PDF Ingestion Strategies for Retrieval-Augmented Generation: A Comparative Analysis of Chunking Methodologies

Calvin Laughlin
calvin3@stanford.edu

June 3, 2025

Abstract

Retrieval-Augmented Generation (RAG) systems have emerged as a powerful paradigm for enhancing large language models with external knowledge sources. However, the effectiveness of these systems critically depends on how documents are processed and chunked before indexing. This paper presents a comprehensive comparative analysis of three distinct PDF ingestion strategies: structured chunking (Reducto-style), fixed-size rolling windows, and sentence-level overlap chunking. I implement a modular evaluation framework that assesses these strategies across multiple dimensions including retrieval accuracy, answer quality, and factual consistency. Experimental results across six diverse documents reveal surprising findings: fixed-size rolling windows achieve superior performance with 28.5% retrieval accuracy and 53.1% answer quality, outperforming structured chunking (9.5% retrieval accuracy, 13.3% answer quality) and sentence-level overlap (14.2% retrieval accuracy, 21.5% answer quality). This challenges conventional assumptions about the benefits of preserving document structure in chunking strategies. I also present an integration of CLIP-based image processing for multimodal PDF understanding. This research provides empirical evidence for chunk size optimization trade-offs and establishes a foundation for future work in intelligent document ingestion.

1 Introduction

Retrieval-Augmented Generation (RAG) has revolutionized how large language models (LLMs) interact with external knowledge sources, enabling them to provide more accurate, up-to-date, and contextually relevant responses [?]. The fundamental premise of RAG lies in its two-stage architecture: first retrieving relevant information from a knowledge base, then using this context to generate informed responses. However, the effectiveness of this paradigm is intrinsically linked to how documents are processed and segmented before being indexed in the retrieval system.

The challenge of document chunking in RAG systems extends beyond simple text segmentation. Different chunking strategies can dramatically impact the quality of retrieved context and, consequently, the accuracy and relevance of generated responses. While conventional wisdom suggests that fixed-size chunking approaches break semantic boundaries and lead to context fragmentation, our empirical findings challenge this assumption, demonstrating that simpler fixed-size approaches can outperform more sophisticated structure-preserving methods under certain conditions.

Recent advances in document understanding have introduced more sophisticated approaches to text segmentation. Structured chunking methods, exemplified by systems like Reducto, attempt to preserve the semantic organization of documents by identifying headers, paragraphs, and sections [?]. These approaches leverage document layout analysis to create chunks that maintain logical coherence while respecting the original document's hierarchical structure.

This paper addresses a critical gap in the current literature by providing a systematic comparison of PDF ingestion strategies for RAG systems. This research is motivated by three key observations: (1) the lack of comprehensive empirical studies comparing different chunking methodologies, (2) the absence of standardized evaluation frameworks for assessing chunking quality in RAG contexts, and (3) the growing need for multimodal document processing capabilities that can handle both text and visual elements.

1.1 Research Contributions

This work makes several contributions to the field of retrieval-augmented generation:

1. **Comprehensive Comparative Analysis**: I implement and evaluate three distinct chunking strategies using a unified framework, providing the first systematic comparison of their effectiveness in RAG applications.
2. **Multi-dimensional Evaluation Framework**: I develop a robust evaluation methodology that assesses chunking strategies across retrieval accuracy, answer quality, and factual consistency dimensions.
3. **Modular Implementation**: The system is designed as a proper Python package with testing and documentation, making it suitable for both research and production environments.
4. **Multimodal Integration**: We present a novel approach for incorporating CLIP-based image processing into PDF ingestion pipelines, enabling better understanding of documents containing visual elements.
5. **Empirical Insights**: Our experimental results reveal counterintuitive findings about chunk size optimization and provide evidence-based insights into the trade-offs between document structure preservation and retrieval effectiveness.

1.2 Paper Organization

The remainder of this paper is organized as follows: Section 2 reviews related work in retrieval-augmented generation and document chunking strategies. Section 3 describes the methodology, including the implementation of different chunking approaches and the evaluation framework. Section 4 presents the experimental setup and results. Section 5 discusses the implications of the findings and identifies areas for future research. Finally, Section 6 concludes with a summary of the contributions and recommendations for practitioners.

2 Related Work

2.1 Retrieval-Augmented Generation

The concept of retrieval-augmented generation was first formalized by **?**, who demonstrated that combining parametric knowledge stored in language model weights with non-parametric knowledge retrieved from external sources could significantly improve performance on knowledge-intensive tasks. This seminal work established the foundation for a new paradigm in natural language processing that has since been adopted across numerous applications.

