Knowledge and Context Compression via Question Generation

Alex Anvi Eponon *, Moein Shahiki-Tash, Abdullah Abdullah, Luis Ramos, Christian Maldonado-Sifuentes, Gregori Sidorov

Posted Date: 22 January 2026

doi: 10.20944/preprints202601.1757.v1

Keywords: information retrieval; retrieval augmented; generation RAG; question generation; natural language processing

 doi:10.20944/preprints202601.1757.v1

Article

Knowledge and Context Compression via Question Generation

Anvi Alex Eponon *, Moein Shahiki-Tash, Abdullah, Luis Ramos, Christian Maldonado-Sifuentes, and Grigori Sidorov

Centro de Investigación en Computación (CIC), Instituto Politécnico Nacional, Mexico City, Mexico
* Correspondence: aeponon2023@cic.ipn.mx

Abstract

Retrieval-Augmented Generation (RAG) systems face substantial challenges when navigating large volumes of complex scientific literature while maintaining reliable semantic retrieval, a critical limitation for automated scientific discovery where models must connect multiple research findings and identify genuine knowledge gaps. This study introduces a question-based knowledge encoding method that enhances RAG without fine-tuning to address these challenges. Recognizing the lack of syntactic understanding in major Large Language Models, we generate syntactic and semantic-aligned questions and apply a syntactic reranker without training. Our method improves both single-hop and multi-hop retrieval with Recall@3 to 0.84, representing a 60% gain over standard chunking techniques on scientific papers. On LongBenchQA v1 and 2WikiMultihopQA, which contain 2000 documents each averaging 2k-10k words, the syntactic reranker with LLaMA2-Chat-7B achieves F1 = 0.52, surpassing chunking (0.328) and fine-tuned baselines (0.412). The approach additionally reduces vector storage by 80%, lowers retrieval latency, and enables scalable, question-driven knowledge access for efficient RAG pipelines. To our knowledge, this is the first work to combine question-based knowledge compression with explicit syntactic reranking for RAG systems without requiring fine-tuning, offering a promising path toward reducing hallucinations and improving retrieval reliability across scientific domains.

Keywords: information retrieval; retrieval augmented; generation RAG; question generation; natural language processing

1. Introduction

Retrieval-augmented generation (RAG) is a retrieval-based generation system that reduces hallucinations in large language models (LLMs) by instructing the model to retrieve information outside the model boundaries during inference [1,3].

In an attempt to improve semantic understanding, research has mostly concentrated on architectural innovations and domain-specific fine-tuning techniques. Although these guidelines are helpful, the chunking strategy, the process of segmenting documents for retrieval is an important but frequently overlooked element. As demonstrated in the literature review section (Section 2), chunking is crucial in determining retrieval accuracy. If the input context is poorly segmented or not aligned with the query intent, even the most semantically aware models may perform poorly.

In this study, we propose a novel approach that leverages question and query generation as a form of knowledge and context compression to enable more effective single and multi-document retrieval. The experiments conducted aim at answering:

- Can question-based knowledge encoding improve retrieval performance in RAG systems compared to traditional chunking methods?
- What is the impact of syntactic reranking and "paper-card" summaries on the accuracy and efficiency of information retrieval in scientific texts?

 doi:10.20944/preprints202601.1757.v1

- Can question generation serve as an effective form of knowledge compression for scalable, fine-tuning-free RAG architectures?

By addressing the questions above, we suggest a method that integrates the semantic understanding capabilities of current models for retrieval tasks with explicit syntactic reranking without any training, a method that generate questions as context anchors. This entails the creation of both queries and pre-formulated questions that serve as textual semantic boundaries. Finally, we introduce paper-cards, a new document representation format which increases and improves both LLM and BM25 techniques. The LongBench QA v1 [23] dataset is used to compare the performance approach to that of state-of-the-art chunking techniques, specifically using the 2WikiMultihopQA and the 2WikiMultihopQA_e datasets [23]. The evaluation compares the results of multiple models, including LLaMA3, LLaMA2-chat-7B [24,25] and Vicuna-7B and Vicuna-7B [26], GPT-3.5-Turbo-16k (Generated Pre-trained model) [23].

The current paper is structured as follow: section 2 reviews related work, section 3 and 4 present the research objective and approach, section 5 and 6 highlight the experiment details and results, while in 7, we discuss these results and finally section 8 concludes with insights on RAG scalability.

2. Literature Review

Document retrieval and information extraction remain central challenges in RAG systems, particularly due to the need for effective chunking. Chunking involves splitting text into meaningful segments to support accurate query retrieval [5]. Common strategies include fixed-size, structure-based, semantic or contextual, and hybrid approaches [6]. Semantic chunking has gained prominence with the emergence of fine-tuned models that provide more precise, context-aware segmentation.

2.1. Fixed-Size Chunking

Fixed-size chunking was the first widely adopted technique, emerging with the rise of RAG systems around 2020. It involves dividing a document into equally sized spans, often with an overlapping window to help retain context across segments. When implemented thoughtfully, this method has proven to be both robust and reliable. For instance, [7] introduced RAGs using fixed-length passages of 100 words. Later, [8] compared fixed-size and semantic chunking techniques and found that a fixed size of 200 words could match or even outperform semantic chunking on real-world datasets, also supported by [9]. Their findings highlighted that fixed-size chunking remains a competitive option, especially considering its lower computational cost relative to semantic approaches.

2.2. Structure-Based Chunking

Instead of using fixed token sizes, structure-based (recursive) chunking divides documents according to their natural hierarchy. Consistent benefits over fixed-size chunking have been reported in previous work: [5] obtained 84% accuracy on FinanceBench QA with ROUGE and BLEU scores of 0.57 and 0.45. [14] outperformed BERT-large on CoQA and QuAC with F1 scores of 81.8 and 62.0, respectively, by introducing a recurrent RL-based chunking mechanism to handle lengthy texts beyond BERT limits. Using Qwen2 and Baichuan2 variants, [16] more recently proposed meta-chunking, which produced mid-sized, linguistically coherent units that enhanced BLEU/ROUGE and F1 on LongBench 2WikiMultihopQA. Despite these benefits, structure-based methods have drawbacks in practical situations. Many online documents are not formatted consistently or cleanly, dynamic elements, embedded multimedia, and irregular HTML make automated structural parsing less reliable, which restricts its usefulness at the web scale.

