Comparison of Chunking techniques Across Diverse Document Types in NLP Retrieval Tasks

Shruti Jaiswal*
Data and Applied Sciences
Bosch Global Software Technologies
Bangalore, India
shruti.jaiswal@in.bosch.com

Priyank Bisht*
Data and Applied Sciences
Bosch Global Software Technologies
Bengaluru, India
bisht_priyank@yahoo.com

Krity Kansara
Data and Applied Sciences
Bosch Global Software Technologies
Bengaluru, India
krity.kansara@in.bosch.com

MSH Shankar Datta
Data and Applied Sciences
Bosch Global Software Technologies
Bengaluru, India
MSH.ShankarDatta@in.bosch.com

2025 International Conference on Responsible, Generative and Explainable AI (ResGenXAI) | 979-8-3315-3588-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/RESGENXAI64788.2025.11344045

***Abstract*—In retrieval-augmented natural language processing systems, how information is segmented—or chunked—plays a critical role in determining the accuracy, efficiency, and robustness of downstream retrieval performance. Despite the increasing reliance on retrieval-based architectures such as open-domain question answering and RAG (Retrieval-Augmented Generation), there remains limited empirical work comparing different chunking strategies in a systematic, task-specific manner. In this paper, we present a comprehensive comparative study of widely used chunking methods—fixed-size chunking, sentence-based chunking, recursive chunking and semantic similarity-based chunking—evaluated within the context of retrieval performance across diverse document types. We analyze each strategy across multiple metrics including Precision, Recall, MRR(Mean Reciprocal Rank), and chunking efficiency, emphasizing how performance varies significantly with the structural and semantic characteristics of different document types. Our findings highlight key trade-offs between retrieval accuracy and chunking granularity, revealing that while finer-grained semantic chunking often improves precision, it may also introduce computational overhead, especially in certain document types. These insights provide practical guidelines for developers and researchers building retrieval-centric NLP pipelines and lay the groundwork for future work on adaptive and task-aware chunking mechanisms.**

***Index Terms*—Document Chunking, Information Retrieval, Natural Language Processing, Semantic Chunking, Retrieval-Augmented Generation, Document Segmentation**

I. Introduction

The integration of retrieval mechanisms into natural language processing (NLP) workflows has become a key driver of recent advancements in tasks such as open-domain question answering (ODQA), document-based reasoning, and retrieval-augmented generation (RAG). In these systems, large collections of text are preprocessed into smaller units, or chunks, which are then embedded and stored in a vector database. At inference time, relevant chunks are retrieved in response to a query and either used directly or passed to a language model for generation or decision-making. This retrieval-augmented design improves factual accuracy, reduces hallucination, and enables access to dynamic external knowledge sources.

Despite the centrality of chunking in this pipeline, the NLP community has not reached a consensus on the most effective chunking strategy. Different approaches—ranging from fixed-length token chunking to semantically aware segmentation—are often applied ad hoc, without systematic evaluation of their downstream effects on retrieval performance. As model size and context windows expand, understanding the trade-offs between chunking granularity, semantic coherence, retrieval accuracy, and computational efficiency becomes increasingly critical.

The performance of retrieval-based NLP systems is highly sensitive to how input documents are segmented into chunks. Poorly chosen chunking strategies can lead to irrelevant retrievals, redundancy, and increased inference costs. However, a lack of empirical benchmarking across chunking strategies for retrieval tasks has left developers without clear guidance on which methods work best under varying conditions.

This paper aims to address this gap by systematically analyzing and comparing the impact of different chunking strategies on information retrieval performance. Specifically, we seek to:

- Evaluate how different chunking methods affect retrieval quality across representative datasets.
- Quantify the trade-offs between retrieval accuracy, chunking granularity, and system efficiency, considering how these trade-offs vary by document structure and content.
- Identify practical guidelines for choosing chunking strategies tailored to retrieval-based NLP use cases, emphasizing that optimal selection often depends on the underlying document characteristics.

