Beyond Chunking: Discourse-Aware Hierarchical Retrieval for Long Document Question Answering

Anonymous ACL submission

Abstract

Long document question answering systems typically process texts as flat sequences or use arbitrary segmentation, failing to capture discourse structures that guide human comprehension. We present a discourse-aware hierarchical framework that leverages rhetorical structure theory (RST) to enhance long document question answering. Our approach converts discourse trees into sentence-level representations and employs LLM-enhanced node representations to bridge structural and semantic information. The framework involves three key innovations: specialized discourse parsing for lengthy documents, LLM-based enhancement of discourse relation nodes, and structure-guided hierarchical retrieval. Comprehensive experiments on QASPER, QuALITY, and NarrativeQA demonstrate consistent improvements over existing approaches. Ablation studies confirm that incorporating discourse structure significantly enhances question answering across diverse document types.

1 Introduction

Document question answering represents a fundamental challenge in natural language processing, with applications spanning from academic research to enterprise knowledge management (Chen et al., 2017; Karpukhin et al., 2020). To date, large language models (LLMs) have achieved remarkable success on the task, particularly for short documents such as SQuAD (Rajpurkar et al., 2016), which has an average context length of 117 words (Reddy et al., 2019), reaching human-level performance with F1 scores exceeding 85% (Chowdhery et al., 2023; Malladi et al., 2023). However, as document length increases, their performance degrades significantly. For example, on challenging long document datasets such as QASPER, state-of-the-art LLM models only achieve performance with less than 50% F1 scores (Shaham et al., 2022; Bai et al., 2024). This performance gap highlights long document question answering as a critical and underexplored research frontier.

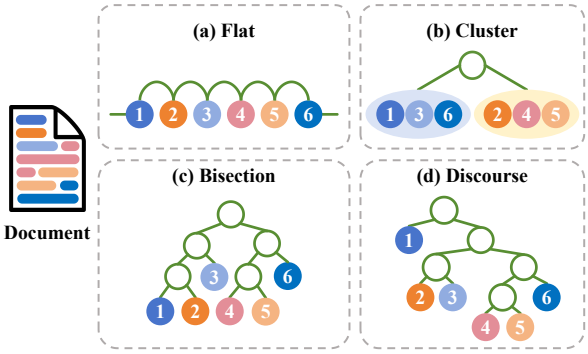

Figure 1: Comparison of document modeling approaches for long document QA. Numbers (1-6) show sentence order in original document, with similar colors indicating semantic relationships. Four approaches are compared: (a) Flat sequential modeling, (b) Bottom-up semantic clustering, (c) Bisection-based adjacent grouping, and (d) Our discourse-aware approach that preserves both semantic and discourse structures.

The most straightforward approach to handling long documents involves flat sequential modeling, where texts are processed as linear sequences or divided into fixed-size chunks (Lewis et al., 2020; Guu et al., 2020a). This chunking-based paradigm has gained widespread adoption due to its simplicity and computational efficiency, enabling scalable processing of large document collections. Building upon this foundation, recent advances have introduced tree-based structures to better capture document organization and move beyond simple chunking. These innovations include semantic clustering methods like RAPTOR (Sarthi et al., 2024) that recursively group similar content, and bisection-based approaches that maintain local coherence through adjacent segment grouping (Liu and Lapata, 2019; Karpukhin et al., 2020). These hierarchical methods represent significant improvements by capturing semantic similarity and textual coherence, respectively, offering more sophisticated

document modeling capabilities.

Discourse structure provides a principled linguistic foundation for document organization that goes beyond surface-level similarities. Consider a document where the first sentence serves as a topic statement, followed by contrasting ideas (sentences 2-3), parallel supporting evidence (sentences 4-5), and a concluding statement (sentence 6). These discourse relationships naturally capture how humans organize and comprehend information in documents, providing guidance for more semantically appropriate chunking and hierarchical organization in retrieval. As a mature theoretical framework, discourse analysis has demonstrated effectiveness across various NLP tasks (Feng and Hirst, 2014; Cohan et al., 2018; Søgaard et al., 2021), presenting an opportunity to develop more principled retrieval methods that align with human document comprehension patterns.

Therefore, in this work, we present **DISRetrieval** (**DIS**course-aware hierarchical **Retrieval**), the first systematic approach that leverages rhetorical structure theory (RST) to enhance long document question answering through discourse-aware modeling. Our approach tackles three key technical challenges: First, we develop an effective adaptation strategy that enables traditional RST to work at appropriate granularity levels for computational efficiency while preserving semantic coherence. Second, we introduce LLM-based node enhancement that enriches intermediate nodes with both discourse structure and semantic content. Third, we design structure-guided evidence retrieval mechanisms that leverage discourse organization for more effective information extraction.

Comprehensive experiments across three challenging benchmarks: QASPER (Dasigi et al., 2021) for research paper understanding, QuALITY (Pang et al., 2022) for reading comprehension, and NarrativeQA (Kočiský et al., 2018) for book-length narrative analysis demonstrate substantial improvements over existing methods. Detailed ablation studies validate the effectiveness of our discourse-aware modeling and structure-guided processing, showing consistent gains across diverse scenarios. Our framework successfully captures both fine-grained semantic details and document-level organizational patterns, providing a robust solution for discourse-informed question answering.

Our main contributions can be summarized as follows:

- A novel discourse-aware hierarchical framework that systematically incorporates RST for long document question answering.
- An innovative LLM-enhanced hierarchical retrieval mechanism that enriches intermediate nodes with semantic content and enables multi-granularity retrieval.
- Comprehensive empirical validation demonstrating consistent improvements across diverse datasets and model architectures.

Our code and model implementations will be publicly available to facilitate future research.

2 Related Works

Recent advances in natural language processing have achieved remarkable progress in document processing and understanding, with large language models demonstrating impressive capabilities across diverse text comprehension tasks (Brown et al., 2020; Chowdhery et al., 2023). However, these successes primarily focus on shorter texts, as extended documents present fundamental challenges due to context length constraints and computational complexity (Tay et al., 2021; Zaheer et al., 2020; Beltagy et al., 2020; Ding et al., 2023; Ainslie et al., 2023; Chen et al., 2024). Traditional approaches typically operate on short text segments without considering broader document context, leading to fragmented understanding (Liu and Lapata, 2019; Guo et al., 2022; Rae et al., 2020). Recent innovations such as SEER's chunking-free key information extraction (Zhao et al., 2024) and CFIC's in-context retrieval approach (Qian et al., 2024) highlight the ongoing evolution in document comprehension strategies for extended content (Wang et al., 2023; Izacard et al., 2023; Yu et al., 2023).

To tackle these challenges, retrieval-based methods have emerged as a promising paradigm, segmenting documents into manageable pieces before retrieving the most relevant evidence for downstream tasks (Lewis et al., 2020; Karpukhin et al., 2020; Guu et al., 2020a; Izacard and Grave, 2020). The evolution of retrieval-based approaches has progressed from simple flat segmentation strategies (Guu et al., 2020b) to sophisticated hierarchical organization methods. RAPTOR pioneered bottom-up semantic clustering for tree-based document representation (Sarthi et al., 2024), while bisection-based methods preserve local coherence through adjacent segment grouping (Zhang et al., 2020; Ivgi et al., 2023). Recent innovations include hybrid hi-

erarchical retrieval systems that bridge document-level and passage-level representations (Liu et al., 2021; Arivazhagan et al., 2023), and LongRefiner's XML-format structure conversion using fine-tuned LLMs for top-down tree construction, though its length sensitivity creates challenges for long documents (Jin et al., 2025). While these hierarchical approaches represent significant progress in document modeling, they primarily rely on surface-level semantic similarity or structural heuristics, overlooking the rich linguistically-informed discourse structures that naturally organize human-authored documents (Liu et al., 2019; Wolf et al., 2020).

3 Method

Discourse analysis provides systematic frameworks for understanding how textual segments relate beyond surface-level semantics, offering principled approaches to document organization (Mann and Thompson, 1988; Carlson et al., 2001). RST formalizes this understanding by representing documents as hierarchical trees where leaf nodes correspond to elementary discourse units (EDUs) and internal nodes encode rhetorical relations such as contrast, elaboration, and summary (Marcu, 2000). Recent advances in neural discourse parsing (Yu et al., 2018; Kobayashi et al., 2020; Yu et al., 2022a) have enabled automatic RST construction at scale, facilitating its application across diverse NLP tasks (Maekawa et al., 2024; Yuan et al., 2025).

To address the challenge of bridging between discourse theory and document retrieval, we propose **DISRetrieval**, a discourse-aware framework that systematically incorporates RST structure into document retrieval as shown in Figure 2. The complete algorithm is detailed in Algorithm 2 in Appendix A.

3.1 Discourse-Aware Tree Construction

The first stage of our framework aims to construct a hierarchical tree structure that captures semantic content and organizational structure of documents. We leverage discourse analysis to identify meaningful relationships between text segments and organize them into a coherent tree representation.