Despite these advantages, structure-based approaches face challenges in real-world scenarios. Many online documents lack clean or consistent formatting; irregular HTML, dynamic elements, and embedded multimedia reduce the reliability of automated structural parsing, limiting practical applicability at web scale.

 doi:10.20944/preprints202601.1757.v1

2.3. Advances in Multihhop Reasoning in RAG Systems

In RAG systems, recent non-finetuned methods aim to enhance multi-hop retrieval and reasoning without supervised adaptation. Semantic dispersion scores are used to iteratively re-rank passages in Vendi-RAG [36] which is a diversity-aware retrieval objective. In comparison to relevance-only baselines, this results in up to +4.1% on 2WikiMultiHopQA. Although this reduces redundancy, it still relies on large chunk inventories and frequent retrieval iterations, which raises latency and offers no way to compact knowledge representations. In a similar vein, Dr3 [37] corrects off-topic reasoning using a multi-stage discriminate–recompose–resolve loop. The method improves exact-match by 3% and reduces off-topic errors by 13%. Even if it is successful in lowering obvious failures, its corrective actions don't deal with the fundamental problem of missing entities during retrieval, and mistakes build up over time. DEC [38] proposes dynamic question decomposition for lightweight LLMs, yet its performance remains highly sensitive to initial retrieval quality, making it vulnerable whenever key phrases or entities are dropped.

Stronger benchmarks are produced by finetuned methods, but they come at a high system cost and an observed time delay that is associated with the complexity of the models. To stabilize multi-hop chains, specialized rerankers, summarizers, or retrievers are trained using Summarize and Plan [39] and data augmented reasoning frameworks [40]. Decomposition quality is strongly dependent on supervised anchoring of entities, as demonstrated by GenDec [41] and Lost-in-Retrieval [42]. However, such finetuned retrievers require significant annotation efforts and do not generalize well across domains without retraining.

Across these systems, three limitations directly affect our target problem. First, none of the existing non-finetuned techniques compress context and knowledge through structured question generation: they operate over full document-chunk indices, leading to heavy storage, slower search, and redundancy. Second, retrieval brittleness persists, especially in decomposition-based methods where entity drift causes entire reasoning chains to collapse. An observation proved that the problem is central to syntactic understanding of models. Third, iterative multi-step loops accumulate error and latency, making them costly for large research corpora that evolve quickly.

Our approach addresses these gaps by introducing a non-finetuned method based on (i) question-driven knowledge compression ("paper-cards") for scientific papers but can be adapted to broader cases, (ii) syntactic reranking for anchor preservation, and (iii) compact vector representations. This reduces embedding storage by roughly 80% while improving syntactical and dense retrieval (e.g., MRR from 0.56 to 0.85). Unlike prior work, our system maintains effectiveness without supervised retrievers, avoids multi-stage error accumulation, and ensures stability through structured syntactic anchors. As such, it provides a scalable, low-maintenance alternative to both heavy fine-tuned pipelines and fragile zero-shot decomposition chains.

2.4. Semantic Chunking Strategies and Syntactic Limitations in RAG Systems

Semantic chunking, the dominant approach in chunk optimization, groups text segments via embeddings and clusters them by semantic similarity. However, as noted in [7], it rarely yields substantial retrieval gains while significantly increasing computational cost, leading many studies to treat it as a baseline [10].

Recent work addresses these limitations through various strategies. LumberChunker [11] uses LLMs to detect natural transition points, outperforming fixed-size chunking. The Chunking-Free Retrieval approach [12] eliminates chunks entirely: its CFIC model encodes full documents and extracts coherent spans via constrained decoding, achieving higher F1 scores on LongBench QA. Mixture of Chunkers (MoC) [10] employs a routing mechanism selecting among multiple meta-chunkers, introducing intrinsic metrics like "Boundary Clarity and Chunk Stickiness" to evaluate chunk quality beyond downstream accuracy.

Despite these advances, effective chunking requires syntactic awareness, yet LLMs lack deep syntactic competence. Probing studies [15] show that pre-trained transformers capture only partial

 doi:10.20944/preprints202601.1757.v1

syntactic structure, with inconsistency across layers. Larger models may exhibit catastrophic forgetting [16], and attention patterns frequently diverge from expected hierarchical structures [17].

There are multiple strategies to compensate for this limitation. Syntax-BERT [18] integrates syntax trees into BERT, RoBERTa, and T5, improving NLU benchmarks. Other work shows that syntactic rather than semantic similarity in few-shot selection enhances Automatic Term Extraction [19], and encoding universal dependency structures improves cross-lingual transfer [20]. Syntactic re-ranking using Tree Kernels has outperformed cosine similarity and BM25 [21].

However, these techniques are rarely deployed in RAG settings and often require heavy models or fine-tuning. Our knowledge and context compression approach aims to enhance syntactic sensitivity without costly training, integrating a syntactic re-ranker to reduce latency while preserving scalability. This study investigates whether generating high-quality questions can indirectly enhance RAG retrieval capabilities.

3. Research Objectives

The present research is motivated by the need to improve large language models (LLMs) in their ability to facilitate scientific discovery. Retrieval-Augmented Generation (RAG) systems offer significant value to the scientific community by providing factual information with reduced hallucinations. However, they still face substantial challenges when navigating large volumes of complex literature while maintaining reliable semantic retrieval. This limitation is critical in automated scientific discovery, where models must connect multiple research findings and identify genuine knowledge gaps that researchers can investigate.

In this context, the current work aims to establish the foundations for a series of studies focused on developing a method to address this challenge. Our research first focuses on designing a universal knowledge compression mechanism that simultaneously compresses scientific information and preserves its contextual meaning. This is achieved by generating questions derived from research experiment papers.

To strengthen this compression-based representation and enhance retrieval quality and paliate to semantical performances of models, we introduce a syntactic reranker, which demonstrates substantial improvements in retrieval performance. Finally, we conduct experiments using our full methodology to produce a new format of scientific abstract called paper-cards, a structured representation intended to better support downstream retrieval. Our results show that paper-cards significantly improve retrieval performance for both LLM-based retrieval systems and traditional methods such as BM25.

Overall, our findings demonstrate that model retrieval can be substantially enhanced by providing stronger syntactic structure and explicit contextualized queries. By replacing document chunking and finetuning with question-based representations, we supply models with a unique form of context that is generally missing at scale. Although our current experiments focus on NLP research papers, the approach exhibits universal characteristics and suggests a promising path toward reducing hallucinations and improving retrieval reliability across scientific domains.

