1755	0.9049	0.8413	0.8696	0.6513	0.5225	0.5994
	4	0.3099	0.1547	0.2345	0.2852	0.1326	0.1935	0.9212	0.8422	0.8758	0.6721	0.5207	0.6083
Structure-based chunking	1	0.1471	0.1471	0.1471	0.1209	0.1209	0.1209	0.8341	0.8341	0.8341	0.4593	0.4593	0.4593
	2	0.2092	0.1453	0.1818	0.1902	0.1235	0.1426	0.8851	0.8347	0.8597	0.5457	0.4647	0.5138
	3	0.2500	0.1443	0.2013	0.2347	0.1239	0.1651	0.9026	0.8267	0.8667	0.5931	0.4700	0.5375
	4	0.2814	0.1432	0.2146	0.2685	0.1237	0.1813	0.9160	0.8289	0.8718	0.6197	0.4709	0.5490
MultiDocFusion	1	**0.2029**	**0.2029**	**0.2029**	**0.1773**	**0.1773**	**0.1773**	**0.8622**	**0.8622**	**0.8622**	**0.6153**	**0.6153**	**0.6153**
	2	**0.2791**	**0.2004**	**0.2459**	**0.2566**	**0.1762**	**0.2016**	0.9021	**0.8655**	**0.8832**	**0.6828**	**0.6281**	**0.6629**
	3	**0.3277**	**0.1993**	**0.2692**	**0.3059**	**0.1753**	**0.2275**	**0.9186**	**0.8688**	**0.8902**	**0.7006**	**0.6183**	**0.6718**
	4	**0.3612**	**0.1978**	**0.2838**	**0.3422**	**0.1749**	**0.2460**	0.9254	**0.8638**	**0.8919**	**0.7122**	**0.6121**	**0.6768**

Table 12: Retrieval summary by Chunking Method — Comparison of VQA Datasets (top-$k = 1 \sim 4$)

Chunking Method	OCR	DUDE			MPVQA			CUAD			MOAMOB		
		R	P	nDCG	R	P	nDCG	R	P	nDCG	R	P	nDCG
Length chunking	easyocr	0.2836	0.1805	0.2289	0.2701	0.1658	0.2042	0.9076	0.8756	**0.8923**	0.8511	0.7689	0.8223
	tesseract	0.2511	0.1617	0.2082	0.2804	0.1805	0.2168	0.8919	0.8469	0.8644	0.6489	0.5919	0.6301
	trocr	0.2538	**0.1636**	**0.2125**	0.2064	0.1298	0.1589	**0.9037**	**0.8386**	**0.8760**	**0.4385**	0.3421	**0.4103**
Semantic chunking	easyocr	0.1050	0.0593	0.0845	0.1037	0.0572	0.0757	0.7392	0.6517	0.6869	0.4437	0.3241	0.4078
	tesseract	0.0909	0.0516	0.0739	0.1014	0.0556	0.0731	0.7573	0.6584	0.7057	0.2785	0.2038	0.2502
	trocr	0.0907	0.0537	0.0742	0.0765	0.0443	0.0551	0.8088	0.7055	0.7618	0.0989	0.0571	0.0779
LumberChunker	easyocr	0.2543	0.1664	0.2114	0.2272	0.1334	0.1669	0.9021	0.8670	0.8813	0.8111	0.7125	0.7631
	tesseract	0.2146	0.1351	0.1765	0.2303	0.1410	0.1738	0.9046	0.8637	0.8817	0.6189	0.5474	0.5863
	trocr	0.2497	0.1586	0.2078	0.1881	0.1151	0.1421	0.9028	0.8421	0.8770	0.4089	0.3015	0.3582
Perplexity chunking	easyocr	0.2482	0.1603	0.2051	0.2230	0.1331	0.1667	0.8979	0.8647	0.8823	0.8293	0.7370	0.7918
	tesseract	0.2422	0.1584	0.2029	0.2396	0.1486	0.1816	0.9000	0.8632	0.8795	0.6337	0.5665	0.6057
	trocr	0.2379	0.1491	0.1981	0.1851	0.1138	0.1403	0.8627	0.7904	0.8191	0.3889	0.2687	0.3380
Structure-based chunking	easyocr	0.2633	0.1759	0.2231	0.2439	0.1451	0.1819	0.8996	0.8600	0.8789	0.8318	0.7279	0.7938
	tesseract	0.2476	0.1631	0.2078	0.2456	0.1542	0.1884	0.8990	0.8553	0.8765	0.6230	0.5464	0.5871
	trocr	0.1548	0.0960	0.1277	0.1212	0.0697	0.0871	0.8547	0.7781	0.8188	0.2085	0.1243	0.1637
MultiDocFusion	easyocr	**0.3305**	**0.2350**	**0.2871**	**0.3034**	**0.1974**	**0.2401**	**0.9077**	**0.8787**	0.8911	**0.8859**	**0.8081**	**0.8542**
	tesseract	**0.2938**	**0.2020**	**0.2524**	**0.2963**	**0.1949**	**0.2347**	**0.9077**	**0.8794**	**0.8896**	**0.6644**	**0.6107**	**0.6504**
	trocr	**0.2539**	0.1633	0.2119	**0.2117**	**0.1354**	**0.1645**	0.8908	0.8370	0.8650	0.4109	**0.3756**	0.3972