3.1.1 Granularity-Adapted RST Processing

Traditional RST operates on fine-grained EDUs, which can create computational overhead and semantic fragmentation when processing long documents. We develop a granularity adaptation strategy that shifts RST processing to the sentence level, achieving better semantic coherence and efficiency while maintaining discourse interpretability. Our adaptation involves training a sentence-level discourse parser by systematically converting EDU-based datasets: we merge intra-sentence EDUs into coherent sentence units and determine inter-sentence relationships through lowest common ancestor analysis in the original EDU trees. This produces a specialized parser $f_{\text{discourse}}$ that operates directly on sentence-level discourse relationships, enabling efficient processing of long documents.

3.1.2 Complete Tree Construction Process

Our core innovation lies in developing a hierarchical discourse modeling framework that constructs complete document-level discourse trees through a systematic two-phase process, effectively bridging local sentence-level relationships with global document organization while preserving discourse coherence at multiple granularity levels.

Phase 1. Paragraph-Level Tree Construction. The initial phase processes each paragraph p_i by transforming its sentence sequence $\boldsymbol{S_i} = \{s_{i,1}, s_{i,2}, ..., s_{i,m}\}$ into a local discourse tree:

$$T_i = f_{\text{discourse}}(\boldsymbol{S_i}), \tag{1}$$

where each tree T_i captures how sentences connect and build upon each other within the paragraph, reflecting both semantic relationships and logical flow between adjacent sentences. This phase ensures that fine-grained intra-paragraph discourse structure is preserved.

Phase 2. Document-Level Tree Construction. The second phase constructs the global document structure by first obtaining simplified representations of paragraphs through bottom-up LLM enhancement. For each internal node v in paragraph-level trees, we apply adaptive processing:

$$v^* = \begin{cases} f_{\text{LLM}}(v_l, v_r), & \text{if } |v_l| + |v_r| \geqslant \tau \\ f_{\text{merge}}(v), & \text{otherwise} \end{cases} \tag{2}$$

where v_l and v_r denote the left and right children of node v. The function f_{LLM} generates concise summaries for larger text spans that align with our sentence-level discourse parser, while $f_{\text{merge}}(v)$ merges smaller, coherent subtrees into single leaf nodes to maintain semantic coherence.

The root representations obtained from each paragraph tree serve as input for constructing the document-level discourse tree:

$$T_{\text{doc}} = f_{\text{discourse}}(\{t^1_{\text{root}}, t^2_{\text{root}}, ..., t^n_{\text{root}}\}). \tag{3}$$

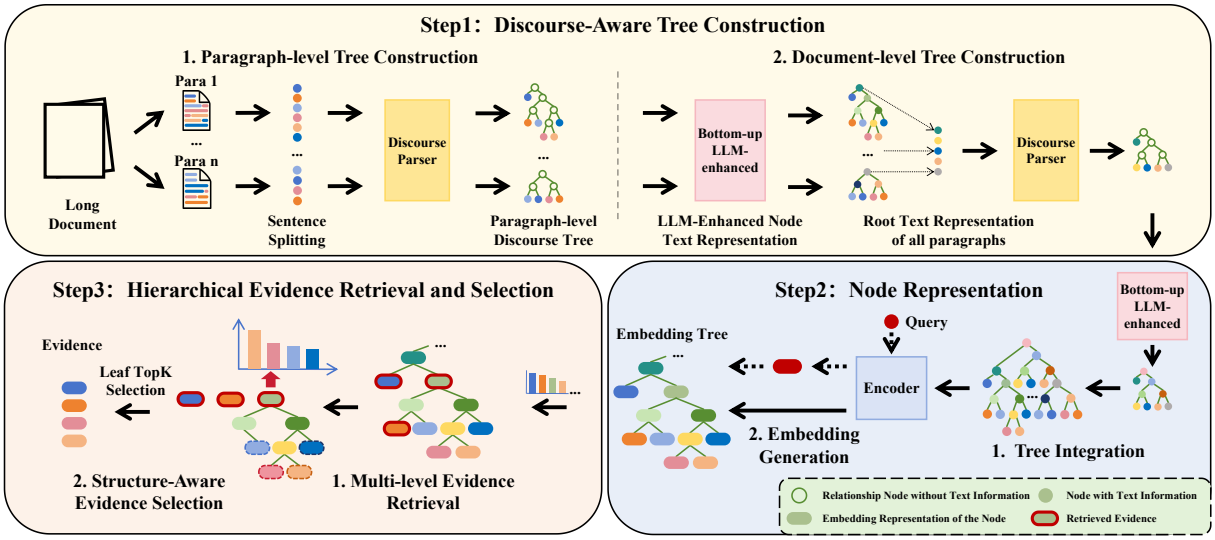

Figure 2: Overview of the DISRetrieval framework. The framework consists of three main steps: (1) Discourse-Aware Tree Construction that builds a hierarchical discourse structure through two phases: constructing paragraph-level discourse trees via discourse parsing, and generating document-level tree with LLM-enhanced node representations; (2) Node Representation that integrates the trees and converts text content into dense vector representations via an encoder; (3) Hierarchical Evidence Retrieval and Selection that performs multi-level evidence retrieval and structure-aware selection to identify relevant text segments.

This approach transforms rhetorical relations into concrete semantic summaries that effectively represent the content at different levels of the discourse hierarchy (detailed illustration provided in Appendix A.3).

3.2 Discourse-Aware Node Representation

A critical challenge in discourse-based retrieval lies in bridging the semantic gap between abstract rhetorical relations and concrete textual content required for neural retrieval. We develop a semantic enhancement framework that transforms discourse structures into dense representations while preserving hierarchical properties through three key stages.

Bottom-up Semantic Enhancement. The document-level discourse tree T_{doc} initially contains only abstract structural relationships at internal nodes, lacking concrete semantic content necessary for retrieval. We apply bottom-up LLM enhancement following Equation 2 to generate meaningful textual representations for these structural nodes. This enhancement process systematically transforms each internal node from abstract structural placeholders into semantically rich representations that capture the combined content of their subtrees, ensuring all nodes possess concrete textual content suitable for downstream semantic matching operations.

Multi-Level Tree Integration. We implement a substitution-based integration mechanism that combines the enhanced document-level tree T^*_{doc} with paragraph-level trees T_i into a unified discourse structure. Each leaf node in T^*_{doc} is systematically replaced with its corresponding paragraph-level discourse tree:

$$T_{\text{D}} = T^*_{\text{doc}}[l_i \leftarrow T_i, \forall i \in \{1, 2, ..., n\}]. \quad (4)$$

This integration enables multi-granularity discourse modeling, where fine-grained intra-paragraph sentence relationships coexist with coarse-grained inter-paragraph discourse connections within a single hierarchical structure.

Node Encoding. To enable efficient retrieval, we transform the integrated discourse tree T_{D} into dense embeddings using a pre-trained encoder:

$$\mathbf{e}_v = f_{\text{enc}}(v), \quad \forall v \in \mathcal{N}(T_{\text{D}}). \quad (5)$$

The resulting embedding tree $\mathcal{T}_{\text{emb}} = (T_{\text{D}}, \{\mathbf{e}_v\})$ preserves both the hierarchical discourse structure and semantic information, enabling structure-aware retrieval through dense vector operations.

3.3 Structure-Guided Evidence Retrieval

We introduce a structure-aware retrieval mechanism that leverages the hierarchical nature of discourse trees to select evidence at multiple granularity levels. Unlike flat retrieval methods that operate

Algorithm 1: Structure-Guided Evidence Retrieval

Input: Query q, discourse tree T, evidence limit K, subtree selection size k
Output: Evidence set E

```
/* Compute node similarities and rank */
```
$\mathbf{e_q} \leftarrow \text{Encoder}(q)$;
$\text{scores}[v] \leftarrow \text{cosine}(\mathbf{e_q}, \mathbf{e_v})$ for all $v \in T$;
$V_{\text{ranked}} \leftarrow \text{sort}(\text{scores}, \text{descending})$;
```
/* Structure-aware selection */
```
$E \leftarrow \{\}$, used $\leftarrow \{\}$;
for $v \in V_{ranked}$ **do**
  if *v is leaf and $v \notin$ used* **then**
    $E \leftarrow E \cup \{v\}$, used $\leftarrow$ used $\cup \{v\}$;
  else
    $L \leftarrow \text{leaves}(v) \setminus \text{used}$; `// Unused leaves in subtree`
    if $L \neq \emptyset$ **then**
      $L_k \leftarrow \text{top_k}(L, \text{scores}, k)$;
      $E \leftarrow E \cup L_k$, used $\leftarrow$ used $\cup L_k$;
    end
  end
  if $|E| \geq K$ **then**
    break
  end
end
return E

on uniform text segments, our approach exploits discourse structure to implement a dual-selection strategy that balances local semantic relevance with global discourse coherence.

