DiscoRAG: A Discourse-Aware Agent for Query-Based Summarization of Long Documents

Alexander Chernyavskiy, Lidiia Ostyakova, and Dmitry Ilvovsky
HSE University, Russia
alschernyavskiy@gmail.com, ostyakova.ln@gmail.com
dilvovsky@hse.ru

Abstract

Query-based summarization of long documents is often tackled with retrieval-augmented generation (RAG). However, conventional RAG models exhibit limitations when applied to narrative texts, where crucial evidence is often implicit and distributed. This exposes a distinct class of "discourse-aware" queries that require specialized, structure-aware models. To address this, we introduce DiscoRAG, a framework that leverages Rhetorical Structure Theory (RST). By modeling the document as a discourse tree, DiscoRAG navigates its structure, explicitly using rhetorical relations to focus on and aggregate evidence from globally related segments. Furthermore, our pipeline integrates a classifier that assesses query complexity to dynamically select the most efficient retrieval strategy. We evaluate our DiscoRAG against standard and extended-context RAG pipelines on the SQuALITY dataset, which we release augmented with questions requiring deep discourse reasoning and integration of the global narrative. Our results demonstrate that this method sizeably outperforms these baselines, demonstrating its superior ability to assemble a coherent, contextually rich evidence base by interpreting the global narrative structure rather than relying on local semantic similarity.

Keywords: RAG, RST, discourse analysis

1. Introduction

The challenge of extracting precise, query-relevant knowledge from long documents, such as scientific literature, legal contracts, or corporate reports, has established Query-Based Summarization (QBS) as a critical research area. The dominant approach, Retrieval-Augmented Generation (RAG), addresses this by first retrieving relevant passages and then synthesizing an answer (Lewis et al., 2020). This paradigm has recently been enhanced by agent-based frameworks that dynamically interact with documents through multi-step tool use, promising more sophisticated reasoning (Schick et al., 2023; Singh et al., 2025; Huang et al., 2025; Qu et al., 2025). Despite their advancements, these systems are fundamentally constrained by a "flat" view of text. Their retrieval and reasoning tools operate at the level of lexical or shallow semantic similarity, processing documents as an unstructured collection of passages.

This structural blindness renders them ineffective for complex queries that require reasoning over the document's rhetorical organization. For instance, an agent might fail to connect an anaphoric reference like "this theory" to its definition sections away, or miss contrastive relations critical for queries about conflicting viewpoints. As illustrated conceptually in Figure 1, a standard RAG model may retrieve a surface-level fact but completely miss its underlying cause, as there is no explicit structural link to follow.

Narrative QA represents a particularly challenging form of long-context reasoning, as answering a query often requires integrating information distributed across distant parts of a story. Linguistic analysis, particularly discourse structure, offers a powerful paradigm to overcome these challenges. Recent work has therefore explored incorporating structural signals into retrieval pipelines. For example, ChronoRAG organizes retrieved passages according to their temporal order, demonstrating that maintaining chronological coherence improves performance on narrative QA benchmarks(Aguilar,

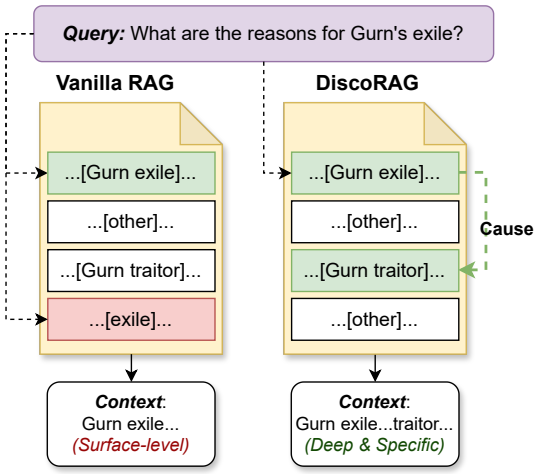

Figure 1: Comparison of retrieval strategies. (a) Standard (Vanilla) RAG retrieves a lexically relevant but incomplete fragment. (b) DiscoRAG navigates from an initial node, following a "Cause" relation to uncover the underlying reason.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 2062–2075
11–16 May 2026.

2025). Similarly, Laitenberger et al. (2025) show that even a simple retrieval strategy that preserves the original document order of retrieved passages (DOS-RAG) can outperform more complex multi-stage pipelines on long-context QA datasets, including NarrativeQA. By representing a document's logical flow as a tree of rhetorical relations, we can equip an agent with the tools for more sophisticated navigation. While discourse-aware methods have improved related tasks like QA (Nair et al., 2023; Du et al., 2023) and text understanding (Chernyavskiy et al., 2024a), this approach remains critically underexplored in agentic summarization systems.

Our Contribution. We bridge this gap by introducing a novel agent framework with a linguistically structured environment. Our key innovation is an environment state explicitly modeled as a discourse tree, enabling three fundamental capabilities:

- **Dynamic Context Focus**: The agent navigates discourse trees to zoom into relevant subtrees (e.g., focusing specifically on evidence-bearing segments for "supporting arguments" queries).
- **Relation-Aware Reasoning**: The agent leverages explicit rhetorical relations (e.g., Contrast, Elaboration) to address complex, non-factoid queries. We show that this leads to superior end-to-end answer quality where methods reliant on lexical similarity fail.
- **Discourse-Aware Problem Space**: Establishing that "discourse-aware" queries form a distinct, identifiable class that justifies specialized, structure-aware models

To our knowledge, this is the first integration of discourse structure as a core agent environment for summarization. Our comprehensive empirical validation shows this approach leads to higher-quality context and superior end-to-end performance.

2. Related Work

2.1. Preliminaries: RST

Rhetorical Structure Theory (RST), introduced by Mann and Thompson (Mann et al., 1989), provides a framework for analyzing the functional organization of texts. Central to RST is the representation of documents as hierarchical tree structures constructed through recursive composition.

The discourse tree construction process involves two fundamental stages. It starts with segmenting the text into elementary discourse units (EDUs), which are minimal coherent propositions that function as the leaf nodes of the discourse tree. These units are then connected by rhetorical relations, classified as either mononuclear or multinuclear.