Recent advances in RAG systems have focused on improving various components of the pipeline. **?** introduced dense passage retrieval (DPR), which uses dense vector representations for more effective document retrieval compared to traditional sparse methods like TF-IDF or BM25. **?** proposed Fusion-in-Decoder (FiD), which processes multiple retrieved passages jointly to generate more coherent responses.

The evaluation of RAG systems has emerged as a critical research area. **?** conducted a comprehensive study of best practices in RAG evaluation, identifying key metrics and methodologies for assessing system performance. Their work highlighted the importance of considering both retrieval quality and generation quality when evaluating RAG systems, a principle that guides our evaluation framework.

2.2 Document Chunking Strategies

Document chunking has been studied extensively in the context of information retrieval and text processing. Traditional approaches have focused on simple heuristics such as fixed-size windows or sentence-based segmentation [**?**]. However, these methods often fail to capture the semantic structure of documents, leading to suboptimal retrieval performance.

Recent work has explored more sophisticated chunking strategies that leverage document structure and semantic understanding. **?** introduced Meta-Chunking, a framework that learns optimal segmentation points through a dual strategy combining text segmentation and semantic completion. Their approach demonstrates the potential for learned chunking strategies to outperform traditional heuristic methods.

The importance of preserving document structure in chunking has been emphasized by several recent studies. **?** showed that semantic chunking, which identifies breaking points based on embedding distances between sentences, can maintain meaningful and coherent chunks. However, their analysis also revealed that the computational overhead of semantic chunking may not always justify the performance gains.

2.3 Multimodal Document Understanding

The integration of visual information in document processing has gained significant attention with the advent of vision-language models. CLIP (Contrastive Language-Image Pre-training) [**?**] has emerged as a powerful tool for understanding visual content in documents. Recent work has explored the application of CLIP and similar models to document analysis tasks, including figure classification and visual question answering.

? demonstrated the benefits of layout-aware document processing in RAG systems, showing that preserving spatial relationships between text and visual elements can improve retrieval accuracy. Their work provides a foundation for our multimodal integration approach.

2.4 Evaluation Metrics for RAG Systems

The evaluation of RAG systems requires careful consideration of multiple dimensions. Traditional information retrieval metrics such as precision, recall, and F1-score provide insights into retrieval quality but may not capture the nuanced effects on generation quality [**?**].

Recent work has proposed specialized metrics for RAG evaluation. Faithfulness measures whether generated answers are grounded in the retrieved context, while relevance assesses how well the retrieved information addresses the user query [**?**]. Answer quality metrics evaluate the completeness, coherence, and accuracy of generated responses.

3 Methodology

3.1 Chunking Strategies

I implement and evaluate three distinct chunking strategies, each representing a different approach to document segmentation:

3.1.1 Structured Chunking

The structured chunking implementation attempts to preserve the semantic structure of documents by identifying headers, paragraphs, and sections. The algorithm uses regular expression patterns to detect section boundaries and creates chunks that maintain the document's logical organization.

Algorithm 1 Structured Chunking Algorithm

```
Input: Document text $T$, header pattern $P$
Output: List of structured chunks $C$
Initialize $chunks \leftarrow []$, $current\_chunk \leftarrow []$, $current\_header \leftarrow$ "Introduction"
for each line $l$ in $T.split('\n')$ do
    if $l$ matches pattern $P$ then
        if $current\_chunk$ is not empty then
            Add chunk to $chunks$ with header $current\_header$
            Reset $current\_chunk \leftarrow []$
        end if
        Update $current\_header \leftarrow l.strip()$
    end if
    if $l.strip()$ is not empty then
        Add $l$ to $current\_chunk$
    end if
end for
Add final chunk if $current\_chunk$ is not empty
return $chunks$
```

This approach creates chunks that respect document hierarchy, ensuring that related content remains grouped together. Each chunk includes metadata about its position in the document structure, enabling more sophisticated retrieval strategies.

3.1.2 Fixed-Size Rolling Window

The fixed-size rolling window strategy creates overlapping chunks of predetermined size with configurable stride. This approach ensures consistent chunk sizes, which can be beneficial for LLMs with fixed context windows.