4. Material and Research Methodology

The research experiments were organized into two main tasks to evaluate the proposed approaches: single-hop and multi-hop document retrieval. The latter poses a greater challenge, as it requires handling both long-range dependencies and the preservation of semantic context.

4.1. Single-Hop Document Retrieval

The objective of this task is to retrieve a single document in a response to a user query. Within the scope of this study, the system must retrieve relevant paper IDs corresponding to the input query, without requiring any fine-tuning of the selected model. Among the retrieved IDs, only one should be an exact match to the input query.

 doi:10.20944/preprints202601.1757.v1

The task can be formally defined as shown in Equation (1):

$$\text{Input Query} \rightarrow [Topk\_ids] \tag{1}$$

Critical Constraint: Only one ID represents a relevant or exact match within the returned set. This restriction ensures that the model functions purely as a retrieval system, returning paper IDs without generating or summarizing textual content.

4.2. Multihop Document Retrieval

This task, which is in line with the LongBench QA benchmark [23], asks the system to use reasoning over a maximum of four passages in a single document to respond to a generally straightforward input query. In other words, it constitutes a four-hop retrieval task as shown in Equation (2):

$$\text{Input Query} \rightarrow [\text{Top-}k\text{\_answers}] \tag{2}$$

4.3. Proposed Approach

The methodology consists of two main components. The first component focuses on the proposed knowledge compression technique and the second component focuses on the retrieval technique developed to improve RAG systems.

4.3.1. Knowledge and Context Compression Pipeline

The knowledge compression pipeline begins by extracting key sections of scientific papers that address the research hypotheses. From the sections, both conceptual referred as broader queries containing few or not technical words and technical queries along with corresponding queries are generated. These elements serve as compressed knowledge units, enabling the RAG system to retrieve concise, high-relevance information while maintaining vector database scalability and maximizing online inference speed Figure 1.

4.3.2. Retrieval Process

The retrieval part is constructed mainly on two main aspects: A lexical filtering and a syntactical reranking followed by a semantical matching using an embedding model.

The syntactic reranking used to boost the semantic retrieval of models in the second evaluation is described in the following section.

4.3.3. Syntactic Re-Ranking

The syntactic re-ranking represents the 3rd step of the retrieval, the syntactic re-ranking algorithm implements a passage filtering and ranking system based on recursive syntactic splitting that maintains semantic coherence through order preservation. The algorithm operates through the following steps:

1. **Recursive Syntactic Splitting:** The query goes through recursive splitting at syntactic boundaries defined by part-of-speech tags (ADP, CCONJ) and specific punctuation marks (colons). This process continues iteratively until no further split points are found, decomposing the query into minimal semantic units while preserving content-bearing segments.
   It can be defined mathematically by defining $\mathcal{P} = \{\text{ADP}, \text{CCONJ}\}$ and $\mathcal{Z} = \{":"\}$ be the split trigger sets. For a text segment s, we define the split points as shown in Equation (3):

$$\text{SplitPoints}(s) = \left\{ i \;\middle|\; \text{token}_i \in s \wedge \left(\text{POS}(\text{token}_i) \in \mathcal{P} \vee \text{token}_i \in \mathcal{Z}\right) \right\}. \tag{3}$$

   The recursive splitting process produces query segments as shown in Equation (4):

$$\mathcal{S}_q = \text{RecursiveSplit}(q, \mathcal{P}, \mathcal{Z}). \tag{4}$$

 doi:10.20944/preprints202601.1757.v1

2. **Document Frequency Calculation:** For each word w in each query segment $s \in \mathcal{S}_q$, calculate its document frequency (number of passages containing the word) as shown in Equation (5):

$$\mathrm{df}(w) = \left| \{ j \mid w \in p_j \} \right|. \tag{5}$$

3. **Word Frequency Counting:** For each word w in the original query, count its raw frequency in every passage p_j as shown in Equation (6):

$$c_{w,j} = \mathrm{Count}(w, p_j). \tag{6}$$

4. **Top-L Frequency Value Selection:** For each word w in each query segment $s \in \mathcal{S}_q$, identify the L highest frequency values across all passages. Let $\mathrm{TopFreqs}_L(w)$ be the set of the L highest unique frequency values for word w. The valid passage indices[1] for word w are as shown in Equation (7):

$$\mathcal{I}_w^L = \{ j \mid c_{w,j} \in \mathrm{TopFreqs}_L(w) \wedge c_{w,j} > 0 \}. \tag{7}$$

5. **Phrase-Level Aggregation:** For each query segment (phrase) s, collect all passage indices that appear in the top-L for any word in that phrase as shown in Equation (8):

$$\mathcal{I}_s^L = \bigcup_{w \in s} \mathcal{I}_w^L. \tag{8}$$

6. **Reverse-Order Union:** Process query segments in reverse order and take the union of all valid indices as shown in Equation (9):

$$\mathcal{P}_{\text{valid}} = \bigcup_{s \in \mathrm{reverse}(\mathcal{S}_q)} \mathcal{I}_s^L. \tag{9}$$

7. **Order Preservation:** Rather than ranking by frequency scores, selected passages are returned according to their original document order. This maintains the semantic and logical flow of the document collection, which is crucial for coherent information retrieval as shown in Equation (10):

The complete reranking function is therefore:

$$\textsc{SyntacticRerank}(q, L) = \mathrm{sort}(\mathcal{P}_{\text{valid}}). \tag{10}$$

Where the sorting preserves the original document order of the selected passages.

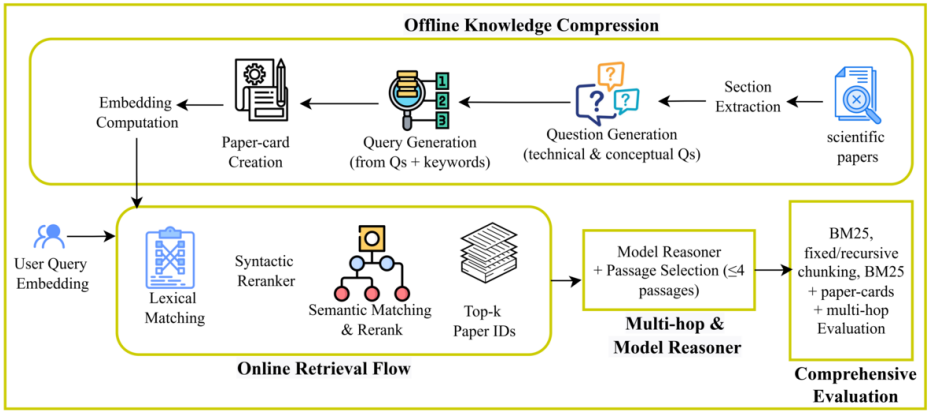


Figure 1. Knowledge Compression and Retrieval Workflow.

