Towards Multi-Document Question Answering in Scientific Literature: Pipeline, Dataset, and Evaluation

Hui HUANG[1,2], Julien Velcin[3], Yacine Kessaci [2]
[1]Université de Lyon, Lyon 2, ERIC UR 3083, France
[2]Worldline S.A., France
[3]École Centrale de Lyon, LIRIS CNRS UMR 5205, France
hui.huang@univ-lyon2.fr, julien.velcin@ec-lyon.fr
yacine.kessaci@worldline.com

Abstract

Question-Answering (QA) systems are vital for rapidly accessing and comprehending information in academic literature. However, some academic questions require synthesizing information across multiple documents. While several prior resources consider multi-document QA, they often do not strictly enforce cross-document synthesis or exploit the explicit inter-paper structure that links sources. To address this, we introduce a pipeline methodology for constructing a **M**ulti-**D**ocument **A**cademic **QA** (MDA-QA) dataset. By both detecting communities based on citation networks and leveraging Large Language Models (LLMs), we were able to form thematically coherent communities and generate QA pairs related to multi-document content automatically. We further develop an automated filtering mechanism to ensure multi-document dependence. Our resulting dataset consists of 6,804 QA pairs and serves as a benchmark for evaluating multi-document retrieval and QA systems. Our experimental results highlight that standard lexical and embedding-based retrieval methods struggle to locate all relevant documents, indicating a persistent gap in multi-document reasoning. We release our dataset and source code for the community. [1]

1 Introduction

In academic research, question-answering (QA) systems are widely used to help researchers and students quickly access and understand complex document information (Hermann et al., 2015). However, as scientific literature expands, extracting key content efficiently and accurately from a vast library and providing relevant answers has become a significant challenge (Chen et al., 2017).

Most existing QA datasets focus on a single document or use multi-hop reasoning (Kwiatkowski et al., 2019; Yang et al., 2018; Kočiskỳ et al., 2018). Although earlier multi-hop QA tasks involve several documents, they primarily rely on logical or deductive chains of reasoning. In the end, the final answer to the question typically can be found in a single document (Zhu et al., 2024). Beyond single-document and multi-hop settings, several datasets explore multi-document QA (Bolotova-Baranova et al., 2023; Fan et al., 2019; Han et al., 2024; Li et al., 2024). However, these methods do not strictly require synthesizing evidence across sources. (Li et al., 2024) leverages structural sig-

Example 1:
Question: How does point supervision strategy differ between infrared small target detection and crowd counting tasks in terms of network design and performance?
Answer: In infrared target detection, point supervision uses single points per target and evolves them into masks through label evolution, achieving 70% IoU of fully supervised methods. For crowd counting, point supervision uses multiple points (typically 5-20) per image and employs multiscale fusion networks to handle varying crowd densities, reaching 95% performance of fully supervised approaches. The key difference lies in the supervision density - infrared targets require minimal points due to target simplicity, while crowd counting needs denser point annotations to capture crowd distribution variations.
Sources: [2304.01484], [2202.09942]

Example 2:
Question: What are the different approaches to achieving differential privacy in language models and their relative trade-offs?
Answer: There are three main approaches: 1) Local DP that anonymizes each text sample individually but requires high noise leading to incoherent language, 2) Global DP that trains language models privately on the full dataset allowing better utility but needs careful privacy accounting, and 3) Selective DP that protects only sensitive portions allowing better utility-privacy trade-offs. The choice depends on trust assumptions and utility requirements.
Sources: [2210.13918], [2108.12944], [1712.05888]

Table 1: Two representative examples of multi-document QA pairs from MDA-QA. Each pair requires the use of information from two or more articles (cf. Sources). Example 1 shows a comparison of different methods. Example 2 shows synthesizing evidence from multiple sources.