The algorithm divides text into tokens and creates chunks using a sliding window approach:

$$chunk_i = tokens[i \times stride : i \times stride + window\_size] \tag{1}$$

where i represents the chunk index, $stride$ determines the overlap between consecutive chunks, and $window\_size$ defines the maximum number of tokens per chunk.

While this method provides predictable chunk sizes, it may break semantic boundaries, potentially splitting paragraphs or sentences across different chunks.

3.1.3 Sentence-Level Overlap Chunking

Sentence-level overlap chunking respects sentence boundaries while creating chunks with a specified number of sentences and overlap. This approach preserves the integrity of sentences, which can improve the coherence of retrieved content.

The algorithm first segments the text into sentences using simple punctuation-based splitting, then creates overlapping chunks:

$$chunk_i = sentences[i \times (chunk\_size - overlap) : i \times (chunk\_size - overlap) + chunk\_size] \tag{2}$$

The overlap parameter helps maintain context across chunk boundaries, potentially improving retrieval for queries that span multiple chunks.

3.2 Retrieval System

The retrieval system uses sentence transformers for embedding generation and supports both FAISS and NumPy implementations for vector similarity search. The system is designed to be modular and can be easily extended to support different embedding models or similarity search implementations.

3.2.1 Embedding Generation

I use the sentence-transformers library with the all-MiniLM-L6-v2 model as our default embedding model. This model provides a good balance between performance and computational efficiency for document retrieval tasks.

Embeddings are L2-normalized to enable cosine similarity computation via dot product:

$$similarity(q, d) = \frac{q \cdot d}{||q||_2 \times ||d||_2} \tag{3}$$

where q represents the query embedding and d represents a document chunk embedding.

3.2.2 Similarity Search

For efficient approximate nearest neighbor search, we use FAISS when available, with a pure NumPy implementation as fallback. The FAISS implementation uses IndexFlatIP for exact inner product search on normalized vectors.

3.3 Multimodal Integration with CLIP

To handle documents containing visual elements, I integrate CLIP-based image processing into the ingestion pipeline. This component classifies images into predefined categories (bar chart, line graph, pie chart, etc.) and extracts text using Optical Character Recognition (OCR).

Algorithm 2 CLIP-based Image Processing

```
Input: Image bytes $I$, context text $C$
Output: Structured image description $D$
Load CLIP model and processor
Convert $I$ to PIL image format
Generate image features using CLIP
Compute similarity with predefined label embeddings
$image\_type \leftarrow$ label with highest similarity
$ocr\_text \leftarrow$ extract text using Tesseract OCR
Construct description $D$ combining type, confidence, and OCR text
return $D$
```

The processed image information is converted to structured text that can be ingested alongside regular document chunks, enabling multimodal retrieval capabilities.

3.4 Evaluation Framework

The evaluation framework assesses chunking strategies across three key dimensions:

3.4.1 Retrieval Accuracy

I measure retrieval quality using standard information retrieval metrics:

$$Precision = \frac{|relevant \cap retrieved|}{|retrieved|} \tag{4}$$

$$Recall = \frac{|relevant \cap retrieved|}{|relevant|} \tag{5}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{6}$$

These metrics assess how effectively each chunking strategy enables the retrieval of relevant information for given queries.

3.4.2 Answer Quality

Answer quality is evaluated using LLM-based assessment across three dimensions:

- **Relevance**: How well the answer addresses the question
- **Completeness**: The thoroughness and detail of the response
- **Coherence**: The logical flow and readability of the answer

Each dimension is scored on a scale of 0-5, with scores aggregated to provide an overall answer quality metric.

3.4.3 Factual Consistency

Factual consistency measures whether generated answers are grounded in the retrieved context and free from hallucinations. We use a combination of automated metrics and human evaluation to assess this dimension.

4 Experimental Setup and Results

4.1 Dataset and Experimental Configuration

The experiments are conducted using six diverse documents spanning multiple domains and scales: (1) a PDF ingestion strategies research paper (5K characters), (2) Les Misérables classic novel (3.25M characters), (3) a neural networks research paper (50K characters), (4) an NLP transformers paper (60K characters), (5) a computer vision paper (55K characters), and (6) a reinforcement learning paper (58K characters). This diversity in both content type and document size ensures comprehensive evaluation across different scenarios, from short technical documents to massive literary works.

