applied sciences

MDPI

Article

HiSem-RAG: A Hierarchical Semantic-Driven Retrieval-Augmented Generation Method

Dongju Yang and Junming Wang *

Large-Scale Stream Data Integration and Analysis Technology Beijing Key Laboratory, School of Artificial Intelligence and Computer Science, North China University of Technology, No. 5 Jinyuanzhuang Road, Beijing 100144, China; yangdongju@ncut.edu.cn

* Correspondence: cdd@mail.ncut.edu.cn

Abstract

Traditional retrieval-augmented generation (RAG) methods struggle with hierarchical documents, often causing semantic fragmentation, structural loss, and inefficient retrieval due to fixed strategies. To address these challenges, this paper proposes HiSem-RAG, a hierarchical semantic-driven RAG method. It comprises three key modules: (1) hierarchical semantic indexing, which preserves boundaries and relationships between sections and paragraphs to reconstruct document context; (2) a bidirectional semantic enhancement mechanism that incorporates titles and summaries to facilitate two-way information flow; and (3) a distribution-aware adaptive threshold strategy that dynamically adjusts retrieval scope based on similarity distributions, balancing accuracy with computational efficiency. On the domain-specific EleQA dataset, HiSem-RAG achieves 82.00% accuracy, outperforming HyDE and RAPTOR by 5.04% and 3.98%, respectively, with reduced computational costs. On the LongQA dataset, it attains a ROUGE-L score of 0.599 and a BERT_F1 score of 0.839. Ablation studies confirm the complementarity of these modules, particularly in long-document scenarios.

Keywords: hierarchical document representation; semantic enhancement; adaptive threshold retrieval; long-form question answering; similarity distribution modeling; large language model

1. Introduction

Retrieval-Augmented Generation (RAG) has emerged to address Large Language Model (LLM) limitations regarding knowledge timeliness, domain specificity, and generation stability. By combining external retrieval with generative capabilities, RAG allows models to access dynamic, domain-specific information during inference, overcoming the boundaries of closed parameter learning.

Since its proposal by Lewis et al. [1], RAG has been widely applied in question answering, summarization, and dialogue systems. Early methods, such as Dense Passage Retrieval (DPR) [2], relied on dual-encoder architectures for vector-based semantic matching. Subsequently, RAG systems evolved in two key areas: semantic indexing (advancing from fixed-length chunks to structure-aware forms) and retrieval mechanisms (expanding to hybrid [3] and adaptive approaches). For instance, Self-RAG [4] employs self-reflection to dynamically invoke external knowledge. However, reliance on external knowledge presents challenges. Mansurova et al. [5] explored LLM reliance on external

Check for updates

Academic Editors: Dickson K. W. Chiu and Stuart So

Received: 1 December 2025
Revised: 1 January 2026
Accepted: 2 January 2026
Published: 15 January 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

vector indices (QA-RAG), highlighting persistent issues with noise robustness and external truth integration.

Despite progress, RAG faces bottlenecks when processing complex, long, or information-dense documents:

1. Semantic Structure Loss: Most methods mechanically split documents into fixed-length segments, ignoring natural hierarchies like headings and sections. This disrupts semantic boundaries, resulting in fragmented retrieval and weak context, making it difficult to capture key information in long documents.
2. Rigid Retrieval Quantity: Mainstream approaches typically use a fixed top-k strategy, ignoring the actual knowledge distribution. A small k risks omitting critical information, while a large k introduces noise and increases computational overhead, leading to generation interference.

To improve accuracy and efficiency, we propose HiSem-RAG, a Hierarchical Semantic-Driven Retrieval-Augmented Generation method. This approach leverages document hierarchy and semantic distribution through three designs:

- Hierarchical Semantic Indexing: Constructs multi-granularity indices based on natural document structures (sections, paragraphs), preserving original boundaries for hierarchical retrieval.
- Bidirectional Semantic Enhancement: Introduces titles and summaries to facilitate information flow across layers, strengthening semantic connections.
- Distribution-Aware Adaptive Thresholding: Dynamically sets retrieval thresholds based on similarity distributions, replacing the fixed top-k approach to balance completeness and efficiency.

Comprehensive evaluations on the EleQA and LongQA datasets demonstrate that HiSem-RAG outperforms baselines in both accuracy and resource efficiency. Ablation studies further confirm the synergistic effects of the core modules.

Further ablation studies confirm the synergistic effects among the three core modules, making HiSem-RAG particularly suitable for question-answering scenarios involving complex hierarchical structures and information-dense content.

To systematically address the challenges of structural loss and rigid retrieval strategies, this study focuses on the following three research questions (RQs):

- RQ1: How can we construct an indexing mechanism that preserves the natural hierarchical boundaries of documents to prevent context detachment and structural ambiguity?
- RQ2: How can we enable bidirectional information flow between document layers to enhance the semantic completeness of retrieved fragments?
- RQ3: How can we design a dynamic retrieval strategy that automatically adjusts the retrieval scope based on information density, balancing coverage and efficiency?

2. Motivating Example and Design Rationale

To illustrate the limitations of current approaches, we present a real-world example from the LongQA dataset in Figure 1. The document structure (Panel A) features identical subtitles, "2. Process Standards," under two different chapters: "Main Transformer" and "Neutral Point System."

- Structural Ambiguity and Context Loss: Traditional RAG methods mechanically split documents into flat chunks (Panel B). This severs the connection between the subtitle and its parent chapter. Consequently, when a user asks about "Main Transformer standards," the retriever, lacking structural awareness, is easily distracted by the similar content in the "Neutral Point" section (Chunk 7), leading to hallucinations.

- Fragmentation vs. Adaptive Retrieval: Furthermore, the correct answer spans 23 detailed items (over 3000 tokens), which are inevitably split into multiple fragments (Chunks 3–8). A rigid top-k retrieval strategy (e.g., $k = 3$) would fetch only the initial fragments, missing the critical details in the tail.

Our Solution (HiSem-RAG): As shown in Panel C, HiSem-RAG preserves the hierarchical tree to address these issues:

- Bidirectional Enhancement: We explicitly embed the title path (Top-Down) to disambiguate the context. Simultaneously, detailed rules are compressed into a summary at the parent node (Bottom-Up Aggregation, shown as the blue "Summary" box).
- Adaptive Threshold: Instead of a fixed top-k, our distribution-aware threshold detects the semantic coherence of the parent node and automatically retrieves the entire cluster of relevant chunks (Standards 1–23) as a complete context unit (indicated by the dashed "Adaptive Scope" box).

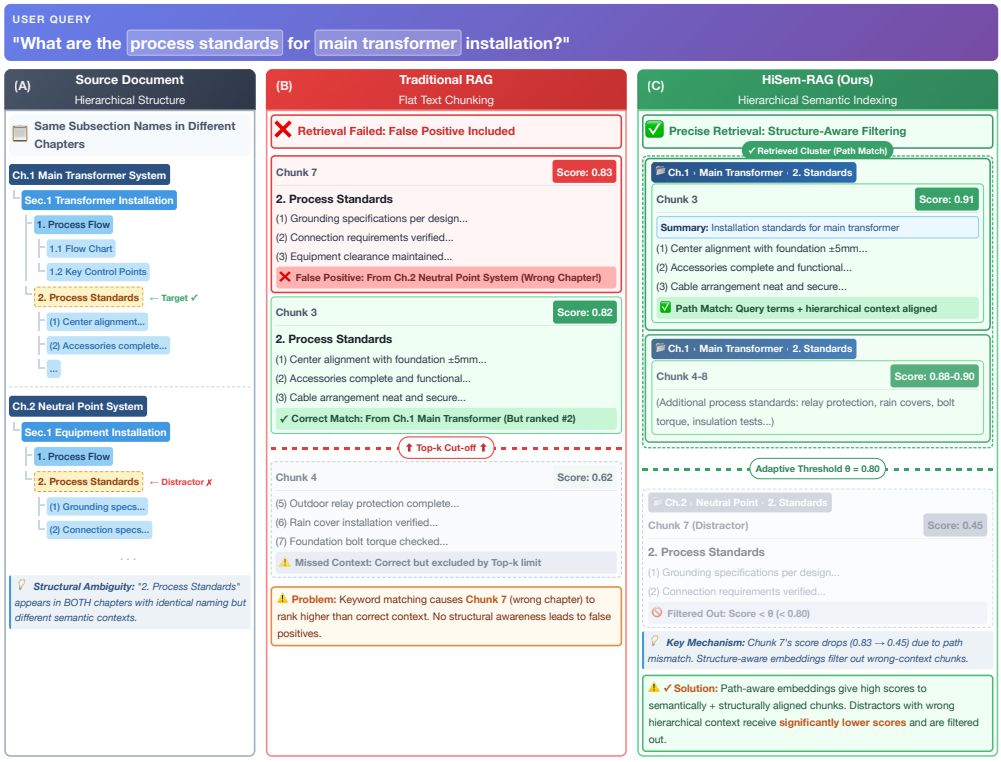


Figure 1. A motivating example illustrating the limitations of traditional RAG versus the proposed HiSem-RAG.

3. Related Work

3.1. Index Structures in RAG Systems

The index structure is a core component of RAG systems, directly determining the organization and retrieval mode of the knowledge base. With the development of RAG technology, index structures have gradually evolved from simple flat structures to more complex forms with hierarchical and semantic associations.