[1] The indices help in tracking the correct passages for good ordering.

 doi:10.20944/preprints202601.1757.v1

4.4. Dataset

The first experiment focuses on constructing and evaluating the proposed knowledge and context compression method through question–query generation. This task is framed as a document-retrieval problem based on an input query, using a dataset of 109 scientific papers. These papers were randomly sampled from the ArXiv platform within the Computer Science domain, specifically the Natural Language Processing category. Although the dataset contains 109 papers, chunking results in 30k–44k text segments, which aligns with standard RAG preprocessing pipelines which is adequate for the purposes of this experiment: it enables reliable evaluation of the question-generation process while allowing rapid iteration during method development. All retrieval techniques are evaluated on this common dataset as shown in Table 1.

To conduct the second task (Multi-hop retrieval), we selected the Long-BenchQA v1 dataset, specifically 2WikiMultihopQA and 2WikiMultihopQA_e subsets, which contain 200 and 300 passages rows resulting in 2000 to 3000 documents for testing, respectively. Notably, the 2WikiMultihopQA_e subset includes an additional 100 documents.

On average, each document row contains approximately 6,146 words in the 2WikiMultihopQA_e subset and 4,887 words in the 2WikiMultihopQA subset.

Table 1. Number of scientific papers used for each retrieval technique.

Technique	Number of Papers
Fixed-sized Chunking	109
Recursive Chunking	109
BM25	109

To conduct the second task (Multi-hop retrieval), we selected the LongBenchQA v1 dataset, specifically `2WikiMultihopQA` and `2WikiMultihopQA_e` subsets, which contain 200 and 300[2] document rows for testing, respectively. Notably, the `2WikiMultihopQA_e` subset includes an additional 100 documents [22].

Table 2. Data distribution for the model evaluation on the datasets.

Models	Samples of both datasets
Llama2-7B-chat-4k	200–200
Vicuna-7B	200–200
Llama3-8B-8k	200–200

On average, each document row contains approximately 6,146 words in the 2WikiMultihopQA_e subset and 4,887 words in the 2WikiMultihopQA subset.

5. Experimental Setup

This section outlines the evaluation protocol, models, and task configurations we conducted.

5.1. Evaluation Methodology

Each query has only one correct document ID among the top-k results, then traditional F1-score is unsuitable. We adopt Accuracy at top k (or equivalently, Recall at top k) as our main metric.

5.1.1. Metric Rationale and Evaluation Scope

The retrieval evaluation metrics selected reflect both relevance and ranking. A single-answer precision remains constant at $1/k$ and provides limited insights, so we focus on Accuracy at top k and Recall at top k, which measure whether the correct document appears within the top-k results, offering

[2] For the current study, the first 200 documents were selected for the 2WikiMultihopQA_e

 doi:10.20944/preprints202601.1757.v1

a clear and interpretable success measure. The Mean Reciprocal Rank (MRR) is also reported, which rewards higher ranks for correct documents and complements Recall at top k.

The evaluation is conducted in two stages: first, a comparison of the question-based retrieval with structure-based retrieval, fixed-size chunking, and BM25 is done, as well as comparing paper-cards to traditional abstracts in document-level retrieval, then second, we benchmark the approach on LongBenchQA v1 datasets to assess performance relative to state-of-the-art methods.

5.2. Models

We used `Llama 3.2 3B Instruct`[3] [27] to develop and compare against baseline retrieval strategies (BM25, recursive, and fixed-size chunking).

For model benchmarking, three additional LLMs were employed which are Llama2-7Bchat-4k [25], Vicuna-7B (16K context) [26], and Llama3-8B (8K context) [24].

5.3. Evaluation Tasks

We evaluate our approach through single-hop and multi-hop retrieval. For single-hop, we compare question-centric methods against fixed-size, recursive, and BM25-based retrieval using V2 (technical) and V3 (conceptual) query sets, testing traditional abstracts versus question-centric paper-cards. For multi-hop, following LongBenchQA v1, we use F1-score and test our syntactic reranker with $L \in \{2,3\}$ (equation 7). All experiments use Llama3.2 3B Instruct with 6000-token context and temperature 0.5.

6. Results

This section presents the evaluation of our retrieval approaches across single-hop and multi-hop tasks, highlighting comparative performances and key insights.

6.1. Task 1: Single-Hop Retrieval

We first assess single-hop document retrieval, analyzing accuracy and ranking metrics to benchmark our proposed approach against established baselines.

6.1.1. Global Performances

Table 3 presents the average performances of each traditional techniques against the question-centric approach developped.

Table 3. Average performance metrics across retrieval approaches.

Approach	Acc.	MRR
Our Approach	**0.844**	**0.803**
Recursive Chunking	0.256	0.217
Fixed-size Chunking	0.231	0.198
BM25	0.789	0.677

Furthermore, the evaluations have been conducted on two dimensions (technical and conceptual queries) to evaluate the performances of the techniques at a granular level based on the nature of the queries as shown in Tables 4 and 5).

[3] A fine-tuned model developed by Tethys Research for educational purposes; excluded from the second evaluation for transparency.

 doi:10.20944/preprints202601.1757.v1

Table 4. Accuracy and MRR for technical (V2) queries at Top-3 and Top-5 retrieval.

Method	Top-3		Top-5	
	Accuracy	MRR	Accuracy	MRR
Approach	0.880	**0.857**	0.917	**0.866**
Recursive	0.165	0.148	0.183	0.152
Chunk	0.146	0.131	0.165	0.135
BM25	**0.899**	0.848	**0.935**	0.856

Table 5. Accuracy and MRR for conceptual (V3) queries at Top-3 and Top-5 retrieval.

