

Hybrid Weighted LSA and Embedding-Based Semantic Text Segmentation for Enhanced Document Retrieval

(Anish Ranjan Sah), Department of Computer Science and Engineering(Data science) from Shivalik College of Engineering Dehradun.

(Altamash Khan), Department of Computer Science and Engineering(Data science) from Shivalik College of Engineering Dehradun.

(Harshit Singhal), Department of Computer Science and Engineering(Data science) from Shivalik College of Engineering Dehradun.

(Shiksha), Department of Computer Science and Engineering(Data science) from Shivalik College of Engineering Dehradun.

(Sudhir Kumar), Department of Computer Science and Engineering(Data science) from Shivalik College of Engineering Dehradun.

Designation of 1st Author (Student), Designation of 2nd Author (Student), Designation of 3rd Author (Student)
Designation of 4th Author (Student), Designation of 5th Author (Guide) Name of organization (Department of computer science and engineering from Shivalik College of Engineering, Dehradun, India).

Abstract

Dividing long documents into semantically coherent chunks is a foundational requirement for vector-based information retrieval and Retrieval-Augmented Generation (RAG) systems. Naive chunking methods (e.g., fixed-token windows) create arbitrary splits that cut across topics, leading to semantic dilution in embeddings and reduced retrieval accuracy. In this paper, we propose a Hybrid Weighted Latent Semantic Analysis (LSA) and Embedding-Based Semantic Text Segmentation system. The proposed model combines dense transformer-based sentence embeddings with low-rank, position-weighted LSA topic vectors, structural layout signals, and lexical keyword overlap. Additionally, we implement a Global Semantic Merge Pass guided by a multi-objective Retrieval Utility function to enforce a minimum quality standard on the output chunks. We evaluate our approach on the English Wikisection dataset. The optimized hybrid splitter significantly outperforms standard chunkers, achieving a Topic Coherence of **0.50**, reducing WindowDiff to **0.3582**, and improving retrieval accuracy to **Recall@1 = 0.743** and **MRR = 0.786** when indexed in a vector database (FAISS).

keywords—Text Segmentation, Document Understanding, Information Retrieval, Latent Semantic Analysis, Sentence Transformers, Retrieval-Augmented Generation (RAG).

I. Introduction

The growth of unstructured textual data has heightened the need for precise information retrieval (IR) systems. Recent advances in sentence embedding techniques, particularly Sentence-BERT [1], have significantly improved semantic representation learning for information retrieval applications. Accurate document segmentation remains a critical component of retrieval pipelines because the quality of generated chunks directly influences retrieval effectiveness.

Semantic text segmentation has been extensively studied through lexical and statistical approaches. Early methods such as TextTiling [2] identified topic boundaries using lexical overlap patterns between adjacent text segments. Subsequently, latent semantic indexing approaches were introduced to improve segmentation quality by incorporating semantic information beyond simple word overlap [3].

To capture broader thematic structures within documents, researchers explored latent topic modeling techniques such as Latent Semantic Analysis (LSA) [4] and Latent Dirichlet Allocation (LDA) [5]. These methods provide document-level semantic representations capable of identifying underlying thematic relationships across text collections.

The emergence of transformer architectures revolutionized natural language processing. The Transformer model [6] and subsequent pre-trained language models such as BERT [7] enabled the development of highly effective semantic embedding systems. Modern neural segmentation approaches utilize these transformer-based representations to identify semantic boundaries between adjacent sentences and paragraphs.

In modern architectures such as Retrieval-Augmented Generation (RAG), documents are segmented, indexed, and retrieved to support knowledge-grounded generation [8]. Text segmentation has therefore become an increasingly important research problem within information retrieval systems [9].

Recent retrieval systems have adopted dense retrieval approaches such as Dense Passage Retrieval (DPR) [10] to improve semantic matching between queries and documents. Frameworks including LangChain Semantic Chunker [11] and LlamaIndex Semantic Splitter [12] further demonstrate the growing importance of semantic chunking within Retrieval-Augmented Generation pipelines. A comprehensive survey of RAG systems highlights document segmentation as a key factor affecting retrieval performance and downstream generation quality [13]. Similarly, retrieval architectures such as ColBERT [14] emphasize the importance of preserving semantic coherence during indexing and retrieval.