The key contributions of this paper are:

- A unified evaluation framework for comparing chunking

*These authors have equal contribution

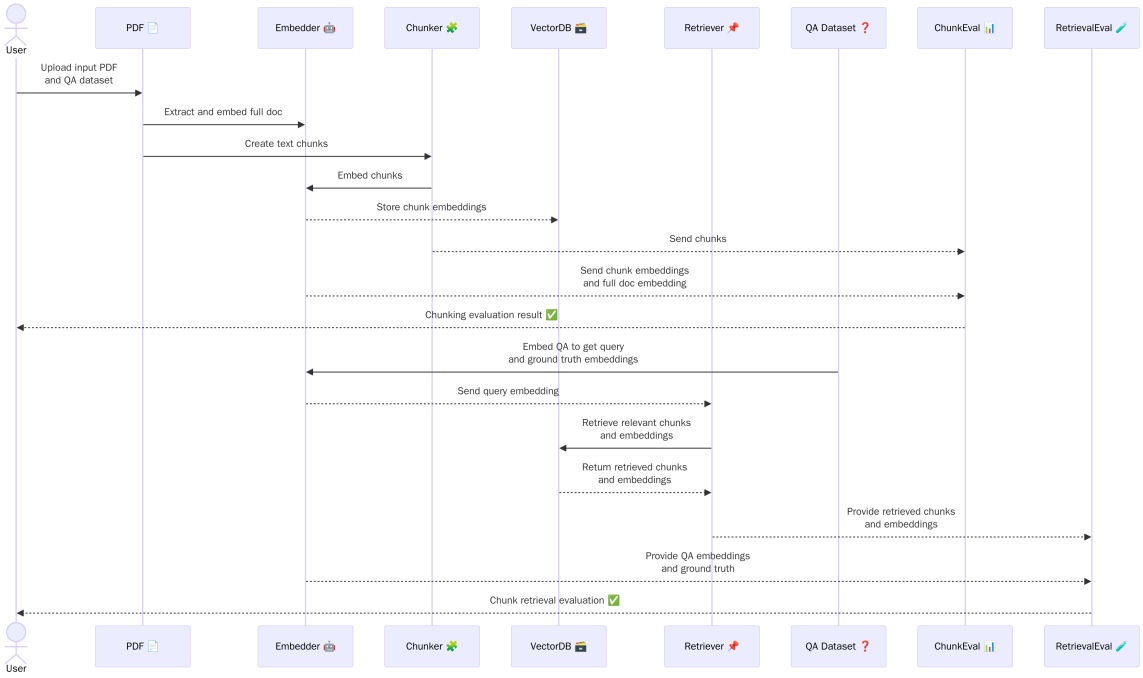

Fig. 1. System architecture for chunking and retrieval evaluation pipeline

strategies in retrieval-based NLP.

- A comprehensive empirical study of four prominent chunking methods: fixed-size, sentence-based, recursive chunking and semantic similarity-based chunking—across multiple document types.
- Quantitative and qualitative insights into how chunking affects retrieval performance, using metrics such as Precision, Recall, MRR, and computational overhead, with an emphasis on how these metrics vary with document characteristics.
- A set of actionable recommendations for selecting chunking strategies in real-world applications.

The rest of this paper is organized as follows:

- Section 2 reviews related work in retrieval-based NLP systems and chunking strategies.
- Section 3 details the experimental setup, including datasets, chunking methods, and evaluation metrics.
- Section 4 presents and analyzes the experimental results.
- Section 5 discusses implications, limitations, and opportunities for future research.
- Section 6 concludes the paper.

II. Related Work

The effectiveness of chunking strategies has a direct influence on the performance of retrieval-based NLP systems, particularly in open-domain question answering, document retrieval, and retrieval-augmented generation. As large language models increasingly rely on external knowledge sources, it becomes essential to understand how chunking affects retrieval precision, recall, and efficiency.