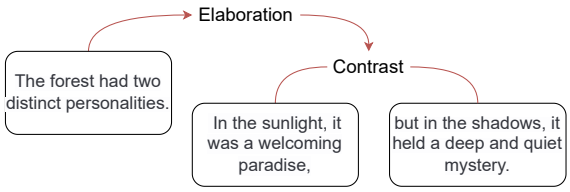

Figure 2: Discourse tree example. Arrows are drawn from nuclei to satellite nodes.

A mononuclear relation links a core Nucleus to a subordinate Satellite (e.g., Evidence), while a multinuclear relation connects multiple nuclei of equal importance (e.g., Contrast).

Figure 2 illustrates this structure using a scientific abstract drawn from our dataset. The initial statement, *"The forest had two distinct personalities,"* serves as the Nucleus or the main idea. This claim is then illustrated by a Satellite whose internal structure forms a Contrast relation, supporting the overall thesis.

2.2. Linguistic Approaches for Long-Document Understanding

Linguistically-informed methods have demonstrated significant value in query-based summarization and long-document QA. Among them, discourse structure analysis, particularly Rhetorical Structure Theory (RST), has proven essential for modeling information salience and coherence in extended texts and dialogues (Chernyavskiy et al., 2024b). Xu et al. (2020) integrated discourse trees into extractive summarization, showing 3-5 point ROUGE improvements on scientific papers by leveraging rhetorical relations. For QA, Chen et al. (2025) demonstrated that discourse-aware passage selection improves answer precision for long documents.

Recent benchmarks like ZeroSCROLLS (Shaham et al., 2023) explicitly highlight limitations of current systems on discourse-sensitive tasks such as SQUALITY (Wang et al., 2022a), where models struggle with query-dependent synthesis of distributed arguments.

2.3. Evolution of Agentic RAG Systems

Retrieval-Augmented Generation (RAG) architectures have evolved through four distinct phases:

1. Vanilla RAG: Static retrieval from fixed corpora followed by generation (Lewis et al., 2020);
2. Modular RAG: Decomposing the retrieval and generation pipeline into independent, reusable components, enabling domain-specific optimization and task adaptability (Gao et al., 2023);

3. Advanced RAG: Graph-based retrieval and RL for context selection (Peng et al., 2024);

4. Agentic RAG: Systems with dynamic planning, tool invocation, and stateful reasoning (Singh et al., 2025).

Deep Research agents (Huang et al., 2025) represent the state-of-the-art in agentic RAG, characterized by multi-iteration tool use (web search, APIs, calculators) and dynamic task decomposition via frameworks such as ReAct (Yao et al., 2023).

However, these agents fundamentally rely on lexical or semantic similarity for text processing tools (chunking, keyword search), lacking mechanisms to leverage discourse structure or rhetorical relations, a critical limitation for queries requiring understanding of argument flow.

Recent research has explored hybrid RAG systems that dynamically route queries between simple and complex processing pipelines based on estimated difficulty. Zhang et al. (2025) describe methods to assign tasks across different models depending on query characteristics. Similarly, Ding et al. (2025) propose the BEST-Route framework, which adaptively routes queries based on their estimated complexity, enabling optimal allocation of computational resources while maintaining answer quality. We adopt a similar hybrid approach by activating our discourse-aware agent only when linguistic complexity warrants it. Our routing classifier considers discourse-specific features: discourse relation density (queries referencing more than two rhetorical relations), cross-sectional dependencies, and so forth. Simple factoid queries (e.g., "What year was the experiment conducted?") bypass agent processing entirely.

This hybrid design directly addresses a **critical research gap**: while agentic systems excel at tool orchestration and reasoning over retrieved content, they still process text as isolated segments, lacking awareness of its rhetorical structure and limiting their ability to perform nuanced content selection in long-document QBS tasks. Given the solid foundations of linguistic approaches, integrating discourse representations as a core reasoning framework constitutes a promising yet underexplored direction toward achieving coherent understanding of long documents.

3. DiscoRAG

3.1. Processing Pipeline

Our hybrid query processing pipeline is illustrated in Figure 3. The system is designed to flexibly handle a wide range of user queries by dynamically selecting the most appropriate processing strategy based on the linguistic complexity of the input.

The pipeline begins with a classifier that receives the user's query and evaluates its discourse complexity using a RoBERTa-based model. This initial assessment determines whether the query requires advanced discourse-aware reasoning or can be addressed with a more straightforward retrieval approach.

If the classifier determines that the query is simple and does not require discourse-level analysis, it is routed directly to the **Base RAG** module. Here, the system performs standard passage retrieval from the document collection, followed by answer generation. This path ensures efficient processing for routine queries, minimizing computational overhead while maintaining high-quality responses.

For queries identified as complex, those that exhibit discourse phenomena or require integrating information across multiple parts of the document, the pipeline activates a more advanced processing branch. In this case, the query is passed to the **DiscoRAG Agent**, which operates within a discourse-based environment. The agent is equipped to navigate the document's discourse structure, leveraging specialized tools and memory components to access and organize relevant information. The agent's actions are guided by the current state within the discourse tree, allowing it to move between different relations and nodes as needed.

Throughout the process, a **memory** block is available to store and retrieve intermediate results. Once the necessary information has been collected, either through the Base RAG module or through the agent's discourse-aware exploration, the pipeline moves to the summary generator. This component synthesizes the retrieved content into a coherent, query-focused summary that is then returned to the user.

By combining a fast, direct retrieval path for simple queries with a more sophisticated, discourse-aware approach for complex cases, our system achieves both efficiency and depth.

3.2. Discourse Environment Construction

Environment construction produces a navigable discourse structure that serves as the foundation for agent-based reasoning. The process consists of the following steps:

1. Pre-processing: Clean the document by removing HTML tags and irrelevant formatting.

2. Segmentation: Divide the text into Elementary Discourse Units (EDUs), which represent the minimal building blocks of discourse.