3.1.1. Flat Indexing Methods

Early RAG systems primarily adopted flat indexing structures, representing documents as independent sets of vectors. This approach is easy to implement but struggles to capture hierarchical relationships and semantic coherence within document content. A typical example is DPR, which uses a dual-encoder to generate vector representations for

queries and documents, enabling retrieval through similarity computation. Compared with traditional keyword-matching methods, DPR better captures semantic relevance but ignores the internal structure of documents.

3.1.2. Hierarchical Index Structures

To overcome the limitations of flat indexing, researchers have proposed various hierarchical indexing methods. RAPTOR [6] constructs document trees through recursive abstraction, clustering and summarizing sentences layer by layer from leaf nodes to the root. This structure improves retrieval flexibility and precision, supporting multi-level retrieval with adaptive granularity based on query complexity. While RAPTOR's bottom-up approach may face challenges in tasks requiring complete context, its concept has inspired subsequent optimizations. Recent domain-specific applications, such as the paradigm for construction management proposed by Wu et al. [7], further demonstrate the importance of hierarchical structures in RAG systems. This approach parses project documents into hierarchical structures, combining document-level features with original content to enhance granularity alignment and retrieval effectiveness. These developments have significantly influenced the hierarchical indexing approach proposed in this paper.

3.1.3. Graph-Based Indexing Methods

In recent years, graph-based index structures have shown significant potential. GraphRAG [8] constructs knowledge graphs by extracting entities and relationships using large language models, addressing complex problems requiring global understanding. KA-RAG [9] integrates structured knowledge graphs with agentic workflows, employing a dual-retrieval strategy that combines symbolic graph reasoning with dense semantic retrieval to enhance interpretability and precision in educational scenarios. Several researchers [10–12] have extended these approaches to specialized domains, retrieving relevant triples from knowledge graphs and converting them into natural language prompts for applications ranging from general knowledge enhancement to forensic injury assessment and failure mode analysis. Additionally, Choi et al. [13] proposed integrating Knowledge Graph encoders with pre-trained models (like ALBERT) to utilize explicit external information, demonstrating that schema graph expansion effectively improves performance in commonsense question answering. Graph-based indexes naturally represent complex inter-document associations and provide structural support for advanced reasoning.

In summary, RAG indexing has evolved from flat to hierarchical and graph-based forms, increasingly emphasizing semantic association and structural preservation. While flat structures are simple, hierarchical approaches better preserve document organization, and graph structures excel at representing complex associations. However, simultaneously maintaining structural integrity and supporting efficient retrieval remains challenging. The proposed HiSem-RAG addresses this by constructing a hierarchical index that retains the original document hierarchy while facilitating precise, adaptive retrieval.

3.2. Optimization of Text Chunking Strategies

Text chunking is a foundational step in index construction, directly affecting the quality of the index and retrieval performance. Based on their core principles, chunking strategies can be classified as follows:

3.2.1. Length-Based Chunking Methods

The most basic and widely used strategy is length-based chunking. Fixed-size chunking mechanically splits documents according to a preset number of tokens or characters. While simple to implement, this method often disrupts semantic integrity. TEJA R., through the LLAMAi framework [14], systematically studied the impact of chunk size on RAG

performance, showing that an appropriate chunk size can balance efficiency and semantic completeness. As large language models' capabilities for long-text processing have improved, research by Xu et al. [15] demonstrated that expanding the context window from 4K to 32K and combining with retrieval augmentation can significantly enhance long-text task performance.

3.2.2. Semantics-Based Chunking Methods

Semantics-based chunking methods aim to preserve the semantic integrity of text. By identifying semantic boundaries for segmentation, they retain key information and logical structure. The Small2Big approach by Yang et al. [16] uses sentences as basic retrieval units and supplies context sentences to the language model, improving retrieval efficiency while maintaining semantic coherence. More recently, Krassovitskiy et al. [17] investigated the use of LLM embeddings to capture semantic coherence and topic closure. Their work demonstrates that integrating embeddings with classical clustering and graph-based methods—including a proposed magnetic clustering algorithm—significantly improves segmentation quality, providing valuable insights for optimizing RAG systems. Such methods focus on semantic cohesion within the text and are suitable for unstructured but semantically rich content.

3.2.3. Structure-Aware Chunking Methods

With increasing needs for processing domain-specific documents, structure-aware chunking methods have gained attention. Meta-Chunking [18] employs dynamic chunking to balance fine and coarse granularity, using metadata to enhance contextual information within chunks. Wang et al. [19] examined the role of document structure in paragraph retrieval, showing that considering document structure helps systems understand paragraph significance within the global context. These methods fully leverage structural information—such as titles and sections—and are highly relevant to the section-title-based chunking strategy proposed in this paper. Zheng et al. [20] demonstrated how specialized text mining approaches can enhance domain-specific chunking strategies for chemistry literature. Bi et al. [21] proposed a sliding window strategy that effectively addresses limitations of traditional fixed-size context windows, dynamically adjusting window size and step length to preserve contextual information.

The evolution of text chunking strategies reflects the deepening understanding of semantics and structure in RAG systems. From initial fixed-length chunking, to semantic-aware chunking, and then to structure-aware chunking, each step addresses the limitations of the previous stage. Length-based methods are simple but cannot ensure semantic integrity; semantic chunking values content cohesion but has limited structural awareness; structure-aware chunking leverages inherent document organization but is less effective for unstructured text. Designing chunking strategies that retain the natural hierarchy of documents while maintaining paragraph-level semantic integrity is key to enhancing RAG systems' ability to process complex documents.

3.3. Optimization of Retrieval Mechanisms and Strategies

Retrieval mechanisms directly determine the quality of information provided to the generation model. Retrieval optimization primarily focuses on query processing, retrieval strategies, and result optimization.

3.3.1. Query Optimization and Decomposition

Query optimization studies how to improve user input to enhance retrieval quality. Various approaches have emerged to address this challenge. The HyDE method [22] generates hypothetical documents for retrieval, while query rewriting strategies like BEQUE [23]

focus on enhancing sparse queries in specialized domains. Several researchers [24–26] have developed systems that identify user intent through specialized models and route queries to appropriate retrieval methods, significantly improving performance in contexts ranging from multi-agent QA systems to geographic question answering and military applications. For complex queries, advanced frameworks [27–29] employ techniques such as query decomposition, unified reading comprehension, and chain-of-thought reasoning with recursive retrieval, allowing systems to break down complex questions and retrieve information iteratively as needed.

3.3.2. Hybrid Retrieval and Dynamic Strategies

Hybrid retrieval combines methods such as BM25, dense vector retrieval, and sparse encoders, leveraging their complementary strengths for both keyword matching and semantic understanding. Multiple researchers [30–32] have validated hybrid retrieval approaches in professional domains including power knowledge bases, industrial knowledge management, and building engineering. These systems typically combine traditional search methods with embedding processing and specialized ranking mechanisms.

Domain-specific applications have further enhanced these approaches through specialized frameworks [33,34] that incorporate dynamic fine-grained methods and query rewriting techniques. Recent advances in adaptive retrieval strategies [35–37] have demonstrated promising results in specialized domains ranging from multi-objective math problems to dialogue state correction and equipment maintenance, often incorporating chain-of-thought reasoning or block-based retrieval. These approaches highlight the growing importance of domain-adaptive retrieval strategies that can dynamically determine when and how to retrieve external knowledge based on content quality and user needs.

3.3.3. Retrieval Result Re-Ranking

Re-ranking plays a crucial role in improving retrieval accuracy. The Re2G framework by Glass et al. [38] merges results from multiple retrieval methods and applies neural re-ranking to optimize initial retrieval results, greatly enhancing final generation quality. The RankRAG method by Yu et al. [39] uses instruction tuning to enable a single LLM to perform both context ranking and answer generation, reducing reliance on separate ranking models. Ren et al. [40] introduced an adaptive two-stage retrieval augmented fine-tuning method for interpreting research policy, combining a two-stage retrieval mechanism with adaptive hard negative sampling. Xu et al. [41] proposed a two-stage consistency learning approach for retrieval information compression in RALMs to enhance performance while maintaining semantic alignment with the teacher model's intent. These approaches focus on improving the quality and relevance of retrieval results, providing a stronger knowledge foundation for the generation stage.

Research on optimizing retrieval mechanisms demonstrates diverse technical approaches, from query improvement, to hybrid retrieval, and result re-ranking, all contributing to advances in RAG retrieval. Query optimization enhances systems' ability to understand user intent through rewriting and decomposition; hybrid retrieval combines the advantages of different methods to improve overall retrieval; re-ranking further refines results to enhance the quality of information provided to the generation model. However, most existing methods still use fixed retrieval quantities, lacking dynamic adjustment capabilities and struggling to adapt to varying query information needs.

In summary, existing indexing methods struggle to simultaneously preserve document hierarchical structure and paragraph-level semantic integrity. They also lack effective mechanisms to represent and utilize hierarchical semantic information. Most methods use fixed top-k retrieval strategies, which cannot dynamically adjust to query characteristics

and document distribution, resulting in insufficient knowledge coverage or redundant information. The HiSem-RAG method proposed in this paper constructs a hierarchical semantic index structure to preserve the natural hierarchy and semantic integrity of documents, and introduces a distribution-aware adaptive threshold mechanism to dynamically adjust the retrieval range, balancing the breadth and precision of knowledge coverage. By organically integrating chunking strategies, index structures, and retrieval mechanisms, HiSem-RAG provides a new solution for QA tasks involving structurally complex documents.