Retrieval Process. Given a query q, we first compute semantic similarities for all nodes in the embedding tree:

$$\text{score}(v) = \cos(f_{\text{enc}}(q), \mathbf{e}_v), \forall v \in \mathcal{N}(T_{\text{D}}). \quad (6)$$

Dual-Selection Strategy. Nodes are ranked by relevance and processed through two complementary selection mechanisms:

- **Direct leaf selection**: High-relevance sentences selected immediately
- **Hierarchical expansion**: Internal nodes trigger top-k subtree selection
- **Redundancy elimination**: Automatic tracking prevents duplicate selection

For leaf nodes with high relevance, we select them directly. For internal nodes, we perform controlled subtree expansion by selecting the top-k most relevant unused leaves from their subtrees. This ensures both specific sentence-level evidence and coherent discourse segments are captured while eliminating redundancy.

The complete algorithm is detailed in Algorithm 1, which naturally balances local semantic relevance with global discourse coherence, enabling retrieval at appropriate granularity levels determined by the document's discourse structure.

4 Experiments

4.1 Experimental Setup

Datasets and Evaluation Metrics. We evaluate our approach on three challenging long document QA datasets: QASPER (Dasigi et al., 2021) for research paper understanding (averaging 4170 words), QuALITY (Pang et al., 2022) for long-form reading comprehension (averaging 5022 words), and NarrativeQA (Kočiský et al., 2018) for QA over exceptionally long narrative documents (averaging 51,372 words). For QASPER, we use F1-Match for QA performance and token-level F1/Recall for retrieval evaluation. For QuALITY, we report accuracy as the primary metric. And for NarrativeQA, we measure performance using BLEU(B-1), ROUGE(R-L) and METEOR(M) metrics. To ensure fair comparison, we fix the total length of retrieved context across all methods.

Baselines. We evaluate our approach against several strong retrieval baselines:

- Flatten-chunk splits articles into chunks of maximum 100 words while preserving sentence boundaries for semantic coherence.
- Flatten-sentence adopts sentence-level splitting and direct retrieval, providing a direct comparison baseline for our hierarchical approach.
- RAPTOR constructs a semantic tree through recursive embedding, clustering, and summarization, with retrieval performed on a collapsed tree structure following Sarthi et al. (2024).
- Bisection shares our LLM-enhanced representations and hierarchical retrieval mechanism but uses a different tree construction method - recursively dividing sentence set into balanced binary subtrees. This baseline helps isolate benefits of our discourse-aware tree structure.

Implementation details. For discourse analysis, we construct our sentence-level RST training data based on RST-DT (Carlson et al., 2001). We then train our discourse parser following Yu et al. (2022b), adopting XLNet-base-cased as the pretrained language model backbone. All summarization tasks, including both our approach (as defined in Equation 2) and RAPTOR baseline, utilize Llama3.1-8B-Instruct for consistency. We set the threshold parameter τ in Equation 2 to 0 for QASPER and 50 for QuALITY and NarrativeQA (detailed in Appendix C.4). For sentence encoder

Method	F1-Match (QASPER) / %						Accuracy (QuALITY) / %					
	200 (s)	200 (o)	300 (s)	300 (o)	400 (s)	400 (o)	200 (s)	200 (o)	300 (s)	300 (o)	400 (s)	400 (o)
UnifiedQA-3B												
flatten-chunk	33.97	37.50	35.41	38.46	36.13	39.03	52.97	56.28	54.46	57.53	55.23	57.62
flatten-sentence	35.16	38.43	37.24	39.31	37.99	40.06	53.16	56.04	54.55	56.71	55.27	57.38
RAPTOR	33.57	37.46	34.95	39.96	37.00	39.53	53.60	55.99	54.75	57.00	55.51	58.53
Ours												
Bisection	36.17	37.41	37.66	39.49	38.84	39.70	55.13	57.00	56.04	58.53	57.24	60.21
DISRetrieval	**37.59**	**38.71**	**39.65**	**40.51**	**40.57**	**41.14**	**56.90**	**58.01**	**59.16**	**59.92**	**59.64**	**60.64**
GPT-4.1-mini												
flatten-chunk	37.37	41.13	41.38	43.34	42.72	44.78	60.16	65.77	63.66	69.27	67.69	71.05
flatten-sentence	39.82	43.08	41.84	45.28	42.44	45.78	60.12	64.43	63.09	66.68	65.24	69.94
RAPTOR	37.55	40.88	39.95	43.26	42.50	43.85	61.31	64.77	63.81	67.79	67.26	70.71
Ours												
Bisection	40.98	43.12	42.74	44.54	43.49	45.69	63.71	67.88	65.68	70.71	67.93	72.00
DISRetrieval	**41.04**	**43.59**	**43.02**	**45.47**	**44.57**	**46.43**	**64.77**	**69.12**	**66.97**	**72.58**	**69.32**	**73.78**
Deepseek-v3												
flatten-chunk	35.29	38.14	37.99	40.78	40.57	42.26	65.68	71.52	69.65	75.02	73.25	76.56
flatten-sentence	36.51	39.27	39.41	41.57	40.82	43.04	64.14	68.74	68.41	72.24	69.65	73.63
RAPTOR	34.66	37.84	37.30	40.50	39.77	42.17	65.53	69.13	68.65	72.24	71.19	75.22
Ours												
Bisection	37.28	39.27	39.80	41.81	40.96	**43.57**	67.83	71.09	70.81	74.40	73.39	76.94
DISRetrieval	**38.34**	**40.03**	**40.33**	**42.10**	**41.61**	43.30	**68.94**	**72.96**	**72.39**	**76.70**	**73.78**	**78.00**

Table 1: Generation performance comparison of different methods under varying retrieved context lengths (200-400 words) and different embedding models (SBERT: s, OpenAI: o) across two datasets. Best results are **bolded**, runner-ups are underlined.

(Section §3.2), we experiment with both Sentence-BERT[1] and the latest embedding model of OpenAI[2] to evaluate the effectiveness of our method across different representation spaces. The Top-K parameter in leaf selection is set to 5 based on preliminary experiments (detailed in Appendix C.3). For generation models, we employ UnifiedQA-3B, GPT-4.1-mini and Deepseek-v3 to demonstrate the robustness of our retrieval improvements across different generation capabilities. More details are provided in Appendix B.

4.2 Main Results

Generation Performance. We evaluate our approach across different context lengths (200-400 words) and embedding models (SBERT and OpenAI) on QASPER and QuALITY datasets. Table 1 presents the comparative results. The results reveal three key insights:

Consistent superiority across settings. DISRetrieval outperforms all baselines regardless of context length, embedding model, or generation architecture (UnifiedQA-3B, GPT-4.1-mini, and Deepseek-v3). For instance, with 400-word contexts and UnifiedQA-3B, we achieve +2.58% F1-Match on QASPER and +4.37% accuracy on QuALITY over the flatten-sentence baseline.

Discourse structure surpasses semantic clustering. Compared to RAPTOR's semantic clustering approach, our discourse-aware method demonstrates clear advantages (40.24% vs. 39.96% F1-Match on QASPER with OpenAI embeddings), confirming that linguistic discourse structure provides more principled document organization than purely semantic-based methods.

Linguistic discourse essential for hierarchical modeling. The Bisection ablation validates our core hypothesis: while hierarchical organization helps (Bisection > flatten baselines), incorporating discourse structure is crucial (DISRetrieval > Bisection consistently), providing substantial benefits beyond simple tree-based document modeling.

Retrieval Performance. We evaluate DISRetrieval's retrieval effectiveness on QASPER using token-level F1 and Recall metrics. The Retrieval Result in Table 2 reveals three key findings:

First, DISRetrieval consistently outperforms baselines across all settings, achieving the highest F1 and Recall scores with both SBERT and OpenAI embeddings. This demonstrates that our discourse-aware context modeling effectively cap-

[1] multi-qa-mpnet-base-cos-v1, https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-cos-v1

[2] OpenAI text-embedding-3-large

Retrieval Result	200 (SBERT)		200 (OpenAI)		300 (SBERT)		300 (OpenAI)		400 (SBERT)		400 (OpenAI)	
	F1 / %	Recall / %	F1 / %	Recall / %	F1 / %	Recall / %	F1 / %	Recall / %	F1 / %	Recall / %	F1 / %	Recall / %
flatten-chunk	26.13	56.05	29.17	62.16	23.10	63.92	25.12	69.04	20.38	68.75	21.91	73.38
flatten-sentence	26.15	57.80	28.25	62.78	22.68	64.63	24.04	68.43	19.98	69.02	21.06	72.42
RAPTOR	24.57	52.42	27.18	58.49	21.71	60.00	23.57	65.63	19.27	65.11	20.64	70.04
Ours												
Bisection	27.63	59.82	29.29	63.11	23.83	66.69	25.16	69.94	21.10	71.52	21.98	74.18
DISRetrieval	**28.66**	**61.76**	**30.57**	**65.51**	**24.74**	**68.30**	**26.20**	**72.20**	**21.68**	**72.92**	**22.76**	**76.03**

Generation Result	Full Document + Retrieved Evidence						Full Document	Gold Evidence
	200 (SBERT)	200 (OpenAI)	300 (SBERT)	300 (OpenAI)	400 (SBERT)	400 (OpenAI)	Avg. 4170 wrods	Avg. 129 words
F1 / %	49.56 (+0.75)	49.83 (+1.02)	49.63 (+0.82)	49.72 (+0.91)	49.71 (+0.90)	49.56 (+0.75)	48.81	50.71 (+1.90)

Table 2: Retrieval and generation results (token-level F1 and Recall) on the QASPER dataset using SBERT and OpenAI Embedding as embedding models and Llama3.1-8B-Instruct as generation model. Best results are **bolded** and runner-ups are underlined.

tures the semantic relationships within documents.