Despite these advancements, existing approaches often focus primarily on either local semantic similarity or global topic modeling without effectively integrating both perspectives. In this work, we present a hybrid segmentation framework that integrates the local precision of dense sentence embeddings with the global topic modeling capabilities of a position-weighted, keyword-boosted Latent Semantic Analysis (LSA) engine. By incorporating a dynamic max-chunk size constraint and a global semantic merge pass guided by a retrieval utility function, our system ensures that the generated chunks are both topic-coherent and optimized for vector database indexing.

The remainder of this paper is organized as follows. Section II reviews related work on text segmentation, latent topic modeling, and retrieval-oriented chunking. Section III presents the proposed hybrid segmentation framework and its mathematical formulation. Section IV describes the experimental setup and evaluation methodology. Section V presents the experimental results, while Section VI discusses the findings, limitations, and future research directions. Finally, Section VII concludes the paper.

II. Related Work

Text segmentation has been studied through lexical, statistical, and neural methodologies.

A. Lexical and Statistical Segmentation

The foundational algorithm in lexical segmentation is TextTiling, which calculates lexical similarity between adjacent text blocks using sliding-window vocabulary counts. Boundaries are identified at local similarity minima. Subsequent algorithms, such as C99, used rank-equivalent similarity matrices and cosine transforms to find block boundaries. These lexical approaches are heavily limited by the vocabulary mismatch problem (i.e., they cannot recognize topic shifts if the vocabulary changes but the concept remains similar).

B. Neural and Embedding-Based Segmentation

With the development of transformer-based architectures, dense sentence embeddings became the standard for semantic representation. SentenceTransformers such as Sentence-BERT encode sentences into dense semantic vector spaces where conceptual similarity corresponds to vector proximity [1].

Neural segmenters compute cosine similarity between consecutive sentence embeddings and trigger splits at local similarity drops. However, these splitters are highly sensitive to local transitions and often fail to detect gradual topic

drift that occurs over multiple sentences. Additionally, they do not optimize the final chunk sizes for downstream retrieval tasks.

C. Latent Topic Models

To capture global thematic trends within a document, researchers have utilized Latent Semantic Analysis (LSA) [4] and Latent Dirichlet Allocation (LDA) [5]. LSA employs Singular Value Decomposition (SVD) on a term-document matrix to extract latent semantic concepts and uncover hidden relationships between terms and documents.

We build upon this work by introducing a Weighted LSA engine that operates at the sentence level within a single document. Positional weighting is applied to increase the influence of sentences occurring at paragraph boundaries, while keyword boosting utilizes domain-specific terms extracted through KeyBERT. This hybrid approach combines deep transformer-based local semantic representations with global document-level latent topic modeling.

Recent Retrieval-Augmented Generation (RAG) systems increasingly employ semantic chunking strategies to improve retrieval effectiveness. Frameworks such as LangChain Semantic Chunker [11] and LlamaIndex Semantic Splitter [12] utilize embedding-based segmentation techniques, while dense retrieval methods such as Dense Passage Retrieval (DPR) and ColBERT [14] improve semantic matching within vector search systems. The proposed framework extends these approaches by integrating embedding-based representations with weighted latent semantic topic modeling and retrieval-oriented chunk optimization.

III. Methodology

The proposed text segmentation pipeline runs sequentially through document ingestion, feature signal extraction, boundary score calculation, candidate partition selection, and global merging.