A. Retrieval-Augmented Systems

Retrieval-augmented language models integrate information retrieval modules into neural architectures to extend context beyond the model's input length. For instance, REALM [1] demonstrated the importance of pre-training language models with retrieval modules. Similarly, RAG [2] showed that document retrieval quality significantly impacts downstream generation quality. Both approaches required the corpus to be divided into manageable "chunks," usually based on token count, highlighting the importance of chunk design.

B. Evolution of Chunking Strategies

Initial systems favored fixed-length token chunking, where documents were segmented into uniform token blocks (e.g., 100 or 512 tokens). While simple, this method often caused semantic boundaries to be broken, reducing the effectiveness of embeddings [3].

To preserve coherence, sentence-based and paragraph-based chunking were introduced. Lee et al. [4] employed sentence-aligned chunks for better context retention in ORQA. Sliding window chunking with overlaps [5] improved recall by increasing coverage of relevant spans, though at the cost of redundancy and computational overhead.

Recently, semantic-aware chunking methods have emerged. These approaches use embedding similarity (e.g., via SBERT or MiniLM) to determine logical chunk boundaries [6], [7]. Semantic chunking improves retrieval quality by aligning chunk boundaries with conceptual completeness rather than arbitrary size constraints.Beyond retrieval, similar design considerations for explainability and robustness have been explored in health imaging [8], spectroscopy-based classification [9],

TABLE I
EVALUATION METRICS USED FOR CHUNKING STRATEGY COMPARISON

Metric	Description
Precision & Recall	Computed using cosine similarity (≥ 0.7). True: $\cos(\text{chunk}, \text{GT}) \geq 0.7$; Predicted: $\cos(\text{chunk}, \text{query}) \geq 0.7$; Precision = TP / (TP + FP), Recall = TP / (TP + FN).
Mean Reciprocal Rank (MRR)	Average reciprocal rank of the first relevant chunk retrieved for each query. Higher MRR reflects better rank ordering.
NDCG	Normalized Discounted Cumulative Gain at top-5 ranks. Measures ranked retrieval quality with higher weight for top positions.
ss2fd (Semantic Similarity to Full Document	Average cosine similarity between each chunk and the full-document embedding. A higher ss2fd indicates broader, overlapping chunks; a lower value suggests more focused, specific chunks.
Semantic Relevancy to Ground Truth (SRGT)	Average cosine similarity between retrieved chunks and the ground-truth answer embedding. Indicates alignment with correct answer.
Retrieval Token Cost	Average total number of tokens in top-5 retrieved chunks per query. Lower values indicate more efficient retrieval for LLMs.
QCS(Query–Chunk Similarity)	Average cosine similarity between the query embedding and top-5 retrieved chunks. Ensure semantically aligned retrievals.
Chunking Time	Average preprocessing time required to chunk each document. Reflects computational cost of the strategy.

and Responsible AI frameworks for fairness-aware computer vision [10] and numeric data classification [11]. These parallels underscore that careful segmentation—whether in text, images, or signal data—directly affects both performance and interpretability.

C. Comparative Evaluations and Benchmarks

Despite their significance, chunking strategies are rarely evaluated in isolation. Most studies assess end-to-end systems, making it difficult to quantify the chunking component's impact. The BEIR benchmark [12] enables zero-shot evaluation of retrieval systems but uses pre-processed corpora with fixed chunking. Longformer [13] and BigBird [14] partially address chunking through attention window manipulation rather than preprocessing.

Other efforts like FiD [15] and ColBERT [16] showed that retrieval performance is sensitive to chunk granularity, but lacked systematic comparisons of chunking strategies themselves.

To the best of our knowledge, there has been no unified study that systematically compares different chunking strategies—fixed-length, sentence-based, recursive, and semantic-based—within a controlled retrieval setup using consistent retrievers and evaluation metrics.

III. METHODOLOGY

A. System Architecture

The overall system is a pipeline that takes a PDF document and a set of question–answer (QA) pairs as input, then produces a chunked document index and evaluates its performance on retrieving answers as shown in Fig. 1.