3. Aggregation: Group EDUs into paragraphs to preserve the local context.

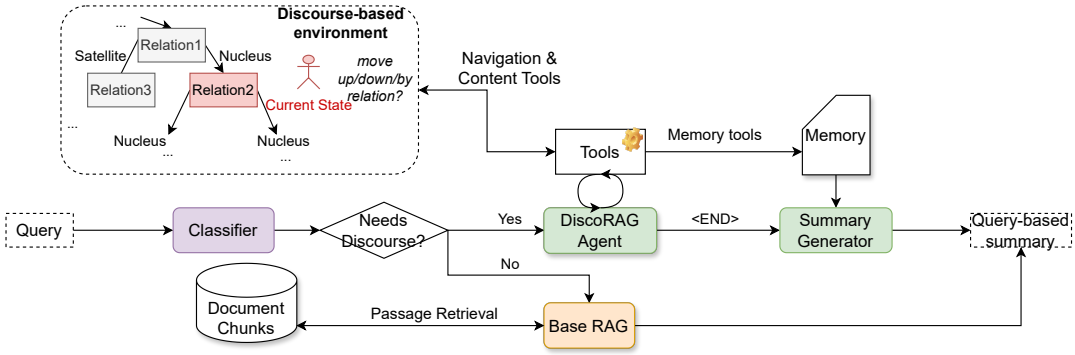

Figure 3: Overview of the **DiscoRAG** query processing pipeline. Queries are first assessed for linguistic complexity by a RoBERTa-based classifier. Simple queries are routed to a basic RAG module, while complex queries activate a discourse-aware agent.

4. Tree Construction: Build an RST-based tree where the nodes are connected by rhetorical relations.

Node Representation: Each node in the tree is represented as a structured object containing:

- `id`: A unique identifier for the node
- `content`: The text span associated with the node (typically 2-5 sentences)
- `path`: The discourse path to the root, represented as a sequence of tuples `(Nuclearity, Relation, ParentID)`
- `relations`: Incoming and outgoing edges (in the form of Relation and TargetID), capturing the node's rhetorical connections
- `metadata`: Additional information such as character span and structural position within the document (e.g. path by HTML headers)

3.3. Agent Toolset

To enable effective discourse navigation and content synthesis, our agent is equipped with a suite of specialized tools, grouped into three main categories (see Table 1):

- **Navigation Tools:** Allow the agent to traverse the discourse structure, move between nodes, and explore relations within the discourse tree.
- **Content Tools:** Provide mechanisms for extracting relevant information from specific nodes or subtrees, including semantic search retrieval.
- **Memory Tools:** Support the accumulation and organization of evidence by enabling the agent to store and reference information gathered during execution.

Table 1 summarizes the available tools and their core functionalities. Navigation tools such as `Move`, `RelationsFrom`, and `Ancestors` facilitate flexible exploration of the discourse environment. Content tools like `Retrieve` and `Text` enable targeted information extraction, while memory tools such as `Write` allow the agent to record and manage evidence throughout the reasoning process. This modular toolset empowers the agent to adapt its strategy dynamically, efficiently gathering and synthesizing information to address complex queries.

Tool	Functionality
`Move(id)`	Focus on specific node
`RelationsFrom(id)`	Nodes accessible via outgoing relations
`RelationsTo(id)`	Nodes with incoming relations to current
`Ancestors(id)`	Path to root (as `path` field)
`Retrieve(query)`	Semantic search in current subtree
`Text(id)`	Get full text of subtree
`Write(text|ids)`	Store evidence (text or node references)

Table 1: Discourse navigation and content tools

3.4. Execution Workflow

Building on the toolset described above, the agent operates through iterative ReAct (Reason + Act) cycles to process complex queries within the discourse environment. Each cycle consists of the following stages:

1. **Reason:** The agent examines the current state of the discourse environment and its accumulated memory. It identifies which parts of the discourse structure are most relevant to the query and determines what information is still missing.
2. **Act:** Based on its reasoning, the agent selects an appropriate tool from its toolkit and specifies the necessary parameters (e.g., node identifiers or search queries).

3. **Observe:** The agent receives and interprets the output produced by the selected tool. This may include new textual evidence, a list of related nodes, or updated context within the discourse tree.

4. **Update:** The agent integrates the newly acquired information into its memory, updating its internal representation of the environment and tracking progress toward answering the query.

This cycle is repeated as needed, allowing the agent to iteratively explore the discourse structure, gather supporting evidence, and refine its understanding of the information landscape. The detailed agent trajectory example is presented in Appendix A, while the full agent prompt is provided in Appendix B. Once sufficient evidence has been collected, a final generation step synthesizes the accumulated information into a coherent, query-focused response.

4. Dataset

This paper addresses narrative QA summarization, a challenging task requiring the synthesis of evidence dispersed throughout long stories with complex structures (Sang et al., 2022; Zhu et al., 2023). RAG is particularly beneficial in this setting by grounding generation in retrieved narrative segments. Recent advancements, such as Graph RAG (Edge et al., 2024), extend this capability by modeling global textual organization, highlighting the promise of structure-aware approaches for improving narrative understanding.

To evaluate our hypothesis that a RAG approach enriched with discourse analysis improves narrative QA, we adapted the SQuALITY dataset (Wang et al., 2022b). While suitable for long-context summarization, its original questions are predominantly fact-based, offering limited coverage of queries that require deep narrative understanding. To address this, we enriched the dataset with new question categories designed to probe more complex reasoning.

For each text from the SQUALITY dataset, following question categories were generated: plot-specific, thematic analytical, worldbuilding contextual analysis, hidden connections and subtle clues, cause-effect, and character interpretation (Table2). Most existing datasets focus primarily on event-centric assessments, while tasks involving other narrative elements remain relatively underexplored (Sang et al., 2022). To ensure comprehensive coverage, we designed the categories based on a two-dimensional schema. The first dimension specifies the target narrative element (e.g., event, character, setting, or functional structure), while the second defines the scope of reasoning (from local to global).

Question generation for the predefined categories was carried out using the GPT-4o-mini API, followed by additional manual filtering. Common issues in the generated questions included overlap across categories, subjective or opinion-based formulations, omission of character names, and substitution with generic terms like “character”. In total, 111 additional questions were generated for 6 categories.