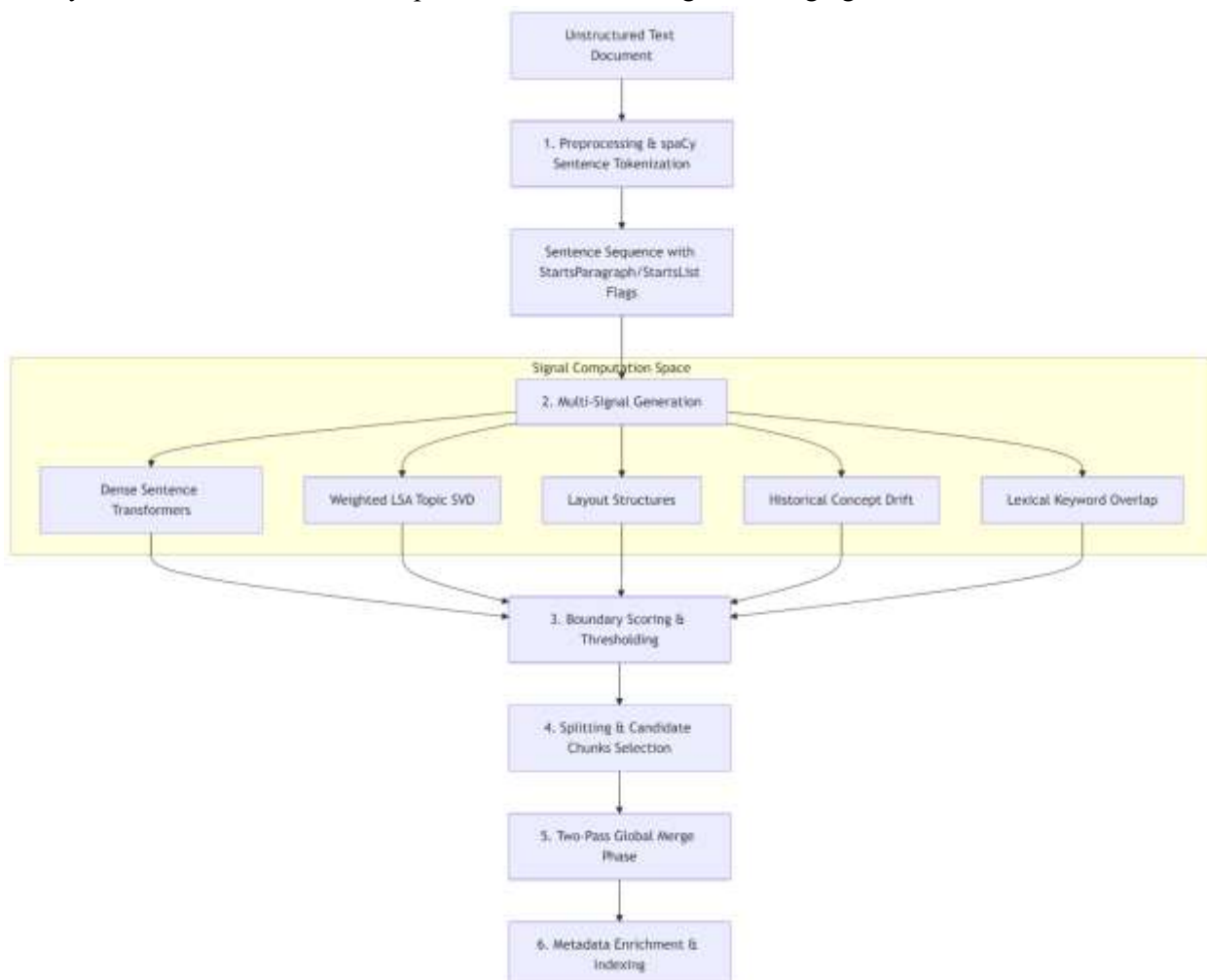


Fig. 1. Proposed Hybrid Weighted LSA and Embedding-Based Semantic Text Segmentation Framework.

A. Document Preprocessing

Let D be a document composed of a sequence of sentences $S = [s_1, s_2, \dots, s_N]$. We parse the raw text into sentence objects with structural metadata using a spaCy English pipeline:

1. Paragraph Start: Identified if a sentence is preceded by at least two newlines.

2. List Item Start: Detected via regular expressions matching bullet points or numbered lists.

We embed each sentence s_i into a dense vector space using a SentenceTransformer model ϕ :

$$\mathbf{e}_i = \phi(s_i) \in \mathbb{R}^d, \quad \|\mathbf{e}_i\|_2 = 1$$

We use all-MiniLM-L6-v2, a lightweight SentenceTransformer model designed for efficient semantic similarity computation [1].