Second, for longer contexts (300-400 words), while all methods show some F1 score degradation, DISRetrieval maintains superior performance, particularly in Recall metrics. This robust performance on longer contexts validates the capability of our method in handling complex document structures through discourse-guided retrieval.

The ablation with Bisection shows that while hierarchical organization helps, the full discourse-aware approach provides additional benefits. Additionally, OpenAI embeddings consistently outperform SBERT across all settings, suggesting that strong semantic representations are fundamental to the effectiveness of discourse-aware retrieval.

	BLEU / %	ROUGE / %	METEOR / %
flatten-chunk	24.24	28.42	19.70
flatten-sentence	21.53	28.22	18.33
RAPTOR	25.05	30.24	20.92
Ours			
Bisection	24.71	28.85	20.41
DISRetrieval	**25.59**	**30.26**	**21.21**

Table 3: Performance on the NarrativeQA dataset. Best results are **bolded** and runner-ups are underlined.

4.3 Discussions

RQ1: Is DISRetrieval effective for handling extremely long documents? Tab 3 We evaluate our method on NarrativeQA, which contains exceptionally long documents (average: 51,372 words, maximum: 346,902 words) that exceed most generative models' context limits. DISRetrieval achieves superior performance across all metrics, outperforming the widely-used flatten-chunk baseline by +1.35% BLEU, +1.84% ROUGE, and +1.51% METEOR. Notably, both DISRetrieval and RAPTOR substantially outperform flatten-based methods, confirming that structured approaches are more effective than flat for extremely long document retrieval. However, DISRetrieval's discourse-aware structure provides consistent advantages over RAPTOR's semantic clustering approach.

RQ2: Is precise evidence retrieval essential for effective question answering? Tab 2 To evaluate the importance of precise evidence retrieval, we compare three context settings using Llama3.1-8B-Instruct: (i) golden evidence, (ii) full document, and (iii) full document augmented with retrieved evidence (highlighted with **[EVIDENCE] ... [/EVIDENCE]** markers). Golden evidence achieves the highest F1 score of 50.71% with only 129 words on average, significantly outperforming the full document approach (48.81% with 4,170 words). This demonstrates that concise yet accurate retrieval is more effective than using substantially larger context. Furthermore, augmenting the full document with our retrieved evidence yields consistent improvements of 0.75-1.02% across all settings, confirming that precise retrieval enhances performance even when combined with complete document access. These results underscore that effective long-document QA systems should prioritize retrieval quality over context quantity.

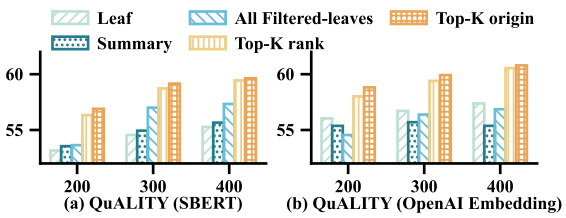

Figure 3: Ablation results of different variants.

RQ3: Is the hierarchical retrieval strategy effective? Fig 3 We conduct ablation experiments on the QuALITY dataset to evaluate our hierarchical retrieval strategy. We compare five variants: leaf-only baseline, summary-based retrieval, all filtered-leaves, Top-K with ranking order, and our Top-K with original order. Three key findings emerge: (1) Summary-based retrieval underperforms the leaf baseline, confirming that preserving original text details is crucial for effective retrieval. (2) While all filtered-leaves shows marginal improvements, selective Top-K methods are superior due to reduced noise from irrelevant content. (3) Our Top-K origin method consistently achieves the best performance by preserving natural document flow. This advantage intensifies with longer contexts and stronger embeddings, validating that maintaining document structure is essential for hierarchical retrieval.

RQ4: Does the scale of LLMs affect the quality of the discourse-aware tree structures in node text enhancement? Tab 4 We investigate whether model scale affects the quality of discourse-aware tree construction by comparing different LLMs. As shown in Table 4, smaller models like Llama-3.1-8B, Qwen2.5-7B, and Mistral-7B achieve comparable performance to larger models, with differences under 0.5%. Notably, Llama-3.1-8B achieves the best recall at 400-word context length. These results indicate that the effectiveness of our discourse-aware retrieval does not heavily depend on LLM scale. This finding is particularly valuable as it enables flexible model selection without requiring extremely large models, making our approach more practical and cost-effective.

	200		300		400	
	F1 / %	Recall / %	F1 / %	Recall / %	F1 / %	Recall / %
Llama-3.1-8B	28.66	61.76	24.74	68.30	21.68	72.92
Qwen2.5-7B	28.32	61.09	24.75	68.11	21.69	72.53
Mistral-7B	28.17	60.86	24.65	68.41	21.65	72.86
GPT-4o-mini	28.27	61.13	24.66	68.31	21.68	72.84
Deepseek-v3	28.89	61.91	25.02	68.57	21.82	72.91

Table 4: Effect of different LLMs for node text enhancement on QASPER dataset retrieval performance (token-level F1 and Recall) across varying context lengths.

RQ5: Does the capability of the discourse parser have a significant impact? Fig 4 We examine how discourse parser capabilities affect downstream performance by training parsers on varying amounts of data (0-100%). As shown in Figure 4, both retrieval recall and answer F1 scores improve consistently as parser training data increases across all context lengths. These results confirm that parser quality directly impacts the effectiveness of our discourse-aware tree structure. This suggests that improving discourse parsing capabilities is a promising direction for enhancing our approach's performance.

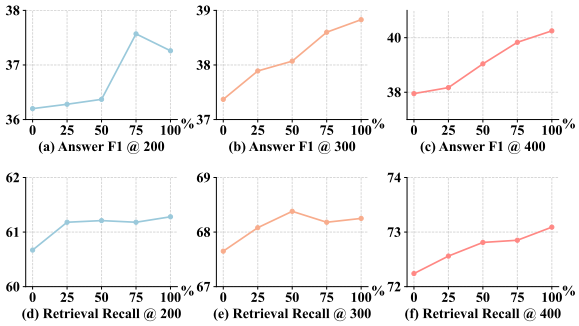

Figure 4: Impact of discourse parser capability on subsequent retrieval and question answering performance. Parsers used for comparative evaluation are trained on varying data scales, ranging from 0% to 100%.

5 Conclusion

In this paper, we presented DISRetrieval, a discourse-aware hierarchical retrieval framework that systematically incorporates rhetorical structure theory into long document question answering. Through three key innovations—sentence-level discourse parsing adapted for lengthy texts, LLM-enhanced node representations, and structure-aware evidence selection—our approach demonstrates substantial improvements over existing methods across multiple challenging benchmarks. Comprehensive experiments on QASPER, QuALITY, and NarrativeQA demonstrate consistent performance improvements across varying context lengths, embedding models, and generation architectures. Our ablation studies confirm that discourse structure provides more principled document organization than semantic clustering approaches, with the framework naturally adapting to different document types through multi-granularity retrieval. Beyond performance improvements, this work demonstrates that linguistically-grounded approaches remain valuable in the era of LLMs, offering principled alternatives to purely data-driven methods. Our framework opens new directions for incorporating structured linguistic knowledge into neural retrieval systems, with potential applications in legal document analysis, scientific literature review, and educational content processing.

Limitations

While DISRetrieval demonstrates substantial improvements across multiple datasets and architectures, we acknowledge several limitations that point to promising future directions. First, our discourse parser's performance inherently bounds the overall system effectiveness, as shown in Figure 4 where parser quality directly impacts downstream performance. However, our results demonstrate that even with a straightforward EDU-to-sentence adaptation, significant gains are achieved, validating the core hypothesis that discourse structure benefits long document retrieval. Future work will explore more sophisticated discourse parsing techniques, including domain-adaptive parsers and multi-granularity discourse analysis. Second, our adaptive summarization strategy using threshold τ is relatively simple but proves effective across diverse document types (τ=0 for academic papers, τ=50 for narratives). While this approach successfully balances computational efficiency with representation quality, more sophisticated dynamic thresholding based on content complexity and hierarchical position could further optimize the trade-off. Third, current evaluation metrics, though comprehensive across three challenging datasets, may not fully capture the nuanced benefits of discourse-aware retrieval, such as structural coherence preservation and hierarchical information flow. We plan to develop evaluation frameworks that better assess discourse-aware retrieval quality across multiple dimensions.