To facilitate reproducibility and support future research on structure-aware reasoning, we release the **DiscoRAG-SQuALITY benchmark** publicly on Hugging Face.[1] The benchmark includes 52 long-context documents, each enriched with the following pre-computed features:

- *RST Parse Trees:* The full Rhetorical Structure Theory trees in bracketed format, generated using a state-of-the-art discourse parser.
- *Flattened Discourse Units:* A representation of the discourse tree tailored for retrieval pipelines. Each unit maps a text span to its nuclearity (Nucleus/Satellite) and rhetorical relation, enabling structure-aware querying.
- *Synthetic Questions:* The set of 111 generated questions categorised by reasoning type (e.g., “Hidden Connections”), designed to probe the model’s ability to aggregate information across distant segments.
- *Original SQuALITY Data:* The baseline questions and reference answers from the SQuALITY benchmark, included for backward compatibility.

By open-sourcing these resources, we aim to lower the computational barrier for discourse-based experimentation and provide a standardized testbed for narrative understanding models.

5. Experiments

To evaluate our framework, we design experiments to answer the following research questions (RQs):

- **RQ1:** Does the DiscoRAG agent retrieve a more relevant and complete context for query-focused summarization compared to baselines?
- **RQ2:** Does this improved context retrieval lead to higher-quality final answers, particularly for queries requiring discourse-level understanding?

[1] https://huggingface.co/datasets/alexchern5757/discorag-squality

Question Type	Description
Plot Specific	Questions focusing on specific events, plot developments, and their logical sequences.
Thematic Analytical	Questions exploring deeper themes, symbolism, character dynamics, and the author's underlying messages.
Worldbuilding / Contextual Analysis	Questions examining the setting, political and societal structures, and their impact on the story.
Hidden Connections / Subtle Clues	Questions requiring identification of indirect references, foreshadowing, or patterns within the text.
Character Interpretation	Questions focusing on interpreting characters' motivations, emotions, and development, requiring an understanding of their implicit intentions and relationships within the narrative.
Cause–Effect	Questions analyzing how different story elements influence each other and requiring an understanding of both explicit and implicit causal relationships.

Table 2: Question types

- **RQ3:** Are discourse-sensitive queries a distinct and identifiable class, justifying the need for specialized frameworks like ours?

5.1. Experimental Setup

5.1.1. Models and Baselines

We compare DiscoRAG against two competitive baselines:

Vanilla RAG A standard Retrieval-Augmented Generation pipeline performs dense retrieval over a flat (unstructured) list of text chunks, where each chunk corresponds to a node from the discourse tree.

ExpansionRAG A stronger baseline designed to test if simply increasing context size is sufficient. It first retrieves the top-M relevant chunks (nodes) and then expands the context by including n neighboring nodes for each retrieved chunk in the original document order. This baseline mimics providing more local context without being aware of the discourse structure.

5.1.2. Implementation Details

Core Architecture. DiscoRAG is implemented as described in Section 3. We found in preliminary experiments that initializing the agent at the discourse tree's root was suboptimal, as it tended to focus on only a few major branches. We therefore employ a more robust two-stage approach:

1) Initial Retrieval: We first run a dense retriever to identify a set of candidate entry-point nodes. In this step, we select M start nodes.

2) Agentic Exploration: The agent is then launched from each of these entry points to navigate the discourse sub-trees and collect a comprehensive context.

Context Aggregation. To ensure a fair comparison, all models use a consistent context aggregation strategy to form the final input for the generator. After the initial retriever provides a ranked list of entry-point nodes, we construct the final context for answer generator as follows:

1. Interleaved Collection: We employ an interleaved aggregation strategy to merge the result lists. The process iterates rank-by-rank: it first gathers all rank-1 nodes from every list, then all rank-2 nodes, and so on. This gives precedence to nodes ranked higher within their respective local results.

2. Deduplication: A node is added to the final context only if it has not been previously included.

3. Sizing and Sorting: The collection stops once K unique nodes are gathered. This final set is then re-sorted according to their original order in the document to preserve textual coherence.

In our experiments, we used the default $M = 8$ and $K = 30$.

Infrastructure. Discourse trees for all documents are built using the publicly available Two-Stage discourse parser (Wang et al., 2017) that demonstrates state-of-the-art performance in discourse parsing.

For all experimental models, we utilize the Llama-3.1 405B model (Dubey et al., 2024) as the core reasoning and generation engine. It powers the decision-making process for our DiscoRAG and serves as the final answer generator for all baselines. This ensures a fair comparison focused on the quality of the retrieved context rather than the generative capabilities of different LLMs.

Metric	Mean	Median	q=0.9	q=0.95
# leaves	134.67	124.00	206.40	214.45
# context	28.49	29.00	36.00	38.00

Table 3: Statistics on the size of the full discourse trees (# leaves) versus the context retrieved by our agent (# context).

5.1.3. Evaluation Metrics

To answer our research questions, we use the following metrics.

To address **RQ1**, we assess context quality using node-level Precision, Recall, and F1-score. These metrics are computed against a manually annotated ground-truth subset of our data (only questions from the original SQUALITY dataset).

To evaluate end-to-end answer quality (**RQ2**), we employ an LLM-as-a-Judge approach (Li et al., 2024). For each query, the judge is presented with a pair of anonymous, randomly ordered answers (one from our model, one from a baseline). It is prompted to select the better response based on its relevance, completeness, and coherence, and to provide a detailed rationale for its choice. We report the resulting win, loss, and tie rates.

5.2. Context Size and Pruning Efficiency

Before evaluating the end-to-end performance, we first analyze the extent to which DiscoRAG can prune the document's search space. Table 3 presents the statistics on the number of leaf nodes in the entire discourse tree compared to the number of nodes selected by our agent to form the final context.

The source documents are substantial, with a mean of 134.67 leaf nodes. In stark contrast, our agent retrieves a much more focused context, selecting only 28.49 nodes on average. This represents an average context reduction of approximately 79%, demonstrating the agent's ability to discard large, irrelevant portions of the text.

5.3. Context Quality Analysis (RQ1)

To evaluate retrieval quality, we analyze model performance on subsets of increasing query complexity. We use the number of passages retrieved by DiscoRAG as a proxy for the perceived difficulty of a query.