Conceptual	at 3		at 5	
	Acc.	MRR	Acc.	MRR
Approach	**0.770**	**0.740**	**0.807**	**0.748**
Recursive	0.321	0.279	0.357	0.288
Chunk	0.293	0.259	0.321	0.266
BM25	0.605	0.489	0.715	0.513

6.1.2. BM25 Boosted

We evaluate the performances of BM25 on both abstract papers and the paper-card generated from our approach. The results can be seen in Tables 6 and 7.

Figure 3 presents a synthesis of the performances of BM25 on both techniques.

Table 6. Accuracy and MRR for BM25 technical queries (v2) at Top-3 and Top-5 retrieval.

Technical	at 3		at 5	
	Acc.	MRR	Acc.	MRR
BM25 (Card)	**0.963**	**0.937**	**0.972**	**0.939**
BM25 (Abs)	0.899	0.845	0.935	0.854

Table 7. Accuracy and MRR for BM25 conceptual queries (v3) at Top-3 and Top-5 retrieval.

Conceptual	at 3		at 5	
	Acc.	MRR	Acc.	MRR
BM25 (Card)	**0.889**	**0.850**	**0.917**	**0.855**
BM25 (Abs)	0.660	0.562	0.715	0.575

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 22 January 2026 doi:10.20944/preprints202601.1757.v1

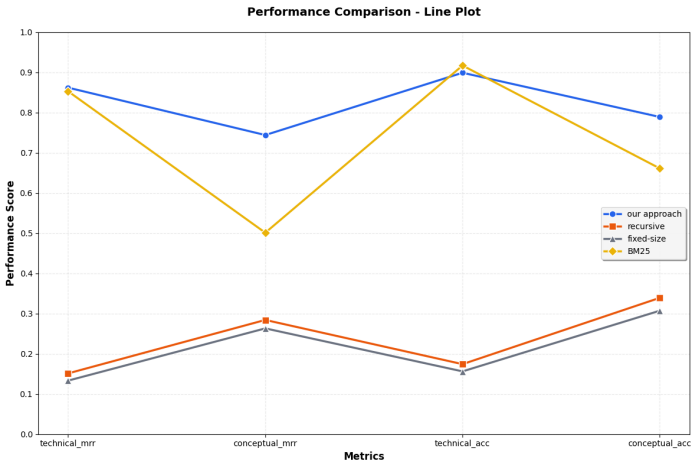


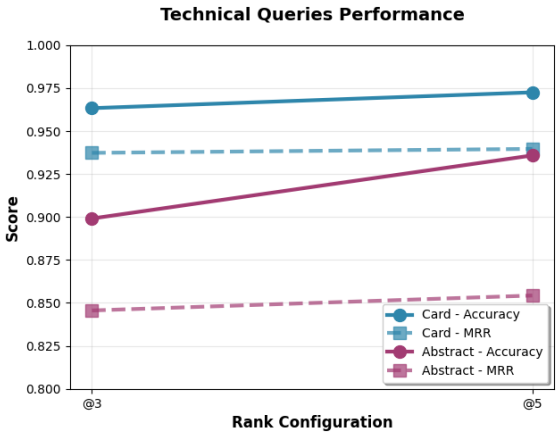
Figure 2. Global perfomances of the techniques.

6.2. Task 2: Multi-Hop Retrieval

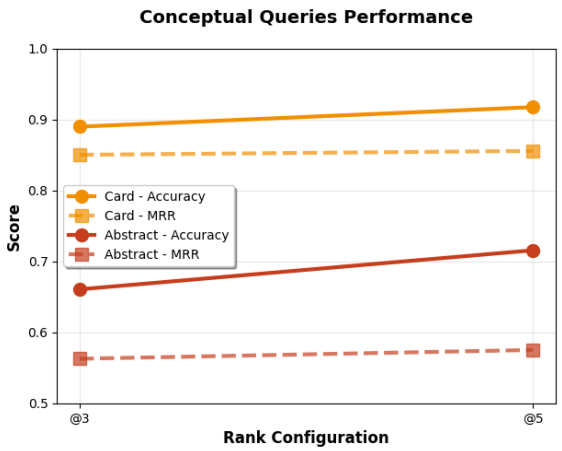
Table 2 presents the evaluation on the second task concerning the multihop retrieval performances of the models on the selected benchmark. The first step is to study the direct performances of the models on the `2WikiMultihopQA` dataset and compare the results against the other implementations mostly using finetuning techniques.

Table 8. Performance comparison across models.

Models	F1
Llama2-7B-chat-4k (ours)	**0.520**
CFIC-7B	0.412
Llama2-7B-chat-4k	0.328
Vicuna-7B (ours)	**0.340**
Vicuna-7B (CFIC)	0.233
Llama3 (ours)	**0.550**
GPT-3.5-Turbo-16k	0.377
Llama_index	0.117
PPL chunking	0.141



BM25 performance on technical queries (v2).



BM25 performance on conceptual queries (v3).

Figure 3. Comparison of BM25 performance across technical (v2) and conceptual (v3) queries.

 doi:10.20944/preprints202601.1757.v1

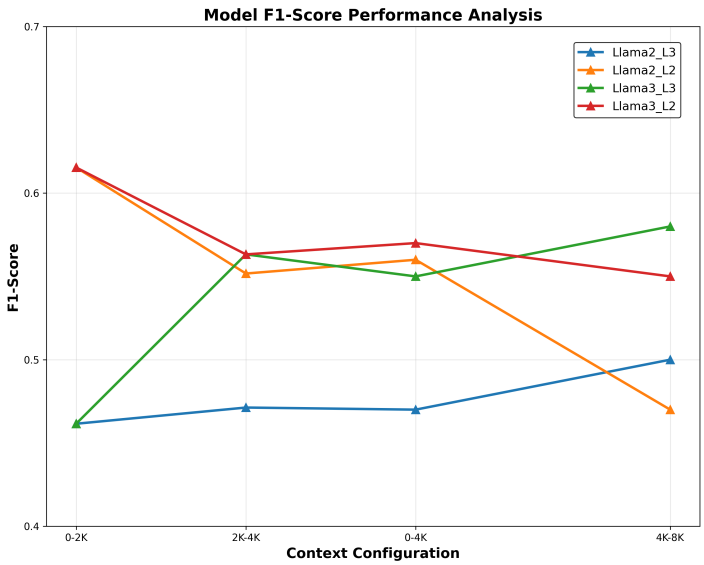


Figure 4. Perfomances on the 2WikiMultihopQA_e. LLama3 and LLama2 correspond to the models using our approach with some specificities on the parameter L which can be either 3 or 2.