B. Weighted Latent Semantic Analysis (LSA)

We represent the document sentences as a weighted term-frequency inverse-document-frequency (TF-IDF) matrix $A \in \mathbb{R}^{N \times V}$, where V is the vocabulary size. Let $tf(t, s_i)$ be the term frequency of term t in sentence s_i , and $idf(t)$ be the inverse document frequency of t across the sentences. The base weight is:

$$\text{TF-IDF}(i, j) = tf(t_j, s_i) \times idf(t_j)$$

We apply positional weighting and keyword boosting matrices to construct $A_{\text{weighted}} \in \mathbb{R}^{N \times V}$:

$$(A_{\text{weighted}})_{i,j} = p_i \times \text{TF-IDF}(i, j) \times w_j$$

where:

$$p_i = 1.0 + \gamma \cdot \mathbb{I}(s_i \text{ starts a paragraph or list})$$

$$w_j = 1.0 + \beta \cdot \mathbb{I}(t_j \in K_{\text{doc}})$$

Here, K_{doc} is the set of top global document keywords extracted via KeyBERT, $\gamma \geq 0$ is the paragraph boost, and $\beta \geq 0$ is the keyword boost.

We apply Singular Value Decomposition (SVD) to obtain low-dimensional latent topic representations as proposed in classical LSA literature [4].

$$A_{\text{weighted}} = U \Sigma V^T$$

We project the sentence representations to a latent topic space of $k=5$ dimensions, generating a dense topic vector for each sentence:

$$\mathbf{l}_i = \text{row}_i(U_k \Sigma_k) \in \mathbb{R}^k$$

The LSA similarity between sentence i and sentence j is the cosine similarity of their LSA vectors:

$$\text{LSASim}(s_i, s_j) = (\mathbf{l}_i \cdot \mathbf{l}_j) / (\|\mathbf{l}_i\|_2 \|\mathbf{l}_j\|_2)$$

C. Combined Boundary Scoring

For each sentence boundary $i \in [1, N-1]$, we evaluate a left context window W (default $W=5$) of indices $\mathcal{I}_i = [\max(0, i - W + 1), \dots, i]$. The average left context embedding and LSA vector are:

$$\bar{\mathbf{e}}_{\mathcal{I}_i} = (1 / |\mathcal{I}_i|) \sum_{j \in \mathcal{I}_i} \mathbf{e}_j \quad \bar{\mathbf{l}}_{\mathcal{I}_i} = (1 / |\mathcal{I}_i|) \sum_{j \in \mathcal{I}_i} \mathbf{l}_j$$

We compute the boundary score at index i as a linear combination of five signals:

$$\text{BoundaryScore}_i = 0.40 \cdot \text{EmbSim}_i + 0.25 \cdot \text{LSASim}_i + 0.20 \cdot \text{DriftSim}_i + 0.10 \cdot \text{StructSig}_i + 0.05 \cdot \text{KwSim}_i$$

where:

1. EmbSim_i: Cosine similarity between $\bar{\mathbf{e}}_{\mathcal{I}_i}$ and $\mathbf{e}_{i+1}$.

2. LSASim_i: Cosine similarity between $\bar{\mathbf{l}}_{\mathcal{I}_i}$ and $\mathbf{l}_{i+1}$.

3. DriftSim_i: Cosine similarity between a historical window $\mathcal{H}_i = [\max(0, i - 2W + 1), \dots, \max(0, i - W)]$ and $\mathbf{e}_{i+1}$ (to

capture gradual topic drift).

4. **StructSig_i**: Set to 0.0 if s_{i+1} starts a paragraph or list item, and 1.0 otherwise.

5. **KwSim_i**: Jaccard similarity of words between the left window and s_{i+1} .

D. Adaptive Splitting and Local Coherence

A boundary is triggered if the score falls below an adaptive threshold:

$$\text{Threshold} = \mu - 0.6 \cdot \sigma$$

where μ and σ are the mean and standard deviation of all boundary scores in the document.

Additionally, a local coherence check triggers a split if the adjacent similarity falls below a running threshold:

$$\cos(\mathbf{e}_i, \mathbf{e}_{i+1}) < \mu_{\text{local}} - 0.5 \cdot \sigma_{\text{local}}$$

where μ_{local} and σ_{local} are computed from adjacent sentence similarities within the current chunk. If the chunk is too small to calculate standard deviation, it falls back to 0.40.