Ethical Statement

This research adheres to ethical principles in natural language processing research and does not raise significant ethical concerns. Our work focuses on improving document retrieval and question answering systems through discourse-aware hierarchical modeling, which has potential positive societal impacts by enhancing information access and comprehension. The datasets used in our experiments (QASPER, QuALITY, and NarrativeQA) are publicly available benchmark datasets that have been previously vetted by the research community and do not contain sensitive personal information or harmful content. Our discourse parsing approach operates on linguistic structures rather than content semantics, minimizing risks of bias amplification or misuse. The proposed DISRetrieval framework is designed as a general-purpose retrieval enhancement technique that can benefit various applications requiring long document understanding, such as academic research assistance, educational content processing, and legal document analysis. We acknowledge that like any information retrieval system, our method could potentially be misused if applied to spread misinformation or manipulate access to information, but such concerns are not specific to our approach and apply broadly to information retrieval technologies. We encourage responsible deployment of our system with appropriate safeguards and human oversight, particularly in high-stakes applications. All experiments were conducted using publicly available computational resources and open-source tools, ensuring reproducibility and transparency in our research process.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. GQA: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901.

Manoj Ghuhan Arivazhagan, Lan Liu, Peng Qi, Xinchi Chen, William Yang Wang, and Zhiheng Huang. 2023. Hybrid hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10680–10689.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lynn Carlson, Daniel Marcu, and Mary Ellen Okurovsky. 2001. Building a discourse-tagged corpus in the framework of Rhetorical Structure Theory. In *Proceedings of the Second SIGdial Workshop on Discourse and Dialogue*.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1870–1879.

Huiyao Chen, Yu Zhao, Zulong Chen, Mengjia Wang, Liangyue Li, Meishan Zhang, and Min Zhang. 2024. Retrieval-style in-context learning for few-shot hierarchical text classification. *Transactions of the Association for Computational Linguistics*, 12:1214–1231.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 615–621.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610.

Jiayu Ding, Shuming Ma, Li Dong, Xingxing Zhang, Shaohan Huang, Wenhui Wang, Nanning Zheng, and Furu Wei. 2023. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*.

Vanessa Wei Feng and Graeme Hirst. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521.

Mandy Guo, Joshua Ainslie, David Uthus, Santiago Ontanon, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. LongT5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 724–736.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020a. REALM: retrieval-augmented language model pre-training. *CoRR*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020b. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938.

Maor Ivgi, Uri Shaham, and Jonathan Berant. 2023. Efficient long-text understanding with short-text models. *Transactions of the Association for Computational Linguistics*, 11:284–299.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24:251:1–251:43.

Jiajie Jin, Xiaoxi Li, Guanting Dong, Yuyao Zhang, Yutao Zhu, Yongkang Wu, Zhonghua Li, Qi Ye, and Zhicheng Dou. 2025. Hierarchical document refinement for long-context retrieval-augmented generation. *arXiv preprint arXiv:2505.10413*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Naoki Kobayashi, Tsutomu Hirao, Hidetaka Kamigaito, Manabu Okumura, and Masaaki Nagata. 2020. Top-down rst parsing utilizing granularity levels in documents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8099–8106.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Yang Liu and Mirella Lapata. 2019. Hierarchical transformers for multi-document summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5070–5081.

Ye Liu, Kazuma Hashimoto, Yingbo Zhou, Semih Yavuz, Caiming Xiong, and Philip Yu. 2021. Dense hierarchical retrieval for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 188–200.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Aru Maekawa, Tsutomu Hirao, Hidetaka Kamigaito, and Manabu Okumura. 2024. Can we obtain significant success in RST discourse parsing by using large

language models? In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2803–2815.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. 2023. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36:53038–53075.

William C Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Daniel Marcu. 2000. *The theory and practice of discourse parsing and summarization*. MIT press.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358.

Hongjin Qian, Zheng Liu, Kelong Mao, Yujia Zhou, and Zhicheng Dou. 2024. Grounding language model with chunking-free in-context retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1311.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. 2024. RAPTOR: recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022. SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Nan Yu, Meishan Zhang, and Guohong Fu. 2018. Transition-based neural rst parsing with implicit syntax features. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 559–570.

Nan Yu, Meishan Zhang, Guohong Fu, and Min Zhang. 2022a. Rst discourse parsing with second-stage edu-level pre-training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4269–4280.

Nan Yu, Meishan Zhang, Guohong Fu, and Min Zhang. 2022b. RST discourse parsing with second-stage EDU-level pre-training. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4269–4280.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2023. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Mengying Yuan, Wenhao Wang, Zixuan Wang, Yujie Huang, Kangli Wei, Fei Li, Chong Teng, and Donghong Ji. 2025. Cross-document cross-lingual

Dataset	Avg. Sentence Length	Avg. Mid Node Depth		Avg. Leaf Num		Avg. Mid Node Percentage / %	
		SBERT	OpenAI	SBERT	OpenAI	SBERT	OpenAI
QASPER	22.32	5.29	5.53	12.64	13.37	54.15	65.13
QuALITY	14.86	13.78	17.09	67.67	88.67	73.93	88.53

Table 5: Statistical analysis of retrieved intermediate node characteristics across QuALITY and QASPER datasets. The table compares various key metrics: *Avg. Sentence Length* means the average sentence length of all documents; *Avg. Mid Node Depth* is the average depth of retrieved intermediate nodes; *Avg. Leaf Num* represents the average number of leaf nodes that each retrieved intermediate node maps to; *Avg. Mid Node Percentage* is the average percentage of intermediate nodes among the Top-20 retrieved nodes.

nli via rst-enhanced graph fusion and interpretability prediction. *arXiv preprint arXiv:2504.12324*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 11328–11339. PMLR.

Xinping Zhao, Dongfang Li, Yan Zhong, Boren Hu, Yibin Chen, Baotian Hu, and Min Zhang. 2024. SEER: Self-aligned evidence extraction for retrieval-augmented generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3027–3041.

A Extension of Technical Details

In this section, we introduce the specific details of our method which we cannot present in the main article due to the space limit.

A.1 Discourse Parser Details

The discourse parser architecture builds upon a transition-based system that constructs discourse trees through a series of well-defined actions. This system is particularly effective for capturing both local and global discourse relationships while maintaining computational efficiency.

The transition system maintains two primary data structures: a stack σ that holds partially built trees, and a queue β that contains unprocessed sentences. This design follows the intuition that discourse relationships often exist between adjacent text spans and can be built incrementally from bottom to top.

The system builds the tree step by step using three basic operations:

1. A "shift" action moves a sentence from the queue to the stack when we need new content to process.

2. A "reduce" action combines two adjacent subtrees on top of the stack into a new subtree by identifying their discourse relationship.

3. A "pop root" action concludes the process when we have successfully built a complete tree.

Each state of the system is represented as $c = (\sigma, \beta)$, starting from $c_0 = ([\,], S_i)$ with all sentences in the queue, and ending at $c_f = ([T_i], [\,])$ with a complete discourse tree T_i.

The transition system follows a deterministic process guided by the neural scoring model:

1. Initialize $\sigma = [\,]$ and $\beta = S_i$.

2. While β is not empty or $|\sigma| > 1$: (a) If $|\sigma| < 2$ and β is not empty, perform a "shift" action to move the next sentence from β to σ; (b) Else if β is empty, perform a "reduce" action to combine the top two subtrees in σ; (c) Else, use the neural scoring model to decide between a "shift" or "reduce" action based on the current state of σ and β.

3. Return the single tree T_i remaining on the stack σ.

The scoring model considers the three topmost subtrees on the stack (s_1, s_2, s_3) and the next sentence in the queue q_1. This design is motivated by several factors:

1. s_1 and s_2 are the immediate candidates for the next potential "reduce" action.

2. s_3 provides crucial context about the recently built structure.

3. q_1 helps determine if we should introduce new content via a "shift" action.

For each tree node v, we compute its representation h_v recursively:

$$\mathbf{h}_v = \begin{cases} \text{PLM}(s_i), & \text{if } v \text{ is } \textit{sentence} \\ \frac{1}{|C(v)|}\sum_{u \in C(v)} \mathbf{h}_u, & \text{if } v \text{ is } \textit{relationship} \end{cases} \tag{7}$$

where $C(v)$ denotes the set of child nodes of v, and PLM($\cdot$) is a pre-trained language model that encodes the semantic meaning of individual sentences. The action scores are then computed as:

$$\mathbf{y}(a) = \mathbf{W}(\mathbf{h}_{s_1} \oplus \mathbf{h}_{s_2} \oplus \mathbf{h}_{s_3} \oplus \mathbf{h}_{q_1}) + \mathbf{b}, \tag{8}$$

where $\oplus$ concatenates the representations to capture their interactions, and $\mathbf{W}$ and $\mathbf{b}$ are learnable parameters. The probability of taking action a is computed using a softmax function over the action scores:

$$p(a|c) = \frac{\exp(\mathbf{y}(a))}{\sum_{a' \in A} \exp(\mathbf{y}(a'))}, \tag{9}$$

where A is the set of valid actions at state c. We train the model using supervised learning with gold-standard discourse trees. The objective function combines cross-entropy loss for action prediction with L2 regularization:

$$\mathcal{L}(\theta) = -\log p(a^*|c) + \frac{\lambda \|\theta\|_2}{2}, \tag{10}$$

where a^* is the correct action derived from gold-standard trees. During inference, we greedily select the highest-scoring action at each step, effectively building the tree in a bottom-up manner while maintaining the discourse relationships between text spans.