Then we evaluated `Llama3` and `Llama2-7B-chat-` on the `2WikiMultihopQA_e` to evaluate at the granular level the performances of the models from 0k-4k to 4k-8k context windows using the syntactic parameter L.

Table 9. F1-Score performance across context ranges on 2WikiMultihopQA_e using parameter L at 2 or 3.

Models	0-4K	4K-8K
Llama2-7B-chat-4k_L3 (ours)	**0.470**	**0.500**
Llama2-7B-chat-4k_L2 (ours)	0.560	0.470
Llama3_L3 (ours)	0.550	**0.580**
Llama3_L2 (ours)	0.550	**0.580**
GPT-3.5-Turbo-16k	0.498	0.451
Llama2-7B-chat-4k	0.333	0.225

7. Discussions

This work set out to examine whether question and query-anchored representations can enhance retrieval effectiveness in document-centric RAG systems. Across both single-hop and multi-hop tasks, our proposed approach, built on generated question anchors and a syntactic reranker demonstrates consistent improvements over standard chunking and semantic retrieval pipelines using finetuning. In this section, we interpret these findings, discuss practical and theoretical implications, evaluate limitations, and outline directions for future work.

7.1. Task 1: Single-Hop retrieval

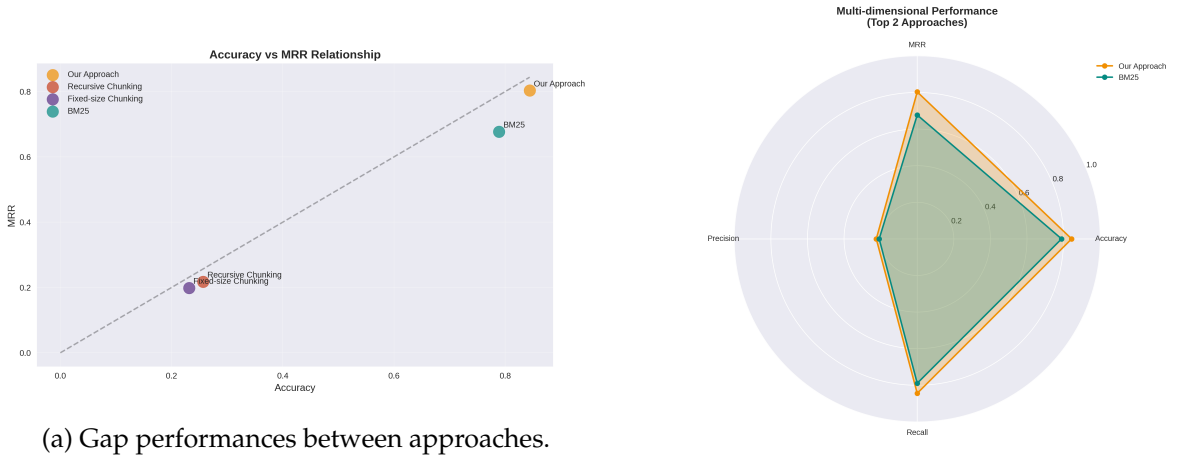
The first evaluation shows that the question-driven approach consistently outperforms standard chunking on scientific NLP papers. As seen in Table 3, it achieves the strongest overall results: BM25 yields with 0.019 higher Accuracy on highly technical queries, while our method attains superior MRR (shown in Figure 2)

Figure 2 highlights this stability: BM25 MRR declines from 84% to 48% between technical and conceptual queries, while our approach falls by only 12%. Fixed-size and recursive chunking remain below 40% Accuracy overall, with only slight gains on conceptual queries.

A BM25 analysis (Figure 3) further shows that incorporating our generated paper-cards boosts and stabilizes retrieval compared to traditional abstracts. Both MRR and Accuracy increase consistently as

 doi:10.20944/preprints202601.1757.v1

k grows from 3 to 5 (Figure 5). The approach also brings substantial storage and computation benefits: for 109 papers, only 218 embedding vectors are stored, versus 38,509 recursive and 44,809 fixed-size chunks (Table 10). Each Markdown paper-card averages under 5 KB, with a maximum of 3.1 KB, reducing storage by roughly 80%.



(a) Gap performances between approaches.

(b) BM25 vs. question-based retrieval.

Figure 5. Comparative performance on Accuracy, MRR, and Recall.

Table 10. Storage efficiency comparison.

Method	Records
Recursive Chunking	38,509
Fixed-size Chunking (350 words)	44,809
Paper-cards (ours)	**109**
Main research queries (ours)	**95**

7.2. Task 2: Multi-Hop Retrieval

In multi-hop settings, the syntactic reranker further improves retrieval quality. As shown in Table 8, applying our syntactic reranker improves F1 scores across models, particularly with $L = 3$ (as shown in Table 9). Figure 4 shows that performance grows with context length under $L = 3$, while $L = 2$ peaks early and declines as longer passages introduce textual noise.

Models equipped with our method show consistent improvement: `Llama3` and `Llama2-7B-Chat-4k` differ by only 3% in F1, yet both outperform their original configurations by over 20%. These results confirm that our syntactic reranker enhances contextual understanding rather than relying on fine-tuning. Table 11 is sharing a complete performance analysis of the current experiment against current state-of-the-art approaches.

Table 11. Performance Analysis of the Approaches

Method / Paper Group	Core Idea	Strengths	Limitations	Example Works
Fixed-size Chunking	Split documents into equal-length segments (often with overlap).	Simple, robust, low computational cost; competitive performance when tuned.	Ignores document structure; can break semantic units; limited adaptability.	[7]: 100-word passages for early RAG; [8,9]: 200-word chunks performing on par with semantic chunking.