4. Method

This section provides a detailed introduction to the core components and technical details of the HiSem-RAG method. As shown in Figure 2, the central idea of HiSem-RAG is to construct an indexing system that preserves the hierarchical structure of documents while supporting intelligent retrieval. The entire method consists of two main parts: the construction of a hierarchical semantic index structure and a distribution-aware adaptive threshold retrieval mechanism based on this structure. This design fully leverages the inherent hierarchical characteristics of documents, providing efficient and precise retrieval services while maintaining semantic integrity.

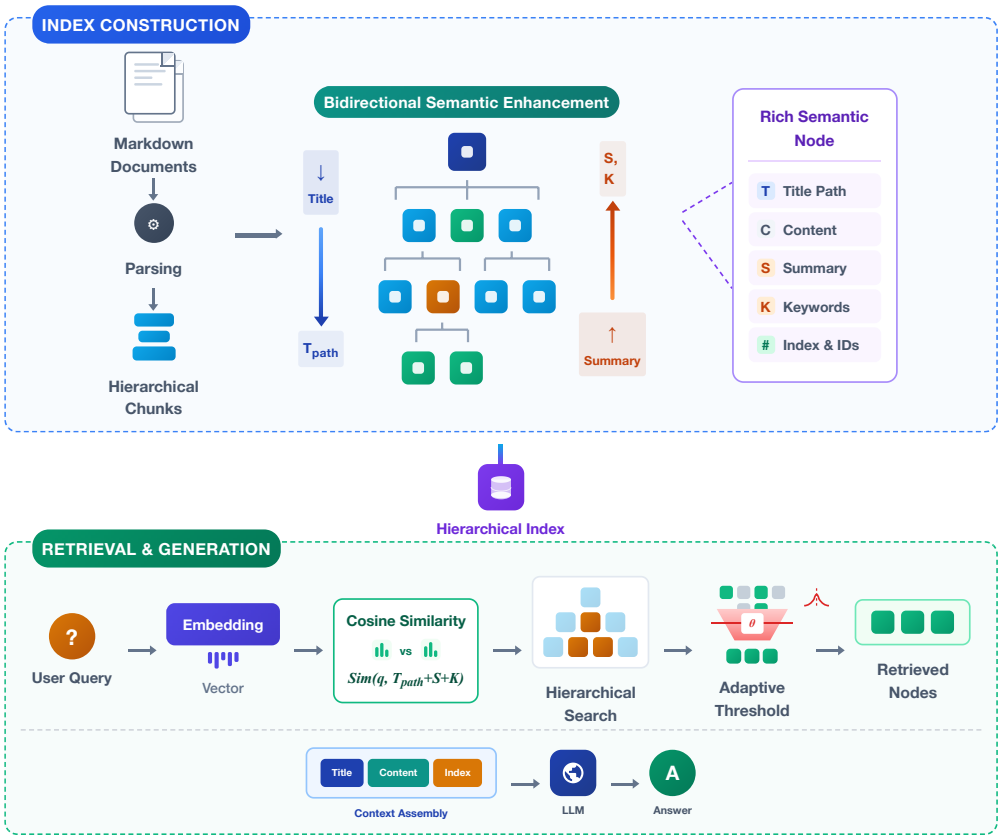


Figure 2. The framework of the HiSem-RAG method.

4.1. Hierarchical Semantic Index Structure

4.1.1. Hierarchical Document Parsing

Hierarchical document parsing is the foundational step of the HiSem-RAG method. Its main goal is to preserve the natural structure of documents, converting original linear text into a tree structure with hierarchical semantic relationships. This process mainly involves three stages: hierarchical title recognition to clarify structural boundaries; tree structure construction to organize documents into multi-level node representations; recursive splitting

of lengthy segments at each level to control retrieval granularity and context length. Essentially, this is a hybrid approach that combines hierarchical modeling and semantic chunking, preserving semantic integrity while providing a structured foundation suitable for long-document processing, which supports subsequent efficient retrieval and generation.

Unlike traditional fixed-length chunking, we parse the multi-level title structure of documents to transform them into tree structures with hierarchical relationships. Formally, given a document d with hierarchical structure, we parse it into a tree $T = \{n_1, n_2, \ldots, n_m\}$, where node n_i represents a semantic unit in the document, such as a section, subsection, or other title-based content fragment. This structure reflects the document's natural hierarchy. Each node can be a content carrier or a container for other nodes, and each node n_i contains the following information:

- Node identifier id: the unique identifier of the node;
- Parent node identifier parent_id: pointer to the parent node;
- Multi-level title path T_{path}: concatenated path from the root to the current node;
- Index: the sequence number after recursive splitting;
- Core knowledge points K: the knowledge points in the node content;
- Content summary S: a concise summary of the node's content;
- Original content: the raw content of the node (empty for non-leaf nodes);
- Children: a list of IDs for all child nodes (empty for leaf nodes).

To handle potentially overlong node content, we employ a recursive character-splitting strategy, using sentence and paragraph boundaries to ensure that segmented content fits model input constraints while preserving semantic integrity as much as possible. For any node n_i with content longer than the preset threshold L_{max}, we recursively split the content, treat the resulting segments as child nodes, and use the index to preserve the original order, ensuring semantic coherence.

4.1.2. Bidirectional Semantic Enhancement

To further improve the expressiveness of semantic units, we design a bidirectional semantic enhancement mechanism. Based on the hierarchical structure, this mechanism models information flow from both directions: top-down to provide structural context and positional awareness; bottom-up to aggregate content summaries and enhance semantic abstraction.

(1) Semantic Block Compression and Key Knowledge Extraction

First, we use a large language model to compress and reconstruct each node's content, extracting knowledge points K and generating summaries S:

$$(K, S) = \text{LLM}(P, \text{content}), \tag{1}$$

where P is the prompt, with specific designs shown in Table 1 (first row). This step effectively filters redundant content, retains important information, and improves the compactness and usability of semantic representations.

Table 1. Overview of prompt design.

Function	Example
Key point and summary extraction	You are a tool for extracting key points and summaries from document paragraphs. Given a text, extract its key points (a set of words separated by commas) and a summary (a paragraph describing the main content). Content: {node.page_content}
Knowledge and summary aggregation	As a document analysis expert, please integrate the key points and summaries of the following subsections to generate the integrated knowledge points and summary for the current section. Current section: {node.title} Current content: {node.page_content} Subsection key points and summaries: {child_summaries} Please generate: 1. Core knowledge points (comma-separated keywords or phrases); 2. Summary (comprehensive summary content). Output format: Key points: [keyword1], [keyword2], . . . Summary: [summary content]
Fill-in-the-blank question evaluation	As an objective and fair expert evaluator, please judge whether the following fill-in-the-blank answer is correct. Question: {question_text} Standard answer: {standard_answer} Answer: {rag_answer} Consider: 1. Semantic consistency; 2. Keyword matching; 3. Terminology accuracy. Only answer "correct" or "incorrect".

(2) Top-Down Semantic Flow

To enhance the structural positioning ability of each node, we construct a multi-level title path T_{path}, which reflects the node's context within the document and aids in structural-aware retrieval. Suppose the path from the root to the current node consists of k nodes $n_1, n_2, \ldots, n_k$, where n_1 is the root and n_k the current node, and $\text{title}(n_i)$ is the title of node n_i. The full title path T_{path} is defined as

$$T_{\text{path}} = \begin{cases} \text{title}(n_1), & \text{if } k = 1, \\ \text{title}(n_1) \rightarrow \text{title}(n_2) \rightarrow \cdots \rightarrow \text{title}(n_k), & \text{if } k > 1, \end{cases} \tag{2}$$

As illustrated in the motivating example (Figure 1), this mechanism explicitly embeds the path "Main Transformer" into the semantic representation of the subsection, effectively distinguishing it from the "Neutral Point" section despite their identical subtitles.

(3) Bottom-Up Semantic Aggregation

To strengthen higher-level nodes' grasp of their subordinate content, we aggregate the knowledge points and summaries of all child nodes:

$$(K_{\text{agg}}, S_{\text{agg}}) = \text{LLM}(P_{\text{agg}}, \{(\text{title}_i, K_i, S_i)\}_{i \in \text{Children}}), \tag{3}$$

where K_{agg} and S_{agg} are the aggregated knowledge points and summary for the parent node, and P_{agg} is a specially designed aggregation prompt (see Table 1, second row). This structure enables higher-level nodes to summarize the core semantics of subordinate nodes, while lower-

level nodes contain detailed information, forming a full semantic hierarchy from abstract to concrete. Referring back to Figure 1, this corresponds to the blue "Summary" box, which aggregates key details from the 23 child items, enabling the parent node to act as a semantic anchor for retrieval. The specific algorithm for hierarchical semantic index construction is shown as Algorithm 1.