The user uploads a PDF and an associated QA dataset. The PDF is parsed and embedded into a full-document vector using a pre-trained language model (e.g., a transformer-based sentence encoder [17]). The text is then chunked using a specified strategy, and each chunk is embedded individually. These chunk embeddings are stored in a vector database (Faiss) for efficient similarity-based retrieval [18].

We also compute the semantic similarity between each chunk and the full document (ss2fd) to evaluate chunking quality. For each query in the QA dataset, the query is embedded and passed to the Retriever, which searches the vector database to return the top-k most similar chunks (we use $k = 5$), filtered by a cosine similarity threshold of 0.70.

Ground-truth answers are also embedded, and retrieval performance is evaluated by comparing retrieved chunks against these ground-truth vectors using metrics such as precision, recall, and MRR.

B. Chunking Strategies

1) Fixed-Size Chunking with Overlap: Fixed-size chunking divides the text into equal-length token segments using a sliding window, so that each segment overlaps the previous one by a constant number of tokens.

2) Sentence-Based Chunking: This strategy splits text at sentence boundaries, grouping complete sentences into chunks up to a token limit (250 or 500) without overlap. It preserves semantic coherence but often yields many small chunks, especially in documents with short sentences.

3) Recursive (Structure-Preserving) Chunking: This method uses document structure (e.g., sections, paragraphs) to define chunk boundaries. Large units are recursively split by sentences if they exceed the token limit, while short sections are kept intact. It preserves logical groupings and avoids unnecessary splits.

4) Semantic Chunking: Semantic chunking uses sentence embeddings and clustering to group semantically similar sentences into coherent chunks. Sentences are embedded using a pretrained model (e.g., Sentence-BERT), then clustered based on embedding similarity. The number of clusters is derived heuristically or optimized using silhouette scores.

Resulting sentence groups are merged into chunks; those exceeding the token limit are split further, while very short chunks are merged with neighbors. This method captures topical cohesion well and improves retrieval quality in RAG

systems, though it is computationally expensive compared to other strategies.

C. Embedding and Vector Database

We use the BAAI/bge-large-en-v1.5 sentence embedding model to generate 768-dimensional dense vectors for all text elements, including chunks, queries, and full documents. These embeddings, which capture semantic meaning, are stored in Faiss—an efficient vector database optimized for large-scale similarity search. Embeddings are L2-normalized, enabling cosine similarity retrieval via fast k-nearest-neighbor search. This setup scales well to chunking strategies that yield numerous segments, such as sentence-based approaches.

D. Query Retrieval and Evaluation

Each query in the QA dataset is embedded using the same model and used to retrieve the top-k ($k = 5$) most similar chunks from Faiss. To improve precision, we discard retrieved chunks with cosine similarity below a relevance threshold of 0.70. Evaluation is based on how well the retrieved chunks align semantically with the ground-truth reference spans.

Retrieved chunks are compared to ground-truth answers. If the ground truth maps to a specific chunk or span, that chunk (or set of chunks) is considered relevant. This allows for multi-span answers to be matched across multiple chunks. We then compute standard information retrieval metrics to evaluate performance against varies metrics of chunking elaborated in table I.

IV. Experiment and Results

A. Experimental Setup

To evaluate the performance of chunking strategies in realistic scenarios, we selected five diverse document types varying in structure, content style, and length, each exhibiting distinct structural and semantic characteristics. These documents vary significantly in hierarchical depth, use of visual elements, content style, and length, enabling a comprehensive evaluation of chunking performance.