4.1.1 Hyperparameter Configuration

The chunking strategies are configured with the following parameters:

- **Fixed-Size Rolling Window**: Window size = 500 tokens, Stride = 250 tokens
- **Sentence-Level Overlap**: Chunk size = 10 sentences, Overlap = 3 sentences
- **Structured Chunking**: Dynamic sizing based on document structure

These parameters are configurable through environment variables, allowing for easy tuning without modifying the source code.

4.2 Performance Comparison

Table 1 presents the performance comparison across all three chunking strategies based on comprehensive evaluation across six diverse documents with six evaluation queries each (36 total evaluations). The results demonstrate significant and counterintuitive differences in performance across all evaluated metrics.

Table 1: Performance Comparison of Chunking Strategies (6 Documents, 36 Total Evaluations)

Strategy	**Retrieval Accuracy**	**Answer Quality**	**Factual Consistency**	**Avg. C**
Fixed-Size Rolling Window	0.285	0.531	0.244	383
Sentence Overlap Chunking	0.142	0.215	0.163	727
Structured Chunking	0.095	0.133	0.102	106

4.3 Key Findings

The experimental results reveal several important and counterintuitive insights:

1. **Fixed-size rolling windows consistently outperform other strategies** across all metrics, achieving the highest scores in retrieval accuracy (28.5%), answer quality (53.1%), and factual consistency (24.4%). This challenges conventional assumptions about the benefits of structure preservation.

2. **Structured chunking unexpectedly performs poorly**, achieving only 9.5% retrieval accuracy and 13.3% answer quality. The high number of small chunks (average 106.2 chunks per document) appears to fragment content excessively, reducing retrieval effectiveness.

3. **Chunk quantity impacts performance significantly**: Fixed-size windows create moderate numbers of chunks (383.8 average), while sentence overlap chunking creates too many chunks (727.5 average), and structured chunking creates too few meaningful chunks relative to content density.

4. **Content coverage trumps structural preservation**: Larger chunks with broader content coverage (fixed-size windows) enable better retrieval than precisely structured but fragmented content, suggesting that retrieval effectiveness depends more on content density than semantic boundaries.

5. **Document scale affects strategy performance**: The inclusion of Les Misérables (3.25M characters) reveals that different strategies scale differently, with structured chunking creating 465 chunks, fixed-size creating 2,275 chunks, and sentence overlap creating 4,294 chunks.

4.4 Qualitative Analysis

Beyond quantitative metrics, the qualitative analysis reveals important differences in the behavior of each strategy:

Fixed-Size Rolling Windows provide consistent content coverage per chunk, ensuring that retrieved chunks contain sufficient context for answering queries. While they may split sentences, the overlap mechanism (250-token stride) maintains reasonable continuity between chunks.

Structured Chunking creates semantically coherent chunks but often produces chunks that are too small or too specialized, reducing the likelihood that any single chunk contains comprehensive information relevant to diverse queries.

Sentence-Level Overlap maintains sentence integrity but creates an excessive number of small chunks, diluting the content density and making it statistically less likely that relevant information will be retrieved in the top-k results.

5 Discussion

5.1 Implications for RAG System Design

Our findings have several important implications for the design of RAG systems:

5.1.1 Rethinking Document Structure Preservation

Contrary to conventional wisdom, our results demonstrate that preserving document structure does not necessarily improve RAG performance. Fixed-size rolling windows, despite potentially breaking semantic boundaries, achieve superior performance by ensuring adequate content coverage per chunk. This suggests that content density and coverage may be more important than structural coherence for retrieval tasks.

5.1.2 The Content Coverage Hypothesis

Our findings support a "content coverage hypothesis": retrieval effectiveness depends more on ensuring that individual chunks contain sufficient diverse content to match varied query types than on preserving precise semantic boundaries. Fixed-size windows naturally achieve this by maintaining consistent chunk sizes with meaningful overlap.

5.1.3 Practical Implementation Considerations

Fixed-size rolling windows offer the best combination of performance and implementation simplicity, achieving 28.5% retrieval accuracy with straightforward implementation. This makes them particularly attractive for production RAG systems where simplicity, predictability, and performance all matter.

5.1.4 Domain-Specific Considerations

Analysis suggests that the optimal chunking strategy may depend on document characteristics and domain requirements. Highly structured documents (academic papers, technical reports) benefit most from structured chunking, while less formal documents may see smaller performance differences between strategies.