Methodology Our evaluation proceeds as follows: for each context size K in the range [20, 30], we first identify the subset of queries for which DiscoRAG retrieved at least K passages. Then, for every query within this specific subset, we evaluate all models — DiscoRAG, Vanilla RAG, and ExpansionRAG — by truncating their respective ranked passage lists to the exact same size K. This design enables a direct, controlled comparison of ranking quality on progressively more demanding tasks. The results are shown in Figure 4.

Results As shown in Figure 4a, DiscoRAG consistently achieves the highest ROUGE-L Precision. The ExpansionRAG model improves upon Vanilla RAG, confirming that incorporating local context is a beneficial heuristic. However, DiscoRAG outperforms ExpansionRAG, which indicates that targeted, relation-aware navigation is substantially more effective at increasing the relevance density of the context than proximity-based expansion.

Conversely, the recall analysis (Figure 4b) shows that Vanilla RAG achieves the highest recall. This is attributable to its unconstrained, broad-net semantic search. Both DiscoRAG and ExpansionRAG exhibit lower recall as a direct consequence of their focusing mechanisms. The ExpansionRAG model is limited by its commitment to a local neighborhood, while DiscoRAG makes a controlled trade-off, prioritizing a coherent discourse path over exhaustive retrieval.

The ROUGE-L F1 score (Figure 4c) provides the most conclusive evidence on context quality. DiscoRAG attains the highest F1 score, demonstrating that its significant gains in precision more than compensate for the controlled reduction in recall.

However, context-level metrics such as F1 are not definitive indicators of the final quality of the generation. The importance of high recall, ensuring that all critical information is available to the generator, cannot be underestimated, and a high F1 score might obscure the omission of a key fact if precision gains are substantial. To address this limitation and assess the ultimate impact on answer quality, we next present our end-to-end evaluation using an LLM-as-a-Judge framework (**RQ2**). Furthermore, to verify that our approach's benefits are most pronounced on the specific class of discourse-aware queries where this precision-recall trade-off is most critical, we develop a classifier to identify such questions (**RQ3**).

5.4. End-to-end Answer Quality (RQ2)

To compare the outputs of Vanilla RAG and DiscoRAG, we employ an LLM-as-a-Judge framework (Gu et al., 2024), utilizing the Gemini 2.5 Pro model as the judge.

To evaluate a model's ability for sophisticated narrative analysis, which extends beyond simple factual recall, we designed a framework to assess the quality of evidence-based reasoning. Our criteria are structured along two axes:

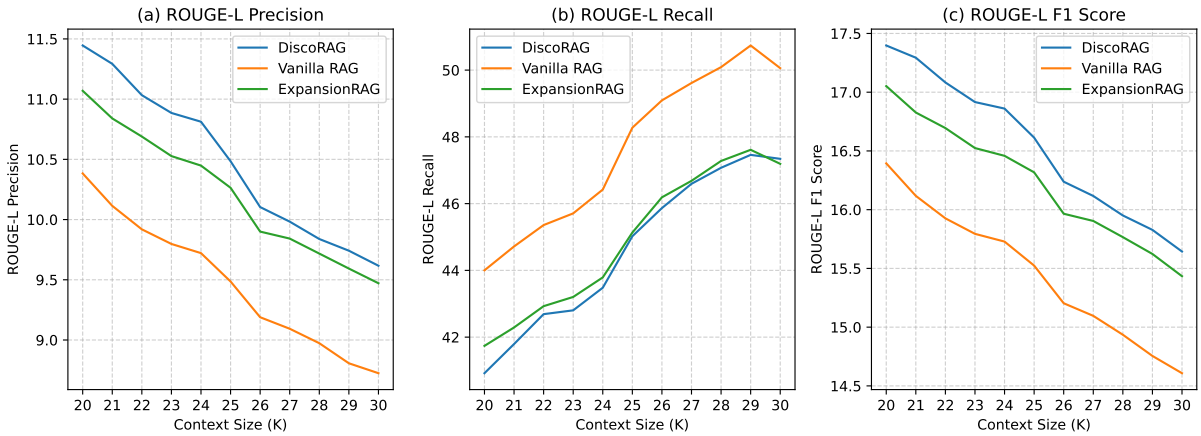

Figure 4: Analysis of context retrieval quality as a function of query complexity. We vary the context size K and evaluate on a corresponding subset of queries. The plots show (a) Precision, which measures relevance density; (b) Recall, which measures information coverage; and (c) the F1 score, which provides a balanced measure of overall effectiveness.

Argument Construction (C1, C2): We first assess the structural integrity of the analytical response.

- Framing the central problem or question rather than simply providing its answer (C1: *Focus on the Question*).
- Grounding claims in direct evidence (dialogue, actions) over abstract narrative summaries (C2: *Evidence over Narration*).

Argument Quality (C3, C4): We then evaluate the coherence and focus of the constructed argument.

- Synthesizing evidence into a single, cohesive thematic point (C3: *Thematic Cohesion*).
- Maintaining strict relevance to the user's query throughout the entire response (C4: *Query Relevance*).

The results of our head-to-head comparison are presented in Table 4. Our DiscoRAG approach significantly outperforms the baseline, achieving an overall win rate of 61.1%. This superiority is particularly evident on the first three criteria—focusing on the question (C1), using direct evidence (C2), and thematic cohesion (C3). Notably, the evaluation for query relevance (C4) resulted in a high tie rate (42.2%), suggesting both models perform comparably on this specific dimension.

A breakdown of performance by question type (Table 5) highlights the strengths of our approach. DiscoRAG significantly outperforms the baseline in complex analytical tasks (Thematic & Analytical, Cause-Effect), while performance is tied for simplerplot-related queries. Although the sample size for some categories is limited, this trend suggests that DiscoRAG excels at reasoning rather than the base Vanilla RAG.

Criteria	DiscoRAG	Vanilla RAG	Tie
C1	**61.2** (68)	35.2 (39)	3.6 (4)
C2	**62.2** (69)	34.2 (38)	3.6 (4)
C3	**59.5** (66)	40.5 (45)	0.0 (0)
C4	**41.5** (46)	18.9 (21)	39.6 (44)
Overall	**61.3** (68)	38.7 (43)	0.0 (0)

Table 4: Head-to-head comparison of DiscoRAG against Vanilla RAG using an LLM-as-a-judge across four evaluation criteria. Criteria: C1 (Question Focus), C2 (Evidence), C3 (Cohesion), C4 (Relevance).