A.2 Iterative Tree Construction Process

The complete algorithm of our DISRetrieval is presented in Algorithm 2. Below we detail the iterative tree construction process, which corresponds to Stage 1 of the algorithm. The construction of discourse trees for long documents presents unique challenges in balancing computational efficiency with structural integrity. We propose an iterative construction strategy that addresses these challenges through a hierarchical, phase-wise approach. This method effectively manages computational resources while preserving discourse relationships at multiple granularity levels. Our iterative process consists of three distinct phases, each designed to handle specific aspects of the tree construction:

Phase 1: Paragraph-level Tree Construction. The first phase focuses on building local discourse trees for individual paragraphs. For each paragraph p_i containing sentences $\boldsymbol{S_i} = \{s_{i,1}, s_{i,2}, ..., s_{i,m}\}$, we construct a local discourse tree T_i using our transition-based parsing system. This phase is particularly efficient as it:

1. Initializes each paragraph's parsing state with an empty stack and sentence queue: $c_0 = ([\,], S_i)$.

2. Processes paragraphs independently, enabling parallel computation.

3. Applies transition actions iteratively until a complete tree is formed.

4. Stores both the resulting paragraph-level tree T_i and its root representation $\mathbf{h}_{T_i}$.

Phase 2: Document-level Tree Construction. The second phase focuses on capturing document-level discourse structure. After obtaining all paragraph-level trees $T_1, T_2, ..., T_n$, we:

1. For each paragraph-level tree T_i, apply bottom-up LLM-enhanced summarization:

   - For each non-leaf node v with children v_l and v_r:

     $$v^* = \begin{cases} f_{\text{LLM}}(v_l, v_r), & \text{if } |v_l| + |v_r| \geqslant \tau \\ f_{\text{merge}}(v), & \text{otherwise} \end{cases} \tag{11}$$

     where $|v_l|$ and $|v_r|$ are the textual content length of child nodes
   - Continue until reaching root node to obtain semantic unit u_i

2. Form the semantic units set $U = \{u_1, u_2, ..., u_n\}$ from root representations

3. Apply the discourse parser to these units to construct a document-level tree T_{doc} using the same transition-based parsing system:

   $$T_{\text{doc}} = f_{\text{discourse}}(U) \tag{12}$$

4. Apply bottom-up LLM-enhanced summarization to T_{doc}:

- For each non-leaf node $v \in T_{\text{doc}}$ with children v_l and v_r:

$$t_v = \begin{cases} f_{\text{LLM}}(t_l \oplus t_r), & \text{if } |t_l \oplus t_r| \geqslant \tau \\ t_l \oplus t_r, & \text{otherwise} \end{cases} \tag{13}$$

- Process nodes level by level from bottom to top until reaching the root of T_{doc}

This step effectively captures the high-level discourse relationships between paragraphs while maintaining computational efficiency by working with LLM-enhanced condensed representations at both paragraph and document levels.

Tree Integration. The final phase integrates the local and global structures into a unified discourse tree. This integration is accomplished through a careful replacement process:

$$T_{\text{D}} = \text{Replace}(T_{\text{doc}}, \{(u_i, T_i) | i \in [1, n]\}) \tag{14}$$

where Replace$(\cdot)$ replaces each unit u_i in the document-level tree with its corresponding paragraph-level tree T_i. This integration phase carefully preserves both local and global discourse relationships by maintaining the internal structure of paragraph-level trees while retaining the document-level relationships established in Phase 2, ultimately creating a seamless hierarchical structure that spans the entire document and effectively captures discourse relationships at all levels of granularity.

The resulting tree structure effectively captures discourse relationships at multiple levels of granularity, from sentence-level connections within paragraphs to broader document-level organizational patterns. This approach not only ensures computational efficiency through its phase-wise processing but also maintains the integrity of discourse relationships throughout the document hierarchy.

A.3 Detailed LLM Enhancement Process

This section provides a detailed illustration of the bottom-up LLM enhancement process described in Phase 2 of our discourse tree construction methodology (Section 3.1.2).

A.3.1 LLM Enhancement Workflow

Figure 5 demonstrates the complete workflow of our LLM enhancement process using a concrete example from a research paper paragraph. The process consists of three main components:

Input Structure. The top portion shows the input paragraph with its initial sentence-level discourse tree structure. Each sentence is represented as a leaf node, connected through discourse relations such as *Contrast*, *List*, *Summary*, and *Topic*. This hierarchical structure captures the rhetorical organization of the paragraph content.

Adaptive Processing Strategy. The center portion illustrates our two-way processing strategy based on text length thresholds. When the combined length of child nodes ($|t_l| + |t_r|$) falls below the threshold τ, we apply direct concatenation ($\bigoplus$) to preserve the original textual details. Conversely, when the combined length exceeds τ, we employ LLM-based summarization to generate more concise representations while retaining essential semantic information.

Enhanced Tree Structure. The bottom portion shows the resulting enhanced discourse tree with progressively generated semantic summaries. Each internal node now contains meaningful textual representations that capture the essence of its subtree content. For example, the *Contrast* relation between different learning methods is transformed into the concrete summary "Different learning methods suit different applications", while maintaining the hierarchical discourse structure.

A.3.2 Key Benefits

This LLM enhancement process provides several advantages:

- **Semantic Preservation**: Maintains essential meaning while reducing textual complexity
- **Computational Efficiency**: Adaptive thresholding prevents unnecessary LLM calls for short text segments
- **Hierarchical Coherence**: Preserves discourse relationships at multiple granularity levels
- **Downstream Compatibility**: Generates representations suitable for sentence-level discourse parsing

The transformation from abstract rhetorical relations to concrete semantic representations enables our framework to effectively bridge the gap between discourse structure and semantic understanding, facilitating improved retrieval performance in long document question answering tasks.