5.2 Multimodal Integration Benefits

The integration of CLIP-based image processing provides several benefits:

- **Enhanced Context Understanding**: Visual elements often contain critical information that complements textual content
- **Improved Retrieval Coverage**: Queries related to visual content can be answered more effectively
- **Structured Metadata**: Image classification and OCR provide structured metadata that can be leveraged for advanced retrieval strategies

5.3 Limitations and Future Work

This study has several limitations that present opportunities for future research:

5.3.1 Limited Document Types

While the dataset includes diverse document types, it is primarily focused on English-language documents. Future work should explore the generalizability of the findings to other languages and document formats.

5.3.2 Computational Cost Analysis

The evaluation focuses on effectiveness metrics but does not comprehensively analyze the computational costs associated with different chunking strategies. Future research should include detailed performance profiling and cost-benefit analysis.

5.3.3 Dynamic Chunking Strategies

The current implementation uses static chunking strategies. Future work could explore adaptive approaches that adjust chunking parameters based on document characteristics or query patterns.

5.3.4 Advanced Multimodal Integration

While the CLIP integration provides basic image understanding, more sophisticated multimodal approaches could leverage recent advances in vision-language models for deeper document understanding.

5.4 Recommendations for Practitioners

Based on the findings, I provide the following recommendations for practitioners implementing RAG systems:

1. **Start with fixed-size rolling windows** as the default chunking strategy due to their superior performance and implementation simplicity
2. **Use 500-token windows with 250-token stride** as evidenced by our experimental configuration achieving the best results
3. **Avoid over-fragmentation** by ensuring chunks contain sufficient content density rather than optimizing for semantic boundaries
4. **Implement comprehensive evaluation frameworks** that assess multiple dimensions of system performance across diverse document types
5. **Consider document scale** when choosing strategies, as performance characteristics change dramatically with document size (as demonstrated by the Les Misérables evaluation)
6. **Invest in multimodal capabilities** for documents containing significant visual content

6 Future Work

Several promising directions emerge from the research:

6.1 Advanced Evaluation Metrics

Future work should develop more sophisticated evaluation metrics that capture nuanced aspects of RAG performance, including:

- Context coherence across chunk boundaries
- Semantic similarity between retrieved and ideal contexts
- User satisfaction and task completion rates

6.2 Learned Chunking Strategies

Machine learning approaches to chunking could potentially outperform rule-based methods by learning optimal segmentation strategies from data. This could include:

- Reinforcement learning approaches that optimize chunking for downstream task performance
- Neural segmentation models trained on document structure data
- Transfer learning approaches that adapt chunking strategies to new domains

6.3 Real-time Adaptive Chunking

Dynamic systems that adjust chunking strategies based on query patterns and user feedback could provide personalized optimization:

- Query-aware chunking that considers likely information needs
- Feedback-driven optimization that learns from user interactions
- Multi-objective optimization balancing multiple performance dimensions

6.4 Large-Scale Empirical Studies

Comprehensive studies across diverse domains, languages, and document types would strengthen the generalizability of our findings:

- Cross-lingual evaluation of chunking strategies
- Domain-specific optimization studies
- Large-scale user studies measuring real-world performance

7 Conclusion

This paper presents a comparative analysis of PDF ingestion strategies for retrieval-augmented generation systems. Through systematic evaluation of structured chunking, fixed-size rolling windows, and sentence-level overlap chunking across six diverse documents, this research reveals counterintuitive findings that challenge conventional assumptions about the benefits of document structure preservation.

Key contributions include: (1) the first systematic comparison of chunking strategies in RAG contexts with comprehensive multi-document evaluation, (2) a robust evaluation framework assessing multiple performance dimensions, (3) a modular implementation suitable for both research and production use, and (4) integration of CLIP-based multimodal processing.

The experimental results demonstrate the unexpected superiority of fixed-size rolling windows, achieving 28.5% retrieval accuracy and 53.1% answer quality, significantly outperforming structured chunking (9.5% retrieval accuracy, 13.3% answer quality) and sentence-level overlap (14.2% retrieval accuracy, 21.5% answer quality). These findings challenge conventional wisdom and provide evidence-based guidance for practitioners implementing RAG systems.

As RAG systems continue to evolve and find applications across diverse domains, the importance of effective document ingestion strategies will only grow. This work contributes to this critical area by providing both theoretical insights and practical tools for building more effective RAG systems.

The modular design of the implementation and the comprehensive evaluation framework we present enable future researchers to build upon the work and explore new directions in document chunking and multimodal integration. We believe this research represents an important step toward more intelligent and effective retrieval-augmented generation systems.