- **Type-1: Research Report**
  A 13-page thematic report characterized by multi-level hierarchical formatting (chapters, sections, sub-sections) and embedded visual elements such as charts and tables. The content is predominantly paragraph-based with limited structured points, making it suitable for analyzing how chunking handles rich textual narratives.
- **Type-2: Training Manual**
  A 154-page instructional guide organized into single-layer chapters and numbered modules. It contains a mix of diagrams, exercises, and comparison tables, with content dominated by structured points and bullet lists. This format challenges chunking strategies in handling repetitive and modular content.
- **Type-3: Technical User Guide**
  A 26-page document with a flat structure, primarily containing top-level section titles and appliance part illustrations. Its content consists largely of short, single-sentence instructions—frequently bulleted or numbered—making it ideal for evaluating sentence-based or fine-grained chunking approaches.
- **Type-4: Opinion Essay**
  A brief 3-page academic-style essay with minimal headings and virtually no visual aids. The content is purely narrative, consisting of long paragraphs and academic citations. This document helps assess chunking performance on compact, semantically dense text without visual or structural cues.
- **Type-5: Strategic Technical Report**
  A 41-page whitepaper featuring numbered headings, subheadings, and bulleted subtopics, along with extensive visual elements such as charts, model outputs, and appendices. Its content blends paragraph-style narrative with occasional structured lists, offering a hybrid layout for testing chunking adaptability.

We applied all four chunking strategies—fixed-size, sentence-based, recursive and semantic—using two size settings (250-token chunks with 50-token overlap; 500-token chunks with 100-token overlap). The size parameter directly governs fixed-size chunking and serves as an upper bound for recursive and semantic methods, while sentence-based chunking uses it as a length cap. Every chunk is embedded with our chosen model and indexed in FAISS. For each document's set of QA pairs, we issue each question as an embedding query, retrieve the top-5 chunks (cosine similarity ≥ 0.70), and label any below threshold as misses. We then compute Precision, Recall, MRR and NDCG for each query, average those metrics per document, and finally compare the resulting scores across the five document types.

B. Results and Analysis

TABLE II
Evaluation metrics for Document Type I (Research Report).

Metric	Fixed		Sentence		Recursive		Semantic	
	250	500	250	500	250	500	250	500
#Chunks	39	20	146	84	60	55	43	23
ss2fd	0.83	0.84	0.80	0.82	0.83	0.82	0.83	0.85
SRGT	0.71	0.69	0.70	0.70	0.70	0.70	0.70	0.69
MRR	0.75	0.80	0.85	0.83	0.90	0.90	0.90	1.00
Precision	0.48	0.50	0.63	0.60	0.69	0.67	0.52	0.45
Recall	0.43	0.45	0.68	0.72	0.80	0.83	0.65	0.57
NDCG	0.89	0.94	0.86	0.89	0.86	0.92	0.87	0.97
Tokens	1241	2467	562	657	752	1086	1131	1854
Sim.	0.69	0.69	0.69	0.69	0.69	0.69	0.69	0.68
Time (s)	0.04	0.04	0.00	0.00	0.05	0.06	59.1	56.2

We evaluated four chunking strategies across five document types. Each document exhibited a preference for a particular strategy depending on its structure, length, and content style.

Type-1: Research Report: Semantic chunking achieved the highest retrieval accuracy (best MRR, NDCG), making it ideal for precision-focused tasks. Recursive chunking offered the best recall and precision trade-off with significantly lower computational cost. Sentence-based was precise but fragmented answers, while fixed-size was the least effective.

TABLE III
EVALUATION METRICS FOR DOCUMENT TYPE II (TRAINING MANUAL).

Metric	Fixed		Sentence		Recursive		Semantic	
	250	500	250	500	250	500	250	500
#Chunks	309	155	1227	627	424	364	297	149
ss2fd	0.78	0.78	0.75	0.76	0.79	0.79	0.77	0.77
SRGT	0.69	0.68	0.68	0.68	0.68	0.68	0.68	0.67
MRR	0.68	0.65	0.56	0.56	0.58	0.60	0.72	0.67
Precision	0.40	0.31	0.46	0.40	0.42	0.42	0.48	0.52
Recall	0.51	0.40	0.38	0.42	0.48	0.50	0.52	0.40
NDCG	0.90	0.98	0.91	0.92	0.91	0.92	0.93	0.95
Tokens	1261	2511	275	452	1178	1727	1093	2206
QCS	0.69	0.68	0.70	0.70	0.69	0.69	0.69	0.68
Time (s)	0.48	0.37	0.01	0.01	0.16	0.13	371.4	368.6

TABLE IV
EVALUATION METRICS FOR DOCUMENT TYPE III (TECHNICAL USER GUIDE).