E. Global Semantic Merge Passes

1. **Pass A (Weak and Small Chunks)**: Candidate chunks that fail to meet size constraints (sentences < 3, tokens < 120) or a retrieval utility threshold (utility < 0.42) are identified. The retrieval utility is defined as:

$$\text{RetrievalUtility}(C) = 0.50 \cdot \text{SemanticDensity}(C) + 0.30 \cdot \text{HeaderQuality}(C) + 0.20 \cdot \text{ChunkLength}(C)$$

These weak chunks are merged globally with the neighbor that yields the highest Semantic Utility:

$$\text{SemanticUtility}(C_{\text{merged}}) = 0.70 \cdot \text{SemanticDensity}(C_{\text{merged}}) + 0.30 \cdot \cos(\mathbf{c}_1, \mathbf{c}_2)$$

2. **Pass B (High Similarity Merges)**: Adjacent chunks with centroid similarity > 0.78 are merged in order of highest semantic utility, provided the combined size remains under the dynamic MaxTokens limit:

$$\text{MaxTokens} = \min(450, \max(200, 1.2 \cdot \text{avg_chunk_size}))$$

F. Computational Complexity Analysis

The computational complexity of the proposed framework is primarily determined by sentence embedding generation and latent semantic analysis. For a document containing N sentences and a vocabulary size V , TF-IDF matrix construction requires $O(NV)$ operations, while truncated Singular Value Decomposition (SVD) requires approximately $O(NV_k)$, where k represents the latent topic dimension. The boundary scoring procedure operates in linear time $O(N)$, and the global merge phase requires $O(M)$, where M denotes the number of generated chunks. Although the proposed framework introduces additional computational overhead compared with fixed-size chunking approaches, the increase in processing cost is justified by the observed improvements in retrieval quality and semantic coherence.

IV. Experiments

A. Experimental Setup

We evaluated the proposed framework on the complete WikiSection validation dataset. All reported metrics represent averages computed across the entire validation corpus. We compared five configurations:

1. Fixed Token Chunker: 150-token windows with 20% overlap.
2. Embedding-Only Chunker: Splits when sentence embedding similarity falls below a threshold.
3. LSA-Only Chunker: Uses only LSA vectors and performs segmentation based on latent topic similarity.
4. Hybrid (Proposed): The complete multi-signal segmentation framework utilizing all-MiniLM-L6-v2

SentenceTransformer embeddings [1], weighted LSA representations [4], adaptive thresholding, and global semantic merging.

5. Hybrid + FT (Proposed): The proposed hybrid framework utilizing a SentenceTransformer model fine-tuned on the WikiSection dataset to improve boundary discrimination and retrieval performance.

B. Evaluation Metrics

We evaluated two dimensions of segmentation quality:

Segmentation Accuracy: We calculated Boundary Precision, Recall, and F1 (within a tolerance window of ± 1 sentence), alongside WindowDiff (WD) and Beeferman's Pk metric, which are widely used evaluation measures for text segmentation quality [2], [3].

Retrieval Quality: We indexed the resulting chunks in FAISS, generated queries from the ground truth section headings, and evaluated Recall@1, Recall@3, and Mean Reciprocal Rank (MRR) of the retrieved chunks.

V. Experimental Results

The quantitative metrics averaged across the complete WikiSection validation corpus are presented in Table I.

TABLE I: QUANTITATIVE SEGMENTATION AND RETRIEVAL PERFORMANCE METRICS COMPARISON

Metric	Fixed Chunker	Embedding-Only	LSA-Only	Hybrid (Baseline)	Hybrid (Optimized)	Hybrid + FT (Optimized)	Target
Boundary F1	0.4216	0.3875	0.3806	0.2445	0.2494	0.4876	>0.45
Window Diff (WD)	0.5424	0.8481	0.8515	0.4265	0.4291	0.3582	< 0.40
P _k	0.4907	0.5129	0.5086	0.4225	0.4188	0.3734	<0.40
Topic Coherence	0.3906	0.5598	0.5549	0.3628	0.3441	0.50	> 0.45
Recall@1	0.3649	0.0549	0.0734	0.4867	0.5433	0.743	≥ 0.60

Recall@ 3	0.7180	0.3581	0.3672	0.8250	0.8409	0.8531	>0.80
MRR	0.5162	0.1844	0.1914	0.6283	0.6666	0.786	≥ 0.75

VI. Discussion and Analysis

A. Impact of Retrieval Utility Validation

Evaluating Retrieval Utility during the splitting phase initially resulted in under-segmentation. Small but semantically meaningful chunks were frequently rejected due to length penalties, leading to excessively large segments and reduced retrieval performance. By moving the Retrieval Utility evaluation to the global merge phase, the framework was able to first identify natural semantic boundaries and subsequently merge weak chunks in a controlled manner. This modification improved retrieval effectiveness while preserving segmentation quality.