Algorithm 1 Build Hierarchical Index

Require: Document collection D, maximum chunk size $L_{\max}$
Ensure: Hierarchical semantic index $\mathcal{I}$

$\mathcal{I} \Leftarrow \emptyset$ ▷ Initialize index
for all $d \in D$ **do**
  $\mathcal{N} \Leftarrow \text{PARSESTRUCTURE}(d)$ ▷ Extract hierarchical nodes
  $r \Leftarrow \mathcal{N}[d.id]$ ▷ Get root node
  for all $n \in \mathcal{N}$ **do**
    if $|n.content| > L_{\max}$ **then**
      $\mathcal{C} \Leftarrow \text{SEMANTICSPLIT}(n.content, L_{\max})$
      $n.content \Leftarrow \epsilon$ ▷ Clear parent content
      for all $c \in \mathcal{C}$ **do**
        $n_c \Leftarrow \text{CREATECHILD}(n, c)$
        $\mathcal{N} \Leftarrow \mathcal{N} \cup \{n_c\}$
      end for
    end if
  end for
  $\text{BUILDTITLEPATHS}(r, \mathcal{N})$ ▷ Top-down enhancement
  $\text{GENERATESUMMARIES}(\mathcal{N})$ ▷ Extract knowledge points
  $\text{AGGREGATEKNOWLEDGE}(r, \mathcal{N})$ ▷ Bottom-up aggregation
  $\mathcal{I}[d.id] \Leftarrow \{r, \mathcal{N}\}$
end for
return $\mathcal{I}$

4.2. Distribution-Aware Adaptive Threshold Retrieval

Based on the hierarchical semantic index, we design a distribution-aware adaptive threshold retrieval mechanism that fully exploits hierarchical document features for intelligent retrieval. Traditional retrieval systems usually use a fixed top-k strategy, returning a fixed number of most similar results regardless of the similarity distribution. This approach has clear limitations: when multiple semantic units are highly relevant to the query, important information might be missed; when most units have low relevance, noise may be introduced.

Our hierarchical recursive retrieval algorithm starts from the root node and explores relevant nodes level by level. For a user query q, we compute the similarity between its vector representation and the semantic-enhanced representations of nodes at the current level, i.e., using "multi-level title path + knowledge points + summary" as the similarity computation object:

$$\text{sim}(q, n_i) = \text{cosine}\Big(\text{Embed}(q), \text{Embed}(T_{\text{path}}(n_i) + K_i + S_i)\Big), \tag{4}$$

where *cosine* is the cosine similarity function, $\text{Embed}(q)$ is the query vector, $T_{\text{path}}(n_i)$ the title path, K_i the knowledge points, and S_i the summary of node n_i. This gives a similarity array reflecting the query's relevance to each semantic-enhanced node.

For these similarity distributions, we propose a distribution-aware adaptive threshold mechanism, which dynamically adjusts the threshold at each level based on statistical characteristics of the similarity distribution. Only nodes with similarity greater than or equal to the threshold are selected as candidate nodes for the next level, continuing

recursively to the leaves. This directly addresses the fragmentation issue shown in Figure 1: instead of a fixed $k = 3$ that would truncate the list, our adaptive threshold detects the high semantic density of the cluster and automatically expands the scope to retrieve all 23 relevant standards as a coherent unit. The threshold is computed as

$$\theta_{\text{raw}} = \beta \cdot s_{\max} - (1 - \gamma \cdot CV) \cdot (s_{\max} - \mu). \tag{5}$$

where

- θ_{raw}: the original dynamic threshold based on similarity distribution;
- $s_{\max}$: the maximum similarity at the current level;
- μ: the average similarity at the current level;
- σ: the standard deviation of similarities at the current level;
- $CV = \sigma/\mu$: the coefficient of variation, measuring distribution dispersion;
- β: base retention coefficient $(0.8 \leq \beta \leq 1)$, controlling the base threshold ratio;
- γ: distribution sensitivity coefficient $(0.8 \leq \gamma \leq 1)$, controlling sensitivity to similarity distribution.

This mechanism sets the initial threshold based on the maximum similarity. When the similarity distribution is more dispersed (large CV), the system adopts a looser filtering criterion; when the distribution is concentrated (small CV), a stricter criterion is used. Intuitively, in a highly concentrated distribution (e.g., $CV < 0.1$), the threshold approaches $\beta \cdot s_{\max}$, resulting in precision-oriented filtering that retains the highly cohesive cluster; conversely, in a highly dispersed distribution (e.g., $CV > 0.5$), the threshold drops more aggressively below $s_{\max}$, allowing a broader set of potentially relevant context nodes to pass through.

However, extreme cases may occur in practice, e.g., all node similarities are low, causing the raw threshold to be too low and selecting too many weakly relevant nodes, which may exceed the model's input limit. To address this, the algorithm incorporates safety mechanisms for computing the final threshold θ:

- Minimum similarity threshold θ_{min} to avoid selecting nodes with excessively low similarity;
- Minimum node count per level k_{min} to prevent over-filtering of potentially relevant nodes;
- Maximum node count per level k_{max} to avoid introducing too much noise;
- Overall maximum result count to control retrieval scale.

After retrieval, HiSem-RAG proceeds to the generation stage, integrating the user query and retrieval results into structured context for the large language model to generate answers. Unlike the retrieval stage, the generation stage uses "multi-level title path + index + original content" as context elements.

Through this distribution-aware recursive retrieval mechanism, the system exhibits significant intelligent adaptability: when the query is highly relevant to certain nodes, the system automatically raises the threshold to retain only highly relevant nodes; when all nodes have similarly high relevance, the system appropriately lowers the threshold to keep more potentially relevant information, dynamically balancing retrieval precision and resource consumption in different query scenarios. The specific algorithm is shown as Algorithm 2.

Algorithm 2 Distribution-Aware Adaptive Threshold Retrieval

Require: Query q, hierarchical index $\mathcal{I} = \{r, \mathcal{N}\}$, parameters β, γ, $\theta_{\min}$, $k_{\max}$
Ensure: Retrieved node set $\mathcal{R}$

$\mathbf{e}_q \Leftarrow \text{EMBED}(q)$ ▷ Query embedding
$\mathcal{R} \Leftarrow \emptyset$; $\mathcal{V} \Leftarrow \emptyset$ ▷ Results and visited sets
SEARCHLEVEL({r}, 0)
return $\mathcal{R}$

function SEARCHLEVEL($\mathcal{N}_\ell, \ell$)
  if $\ell > \ell_{\max}$ **or** $\mathcal{N}_\ell = \emptyset$ **then**
    return
  end if
  $\mathcal{S} \Leftarrow \emptyset$ ▷ Similarity scores at level ℓ
  for all $n \in \mathcal{N}_\ell \setminus \mathcal{V}$ **do**
    $s_n \Leftarrow \text{COSINE}(\mathbf{e}_q, \mathbf{e}_n)$
    $\mathcal{S} \Leftarrow \mathcal{S} \cup \{(n, s_n)\}$
  end for
  if $\mathcal{S} \neq \emptyset$ **then**
    $\theta \Leftarrow \text{ADAPTIVETHRESHOLD}(\mathcal{S}, \beta, \gamma, \theta_{\min})$ ▷ Equation (5)
    for all $(n, s_n) \in \mathcal{S}$ **where** $s_n \geq \theta$ **do**
      $\mathcal{V} \Leftarrow \mathcal{V} \cup \{n\}$
      if n is leaf **then**
        $\mathcal{R} \Leftarrow \mathcal{R} \cup \{(n, s_n)\}$
      else
        SEARCHLEVEL(CHILDREN(n), $\ell + 1$)
      end if
    end for
  end if
end function

By organically combining the hierarchical semantic index structure and the distribution-aware adaptive threshold retrieval mechanism, HiSem-RAG provides a complete solution for processing professional documents with complex hierarchical structures, balancing retrieval comprehensiveness and precision, and offering high-quality context to large language models.

5. Experimental Setup

5.1. Datasets

To verify the effectiveness of the HiSem-RAG method, we evaluate various benchmarks on the public power domain expert dataset EleQA [42] and a self-constructed long-form question answering dataset (LongQA).

EleQA is a publicly available, high-quality dataset in the field of electrical engineering, covering core expert knowledge such as power system operation, equipment maintenance, and safety regulations. This dataset contains 32,610 professional specification clauses and 19,560 QA pairs, with a well-balanced distribution of question types and broad coverage. It features a high degree of professionalism and structuring, making it an important benchmark for assessing models' capabilities in vertical domain knowledge understanding and application.

LongQA focuses on long-form document QA tasks, aiming to test RAG systems' comprehensive ability to handle lengthy texts, multi-level structures, and complex semantic relationships. Based on preset screening criteria, we selected 27 representative long documents from technical manuals and e-books, with average length reaching 480,000 characters per document. These documents are highly hierarchical and structurally complex, covering

multiple professional knowledge domains, and are well-suited for thoroughly evaluating models' abilities in cross-paragraph reasoning and information integration.

For QA pair construction, we adopted a three-stage process: human annotation, large language model-assisted generation, and expert review. First, human annotators designed questions covering different hierarchical levels and information spans to ensure diversity. Then, large language models were used to compress document content and generate preliminary answers. Finally, annotators with domain expertise reviewed each answer, supplementing or correcting model-generated fragments to ensure accuracy and completeness.

In total, we constructed 279 high-quality QA pairs, with an average QA length of 511.23 characters. The main challenges of the LongQA dataset are large document spans, complex semantic relationships between paragraphs, requirements for multi-step reasoning, and cross-level information integration, making it suitable for research on advanced long-text processing.

These two datasets are complementary in scale, structure, knowledge domain, and evaluation focus, together forming a comprehensive experimental foundation for evaluating HiSem-RAG. Table 2 presents their key statistics and comparison dimensions.

Table 2. Statistics of the experimental datasets.

Dataset	Documents Count	QA Pairs	Avg. Doc. Length (Chars)	Question Type Distribution & Avg. QA Length (Chars)
EleQA	32,610	19,560	12,159.20	Single Choice: 6150; Fill-in-the-Blank: 6590; Judgement: 6820 Avg. QA Length: 55.35 chars
LongQA	27	279	478,982.93	QA type: 279 Avg. QA Length: 511.23 chars

5.2. Experimental Design

Three groups of systematic experiments are designed in this study: baseline comparison, resource consumption comparison, and ablation study, to evaluate the overall performance, resource consumption, and component contributions of the HiSem-RAG method, respectively.