Metric	Fixed		Sentence		Recursive		Semantic	
	250	500	250	500	250	500	250	500
#Chunks	42	21	182	82	62	56	41	25
ss2fd	0.85	0.85	0.82	0.83	0.82	0.82	0.84	0.84
SRGT	0.69	0.68	0.72	0.70	0.68	0.68	0.66	0.66
MRR	0.90	0.83	0.78	0.78	0.71	0.77	0.46	0.65
Precision	0.58	0.50	0.66	0.66	0.62	0.67	0.25	0.50
Recall	0.33	0.27	0.55	0.52	0.52	0.42	0.24	0.32
NDCG	0.95	0.95	0.88	0.96	0.88	0.89	0.82	0.91
Tokens	1249	2495	225	479	824	958	1103	2053
QCS	0.66	0.65	0.70	0.68	0.67	0.67	0.65	0.64
Time (s)	0.06	0.05	0.00	0.00	0.07	0.01	69.78	70.14

TABLE V
EVALUATION METRICS FOR DOCUMENT TYPE IV (OPINION ESSAY).

Metric	Fixed		Sentence		Recursive		Semantic	
	250	500	250	500	250	500	250	500
#Chunks	16	8	55	36	13	9	18	11
ss2fd	0.94	0.94	0.90	0.92	0.91	0.90	0.93	0.92
SRGT	0.68	0.68	0.68	0.68	0.67	0.67	0.68	0.68
MRR	0.55	0.70	0.40	0.48	0.52	0.70	0.57	0.65
Precision	0.34	0.36	0.24	0.30	0.16	0.21	0.34	0.32
Recall	0.44	0.52	0.45	0.45	0.40	0.50	0.52	0.52
NDCG	0.89	0.89	0.94	0.97	0.88	0.89	0.91	0.89
Tokens	1226	2380	359	441	1502	2306	978	2055
QCS	0.73	0.73	0.73	0.73	0.71	0.71	0.73	0.72
Time (s)	0.02	0.02	0.00	0.00	0.04	0.06	16.95	17.43

TABLE VI
EVALUATION METRICS FOR DOCUMENT TYPE V (STRATEGIC TECHNICAL REPORT).

Metric	Fixed		Sentence		Recursive		Semantic	
	250	500	250	500	250	500	250	500
#Chunks	80	40	221	130	96	88	73	40
ss2fd	0.88	0.88	0.83	0.86	0.90	0.90	0.88	0.88
SRGT	0.68	0.68	0.71	0.70	0.68	0.69	0.70	0.68
MRR	0.73	0.78	0.58	0.77	0.65	0.65	0.73	0.90
Precision	0.42	0.40	0.40	0.44	0.61	0.59	0.62	0.70
Recall	0.50	0.58	0.36	0.77	0.77	0.77	0.64	1.00
NDCG	0.93	0.96	0.91	0.94	0.91	0.91	0.91	0.83
Tokens	1261	2511	371	503	972	1088	984	2054
QCS	0.70	0.70	0.71	0.71	0.71	0.71	0.71	0.70
Time (s)	0.08	0.10	0.00	0.00	0.07	0.01	88.94	85.54

Overall, semantic is best for accuracy, recursive for efficiency and coverage.

Type-2: Training Manual: Semantic chunking offers the highest retrieval accuracy but is the slowest. Recursive chunking strikes a good balance between accuracy and speed. Fixed-size is reliable and fast, while sentence-based is most efficient in token usage but less accurate. The ideal strategy depends on the desired trade-off between effectiveness and efficiency.