Continued on next page

 doi:10.20944/preprints202601.1757.v1

Table 11 – *Continued from previous page*

Method / Paper Group	**Core Idea**	**Strengths**	**Limitations**	**Example Works**
Structure-based Chunking	Split documents using natural hierarchy (HTML, headings, recursion, RL-based segmentation).	More coherent chunks; improved QA scores (e.g., 84% on FinanceBench; F1 81.8/62.0 on CoQA/QuAC).	Fragile with messy real-world documents, inconsistent formatting, noisy HTML; limited web-scale reliability.	[5], [14]: RL recurrent chunker; [16]: Meta-chunking with Qwen2/Baichuan2.
Semantic Chunking	Group text using embedding-based semantic similarity.	Conceptually adaptive; captures meaning across spans.	High computational cost; often minimal retrieval gains; embedding drift.	[10].
Improved Semantic / Chunk-Free Approaches	Use LLMs or full-text encoding to find natural transitions or remove chunking entirely.	Better retrieval accuracy; coherent boundaries; avoids chunk fragmentation.	Much higher computational cost; not always scalable.	LumberChunker [11]; Chunking-Free Retrieval [12]; Mixture of Chunkers (MoC) [10].
RGV Framework	Retrieve, generate and verify answers to reduce hallucinations.	Improves factual coherence and QA reliability.	Complex multi-step pipeline reduces scalability; does not address chunking or representation.	[35].
Syntactic Competence Limitations in LLMs	Analysis of LLM syntactic knowledge and probing studies.	Reveals partial syntactic awareness that inspires new chunking methods.	Transformers capture syntax inconsistently; cross-layer misalignment; catastrophic forgetting.	Probing studies: [16–18].
Syntax-aware Methods	Inject or leverage syntactic structure (UD trees, tree kernels, syntax-enhanced encoders).	Improves reranking, NLU tasks, cross-lingual transfer, few-shot selection.	Requires tree extraction; sensitive to parser errors; high preprocessing cost.	Syntax-BERT [19]; UD-based transfer [21]; syntactic reranking [22]; ATE selection [20].
Decomposition-Reflection (DR)	Break complex questions into sub-questions and refine via reflection.	Stronger long-form reasoning; improved multi-paragraph answer quality.	Closed-book; no retrieval, no chunking, no semantic compression; not useful for RAG.	[34].
Our Proposed Method	Compress documents into query-aligned representations (paper-cards) for precise retrieval and minimal storage.	Reduces chunking errors; boosts retrieval (MRR drop only 12% vs 50% for BM25); highly storage-efficient; scalable.	Requires controlled generation pipeline; depends on compression quality.	*Current work*.

 doi:10.20944/preprints202601.1757.v1

7.3. Limitations

During experiments, several limitations were observed. Models struggled to generate concise, high-quality questions for long contexts, often producing verbose keywords and shallow multi-hop queries over passages exceeding 10,000 words. The syntactic reranker improved relevance, but current models lack sufficient syntactic understanding to effectively decompose complex queries. Generated questions also lack structural organization, suggesting that graph-based representations could improve retrieval at scale.

The absence of fine-tuning further constrained performance, as syntactic fine-tuning may yield greater gains than purely semantic approaches. Additionally, the reranker relies on a fixed L parameter, and adapting it dynamically could enhance results. Future work will address these issues and expand evaluation to benchmarks such as LongBench QA v2 [32], Qasper [31], FinanceBenchQA [30], NFcorpus [28], and Loong [29].

8. Conclusions

This study demonstrates that query-oriented question generation, viewed as knowledge compression, improves RAG retrieval while preserving semantics. The method outperformed chunking and BM25, with MRR dropping only 12% on complex queries (vs. $\sim$50% for BM25), and showed further gains when combined with BM25 over paper-cards. Storage efficiency was significant: 109 papers required only 218 embeddings, and paper-cards ($\leq$5 KB) enabled near-instant retrieval. Syntactic reranking improved multi-hop retrieval, yielding up to 20% F1 gains on LongBench QA v1, with strongest results on 2WikiMultihopQA and 2WikiMultihopQA_e.

Limitations include the need for broader benchmark evaluation and richer semantic metrics (e.g., BLEU) to assess retrieval and generation quality. Future work will explore graph-based representations of generated queries to enhance interpretability and performance in complex retrieval scenarios.

Author Contributions: All authors contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The work was done with partial support from the Mexican Government through the grant A1- S47854 of CONACYT, Mexico, grants 20250738, of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of Unnumbered acknowledgements section if required. the Laboratorio de Supercómputo of the INAOE, Mexico, and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

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

References

1. Arslan, M.; Ghanem, H.; Munawar, S.; Cruz, C. A Survey on RAG with LLMs. *Procedia Computer Science* **2024**, *246*, 3781–3790.
2. Arslan, M.; Munawar, S.; Cruz, C. Business insights using RAG–LLMs: a review and case study. *Journal of Decision Systems* **2024**, 1–30.
3. Zhu, Z.; Sun, Z.; Yang, Y. HaluEval-Wild: Evaluating Hallucinations of Language Models in the Wild. *ArXiv* **2024**, *abs/2403.04307*.
4. Guțu, B. M.; Popescu, N. Exploring data analysis methods in generative models: from Fine-Tuning to RAG implementation. *Computers* **2024**, *13* (12), 327.
5. Xu, S.; Pang, L.; Shen, H.; Cheng, X. BERM: Training the Balanced and Extractable Representation for Matching to Improve Generalization Ability of Dense Retrieval. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Long Papers), 2023; pp 6620–6635.
6. Jimeno-Yepes, A.; You, Y.; Milczek, J.; Laverde, S.; Li, R. Financial Report Chunking for Effective Retrieval Augmented Generation. *ArXiv* **2024**, *abs/2402.05131*.