5.2.1. Baseline Model Comparison

The baseline comparison experiment aims to comprehensively evaluate the advantages of HiSem-RAG over existing RAG methods. We select five representative RAG methods as baselines, covering both traditional retrieval models and advanced RAG techniques.

For traditional retrieval, we select BM25 and DPR as benchmarks. BM25 calculates relevance based on TF-IDF statistics without relying on neural networks; DPR uses a dual-tower neural encoder to map queries and documents into a shared vector space for similarity computation.

For advanced RAG methods, we select three state-of-the-art approaches: HyDE generates hypothetical documents as retrieval proxies using large language models to enhance query representation; Meta-Chunking adopts an adaptive chunking strategy, combining perplexity and boundary signals to optimize text segmentation; RAPTOR builds document trees via recursive abstraction for multi-level information integration and hierarchical retrieval.

All experiments use the same base model configuration: GLM-4-flash [43] as the large language model and BGE-M3 [44] as the text embedding model.

5.2.2. Resource Consumption Comparison

The resource consumption comparison experiment aims to evaluate the effect of the distribution-aware adaptive threshold mechanism on retrieval efficiency and computational cost. To verify its effectiveness, we design resource consumption comparisons under different retrieval strategies and analyze the trade-off between retrieval quantity and model performance.

In the experiment, we change the fixed retrieval number (topk = 5) of DPR and HyDE to dynamic retrieval based on the adaptive threshold (range 1–7), and also replace HiSem-RAG's adaptive threshold with fixed per-layer retrievals (topk = 2, 3). We then compare the resource consumption differences between fixed and adaptive threshold strategies.

5.2.3. Ablation Study

The ablation study analyzes the independent contribution and synergy of each HiSem-RAG component, validating the rationality of the system's design. We systematically remove or replace key modules and observe changes in system performance to evaluate their importance.

Specifically, we construct three simplified variants: (1) removing the hierarchical index module and using traditional fixed-window chunking; (2) removing the semantic enhancement module, i.e., not using title information propagation and knowledge point extraction; (3) removing the distribution-aware adaptive threshold module and using a fixed retrieval quantity strategy. By comparing the performance of these variants and the full HiSem-RAG on both datasets, we can quantitatively assess the impact of each component on different tasks and gain deeper insight into their mechanisms.

5.3. Evaluation Metrics

For different question types, we design differentiated evaluation metrics for comprehensive and accurate performance assessment. For single-choice, judgement, and fill-in-the-blank questions, we use Accuracy as the main metric, reflecting the model's ability to answer questions correctly. Accuracy is defined as the ratio of correctly answered questions to the total number of questions:

$$\text{Accuracy} = \frac{\text{Number of Correct Answers}}{\text{Total Number of Questions}}. \tag{6}$$

Evaluation standards differ by question type: single-choice and judgement questions are evaluated by strict matching, while fill-in-the-blank questions use semantic judgment, combining large language model and expert review. The prompt for fill-in-the-blank evaluation is shown in Table 1 (third row).

For QA questions, we use BERTScore, ROUGE-L, and MRR@K as evaluation metrics. BERTScore [45] is a semantic similarity-based metric; it compares the generated and reference answers in semantic space, capturing semantically similar but textually different responses. It focuses on deep semantic understanding and is suitable for open-ended QA. The calculation is as follows:

$$\text{BERT}_P = \frac{1}{|\text{BERT}_C|} \sum_{x_i \in \text{BERT}_C} \max_{y_j \in \text{BERT}_{Ref}} \text{sim}(x_i, y_j), \tag{7}$$

$$\text{BERT}_R = \frac{1}{|\text{BERT}_{Ref}|} \sum_{y_j \in \text{BERT}_{Ref}} \max_{x_i \in \text{BERT}_C} \text{sim}(x_i, y_j), \tag{8}$$

$$\text{BERT}_{F1} = 2 \times \frac{\text{BERT}_P \times \text{BERT}_R}{\text{BERT}_P + \text{BERT}_R}. \tag{9}$$

where BERT_C is the set of tokens in the candidate answer, $|\text{BERT}_C|$ is its size, BERT_{Ref} is the set of tokens in the reference answer, $|\text{BERT}_{Ref}|$ is its size, BERT_P is precision, BERT_R is recall, and $\text{sim}(x_i, y_j)$ is the cosine similarity between candidate token x_i and reference token y_j.

ROUGE-L is a sequence-matching metric that computes the longest common subsequence between generated and reference texts, assessing whether the generated answer accurately covers core information and maintains reasonable word order. Compared to n-gram exact matches, ROUGE-L focuses more on structural similarity and can tolerate local word order changes. The calculation is as follows:

$$\text{ROUGE}_R = \frac{\text{LCS}(\text{ROUGE}_X, \text{ROUGE}_Y)}{\text{len}(\text{ROUGE}_Y)}, \tag{10}$$

$$\text{ROUGE}_P = \frac{\text{LCS}(\text{ROUGE}_X, \text{ROUGE}_Y)}{\text{len}(\text{ROUGE}_X)}, \tag{11}$$

$$\text{ROUGE}_L = \frac{(1+\beta^2) \times \text{ROUGE}_R \times \text{ROUGE}_P}{\text{ROUGE}_R + \beta^2 \times \text{ROUGE}_P}. \tag{12}$$

where ROUGE_X is the reference answer, ROUGE_Y is the generated answer, LCS is the longest common subsequence, ROUGE_R is recall, ROUGE_P is precision, and we set the balance factor $\beta = 1$ to balance precision and recall.

Mean Reciprocal Rank (MRR) is an important metric for evaluating the ranking quality of retrieval systems, focusing on the rank position of the correct answer in retrieval results. It gives higher weight to correct answers ranked higher. MRR@K only considers the top K results, and is defined as

$$\text{MRR@K} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i} \cdot I(\text{rank}_i \leq K). \tag{13}$$

where $|Q|$ is the total number of queries, rank_i is the rank of the correct answer for the i-th query, and $I(\cdot)$ is the indicator function to include only answers ranked within K.

5.4. Experimental Environment and Parameter Settings

5.4.1. Hardware Environment

All experiments are conducted on a server equipped with an Intel(R) Xeon(R) Platinum 8375C CPU @2.90 GHz and an NVIDIA A800 80 GB graphics processor.

5.4.2. Parameter Settings

In the HiSem-RAG method, the parameters are configured based on the statistical properties of semantic similarity distributions and hardware constraints, serving as architectural constants rather than sensitive hyperparameters requiring dataset-specific tuning.

- Adaptive Threshold Coefficients ($\beta = 0.9, \gamma = 0.8$): The base retention coefficient β is set to 0.9 to ensure that only chunks with high relative similarity to the top result are retained. The distribution sensitivity coefficient $\gamma = 0.8$ allows the threshold to adapt to the variance of similarity scores. These values are selected as robust statistical heuristics to capture the semantically cohesive "head" of the distribution across different contexts.
- Safety Bounds ($\theta_{min} = 0.3, k_{max} = 7$): The absolute minimum similarity threshold θ_{min} is set to 0.3 to filter out clearly irrelevant noise. The maximum per-layer node count k_{max} is limited to 7. This upper bound is determined primarily by the context window limits of the LLM to prevent context overflow, functioning as an engineering safeguard rather than a retrieval variable.

- Global Constraints: The overall maximum number of retrieved results is set to 15, and the chunk length threshold is 1024 tokens. These settings are defined to balance information completeness with the processing capacity of the underlying model (GLM-4-flash).

6. Experimental Results and Analysis

6.1. Main Results

6.1.1. Baseline Model Comparison Analysis

The results of the baseline model comparison experiment are shown in Tables 3 and 4.

Table 3. Experimental results of baseline model comparison (EleQA).

Method	Single Choice Acc.	Judgment Acc.	Fill-in-the-Blank Acc.	Overall Acc.
BM25	71.28%	24.77%	41.46%	45.16%
DPR	72.35%	67.98%	67.67%	69.27%
HyDE	80.19%	75.42%	**75.48%**	76.96%
Meta-Chunking	79.01%	73.42%	72.62%	74.93%
RAPTOR	82.18%	78.02%	74.10%	78.02%
HiSem-RAG	**83.12%**	**88.62%**	74.13%	**82.00%**

Best results highlighted in bold.

Table 4. Experimental results of baseline model comparison (LongQA).

Method	ROUGE-L	BERT_P	BERT_R	BERT_F1	MRR@1	MRR@3	MRR@5
BM25	-	-	-	-	-	-	-
DPR	0.437	0.805	0.828	0.815	0.311	0.412	0.438
HyDE	0.460	0.811	0.831	0.819	0.333	0.443	0.466
Meta-Chunking	0.508	0.821	**0.843**	0.831	0.379	0.455	0.469
RAPTOR	0.463	0.812	0.831	0.820	0.372	0.480	0.509
HiSem-RAG	**0.599**	**0.837**	**0.843**	**0.839**	**0.458**	**0.585**	**0.587**

Best results highlighted in bold.

The experimental results show that the HiSem-RAG method achieves superior performance across multiple evaluation dimensions. On the EleQA dataset, HiSem-RAG achieves an overall accuracy of 82.00%, outperforming other baseline methods. Analysis by question type shows that HiSem-RAG performs well in both single-choice and judgment tasks, mainly due to the synergy of its three core mechanisms: the hierarchical semantic index helps preserve the integrity of semantic boundaries between paragraphs; the bidirectional semantic enhancement transmits context information via title paths and knowledge point summaries; the adaptive threshold effectively adjusts the retrieval strategy. Notably, in fill-in-the-blank tasks, HyDE slightly outperforms HiSem-RAG and RAPTOR, possibly due to its use of hypothetical document generation as a retrieval proxy, which offers an advantage in precisely locating information fragments. This highlights the need to optimize retrieval strategies for different task types.