Question Type	DiscoRAG	Vanilla RAG
Plot Specific	50.0 (12)	50.0 (12)
Thematic	**77.8** (7)	22.2 (2)
Cause-Effect	**66.7** (16)	33.3 (8)
Worldbuilding	**66.7** (12)	33.3 (6)
Character Interp.	**61.9** (13)	38.1 (8)
Hidden Conn.	**53.3** (8)	46.7 (7)

Table 5: Fine-grained performance analysis across question types. Values are win rates (%) with absolute counts (N).

5.5. Classifier Evaluation (RQ3)

We additionally trained a binary classifier to distinguish between questions suitable for a standard RAG methodology and those requiring discourse level analysis (questions according to the categorization in Section 4). For this purpose, a comprehensive dataset encompassing over 6500 questions of both types was built. Most questions were generated for Project Gutenberg texts included in the NarrativeQA corpus (Kočiskỳ et al., 2018), us-

ing the same approach applied to the SQuALITY dataset and manual assessment. Furthermore, to enhance domain and topic diversity, the dataset was extended with queries referring to popular books, films, and television series.

For the binary classification task, the DeBERTa-v3-small model (He et al., 2020) was fine-tuned using the Hugging Face transformers library. The model's output logits were passed through a softmax layer to obtain normalized class probabilities during the three training epochs, which were subsequently used to compute metrics. Evaluation on a balanced test set including 500 questions (250 per class) showed that the classifier achieved an F1-score of 0.984 demonstrating predictive performance.

Despite its strong overall performance, the classifier, while distinguishing explicit causal (starting with *why/how*) from factual questions, tends to overrely on lexical markers. The most difficult cases for the model involve implicit social or attitudinal reasoning (e.g., *"How did Mary Alice's sheltered upbringing affect her interactions with Jack?"*). Explicit causal questions like *"What caused Tinker Tim to quarrel with Dan?"* are also misclassified as one requiring discourse analysis indicating that the classifier can overestimate the interpretive complexity of questions.

6. Conclusion

In this paper, we suggested DiscoRAG, a novel approach designed to overcome the limitations of standard RAG on complex narrative queries. Our method moves beyond local semantic similarity to reason about a document's global structure using Rhetorical Structure Theory (RST). This pipeline, which integrates a query classifier and a discourse-aware agent, navigates a document's discourse tree to assemble a more coherent and contextually rich evidence base.

Our empirical evaluation confirms the superiority of this approach. DiscoRAG achieves a higher context-level ROUGE-L F1 score and, more importantly, generates higher-quality final answers, as verified by an LLM-as-a-Judge. The model's sizeable outperformance on analytical and cause-effect questions validates our central thesis: "discourse-aware" queries form a distinct problem class that necessitates specialized, structure-aware models. To support future research in this direction, we openly release our augmented benchmark. Future work will focus on optimizing the agent's planning efficiency and extending the framework to multilingual scenarios. Generally, DiscoRAG marks a sizable step towards a new paradigm of structure-aware retrieval, paving the way for more sophisticated and human-like text understanding in generative AI.

Limitations

Although our experimental evaluation focuses on English documents, the DiscoRAG framework is designed to be language-agnostic. Since high-quality discourse parsers are already available for various other languages, our approach can be readily adapted to these linguistic contexts. We leave the explicit evaluation of multilingual performance for future work.

A limitation of our approach lies in the agent's interaction with the discourse graph. Not all language models navigate discourse structure equally effectively: weaker models may repeatedly focus on the same node, fail to explore relevant discourse relations, or retrieve insufficient textual evidence. These issues are often related to limited planning capabilities of models, suggesting that additional control mechanisms may be necessary to ensure stable and effective discourse graph navigation.

Regarding efficiency, the proposed method introduces additional computational cost compared to standard single-step retrieval pipelines. Specifically, the pre-computation of discourse trees and the iterative nature of the agentic workflow increase both inference latency and token consumption. However, we consider that this trade-off is justified by the substantial performance gains in resolving complex, structure-dependent queries.

Ethics and Broader Impact

Our work contributes to the development of more reliable AI systems for understanding long-documents. By explicitly grounding the generation process in the document's rhetorical structure and providing an observable agent trajectory, DiscoRAG improves the transparency of the reasoning process compared to black-box summarization models. This allows users to better verify the provenance of the synthesized information.

Like all systems relying on pre-trained LLMs and discourse parsers, our framework is susceptible to the biases inherent in these upstream components. For instance, discourse parsers trained primarily on standard corpora may perform sub-optimally on non-standard dialects or specialized domains, potentially leading to uneven performance across different text types.

Acknowledgements

This research was supported in part through computational resources of HPC facilities at HSE University. The article was prepared within the framework of the HSE University Basic Research Program.

7. Bibliographical References

Sergio Torres Aguilar. 2025. Chronorag: A self-correcting, context-aware rag for longitudinal inquiry in historical archives.

Huiyao Chen, Yi Yang, Yinghui Li, Meishan Zhang, and Min Zhang. 2025. Disretrieval: Harnessing discourse structure for long document retrieval. *CoRR*, abs/2506.06313.

Alexander Chernyavskiy, Dmitry Ilvovsky, and Preslav Nakov. 2024a. Unleashing the power of discourse-enhanced transformers for propaganda detection. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 1452–1462. Association for Computational Linguistics.

Alexander Chernyavskiy, Lidiia Ostyakova, and Dmitry Ilvovsky. 2024b. GroundHog: Dialogue generation using multi-grained linguistic input. In *Proceedings of the 5th Workshop on Computational Approaches to Discourse (CODI 2024)*, pages 149–160, St. Julians, Malta. Association for Computational Linguistics.

Dujian Ding, Ankur Mallick, Shaokun Zhang, Chi Wang, Daniel Madrigal, Mirian del Carmen Hipolito Garcia, Menglin Xia, Laks V. S. Lakshmanan, Qingyun Wu, and Victor Rühle. 2025. Best-route: Adaptive LLM routing with test-time optimal compute. *CoRR*, abs/2506.22716.