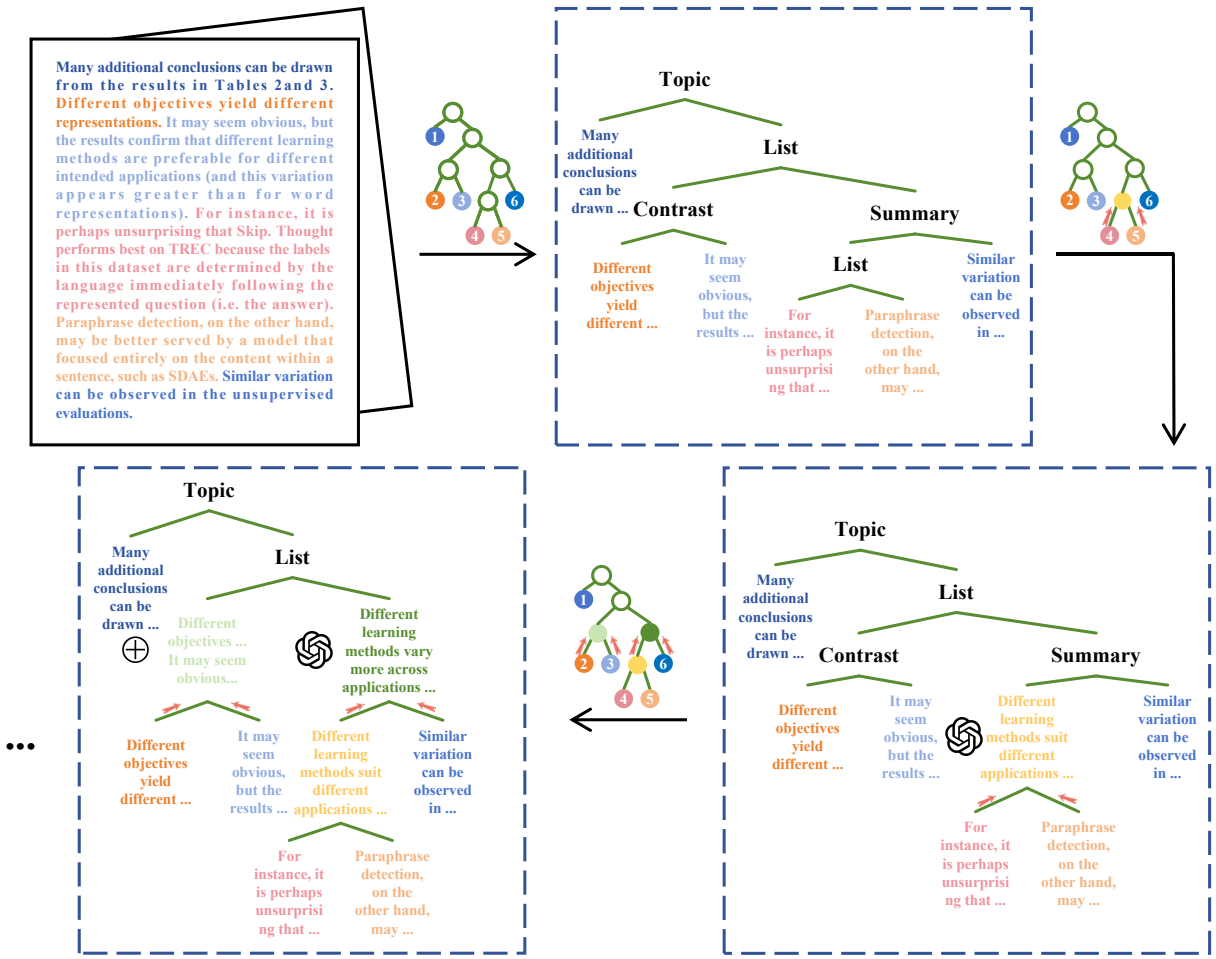

Figure 5: Illustration of bottom-up LLM enhancement in Phase 2 of discourse tree construction. Top: Input paragraph with its initial discourse tree structure. Center: Two-way processing strategy based on text length - using direct concatenation (⊕) when combined length is below threshold τ, and LLM-based summarization when above threshold. Bottom: Enhanced discourse tree with progressively generated semantic summaries following Equation 2, demonstrating the transformation from rhetorical relations to concrete semantic representations.

B Detailed Experiment Settings

B.1 Dataset Specifications

The QASPER dataset is constructed from NLP research papers, containing questions that require deep understanding of technical content. All answerable questions are annotated with multiple reference answers to account for different valid expressions of the same information. The ground-truth evidence spans are carefully annotated by domain experts to ensure the reliability of retrieval evaluation.

Dataset	Used Set	Question Num	Doc. Num	Avg. words	Max. words
QASPER	test	1456	416	4170	21,165
QuALITY	dev	2086	115	5022	5967
NarrativeQA	test	10,577	355	51,372	346,902

Table 6: Detailed infomation of the datasets used in our experiments.

The QuALITY dataset consists of long passages primarily drawn from fiction stories and magazine articles, with an average length longer than typical QA datasets. Each question is accompanied by multiple choice options that test comprehensive understanding of the passage. We utilize the validation set with publicly available labels for our experiments to enable thorough analysis and comparison. Notably, to enable more extensive experimentation, we evaluated on the validation set with publicly available labels rather than the submission-required test set.

The NarrativeQA dataset is a significant advancement in evaluating machine reading comprehension. Unlike traditional datasets that focus on short passages, NarrativeQA emphasizes understanding long-form narratives, such as entire books or full-length movie scripts. One of the defining features of NarrativeQA is the length of its documents. On

average, each document contains 51,372 words. Notably, some individual documents extend up to 346,902 words, making it one of the longest reading comprehension datasets available. More detailed information is presented in Table 6.

B.2 Model Specifications

For semantic embeddings, the Sentence-BERT model (multi-qa-mpnet-base-cos-v1) is based on the MPNet architecture with 768-dimensional representations, specifically optimized for question-answering tasks. The OpenAI embedding model (text-embedding-3-large) represents their latest advancement in semantic representation capabilities.

B.3 Implementation Details

In the discourse analysis process, we preserve sentence boundaries throughout all splitting operations to maintain semantic coherence. For the RAPTOR baseline implementation, we follow the original paper's collapsed tree approach where all nodes are considered simultaneously during retrieval. The tree construction process in our Bisection baseline ensures a nearly balanced binary structure through recursive division of the sentence set.

B.4 Experiments Compute Resources

For the training of the sentence-level Discourse Parser, we used 1 NVIDIA A100-40G GPU. All other experiments incluing document discourse parsing, LLM summarization and node embedding, as well as the retrieval and generation processes, are conducted using 4 NVIDIA A800-80G GPUs.

B.5 Prompts

We provides the specific prompts used in our approach to ensure reproducibility.

In the prompt templates: (1) fixed prompts are displayed in black. (2) Input text is highlighted in deep red. (3) The retrieved context is colored in purple. (4) The output generated by the LLM is presented in green.

Prompt for Intermediate Node Text Summerization

System: You are a helpful assistant.

User: Write a summary of the given sentences, keeps as more key information as possible. Only give the summary without other text. Make sure that the summary no more than 200 words.
Given text: {left child node text} {input cild node text}

LLM: Summerization result.

Prompt for Question Answering on QASPER Dataset

System: You are a helpful assistant.

User: Using the folloing information: {context}. Answer the following question in less than 5-7 words, if possible. For yes or no question, only return 'yes' or 'no'.
question: {question}

LLM: Question Answering result.

Prompt for Question Answering on QuALITY Dataset

System: You are a helpful assistant.

User: Given context: {context}.
Answer the following multiple-choice question:{question}

LLM: The correct answer is (A). The context provided...

C Extended Experiment Analysis

C.1 Computational Efficiency and Scalability

To evaluate the computational efficiency and scalability of our approach, we conducted comprehensive timing analysis across varying document lengths. We systematically selected documents ranging from 10,000 to 100,000 words, performed discourse-aware tree construction and LLM enhancement, and measured the computation time for each process. All experiments were conducted on a single A800 80G GPU, with LLM enhancement performed using the LLaMA3.1-8B-Instruct model accelerated by the vLLM framework. To ensure statistical reliability, we sampled 10 documents for each length range and report the average processing

time with standard deviation. The detailed results are presented in Figure 6.

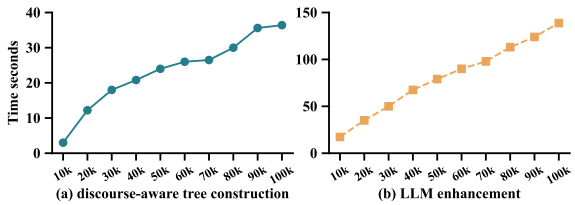

Figure 6: Average processing time of discourse-aware tree construction and LLM enhancement across varying document lengths.

The processing time for both discourse-aware tree construction and LLM enhancement exhibits near-linear scaling with document length, demonstrating favorable computational complexity for long documents. Specifically, tree construction for documents up to 100,000 words requires less than 40 seconds, while LLM enhancement completes within 140 seconds, confirming practical feasibility for real-world applications. The linear scaling behavior indicates that our method maintains consistent per-word processing efficiency regardless of document length. Moreover, as discussed in Section 4.3, the size and performance of the summarization model have no significant impact on the results, which further suggests that our method can be deployed with a modest computational cost.

Crucially, both tree construction and LLM enhancement are query-independent preprocessing operations performed once per document, allowing costs to be amortized across multiple queries on the same document. For instance, processing a 50,000-word document requires approximately 90 seconds of preprocessing but can subsequently serve unlimited queries with minimal additional overhead. In practical deployment scenarios where documents receive multiple queries over time, this one-time preprocessing cost becomes negligible compared to the cumulative benefits of improved retrieval accuracy.

Compared to traditional chunking approaches that require no preprocessing, our method introduces modest upfront costs but delivers significantly better retrieval quality. The preprocessing overhead is comparable to other hierarchical methods like RAPTOR, while providing superior performance through linguistically-grounded discourse modeling.

C.2 Analysis of Multi-granularity Adaptive Retrieval

We conducted a thorough statistical analysis to evaluate the multi-granularity adaptive retrieval capability of our method. For each query, we retrieved the Top-20 nodes from the full discourse-aware tree, and analyzed the characteristics of retrieved intermediate nodes across datasets.

Retrieved Node Characteristics Across Datasets. Table 5 reveals significant differences in the retrieved intermediate nodes between datasets. QuALITY exhibits substantially higher values for intermediate node metrics compared to QASPER: the average depth of retrieved intermediate nodes is 2.6 times greater with SBERT embeddings (13.78 vs. 5.29) and 3.1 times greater with OpenAI embeddings (17.09 vs. 5.53). Similarly, the average number of leaf nodes that each retrieved intermediate node maps to is 5.4 times higher with SBERT (67.67 vs. 12.64) and 6.6 times higher with OpenAI (88.67 vs. 13.37). The percentage of intermediate nodes among Top-20 retrieved results is also higher for QuALITY across both embedding methods (73.93% vs. 54.15% with SBERT; 88.53% vs. 65.13% with OpenAI). These differences exist despite QASPER having a 50% longer average sentence length (22.32 vs. 14.86 words).