Type-3: Technical User Guide: Sentence-based chunking emerges as the most effective strategy for this document—offering the best trade-off between precision, recall, token efficiency, and processing time. Fixed-size performs well in MRR and NDCG but suffers from low recall and high token cost. Recursive is balanced but slightly behind in ranking metrics. Semantic chunking, despite high ss2fd and NDCG, underperforms due to low precision/recall and heavy computational cost.

Type-4: Opinion Essay: Semantic chunking slightly outperformed others at finer granularity by preserving argument coherence. However, fixed-size chunks with larger limits performed nearly as well. Sentence-based splitting retrieved relevant snippets but often lacked broader context.

Type-5: Strategic Technical Report: Semantic chunking yielded the best retrieval accuracy, with top recall, MRR, and precision, making it ideal for context-rich content—though at high computational and token costs. Recursive chunking provided a good balance of accuracy and efficiency. Fixed-size was competitive but token-heavy, while sentence-based was fastest but less effective for complex answers. Overall, semantic chunking excels in accuracy, recursive in practicality.

C. Comparative Insights

Our analysis highlights trade-offs between retrieval performance, efficiency, and implementation complexity. Semantic chunking often yielded the best accuracy but incurred high computational cost, making simpler methods more practical in some settings.

Retrieval Effectiveness: Semantic chunking excelled on long or dense documents (Types 1, 2, and 5) by preserving context through coherent grouping. It improved recall and MRR but was less effective on short or structured content (e.g., Type 3), where over-merging hurt precision.

Efficiency and Cost: Semantic chunking required significantly more preprocessing time (up to hundreds of seconds) and had higher retrieval token costs. In contrast, fixed-size, sentence-based, and recursive strategies executed quickly, with sentence-based minimizing token cost but increasing index size.

Scalability and Simplicity: Fixed-size and sentence-based approaches are fast, easy to implement, and work well for structured or short documents (Types 3 and 4). Recursive chunking offers a good balance of structure-awareness and performance, especially in longer documents (e.g., Type 2).

Strategy by Document Type: Effectiveness varies by document type:

- **Narrative/complex** (Types 1, 5): Semantic chunking performs best.
- **Modular/structured** (Type 2): Recursive and semantic chunking are effective.

- **Bullet-style/concise** (Type 3): Fixed-size or sentence-based preferred.
- **Short essays** (Type 4): Coarse fixed or semantic chunking suffice.

D. Limitations

While this study provides a comprehensive comparison of chunking strategies, there are several limitations. First, the chunking strategies were evaluated using a limited set of document types and sizes, which may not fully capture performance in highly diverse or multilingual corpora. Second, all experiments relied on a single embedding model (Sentence-BERT), which could bias retrieval outcomes toward strategies better aligned with that model's representation capabilities. Third, semantic chunking relied on clustering algorithms that may be sensitive to hyperparameters; a different similarity threshold or clustering approach might yield different outcomes. Finally, the chunking strategies were tested in isolation—hybrid approaches or adaptive mechanisms could further improve performance but were not explored here.

V. Conclusion and Future Work

A. Conclusion

This study presents a comparative analysis of four chunking strategies—fixed-size with overlap, sentence-based, recursive, and semantic chunking—for retrieval-augmented NLP tasks. Evaluated across diverse document types, the results show that semantic chunking generally achieves the highest retrieval accuracy (MRR, Precision, Recall), while fixed-size and recursive chunking offer a strong balance of performance and computational efficiency. Sentence-based chunking, although efficient in terms of token usage, often struggles with context preservation, impacting retrieval accuracy.

A key insight is that no single strategy consistently outperforms the others. Chunking effectiveness varies with document type, structure, and content density. Semantic chunking is ideal for high-accuracy, offline indexing, while fixed-size and recursive strategies are more suitable for scalable, real-time applications. These findings highlight the importance of adapting chunking methods based on document characteristics rather than applying a one-size-fits-all approach.