Table 13: OCR performance by Chunking Method (top-$k = 1 \sim 4$ average)

Chunking Method	DP Model	DUDE			MPVQA			CUAD			MOAMOB		
		R	P	nDCG	R	P	nDCG	R	P	nDCG	R	P	nDCG
Length chunking	detr	0.2918	0.2007	0.2507	0.2588	0.1680	0.2023	0.9097	0.8800	0.8934	**0.6611**	**0.6078**	**0.6345**
	dit	0.2852	0.1836	0.2370	**0.2835**	0.1852	0.2250	0.9047	0.8450	0.8804	0.6278	0.5365	0.6030
	vgt	0.2115	0.1214	0.1620	0.2145	0.1227	0.1526	0.8888	0.8361	0.8590	0.6496	0.5585	0.6252
Semantic chunking	detr	0.1298	0.0784	0.1069	0.1109	0.0605	0.0790	0.8621	0.7873	0.8226	0.3274	0.2374	0.2903
	dit	0.1037	0.0591	0.0845	0.1123	0.0647	0.0833	0.7373	0.6244	0.6746	0.2933	0.2241	0.2667
	vgt	0.0532	0.0271	0.0412	0.0585	0.0318	0.0416	0.7059	0.6039	0.6573	0.2004	0.1234	0.1788
LumberChunker	detr	0.2386	0.1612	0.2027	0.2123	0.1301	0.1596	**0.9213**	0.8866	**0.9049**	0.6178	0.5509	0.5809
	dit	0.2357	0.1462	0.1920	0.2315	0.1444	0.1784	0.9027	0.8420	0.8701	0.6048	0.5168	0.5727
	vgt	0.2442	0.1526	0.2011	0.2018	0.1150	0.1448	0.8854	0.8442	0.8650	0.6163	0.4937	0.5541
Perplexity chunking	detr	0.2527	0.1697	0.2145	0.2098	0.1301	0.1585	0.9055	0.8720	0.8878	0.6315	0.5517	0.5915
	dit	0.2429	0.1556	0.2019	0.2312	0.1447	0.1777	0.8739	0.8093	0.8329	0.5985	0.5089	0.5716
	vgt	0.2327	0.1425	0.1898	0.2068	0.1206	0.1524	0.8813	0.8371	0.8602	0.6218	0.5117	0.5724
Structure-based chunking	detr	0.2390	0.1642	0.2046	0.2013	0.1205	0.1471	0.8941	0.8550	0.8710	0.5859	0.4937	0.5355
	dit	0.2210	0.1440	0.1855	0.2103	0.1293	0.1606	0.8785	0.8075	0.8464	0.5130	0.4267	0.4759
	vgt	0.2058	0.1268	0.1684	0.1991	0.1192	0.1497	0.8806	0.8308	0.8568	0.5644	0.4782	0.5332
MultiDocFusion	detr	**0.3051**	**0.2192**	**0.2662**	**0.2800**	**0.1864**	**0.2235**	0.9144	**0.8882**	0.9006	0.6318	0.5965	0.6152
	dit	**0.3017**	**0.2076**	**0.2588**	0.2671	0.1758	0.2117	0.8990	**0.8527**	0.8760	**0.6711**	**0.5959**	**0.6438**
	vgt	**0.2714**	**0.1736**	**0.2264**	**0.2644**	**0.1654**	**0.2041**	**0.8928**	**0.8543**	**0.8691**	**0.7407**	**0.6773**	**0.7246**

Table 14: DP performance by Chunking Method (top-$k = 1 \sim 4$ average)