Distributional Analysis. Figure 7 further illustrates these distinctions through distributional analysis. The sentence length distribution (Fig. 7a) shows that QuALITY is heavily skewed toward shorter sentences (5-15 words), while QASPER has a broader, more even distribution extending to 40+ words. The node depth distributions (Fig. 7b,c) demonstrate that retrieved intermediate nodes from QuALITY frequently reach depths of 10-15, whereas those from QASPER are predominantly shallower (depths below 10), consistent across both embedding methods.

Retrieval Strategy Adaptation. These patterns reveal how our retrieval approach adapts to different document structures. For narrative fiction in QuALITY with abundant dialogue and shorter sentences, the retrieval system favors higher-level intermediate nodes that aggregate related content, as evidenced by the higher percentage of intermediate nodes and greater node depths. This suggests that hierarchical composition is particularly beneficial for documents where meaning is distributed across multiple short sentences. In contrast, for

research papers in QASPER with longer, more self-contained sentences, the retrieval system relies less on deep hierarchical structures, as shown by the shallower node depths and lower percentage of intermediate nodes. This indicates that relevant information in scientific documents can often be retrieved effectively with less compositional processing.

These findings empirically validate that our approach effectively adapts to diverse document structures and information needs. Our discourse-aware tree construction naturally induces appropriate segmentations based on document characteristics rather than arbitrary chunking, enabling multi-level text composition that overcomes the limitations of fixed-granularity methods. Through these mechanisms, DISRetrieval provides a multi-granularity retrieval framework that adaptively selects appropriate granularity levels across different document types, accommodating their inherent structural variations.

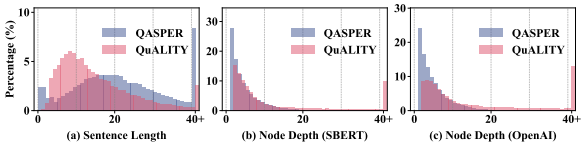

Figure 7: Comparative analysis of distribution difference of two datasets. Figure (a) shows the difference on Sentence Length, while (b) and (c) demonstrate the distribution of retrieved intermediate node depth across different embedding models.

C.3 Ablation study on different Top-K settings

We conduct comprehensive experiments to investigate the impact of varying the value of K from 1 to 20, with results presented in Figure 8.

Our analysis reveals several noteworthy patterns:

Performance Trends. Across all configurations, performance initially improves as K increases, typically peaking around $K = 5$ (marked by the vertical dashed line), after which it either plateaus, gradually declines, or exhibits minor fluctuations. This consistent pattern suggests an optimal balance point where sufficient context is provided without introducing excessive noise.

Dataset-Specific Behaviors. QuALITY demonstrates more pronounced performance variations with changing K values compared to QASPER, with performance differences of up to 2 percentage points between optimal and suboptimal settings. This higher sensitivity likely reflects QuALITY's more complex narrative structure, where precise evidence selection is particularly crucial.

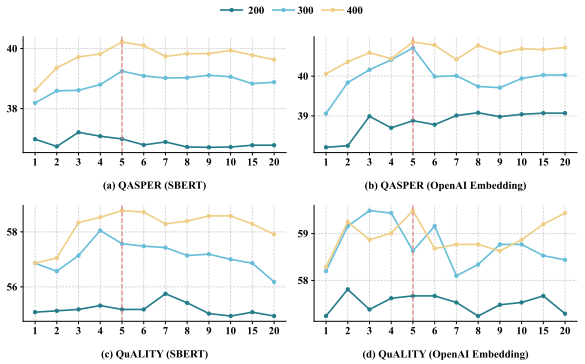

Figure 8: Ablation results with different values of K. The horizontal axis represents different choices of K, and the vertical axis indicates generation performance (F1-match for QASPER and accuracy for QuALITY). All question answering tasks are conducted on the UnifiedQA-3B model.

Context Length Impact. Longer context windows (300 and 400 words) consistently outperform shorter ones (200 words) across all K values. Notably, the performance advantage of longer contexts is most significant when K is small. This suggests that when the system retrieves fewer evidence segments, the comprehensiveness of each individual segment becomes critical, as the model must extract all necessary information from a limited number of passages.

Based on these observations, we adopt $K = 5$ as our optimal setting for all main experiments, as it consistently delivers strong performance across datasets and embedding methods while maintaining computational efficiency.

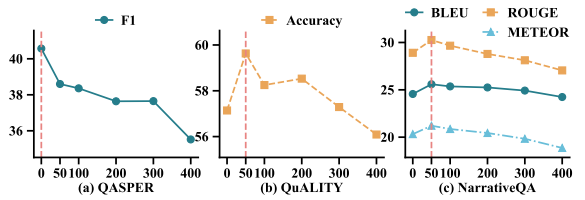

Figure 9: Ablation results with different values of τ across three datasets. Sentence-BERT is used as the embedding model and UnifiedQA-3B as the question-answering model, with a context length of 400 words.

C.4 Ablation study on parameter τ across different datasets

Equation 2 employs a threshold parameter τ to decide whether use an LLM for summarization or directly merge the subtree. To assess the impact of this parameter and identify the optimal settings, we conduct experiments on three datasets. The results are shown in figure 9.

Performance Trends. For QASPER, the F1 score is highest when $\tau = 0$ and gradually declines as τ increases, indicating that summarization is more beneficial than direct merging sentences of subtrees. In contrast, QuALITY and NarrativeQA reach their peak performance at $\tau = 50$ and gradually decline afterward, only approaching the level observed at $\tau = 0$ when τ reaches 300 and 200 respectively, suggesting that a moderate merging strategy is essential for these datasets.

Dataset Analysis These results are consistent with the characteristics of the datasets. QASPER consists of academic papers, where sentences are relatively independent and logically complete. So keeping sentences as the minimal unit allows for more precise information extraction. In contrast, QuALITY and NarrativeQA mainly contain novels and movie scripts, which include numerous dialogues and short sentences that require a relatively complete context to convey the correct semantics. For these two datasets, merging subtrees based on rhetorical structure parsing results enables logically related sentences to form more coherent text segments, preventing semantic fragmentation. At the same time, moderate merging is crucial, as excessively long text segments may introduce redundancy. Therefore, we set $\tau = 0$ for QASPER, $\tau = 50$ for QuALITY and NarrativeQA respectively. This can also provide guidance for other datasets: for documents with more formal and complete sentences, smaller τ values are preferable, whereas for documents with shorter and more fragmented sentences, relatively larger τ values should be used.

Algorithm 2: DISRetrieval: Discourse Structure-based Long Document Retrieval

```
Input: Document D containing sentences S_i, paragraphs P_i; Query q
Output: Retrieved evidence segments E
/* Stage 1: Discourse-Aware Tree Construction                                   */
for each paragraph P_i ∈ D do
    Initialize stack σ and sentence queue β ← S_i;
    while β not empty OR |σ| > 1 do
        /* RST parsing operations                                               */
        a_t ← Select action from {shift, reduce, pop_root};   // Determine next parsing action
        switch a_t do
            case shift do
                Move next sentence from β to σ;                   // Add new sentence to stack
            end
            case reduce do
                s_2, s_1 ← Pop top two elements from σ;
                Determine discourse relation r between s_1 and s_2;   // Using RST relations
                Create new node with relation r;
                Push new node to σ;
            end
            case pop_root do
                Final tree node ← Pop from σ;                     // Complete the tree
            end
        end
    end
    T_i ← Resulting discourse tree;
    /* LLM Enhancement for Paragraph Tree                                       */
    for each non-leaf node v ∈ T_i (bottom-up order) do
        v_l, v_r ← Get left and right children;
        if |v_l| + |v_r| ≥ τ then
            v* ← f_LLM(v_l, v_r);                         // Generate concise summary using LLM
        else
            v* ← f_merge(v);                                  // Merge the subtree rooted at v
        end
    end
    Store root text representation t^i_root for T_i;
end
Initialize stack σ and queue β ← {t^1_root, ..., t^n_root};
while β not empty OR |σ| > 1 do
    /* Similar RST parsing at document level                                    */
    Apply RST parsing operations as in Phase 1;
    Build document tree T_doc;
end
/* Stage 2: Node Representation                                                 */
for each non-leaf node v ∈ T_doc (bottom-up order) do
    t_l, t_r ← Get content from left and right children;
    if |t_l| + |t_r| ≥ τ then
        t_v ← f_LLM(t_l, t_r);                  // LLM-based enhancement for document-level nodes
    else
        t_v ← t_l ⊕ t_r;                // Direct concatenation for short text, don't need to merge subtree
    end
end
T_D ← Replace leaf nodes in T_doc with corresponding T_i;          // Integrate trees into unified structure
for each node v ∈ T_D do
    e_v ← Encoder(t_v);                                   // Generate dense vector representations
end
/* Stage 3: Hierarchical Evidence Retrieval and Selection                       */
e_q ← Encoder(q);                                         // Transform query to embedding space
scores ← {}, E ← {};
for node v ∈ T_D do
    scores[v] ← cosine(e_q, e_v);                             // Compute relevance scores
end
Apply Algorithm 1 for evidence selection;
return E
```