From a practical standpoint, the study offers actionable recommendations for developers and practitioners building retrieval pipelines, especially for use cases like enterprise search, RAG systems, and customer support.

B. Future Work

Future research can extend this work by:

- Exploring hybrid chunking strategies (e.g., combining sentence and semantic methods).
- Including more domain-specific document types (e.g., legal, biomedical, or customer service logs).
- Studying the impact of different embedding models on chunking performance.
- Building adaptive pipelines that dynamically select chunking methods based on document features.

Such directions would enhance generalizability and automation in retrieval-based NLP systems.

Acknowledgment

We would like to express our sincere gratitude to Pavan M Laxmeshwar, heading Data Science team at Bosch Global Software Technologies (BGSW) and Shankar Datta M S H, Director at Bosch Global Software Technologies (BGSW) for their invaluable support and believing in us and in providing the insightful review for this research. Without their generosity and commitment to advancing scientific research, this work would not have been possible.

References

[1] Guu, Kelvin, et al. "Retrieval augmented language model pre-training." International conference on machine learning. PMLR, 2020.

[2] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] M. Joshi, E. Choi, D. Weld, and L. Zettlemoyer, "SpanBERT: Improving Pre-training by Representing and Predicting Spans," in *Proc. ACL*, 2020.

[4] K. Lee, M.-W. Chang, and K. Toutanova, "Latent Retrieval for Weakly Supervised Open Domain Question Answering," in *Proc. ACL*, 2019.

[5] V. Karpukhin et al., "Dense Passage Retrieval for Open-Domain Question Answering," in *Proc. EMNLP*, 2020.

[6] L. Wu et al., "MemPrompt: Memory-assisted Prompt Engineering for Retrieval-Augmented Generation," arXiv preprint arXiv:2205.12689, 2022.

[7] A. Paranjape et al., "Information-Theoretic Chunking for Scientific Texts," in *Findings of EMNLP*, 2021.

[8] S. Jaiswal, G. L. Narasimha, D. Prabhu, et al. Improving Model Performance and Explainability of Attention-Based CNN Models on Health Image Datasets Using Grad-CAM. Advanced Computing and Communications: Responsible AI, 2025.

[9] S. Jaiswal, G. L. Narasimha, K. Sathiyanarayanan, A. K. Chebrolu. Deep Learning for NMR Spectra: CNN Classification Using Autoencoder-Generated Latent Features. 2024 4th International Conference on Artificial Intelligence and Signal Processing, 2024.

[10] S. Jaiswal, S. Sekhar, S. Chakraborty. Responsible AI in Action: Enhancing Fairness in Computer Vision through Model Improvement Strategies. 2024 3rd International Conference on Artificial Intelligence for Internet of Things, 2024.

[11] S. Jaiswal, K. C. Gollapudi, R. Susma. Comprehensive Framework for Robustness Evaluation on Numeric Data Classification. 2024 15th International Conference on Computing Communication and Networking, 2024.

[12] N. Thakur, N. Reimers, A. Suni, and I. Gurevych, "BEIR: A Heterogeneous Benchmark for Zero-shot Evaluation of Information Retrieval Models," in *NeurIPS*, 2021.

[13] I. Beltagy, M. Peters, and A. Cohan, "Longformer: The Long-Document Transformer," arXiv preprint arXiv:2004.05150, 2020.

[14] M. Zaheer et al., "BigBird: Transformers for Longer Sequences," in *NeurIPS*, 2020.

[15] G. Izacard and E. Grave, "Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering," arXiv preprint arXiv:2007.01282, 2020.

[16] O. Khattab and M. Zaharia, "ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT," in *Proc. SIGIR*, 2020.

[17] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proc. EMNLP-IJCNLP*, Hong Kong, China, Nov. 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410/

[18] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini, and H. Jégou, "The Faiss Library," *arXiv preprint arXiv:2401.08281*, 2025. [Online]. Available: https://arxiv.org/abs/2401.08281