In the LongQA QA evaluation, HiSem-RAG not only demonstrates strong performance in generation quality metrics (with a ROUGE-L of 0.599, significantly higher than other methods) but also shows an advantage in semantic similarity (BERT_F1 of 0.839). These results indicate that hierarchical semantic indexing and dynamic similarity thresholding are beneficial for long-document processing. When dealing with documents averaging nearly 480,000 characters, HiSem-RAG preserves document structure well and adjusts retrieval granularity according to query characteristics, alleviating the information fragmentation problem faced by traditional methods.

Analysis of retrieval ranking quality further supports these observations. For MRR@1, HiSem-RAG reaches 0.458, about 8 percentage points higher than the second-best, Meta-Chunking, indicating stronger ability to rank relevant documents at the top. The close values of HiSem-RAG's MRR@3 and MRR@5 suggest its retrieval results are concentrated in the top three, helping reduce retrieval noise. By comparison, RAPTOR's MRR increases more with K, indicating a more dispersed distribution of relevant results.

Through the organic combination of hierarchical semantic indexing, bidirectional semantic enhancement, and distribution-aware adaptive thresholding, HiSem-RAG flexibly adjusts retrieval strategies while maintaining overall document structure. This leads to significant advantages in most tasks, especially long-document processing, making it practically valuable for complex QA and document understanding tasks. Of course, there is still room for improvement in specific tasks such as fill-in-the-blank, pointing to directions for future research.

6.1.2. Resource Consumption Analysis

To validate the effectiveness of the distribution-aware adaptive threshold mechanism, we conducted resource consumption comparison experiments. We applied adaptive thresholds to existing RAG methods and analyzed their impact on HiSem-RAG performance. Specifically, we replaced the fixed retrieval number (topk = 5) of DPR and HyDE with adaptive threshold-based dynamic retrieval (range 1–7), and also replaced HiSem-RAG's adaptive threshold with fixed per-layer retrieval (topk = 2, 3). Experimental results are shown in Figure 3.

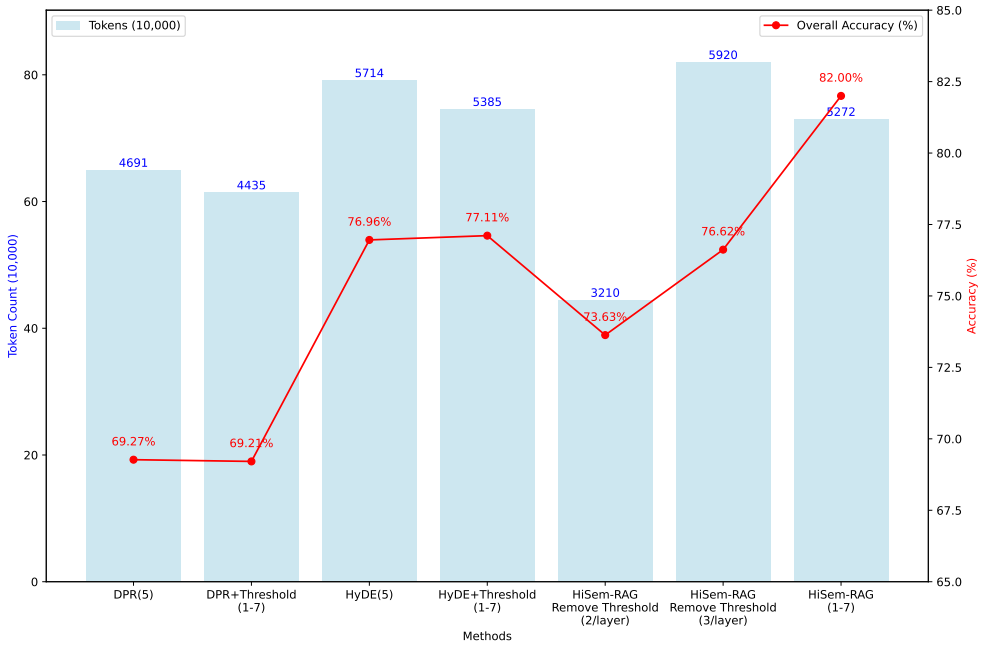


Figure 3. Experimental results on accuracy and resource consumption.

The results show that the adaptive threshold mechanism effectively balances retrieval quality and resource consumption. Applying this mechanism to traditional RAG methods reduces resource consumption while maintaining nearly unchanged accuracy. For example, when DPR uses adaptive thresholds, accuracy drops by just 0.06%, but token consumption decreases by 5.46%. For HyDE, accuracy slightly increases with a 5.75% reduction in token usage. This demonstrates that adaptive thresholds can intelligently adjust retrieval quantity based on query characteristics, avoiding retrieval of excessive irrelevant content.

Analysis of different HiSem-RAG configurations further illustrates this point. With a fixed retrieval of 2 document blocks per layer, token consumption is minimized (32.1 million), but accuracy is limited (73.63%) due to insufficient retrieval. With 3 per layer, accuracy rises to 76.62% but token consumption surges to 59.2 million, indicating clear redundancy. Crucially, even with higher resource use, fixed-3 retrieval's accuracy is still significantly lower than the adaptive threshold scheme (82.00%). This is because a fixed top-*k* not only introduces redundancy but also restricts the flexibility of retrieval: some layers require more blocks for key information, while others need only 1–2.

Adaptive thresholding in HiSem-RAG allows dynamic adjustment of retrieval range per layer and query, expanding it when needed (up to 7 blocks) or narrowing it when information is concentrated, thus saving about 11% computation resources while improving accuracy. This mechanism enables the model to break free from fixed window constraints and explore key regions more extensively, avoiding missing important but not absolutely top-ranked documents. These results demonstrate the limitations of fixed top-*k* retrieval: too small k leads to insufficient information, too large k introduces redundancy and extra computation. The adaptive threshold mechanism dynamically determines the optimal retrieval quantity, ensuring both quality and coverage while avoiding unnecessary resource waste.

6.2. Ablation Study

To assess the effectiveness of each HiSem-RAG component, we conducted systematic ablation experiments, removing the three core modules (hierarchical index, semantic enhancement, and adaptive threshold) in turn. Results are shown in Tables 5 and 6.

Table 5. Ablation study results (EleQA).

Method	Single Choice Acc.	Judgment Acc.	Fill-in-the-Blank Acc.	Overall Acc.
without Hierarchical Index	78.98%	73.69%	73.74%	75.39%
without Semantic Enhancement	77.58%	74.63%	73.22%	75.08%
without Adaptive Threshold	80.32%	86.55%	62.90%	76.62%
HiSem-RAG	83.12%	88.62%	74.13%	82.00%

Table 6. Ablation study results (LongQA).

Method	ROUGE-L	BERT_P	BERT_R	BERT_F1	MRR@1	MRR@3	MRR@5
without Hierarchical Index	0.459	0.809	0.830	0.818	0.333	0.443	0.466
without Semantic Enhancement	0.346	0.768	0.775	0.770	0.147	0.292	0.334
without Adaptive Threshold	0.501	0.799	0.798	0.797	0.383	0.494	0.499
HiSem-RAG	0.599	0.837	0.843	0.839	0.458	0.585	0.587

The ablation results highlight the contributions of each HiSem-RAG component across different datasets. All three core modules play significant roles in both tasks, but their impact and mechanisms vary.

The hierarchical index module is foundational for both datasets. On EleQA, removing it reduces overall accuracy by 6.61%; on LongQA, ROUGE-L drops by 23.4% and MRR@1 by 27.3%. This reflects the limitations of traditional fixed-window chunking, which easily disrupts semantic integrity. Hierarchical indexing preserves natural paragraph boundaries and semantic coherence, enabling the model to better grasp document logic. This is especially important for long documents, where multi-level semantic structure is common and maintaining integrity helps the retrieval system understand organization.

The semantic enhancement module has the most pronounced impact. Its removal causes a 6.92% drop in overall EleQA accuracy, a 42.2% decrease in ROUGE-L, and a 67.9% decrease in MRR@1 on LongQA. This indicates that semantic enhancement is crucial for connecting document fragments and supplementing context: title propagation restores lost semantic links during chunking, while knowledge extraction and summarization filter out irrelevant content. Lack of semantic ties severely impacts retrieval quality, especially in long documents where fragments often need more context for accurate understanding.

The adaptive threshold module shows differentiated effects across task types. On EleQA, its effect is most marked on fill-in-the-blank questions: removing it drops accuracy from 74.13% to 62.90%. On LongQA, MRR@1 drops from 0.458 to 0.383. This validates the limitations of fixed-retrieval-number strategies, while adaptive thresholding can dynamically widen or narrow the retrieval scope within hierarchical structures according to query characteristics. This flexibility is especially important for fill-in-the-blank tasks needing precise information localization, as key information may be in non-topmost but relevant areas. For long documents, the mechanism balances retrieval breadth and depth, improving quality while reducing redundancy.

In summary, the ablation study demonstrates the complementary roles of the three core modules. The hierarchical index provides semantically complete base units, semantic enhancement supplements context and deepens understanding, and adaptive thresholding optimizes retrieval strategy to expand relevant coverage. This combination enables HiSem-RAG to adapt to various QA tasks, delivering high performance with reduced computational cost.