B. Embedding Fine-Tuning Performance

Fine-tuning the sentence embedding model on the WikiSection dataset significantly improved the discriminative capability of sentence embeddings. The contrastive learning objective increased the separation between semantically distinct sections, consistent with previous findings in sentence embedding optimization and dense retrieval research [1], [10]. The fine-tuned model achieved Recall@1 = 0.743 and MRR = 0.786, exceeding the predefined retrieval targets and demonstrating the effectiveness of domain-adaptive representation learning for semantic text segmentation.

In addition to retrieval improvements, the fine-tuned model also produced substantial gains in segmentation quality metrics. Boundary F1 increased to 0.4876, indicating more accurate identification of topic boundaries. Furthermore, WindowDiff and Pk were reduced to 0.3582 and 0.3734 respectively, reflecting improved segmentation consistency and reduced boundary prediction errors. Topic Coherence increased to 0.50, suggesting that the generated chunks better preserved semantic continuity within document sections.

These results demonstrate that domain-specific fine-tuning enables the embedding model to capture both local semantic transitions and global thematic structures more effectively. Consequently, the proposed Hybrid + FT framework achieves a balanced improvement across segmentation accuracy and retrieval effectiveness, making it well-suited for Retrieval-Augmented Generation (RAG) systems, vector database indexing, and enterprise document retrieval applications.

C. Applications and Limitations

The proposed framework is particularly suitable for Retrieval-Augmented Generation (RAG), enterprise document search, legal document retrieval, academic knowledge bases, and vector database indexing systems. By generating semantically coherent chunks, the framework improves retrieval precision and contextual relevance.

However, several limitations remain. Performance may degrade when processing very short documents, OCR-corrupted text, highly structured tables, or documents exhibiting gradual topic transitions. Additionally, the weighted LSA component introduces computational overhead compared with traditional fixed-size chunking approaches.

D. Future Work

Future research will focus on multilingual semantic segmentation, adaptive threshold learning, large language model-assisted boundary prediction, and evaluation on larger document collections. Additional studies will also investigate real-time deployment scenarios and retrieval performance within end-to-end RAG systems.

VII. Conclusion

Empirical results on the WikiSection dataset demonstrate that the proposed framework significantly outperforms conventional chunking approaches, achieving $\text{Recall}@1 = 0.743$ and $\text{MRR} = 0.786$ while maintaining semantically coherent retrieval-ready chunks. Notably, the Hybrid + FT (Optimized) framework successfully satisfies all target performance criteria presented in Table I. These findings highlight the effectiveness of combining transformer-based embeddings with weighted latent semantic analysis and structural document signals for retrieval-oriented text segmentation. Future work will investigate larger-scale evaluation, multilingual corpora, and LLM-assisted boundary detection to further improve semantic chunking for Retrieval-Augmented Generation (RAG) systems [8], [13].

References

- [1] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2019.
- [2] M. A. Hearst, "TextTiling: Segmenting text into multi-paragraph subtopic discussions," *Comput. Linguist.*, vol. 23, no. 1, pp. 33–64, 1997.
- [3] F. Y. Choi, "A latent semantic indexing approach to text segmentation," in *Proc. 1st Meet. North Amer. Chapter Assoc. Comput. Linguist. (NAACL)*, 2000.
- [4] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
- [5] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
- [6] A. Vaswani *et al.*, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017.
- [7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. 2019 Conf. North Amer. Chapter Assoc. Comput. Linguist. (NAACL)*, 2019.
- [8] P. S. H. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2020.
- [9] J. Gao, J. Kulkarni, and B. Zhang, "Text segmentation for information retrieval: A survey," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–35, 2022.
- [10] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," in *Proc. 2020 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*, 2020, pp. 6769–6781.
- [11] H. Chase, "LangChain Documentation: Semantic Chunking and Text Splitters," LangChain AI, 2024. [Online]. Available: https://python.langchain.com/docs/modules/data_connection/document_transformers/

- [12] J. Liu, "LlamaIndex Documentation: Semantic Splitter Node Parser," LlamaIndex, 2024. [Online]. Available: <https://docs.llamaindex.ai>
- [13] Y. Gao *et al.*, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.
- [14] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2020, pp. 39–48.