Chunking Method	Embedding	DUDE			MPVQA			CUAD			MOAMOB		
		R	P	nDCG	R	P	nDCG	R	P	nDCG	R	P	nDCG
Length chunking	bge	0.2655	0.1646	0.2097	0.2810	0.1742	0.2118	**0.9152**	0.8778	0.8960	0.6504	0.5677	0.6162
	e5	0.2644	0.1602	0.2178	0.2469	0.1485	0.1855	0.8915	0.8491	0.8651	0.6467	0.5672	0.6175
	BM25	0.2586	0.1810	0.2222	0.2289	**0.1534**	**0.1827**	0.8965	0.8342	0.8717	0.6415	0.5679	0.6290
Semantic chunking	bge	0.0753	0.0433	0.0619	0.0876	0.0511	0.0653	0.8663	0.8046	0.8361	0.2956	0.2309	0.2824
	e5	0.1184	0.0665	0.0955	0.1152	0.0650	0.0839	0.8380	0.7628	0.8035	0.4126	0.2966	0.3681
	BM25	0.0929	0.0548	0.0752	0.0789	0.0409	0.0547	0.6009	0.4482	0.5148	0.1130	0.0575	0.0853
LumberChunker	bge	0.2615	0.1651	0.2157	0.2456	0.1456	0.1811	0.9055	0.8699	0.8852	0.6378	0.5523	0.6013
	e5	0.1983	0.1132	0.1564	0.1892	0.1065	0.1370	0.8937	0.8441	0.8660	0.6100	0.5230	0.5680
	BM25	0.2587	0.1817	0.2236	0.2108	0.1373	0.1647	**0.9102**	**0.8589**	**0.8889**	0.5911	0.4862	0.5383
Perplexity chunking	bge	0.2673	0.1695	0.2213	0.2561	0.1554	0.1938	0.8858	0.8471	0.8635	0.6418	0.5498	0.6073
	e5	0.2084	0.1204	0.1661	0.1844	0.1030	0.1316	0.8918	0.8485	0.8682	0.6026	0.5138	0.5613
	BM25	0.2526	0.1779	0.2188	0.2073	0.1371	0.1632	0.8831	0.8227	0.8492	0.6074	0.5086	0.5668
Structure-based chunking	bge	0.2611	0.1672	0.2176	0.2560	0.1570	0.1960	0.9003	0.8559	0.8757	0.6211	0.5236	0.5823
	e5	0.1723	0.0977	0.1377	0.1615	0.0901	0.1166	0.8681	0.8047	0.8378	0.5574	0.4734	0.5237
	BM25	0.2323	0.1700	0.2033	0.1932	0.1219	0.1447	0.8850	0.8327	0.8606	0.4848	0.4016	0.4386
MultiDocFusion	bge	**0.3222**	**0.2202**	**0.2771**	**0.3112**	**0.2070**	**0.2501**	0.9125	**0.8815**	**0.8967**	**0.6808**	**0.6276**	**0.6613**
	e5	**0.2762**	**0.1797**	**0.2318**	**0.2682**	**0.1673**	**0.2071**	**0.8952**	**0.8581**	**0.8723**	**0.6663**	**0.6070**	**0.6425**
	BM25	**0.2798**	**0.2005**	**0.2425**	**0.2320**	0.1533	0.1821	0.8986	0.8555	0.8767	**0.7560**	**0.6859**	**0.7325**

Table 15: Performance comparison by Embedding and Chunking Method (top-$k = 1 \sim 4$ average)

Top-k	Chunking Method	MPVQA			DUDE			CUAD			MOAMOB		
		ANLS	ROUGE-L	METEOR	ANLS	ROUGE-L	METEOR	ANLS	ROUGE-L	METEOR	ANLS	ROUGE-L	METEOR
1	Length chunking	0.1299	0.0679	0.0859	0.1573	0.1281	0.1722	0.2675	**0.1761**	0.1616	0.2498	0.0739	0.1054
	Semantic chunking	0.1256	0.0635	0.0794	0.1527	0.1126	0.1420	0.2611	0.1544	0.1409	0.2444	0.0737	0.1009
	LumberChunker	0.1394	0.0737	0.0873	0.1614	0.1264	0.1642	0.2690	0.1698	**0.1657**	0.2588	0.0748	0.1189
	Perplexity chunking	0.1254	0.0544	0.0676	0.1683	0.1357	0.1689	0.2592	0.1629	0.1457	0.2530	0.0868	0.1182
	Structure based Chunking	0.1440	0.0858	0.1103	0.1615	0.1217	0.1527	0.2489	0.1569	0.1592	0.2477	**0.0892**	0.1108
	MultiDocFusion	**0.1473**	**0.1021**	**0.1335**	**0.1726**	**0.1512**	**0.2001**	**0.2692**	0.1739	0.1578	**0.2642**	0.0826	**0.1239**
4	Length chunking	0.1398	0.0966	0.1408	0.1611	0.1444	0.1988	0.2495	0.1593	0.1708	0.2495	0.0906	0.1176
	Semantic chunking	0.1332	0.0805	0.0978	0.1548	0.1261	0.1657	0.2574	0.1438	0.1526	0.2465	0.0955	0.1076
	LumberChunker	0.1307	0.0769	0.0993	0.1531	0.1284	0.1752	0.2623	0.1562	0.1643	0.2483	0.0947	0.1144
	Perplexity chunking	0.1344	0.0751	0.0950	0.1653	0.1390	0.1855	0.2690	0.1662	0.1591	0.2534	0.0919	0.1197
	Structure based Chunking	0.1537	0.0980	0.1278	0.1751	0.1489	0.1921	0.2507	0.1543	0.1590	**0.2524**	**0.1065**	0.1120
	MultiDocFusion	**0.1615**	**0.1316**	**0.1850**	**0.1859**	**0.1692**	**0.2285**	**0.2783**	**0.1785**	**0.1721**	0.2550	0.1005	**0.1274**

Table 16: Average generation performance (*ANLS*, *ROUGE-L*, *METEOR*) of six chunking strategies on MPVQA, DUDE, CUAD, and MOAMOB datasets, separated by top-k settings ($k = 1$ and $k = 4$). Best scores for each metric and dataset are highlighted in **bold**.