6.3. Qualitative Case Study: Retrieval Completeness Analysis

To intuitively demonstrate how HiSem-RAG improves contextual completeness compared to traditional methods, we present a detailed step-by-step analysis of a real-world query from the LongQA dataset: “What are the process standards for main transformer installation? ” (Refer to the visualized comparison in Figure 1).

Scenario Setup: The source document contains structural ambiguity, where the subsection “2. Process Standards” recurs across **multiple chapters** (including Main Transformer System, Neutral Point System, etc.) with identical naming but distinct contexts. The user’s intent specifically targets the standards within the “Main Transformer” chapter.

Step 1: Baseline Retrieval (Traditional RAG) As shown in Figure 1B, traditional flat retrieval relies heavily on keyword matching and fixed truncation strategies.

- False Positive: The chunk from the Neutral Point System (Chunk 7) achieves the highest similarity score (0.83) because its content heavily overlaps with the query terms, despite being semantically irrelevant to the requested equipment.
- Context Fragmentation: The subsequent relevant chunk (Chunk 4), which contains critical installation details like “relay protection,” yields a significantly lower similarity score (0.62) because it lacks direct keyword overlap with the query. Consequently, it is excluded by standard top-k or threshold filtering. Furthermore, even if Chunk 4 were retrieved, it would lack the necessary hierarchical context (i.e., belonging to “Main Transformer”), making it difficult for the LLM to interpret correctly.
- Outcome: The generated answer would likely hallucinate by mixing specifications from the wrong system or provide an incomplete response due to missing details.

Step 2: HiSem-RAG Retrieval (Ours) As shown in Figure 1C, our method employs structural encoding and adaptive thresholding.

- Path-Aware Scoring: By incorporating the title path “Ch.1 > Main Transformer”, the similarity score of the correct Chunk 3 is boosted to 0.91. Conversely, the distractor

(Chunk 7) is penalized due to the path mismatch ("Neutral Point" vs. "Main Transformer"), dropping its score significantly to 0.45.

- Adaptive Expansion: Based on the similarity distribution of the retrieved candidates, the system calculates a dynamic threshold $\theta = 0.80$.
- Outcome: The system successfully includes the entire cluster of relevant chunks (Chunks 3, 4–8, all scores > 0.88) while rigorously filtering out the distractor ($0.45 < \theta$). This ensures the LLM receives the complete, non-fragmented procedure for the correct equipment.

This case study confirms that HiSem-RAG effectively utilizes structural semantics to resolve ambiguity and employs adaptive thresholds to guarantee contextual integrity, addressing the limitations observed in baselines.

7. Discussion

This section interprets the results regarding our research questions, compares HiSem-RAG with related work, and discusses practical implications.

7.1. Answering Research Questions

Regarding RQ1 (Structure Preservation): Results confirm that preserving natural hierarchical boundaries minimizes information fragmentation. Ablation studies (Table 5) show that removing the hierarchical index drops overall EleQA accuracy by 6.61%. This aligns with the "structural ambiguity" issue in Figure 1, where traditional flat chunking fails to distinguish identical subtitles in different chapters. HiSem-RAG eliminates this ambiguity by enforcing physical document structure.

Regarding RQ2 (Semantic Enhancement): The bidirectional enhancement mechanism is critical for long-context understanding. On LongQA (Table 6), removing this module caused a 42.2% drop in ROUGE-L and a 67.9% drop in MRR@1. This answers RQ2: explicitly injecting title paths (Top-Down) and aggregating summaries (Bottom-Up) effectively bridges the semantic gap between distant layers, ensuring retrieved fragments carry full context.

Regarding RQ3 (Adaptive Retrieval): Resource consumption experiments (Figure 3) address RQ3, showing that dynamic retrieval outperforms fixed strategies. The distribution-aware threshold reduced token consumption by approximately 11% while maintaining high accuracy (82.00%). This confirms that the optimal retrieval scope is a dynamic variable determined by semantic density, effectively solving the "fragmentation vs. redundancy" dilemma.

7.2. Comparison with Related Work

Unlike DPR [2] and BM25, which treat documents as flat bags of words, HiSem-RAG explicitly encodes structural signals. This significantly reduces hallucinations caused by overlapping keywords in different sections.

Compared to RAPTOR [6], which builds hierarchy via bottom-up clustering, HiSem-RAG utilizes the explicit document structure. While RAPTOR excels with unstructured text, our method shows superior performance (4% higher accuracy on EleQA) for domain-specific documents where the table of contents provides a logical backbone. This suggests that "following the author's logic" is often more effective and computationally cheaper than "rediscovering logic via clustering" for professional manuals.

HiSem-RAG also outperforms HyDE [22] in precise constraint checking on LongQA. HyDE relies on LLM hallucinations, which can be unstable; HiSem-RAG grounds retrieval in the actual document structure, ensuring higher faithfulness in high-stakes domains.

7.3. Parameter Robustness Analysis

A key motivation for adaptive thresholding is eliminating the dependency on the rigid top-k hyperparameter. Unlike top-k, our parameters (β, γ) operate on the statistical properties of similarity scores.

- Robustness of β and γ: β defines the tolerance range relative to the best match. Setting $\beta \approx 0.9$ ensures only chunks with similarity scores close to the top result are retained. γ adjusts this based on variance (CV). Theoretical analysis suggests that as long as β is in a high confidence range (e.g., 0.85∼0.95), the retrieved set remains stable.
- Role of k_{max}: k_{max} acts solely as an engineering upper bound to protect the LLM context window, not as a primary retrieval logic parameter.

Thus, these parameters act as architectural constants rather than sensitive hyperparameters, allowing HiSem-RAG to generalize across datasets without fine-tuning. We note that preliminary experiments across both EleQA and LongQA datasets confirmed stable performance when β and γ varied within $[0.8, 1.0]$; detailed sensitivity curves are omitted for brevity, as the observed variance in accuracy was less than 1%.

7.4. Applicability to Unstructured Environments

While HiSem-RAG excels with structured documents, it currently relies on explicit markers (e.g., headings). In scenarios with implicit or noisy structures, performance may degrade. However, as noted in future work, this can be mitigated via an "Implicit Structure Induction" pipeline. Semantic segmentation algorithms can identify logical boundaries in unstructured text, and LLMs can generate synthetic hierarchical titles, transforming implicit semantics into the explicit structure HiSem-RAG requires.

7.5. Implications

Theoretical: This study shifts the RAG indexing perspective from "text-based" to "structure–semantic coupled," highlighting document structure as a dense semantic signal essential for precision.

Practical: For industrial applications like maintenance and engineering, HiSem-RAG offers a "safe" retrieval mechanism. By retrieving complete context clusters, it minimizes the risk of missing safety-critical constraints—a common failure in fixed top-k systems. Additionally, reduced token usage lowers API costs and latency.

8. Conclusions

This paper proposed HiSem-RAG, a method designed to enhance retrieval accuracy in complex, hierarchical documents. By constructing a hierarchical semantic index, employing bidirectional semantic enhancement, and introducing a distribution-aware adaptive threshold, HiSem-RAG effectively resolves issues of structural ambiguity and information fragmentation. Experimental results on EleQA and LongQA datasets demonstrate that our method outperforms state-of-the-art baselines in both accuracy and computational efficiency.

Future Work: Future research will focus on three directions: (1) Unstructured Text Adaptation: Developing implicit structure recognition algorithms to apply HiSem-RAG to texts without explicit headings (e.g., chat logs or flat reports). (2) Multimodal Integration: Extending the hierarchical index to include images and tables, which are prevalent in technical manuals. (3) Lightweight Enhancement: Investigating the use of smaller, distilled models for the semantic enhancement step to further reduce the offline indexing cost.