Haowei Du, Yansong Feng, Chen Li, Yang Li, Yunshi Lan, and Dongyan Zhao. 2023. Structure-discourse hierarchical graph for conditional question answering on long documents. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 6282–6293. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Qianyu Guo, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *CoRR*, abs/2312.10997.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Yuxuan Huang, Yihang Chen, Haozheng Zhang, Kang Li, Meng Fang, Linyi Yang, Xiaoguang Li, Lifeng Shang, Songcen Xu, Jianye Hao, Kun Shao, and Jun Wang. 2025. Deep research agents: A systematic examination and roadmap. *ArXiv*, abs/2506.18096.

Tomáš Kočiskỳ, 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.

Alex Laitenberger, Christopher D Manning, and Nelson F Liu. 2025. Stronger baselines for retrieval-augmented generation with long-context language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32547–32557.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *CoRR*, abs/2412.05579.

William Mann, Christian Matthiessen, and Sanda Thompson. 1989. Rhetorical structure theory and text analysis. *Discourse Description: Diverse Linguistic Analyses of a Fund Raising Text*, page 66.

Inderjeet Nair, Shwetha Somasundaram, Apoorv Saxena, and Koustava Goswami. 2023. Drilling down into the discourse structure with llms for long document question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 14593–14606. Association for Computational Linguistics.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. 2024. Graph retrieval-augmented generation: A survey. *CoRR*, abs/2408.08921.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-Rong Wen. 2025. Tool learning with large language models: a survey. *Frontiers Comput. Sci.*, 19(8):198343.

Yisi Sang, Xiangyang Mou, Jing Li, Jeffrey Stanton, and Mo Yu. 2022. A survey of machine narrative reading comprehension assessments. *arXiv preprint arXiv:2205.00299*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *ArXiv*, abs/2302.04761.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

Aditi Singh, Abul Ehtesham, Saket Kumar, and Tala Talaei Khoei. 2025. Agentic retrieval-augmented generation: A survey on agentic RAG. *CoRR*, abs/2501.09136.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R. Bowman. 2022a. Squality: Building a long-document summarization dataset the hard way. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1139–1156. Association for Computational Linguistics.

Alex Wang, Richard Yuanzhe Pang, Angelica Chen, Jason Phang, and Samuel R Bowman. 2022b. Squality: Building a long-document summarization dataset the hard way. *arXiv preprint arXiv:2205.11465*.

Yizhong Wang, Sujian Li, and Houfeng Wang. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, pages 184–188. Association for Computational Linguistics.

Jiacheng Xu, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Discourse-aware neural extractive text summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5021–5031. Association for Computational Linguistics.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jiarui Zhang, Xiangyu Liu, Yong Hu, Chaoyue Niu, Fan Wu, and Guihai Chen. 2025. Query routing for retrieval-augmented language models. *CoRR*, abs/2505.23052.

Lixing Zhu, Runcong Zhao, Lin Gui, and Yulan He. 2023. Are nlp models good at tracing thoughts:

An overview of narrative understanding. *arXiv preprint arXiv:2310.18783*.

A. Detailed Agent Trajectory Example

Here we present a condensed trajectory of **DiscoRAG** for the query "What is Gurn's role in the story?" to illustrate its core mechanism. This trace highlights how the agent moves beyond initial semantic retrieval to discover deeper, structurally-linked information by navigating the discourse tree. The process can be summarized in the following key steps:

1. **Initial Retrieval:** The agent begins with a standard semantic search, retrieving five relevant but surface-level nodes. This represents the context available to a 'Vanilla RAG' model.

2. **Discourse Navigation:** Crucially, the agent identifies the potential for deeper exploration and decides to navigate the discourse tree starting from node 39.

3. **Discovery of Novel Information:** By following an 'Attribution' link, the agent discovers node 37, which contains a physical description of Gurn. This vital information was entirely absent in the initial retrieval set.

4. **Context Enrichment:** Further exploration along an 'Elaboration' link enriches the context with details about character relations (node 41).