 doi:10.20944/preprints202601.1757.v1

7. Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Kuttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; Riedel, S.; Kiela, D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *ArXiv* **2020**, *abs/2005.11401*.
8. Qu, R.; Tu, R.; Bao, F. S. Is Semantic Chunking Worth the Computational Cost? In Findings of the ACL: NAACL 2025, 2025; pp 2155–2177.
9. Merola, C.; Singh, J. Reconstructing Context: Evaluating Advanced Chunking Strategies for Retrieval-Augmented Generation. In Proceedings of ACL 2025, 2025.
10. Zhao, J.; Ji, Z.; Fan, J. Z.; Wang, H.; Niu, S.; Tang, B.; Xiong, F.; Li, Z. MoC: Mixtures of Text Chunking Learners for Retrieval-Augmented Generation System. *ArXiv* **2025**, *abs/2503.09600*.
11. Duarte, A. V.; Marques, J.; Graça, M.; Freire, M. F.; Li, L.; Oliveira, A. L. LumberChunker: Long-Form Narrative Document Segmentation. *ArXiv* **2024**, *abs/2406.17526*.
12. Qian, H.; Liu, Z.; Mao, K.; Zhou, Y.; Dou, Z. Grounding Language Model with Chunking-Free In-Context Retrieval. In Proceedings of the 62nd Annual Meeting of the ACL (Long Papers), 2024; pp 1298–1311.
13. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv* **2018**, *abs/1810.04805*.
14. Goldberg, Y. Assessing BERT's syntactic abilities. *ArXiv* **2019**, *abs/1901.05287*.
15. Liu, N. F.; Gardner, M.; Belinkov, Y.; Peters, M. E.; Smith, N. A. Linguistic knowledge and transferability of contextual representations. *ArXiv* **2019**, *abs/1903.08855*.
16. Iwamoto, R.; Yoshida, I.; Kanayama, H.; Ohko, T.; Muraoka, M. Incorporating syntactic knowledge into pre-trained language model using optimization for overcoming catastrophic forgetting. In Findings of EMNLP 2023, 2023; pp 10981–10993.
17. Wang, X.; Gao, T.; Zhu, Z.; Zhang, Z.; Liu, Z.; Li, J.; Tang, J. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *TACL* **2021**, *9*, 176–194.
18. Bai, J.; Wang, Y.; Chen, Y. Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees. In Proceedings of EACL 2021; pp 3011–3020.
19. Chun, Y.; Kim, M.; Kim, D.; Park, C.; Lim, H. Enhancing Automatic Term Extraction with Large Language Models via Syntactic Retrieval. In Findings of ACL 2025; pp 9916–9926.
20. Ahmad, W.; Li, H.; Chang, K.-W.; Mehdad, Y. Syntax-augmented Multilingual BERT for Cross-lingual Transfer. In Proceedings of ACL AND IJCNLP 2021; pp 4538–4554.
21. Arif, R.; Bashir, M. Question Answer Re-Ranking using Syntactic Relationship. In Proceedings of ICOSST 2021; pp 1–6.
22. Bai, Y.; Lv, X.; Zhang, J.; Lyu, H.; Tang, J.; Huang, Z.; Du, Z.; Liu, X.; Zeng, A.; Hou, L.; Dong, Y.; Tang, J.; Li, J. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. *ArXiv* **2023**, *abs/2308.14508*.
23. Bai, Y.; Lv, X.; Zhang, J.; Lyu, H.; Tang, J.; Huang, Z.; Du, Z.; Liu, X.; Zeng, A.; Hou, L.; Dong, Y.; Tang, J.; Li, J. LongBench: A Bilingual, Multitask Benchmark for Long Context Understanding. In Proceedings of the 62nd Annual Meeting of ACL, 2024; pp 3119–3137.
24. Dubey, A.; Others. The Llama 3 Herd of Models. *ArXiv* **2024**, *abs/2407.21783*.
25. Touvron, H.; Others. Llama 2: Open Foundation and Fine-Tuned Chat Models. *ArXiv* **2023**, *abs/2307.09288*.
26. Zheng, L.; Others. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *ArXiv* **2023**, *abs/2306.05685*.
27. Meta Team. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. Available online: https://tinyurl.com/p9y46bv6 (accessed on 4 November 2025).
28. Boteva, V.; Gholipour, D.; Sokolov, A.; Riezler, S. A Full-Text Learning to Rank Dataset for Medical Information Retrieval. In Proceedings of ECIR 2016.
29. Wang, M.; Others. Leave No Document Behind: Benchmarking Long-Context LLMs with Extended Multi-Doc QA. In Proceedings of EMNLP 2024; pp 5627–5646.
30. Islam, P.; Others. FinanceBench: A New Benchmark for Financial Question Answering. *ArXiv* **2023**, *abs/2311.11944*.
31. Dasigi, P.; Others. A Dataset of Information-Seeking Questions and Answers Anchored in Research Papers. *ArXiv* **2021**, *abs/2105.03011*.
32. Bai, Y.; Others. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. *ArXiv* **2024**, *abs/2412.15204*.
33. Rackauckas, Z. Rag-Fusion: A New Take on Retrieval-Augmented Generation. *ArXiv* **2024**, *abs/2402.03367*.
34. Xiao, J., Wu, W., Zhao, J., Fang, M. Wang, J. Enhancing long-form question answering via reflection with question decomposition. *Information Processing & Management*. **62**, 104274 (2025), https://www.sciencedirect.com/science/article/pii/S0306457325002158

 doi:10.20944/preprints202601.1757.v1

35. Sun, S., Zhang, K., Li, J., Yu, M., Hou, K., Wang, Y. & Cheng, X. Retriever-generator-verification: A novel approach to enhancing factual coherence in open-domain question answering. *Information Processing & Management*. **62**, 104147 (2025), https://www.sciencedirect.com/science/article/pii/S0306457325000883
36. S. Rezaei, A. Frisch, F. Yin, and D. Friedman. Vendi-RAG: Balancing Diversity and Accuracy in Retrieval-Augmented Generation. *arXiv preprint arXiv:2502.11228*, 2025.
37. Z. Gao, Y. Zhang, S. Zhang, L. Wang, and D. Li. Dr3: A Multi-Stage Discriminate–Recompose–Resolve Framework for Reducing Off-Topic Responses. *arXiv preprint arXiv:2403.12393*, 2024.
38. Y. Ji, Z. Huang, R. Cen, and Q. Zhang. DEC: A Resource-Friendly Dynamic Enhancement Chain for Multi-Hop Question Answering. *arXiv preprint arXiv:2506.17692*, 2025.
39. O. Press, L. Cheng, M. Liu, and N. Aharoni. Retrieve–Summarize–Plan: An Iterative Framework for Multi-Hop Reasoning. *arXiv preprint arXiv:2407.13101*, 2024.
40. Y. Cao, R. Zhou, J. Chen, and S. Zhao. Data Augmentation for Real-World Multi-Hop Reasoning via Synthetic Knowledge Graph Expansion. *arXiv preprint arXiv:2504.20752*, 2025.
41. X. Zhang, S. Garg, K. Lin, and D. Jurafsky. GenDec: Robust Generative Multi-Hop Question Decomposition. *arXiv preprint arXiv:2402.11166*, 2024.
42. T. Wang, H. Liu, Y. He, and M. Sun. Lost-in-Retrieval: Diagnosing and Mitigating Retrieval Failures in Multi-Hop QA. *arXiv preprint arXiv:2502.14245*, 2025.