https://doi.org/10.3390/app16020903

Author Contributions: Conceptualization, D.Y.; methodology, D.Y. and J.W.; software, J.W.; validation, J.W.; formal analysis, J.W.; investigation, J.W.; resources, D.Y.; data curation, J.W.; writing—original draft preparation, J.W.; writing—review and editing, D.Y. and J.W.; visualization, J.W.; supervision, D.Y.; project administration, D.Y.; funding acquisition, D.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China International (Regional) Cooperation and Exchange Project, grant number 62061136006.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets and code generated and analyzed during the current study are available in the GitHub repository at https://github.com/CharmingDaiDai/HiSem-RAG (accessed on 23 April 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2020; Volume 33, pp. 9459–9474.
2. Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.S.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.-t. Dense Passage Retrieval for Open-Domain Question Answering. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6769–6781.
3. Sawarkar, K.; Mangal, A.; Solanki, S.R. Blended rag: Improving rag (retriever-augmented generation) accuracy with semantic search and hybrid query-based retrievers. In Proceedings of the 2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR), San Jose, CA, USA, 7–9 August 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 155–161.
4. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
5. Mansurova, A.; Mansurova, A.; Nugumanova, A. QA-RAG: Exploring LLM reliance on external knowledge. *Big Data Cogn. Comput.* **2024**, *8*, 115. [CrossRef]
6. Sarthi, P.; Abdullah, S.; Tuli, A.; Khanna, S.; Goldie, A.; Manning, C.D. Raptor: Recursive abstractive processing for tree-organized retrieval. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
7. Wu, C.; Ding, W.; Jin, Q.; Jiang, J.; Jiang, R.; Xiao, Q.; Liao, L.; Li, X. Retrieval augmented generation-driven information retrieval and question answering in construction management. *Adv. Eng. Inform.* **2025**, *65*, 103158. [CrossRef]
8. Edge, D.; Trinh, H.; Cheng, N.; Bradley, J.; Chao, A.; Mody, A.; Truitt, S.; Metropolitansky, D.; Ness, R.O.; Larson, J. From local to global: A graph rag approach to query-focused summarization. *arXiv* **2024**, arXiv:2404.16130. [CrossRef]
9. Gao, F.; Xu, S.; Hao, W.; Lu, T. KA-RAG: Integrating Knowledge Graphs and Agentic Retrieval-Augmented Generation for an Intelligent Educational Question-Answering Model. *Appl. Sci.* **2025**, *15*, 12547. [CrossRef]
10. Li, X.; Peng, S.; Yada, S.; Wakamiya, S.; Aramaki, E. GenKP: generative knowledge prompts for enhancing large language models. *Appl. Intell.* **2025**, *55*, 464. [CrossRef]
11. Zhang, F.; Luo, Y.; Gao, Z.; Han, A. Injury degree appraisal of large language model based on retrieval-augmented generation and deep learning. *Int. J. Law Psychiatry* **2025**, *100*, 102070. [CrossRef] [PubMed]
12. Bahr, L.; Wehner, C.; Wewerka, J.; Bittencourt, J.; Schmid, U.; Daub, R. Knowledge graph enhanced retrieval-augmented generation for failure mode and effects analysis. *J. Ind. Inf. Integr.* **2025**, *45*, 100807. [CrossRef]
13. Choi, B.; Lee, Y.; Kyung, Y.; Kim, E. ALBERT with Knowledge Graph Encoder Utilizing Semantic Similarity for Commonsense Question Answering. *Intell. Autom. Soft Comput.* **2023**, *36*, 71–82. [CrossRef]
14. Theja, R. Evaluating the ideal chunk size for a rag system using llamaindex. *LLAMAi [Online]* **2023**, *30*, 31. Available online: https://www.llamaindex.ai/blog/evaluating-the-ideal-chunk-size-for-a-rag-system-using-llamaindex-6207e5d3fec5 (accessed on 5 October 2023).
15. Xu, P.; Ping, W.; Wu, X.; McAfee, L.; Zhu, C.; Liu, Z.; Subramanian, S.; Bakhturina, E.; Shoeybi, M.; Catanzaro, B. Retrieval meets long context large language models. In Proceedings of the Twelfth International Conference on Learning Representations, Vienna, Austria, 7–11 May 2024.
16. Yang, S. Advanced Rag 01: Small-to-Big Retrieval. 2023. Available online: https://medium.com/data-science/advanced-rag-01-small-to-big-retrieval-172181b396d4 (accessed on 4 November 2023).

17. Krassovitskiy, A.; Mussabayev, R.; Yakunin, K. LLM-Enhanced Semantic Text Segmentation. *Appl. Sci.* **2025**, *15*, 10849. [CrossRef]
18. Zhao, J.; Ji, Z.; Feng, Y.; Qi, P.; Niu, S.; Tang, B.; Xiong, F.; Li, Z. Meta-chunking: Learning efficient text segmentation via logical perception. *arXiv* **2024**, arXiv:2410.12788. [CrossRef]
19. Wang, K.; Reimers, N.; Gurevych, I. DAPR: A benchmark on document-aware passage retrieval. *arXiv* **2023**, arXiv:2305.13915.
20. Zheng, Z.; Zhang, O.; Borgs, C.; Chayes, J.T.; Yaghi, O.M. ChatGPT chemistry assistant for text mining and the prediction of MOF synthesis. *J. Am. Chem. Soc.* **2023**, *145*, 18048–18062. [CrossRef]
21. Bi, F.; Zhang, Q.; Zhang, J.; Wang, Y.; Chen, Y.; Zhang, Y.; Wang, W.; Zhou, X. A Retrieval-Augmented Generation System for Large Language Models Based on Sliding Window Strategy. *J. Comput. Res. Dev.* **2025**, *62*, 1597–1610.
22. Gao, L.; Ma, X.; Lin, J.; Callan, J. Precise zero-shot dense retrieval without relevance labels. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 1762–1777.
23. Peng, W.; Li, G.; Jiang, Y.; Wang, Z.; Ou, D.; Zeng, X.; Xu, D.; Xu, T.; Chen, E. Large language model based long-tail query rewriting in taobao search. In Proceedings of the Companion Proceedings of the ACM Web Conference 2024, Singapore, 13–17 May 2024; pp. 20–28.
24. Liu, B. Demystifying the black box: AI-enhanced logistic regression for lead scoring. *Appl. Intell.* **2025**, *55*, 574. [CrossRef]
25. Tupayachi, J.; Li, X. Conversational Geographic Question Answering for Route Optimization: An LLM and Continuous Retrieval-Augmented Generation Approach. In Proceedings of the 17th ACM SIGSPATIAL International Workshop on Computational Transportation Science GenAI and Smart Mobility Session, Atlanta, GA, USA, 29 October–1 November 2024; pp. 56–59.
26. Zhang, Y.; Chen, M.; Tian, C.; Yi, Z.; Hu, W.; Luo, W.; Luo, Z. A Multi-Strategy Retrieval-Augmented Generation Method for Knowledge-Based Question Answering in the Military Domain. *Comput. Appl.* **2025**, *45*, 746–754.
27. Ma, X.; Gong, Y.; He, P.; Zhao, H.; Duan, N. Query rewriting in retrieval-augmented large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 5303–5315.
28. Kim, G.; Kim, S.; Jeon, B.; Park, J.; Kang, J. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 996–1009.
29. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv* **2023**, arXiv:2210.03629. [CrossRef]
30. Zhang, J.; Wang, T.; Yao, C.; Xie, H.; Chai, L.; Liu, S.; Li, T.; Li, Z. Construction and Evaluation of an Intelligent Question Answering System for Electric Power Knowledge Base Based on Large Language Models. *Comput. Sci.* **2024**, *51*, 286–292.
31. Chen, L.C.; Pardeshi, M.S.; Liao, Y.X.; Pai, K.C. Application of retrieval-augmented generation for interactive industrial knowledge management via a large language model. *Comput. Stand. Interfaces* **2025**, *94*, 103995. [CrossRef]
32. Wang, Z.; Liu, Z.; Lu, W.; Jia, L. Improving knowledge management in building engineering with hybrid retrieval-augmented generation framework. *J. Build. Eng.* **2025**, *103*, 112189. [CrossRef]
33. Wan, Y.; Chen, Z.; Liu, Y.; Chen, C.; Packianather, M. Empowering LLMs by hybrid retrieval-augmented generation for domain-centric Q&A in smart manufacturing. *Adv. Eng. Inform.* **2025**, *65*, 103212.
34. Zhang, H.; Hao, W.; Jin, D.; Cheng, K.; Zhai, Y. DF-RAG: A Retrieval-Augmented Generation Method Based on Query Rewriting and Knowledge Selection. *Comput. Sci.* **2025**, *52*, 30–39.
35. Sun, J.; Shi, W.; Shen, X.; Liu, S.; Wei, L.; Wan, Q. Multi-objective math problem generation using large language model through an adaptive multi-level retrieval augmentation framework. *Inf. Fusion* **2025**, *119*, 103037. [CrossRef]
36. Su, H.; Xie, H.; Shi, J.; Wu, D.; Jiang, L.; Huang, H.; He, Z.; Li, Y.; Fang, R.; Zhao, J.; et al. RAICL-DSC: Retrieval-Augmented In-Context Learning for Dialogue State Correction. *Knowl.-Based Syst.* **2025**, *317*, 113423. [CrossRef]
37. He, Z.; Jiang, B.; Wang, X. Improved Retrieval Augmentation and LLM Chain-of-Thought for Maintenance Strategy Generation. *Comput. Appl. Softw.* **2025**, *42*, 1–6+83.
38. Glass, M.; Rossiello, G.; Chowdhury, M.F.M.; Naik, A.R.; Cai, P.; Gliozzo, A. Re2G: Retrieve, rerank, generate. *arXiv* **2022**, arXiv:2207.06300. [CrossRef]
39. Yu, Y.; Ping, W.; Liu, Z.; Wang, B.; You, J.; Zhang, C.; Shoeybi, M.; Catanzaro, B. Rankrag: Unifying context ranking with retrieval-augmented generation in llms. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2024; Volume 37, pp. 121156–121184.
40. Ren, R.; Ma, J.; Zheng, Z. Large language model for interpreting research policy using adaptive two-stage retrieval augmented fine-tuning method. *Expert Syst. Appl.* **2025**, *278*, 127330. [CrossRef]
41. Xu, C.; Zhao, D.; Wang, B.; Xing, H. Enhancing Retrieval-Augmented LMs with a Two-Stage Consistency Learning Compressor. In Proceedings of the International Conference on Intelligent Computing, Tianjin, China, 5–8 August 2024; Springer: Singapore, 2024; pp. 511–522.
42. Wang, H.; Wei, J.; Jing, H.; Song, H.; Xu, B. Meta-RAG: A Metadata-Driven Retrieval-Augmented Generation Framework for the Electric Power Domain. *Comput. Eng.* **2024**, 1–11. [CrossRef]

43. GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Zhang, D.; Rojas, D.; Feng, G.; Zhao, H.; et al. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv* **2024**, arXiv:2406.12793.
44. Chen, J.; Xiao, S.; Liu, P.; Zhang, K.; Lian, D.; Xie, X.; Li, D. M3-Embedding: Multi-Linguality, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *arXiv* **2024**, arXiv:2402.03216. [CrossRef]
45. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv* **2019**, arXiv:1904.09675. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.