Listing 1: Full agent trace for the query "What is Gurn's role in the story?"

```
Thought 1: I should use retrieval to find
    ↪ information about Gurn's role in the story.

Action 1: Retrieve[Role of Gurn in the story]

Observation 1: Retrieved 5 nodes:
Node ID 39 -
" "That is Gurn," admitted Noork shortly. "He is
    ↪ also an exile from the walled city of Grath.
    ↪  The city rulers call him a traitor. He has
    ↪ told me the reason. Perhaps you know it as
    ↪ well?
Node ID 35 -
" "I am Noork," the man told her. "For many days
    ↪ have I dwelt among the wild Vasads of the
    ↪ jungle with their golden-skinned chief, Gurn
    ↪ , for my friend.
Node ID 94 -
" "I have stolen the skin of a demon," answered the
    ↪ invisible man. "Go to Gurn. Tell him to fear
    ↪  the demons no longer. Tell him the Misty
    ↪ Ones can be trapped and skinned.
Node ID 297 -
"The message that Ud carried to me was good,"
    ↪ laughed Gurn. "The Misty Ones skin easily.
    ↪ We were trapping the Misty Ones as they came
    ↪  across the lake," he looked at the dying
    ↪ Von Mark, "as were these others. Soon we
    ↪ would have come to your rescue, Noork, my
    ↪ friend.
Node ID 97 -
"Go to save Gurn's ..." and here Noork was stumped
    ↪ for words. "To save his father's woman woman
    ↪ ," he managed at last. "Father's woman woman
    ↪  called Sarna.

Thought 2: I will save all the retrieved nodes
    ↪ related to Gurn's role in the story.

Action 2: Write[39, 35, 94, 297, 97]

Observation 2: The texts are written.

Thought 3: I should explore the discursively related
    ↪  nodes to gain a better understanding of
    ↪ Gurn's role.

Action 3: RelationsFrom[39]

Observation 3: There are the following relations:
Attribution: node ID 40; depth 11
Attribution: node ID 37; depth 10
Attribution: node ID 17; depth 9
Elaboration: node ID 41; depth 8
Contrast: node ID 42; depth 7
Same-Unit: node ID 13; depth 6
Joint: node ID 9; depth 5

Thought 4: I will explore the nodes related to Gurn
    ↪ through the Attribution relation.

Action 4: Move[40]

Observation 4: The transition to node with ID 40 has
    ↪  been completed.
Text - " "Indeed I do," cried Sarna. "My brother
    ↪ said that we should no longer make slaves of
    ↪  the captured Zurans from the other valleys
    ↪ ..

Thought 5: I will retrieve the text related to Sarna
    ↪  and her brother's decision regarding the
    ↪ Zurans.

Action 5: Retrieve[Sarna and her brother's decision
    ↪ regarding the Zurans]

Observation 5: Retrieved 5 nodes:
Node ID 40 -
" "Indeed I do," cried Sarna. "My brother said that
    ↪ we should no longer make slaves of the
    ↪ captured Zurans from the other valleys.
Node ID 27 -
"Tako," she replied fearfully. "Who speaks to Tholon
    ↪  Sarna? Be you hunter or escaped slave?
Node ID 303 -
" He smiled and his brown arm tightened around Sarna
    ↪ 's golden body. "The evil man is dead. My
    ↪ native valley is safe. Now I can live in
    ↪ peace with you, Gurn, and with your sister,
    ↪ here in the jungle.
Node ID 97 -
"Go to save Gurn's ..." and here Noork was stumped
    ↪ for words. "To save his father's woman woman
    ↪ ," he managed at last. "Father's woman woman
    ↪  called Sarna.
Node ID 235 -
Tholon Sarna stumbled through the door and he caught
    ↪  her in his arms. Hurriedly he loosed one of
    ↪  the two robes fastened about his waist and
    ↪ slipped it around her slim shivering
    ↪ shoulders.

Thought 6: The text from node 27 seems relevant to
    ↪ Gurn's role in the story. I will save it.

Action 6: Write[235]

Observation 6: The texts are written.

Thought 7: I should explore the nodes related to
    ↪ Gurn through the Attribution relation from
    ↪ node ID 37.

Action 7: Move[37]

Observation 7: The transition to node with ID 37 has
    ↪  been completed.
```

```
Text - " The girl impulsively took a step nearer. "
     ↪ Gurn!" she cried. "Is he tall and strong?
     ↪ Has he a bracelet of golden discs linked
     ↪ together with human hair? Does he talk with
     ↪ his own shadow when he thinks?.

Thought 8: The text from node 37 describes a
     ↪ description of Gurn. I will save it.

Action 8: Write[37]

Observation 8: The texts are written.

Thought 9: I should explore the nodes related to
     ↪ Gurn through the Elaboration relation from
     ↪ node ID 41.

Action 9: Move[41]

Observation 9: The transition to node with ID 41 has
     ↪  been completed.
Text - " Noork smiled. "I am glad he is your brother
     ↪ ," he said simply..

Thought 10: The text from node 41 suggests that Gurn
     ↪  is Noork's brother. I will save it.

Action 10: Write[39, 40, 41]

Observation 10: The texts are written.

Thought 11: I should explore the nodes related to
     ↪ Gurn through the Contrast relation from node
     ↪  ID 42.

Action 11: Move[42]
```

B. Prompts

In our experiments, we utilized the agentic prompt presented in Figure 5.

```
You should collect the necessary pieces of information from the text, which will subsequently allow you to
     write a general summary according to a given query. The document is presented hierarchically in the
     form of a discourse tree, enriched with chapter html tags.
  You have access to the following actions with the document tree:

  {tools}

  Use chained reasoning Thought | Action | Observation
  The thought should consist of one-two sentences with a plan for what to do next. A thought cannot be empty
     !
  The action must include a call to tools in the format Tool[Argument]. If a tool requires no arguments,
     then it must be used as Tool[].
  The tool to be called must be selected from {tool_names}.
  Pay attention to the Write[text] action, according to which you can write down the necessary information
     to a scratchpad, according to which the final summary will be constructed in the future.
  The last action of the argument should be Finish[Done]. Use it when you think you have accumulated enough
     information to write a final summary.

  [Example]
  =======
  Question: Who are the Bogans, and what happens to their plan?
  Thought 1: I should use information retrieval to fiend relevent text piece about Bogans.
  Action 1: Retrieve[Who are the Bogans?]
  Observation 1: Retrieved 5 nodes:
  Node ID 168 -
  I wonder if the Bogans know that?
  Node ID 234 -
  Do you know the name of the institution these Bogan students are bound for?
  Node ID 19 -
  Magnan nodded. The Bogans have launched no less than four military campaigns in the last two decades. They
      are known as the Hoodlums of the Nicodemean Cluster. Now, perhaps, we shall see them breaking that
     precedent and entering into the cultural life of the Galaxy.
  Thought 2: All nodes seems relevant, so I will save all of them.
  Action 2: Write[19, 168, 234]
  Thought 3: I will explore the discursively related nodes, starting with the nodes related with node 19.
  Action 3: RelationsFrom[19]
  Observation 3:  There are the following relations:
  Elaboration: ID 24; depth 8
  Contrast: ID 25; depth 7
  Thought 4: I can explore further with the help of contrast node.
  Action 4: Move[25]
  Observation 4: Inner node (text is hidden)
  Thought 5: I can call the retrival tool from this node.
  Action 5: Retrieve[Bogans]
  =======

  [Thought examples]
  =======
  - I should use retrieval to find relevant information.
  - I can use Attribution relation from the current node, to explore information that attributes it.
  - Information seems relevant. I should save it using Write action.
  =======

  Continue the reasoning below. Important: use more Write actions to save relevant texts!
  You must cover as much relevant parts of the story as possible!

  Important: You must explore the node obtained after the first Retrieve action!
  Important: use the exact text in the Retrieve actions (texts like "information from..." are restricted)!
  Important: use MORE discourse-based and Write actions and LESS Retrieve actions!
  Important: use Write[index] for the short texts and Write[full text to write] for the long texts.
  Important: try to use Retrieve actions instead of Text for inner nodes.
  Important: in Write save only those nodes that you are confident in.

  Question: {input}{agent_scratchpad}
```

Figure 5: The complete system prompt for the DiscoRAG agent.