[1] https://github.com/YeloDriver/MDAQA

Dataset	QA Generation	Size	Source	Structure	Multiple Docs
PubMedQA (Jin et al., 2019)	Human Experts	1k	Title/Abstract	×	×
QASPER (Dasigi et al., 2021)	Human Experts	5k	Title/Abstract	×	×
QASA (Lee et al., 2023)	Human Experts	1.8k	Full-Text	×	×
SPIQA (Pramanick et al., 2025)	LLMs + Human experts	6k	Full-Text + Figs & Tabs	×	×
SciDQA† (Singh et al., 2024)	LLMs + Human experts	2.9k	Full-Text	×	(✓)
MDA-QA (Ours)	LLMs + Hybrid Filter	6.8k	Full-Text	✓	✓

Table 2: **Comparison of our MDA-QA with existing scientific QA datasets.** SciDQA†: Only 11% of the QA in SciDQA dataset require multiple documents.

nals by using clusters from a single paper and its citations. This anchor-centric method produces asymmetric, topic-mixed neighborhoods and still does not ensure multi-document requirements. The research questions often scatter across articles because they involve shared or contrasting methodologies, or concepts that overlap but are not identical. Existing QA datasets in scientific literature rarely capture these cross-document relationships or ask questions requiring multiple documents (Jin et al., 2019; Tsatsaronis et al., 2015).

To address this issue, we design a pipeline for generating specialized QA that relies on multiple documents and propose MDA-QA, a corresponding academic QA dataset in multi-document scenarios. The core idea is to automatically generate QA pairs from small groups of structurally connected articles (e.g., through citation links), ensuring that each QA pair draws on multiple sources. Specifically, we propose using community detection on citation graphs to identify groups of articles, then leveraging a large language model (LLM) to create these complex QA pairs. Finally, we design an automated quality control process to ensure that the QA requires content from multiple documents for a complete response. Table 1 shows two examples generated by our approach, illustrating how it can address various question types, such as comparing two methods (see Example 1), synthesizing evidence from multiple sources (see Example 2), and even identifying connections across different scientific domains, thereby fostering the development of potential new ideas.

We use the open-access SPIQA dataset (Pramanick et al., 2025), expanding it with structural information to produce multi-document QA pairs. We conduct a community division on over 25,000 open-access machine learning conference papers and generate 6,804 filtered QA pairs, covering over 3,000 small-scale groups of academic papers. While our QA pairs undergo automatic multi-level filtering, we also perform a structured manual check on 30 randomly sampled questions. Three domain experts review each question, confirming that at least two referenced articles are required. This additional validation step ensures that MDA-QA reflects multi-document needs rather than incidental single-source occurrences.

To further verify the capability of existing QA systems on our constructed multi-document dataset, we establish a multi-document retrieval task and evaluate various retrieval methods, including BM25 (Robertson et al., 2009), ColBERT (Khattab and Zaharia, 2020), BGE (Xiao et al., 2023), and MPNET (Song et al., 2020), in chunk retrieval and document retrieval scenarios. Our experiments (see Table 3) show that BM25 achieves a Recall@10 of only 0.35, while our structure-enhanced dense embedding method, BGE-neighbor, can reach around 0.55, indicating a significant challenge for current mainstream retrieval techniques in multi-document scenarios. These findings reveal the complexity of the dataset in terms of multi-document integration and inference requirements, leaving room for future improvements in multi-document retrieval and QA methods.

The main contributions of this research are threefold:

1. We propose a structure-aware framework that constructs complex multi-document QA pairs by detecting thematically coherent communities in the citation graph and guiding LLMs to synthesize evidence across multiple papers.

2. We construct MDA-QA, a high-quality multi-document academic QA dataset, filling the gap in existing datasets for complex multi-document questions. We release our dataset and source code for future research.

3. Experimental results show that current retrieval methods exhibit substantial gaps on

MDA-QA, offering a new research baseline for subsequent development in multi-document integration and deep reasoning.

2 Related Work

2.1 Datasets for QA on Scientific articles

In recent years, researchers have developed diverse datasets of various scales and forms for QA tasks centered on scientific literature. We provide a concise overview and comparison of current scientific QA datasets in Table 2. Earlier work employed automated techniques to extract named entities and their relationships from documents to generate cloze-style academic paper QA (Rajpurkar et al., 2016; Jin et al., 2019). Subsequently, datasets such as PubmedQA (Jin et al., 2019), BIOASQ-QA (Tsatsaronis et al., 2015; Krithara et al., 2023), and Qasper (Dasigi et al., 2021) gradually extended QA generation to include abstracts and parts of the full text, emphasizing richer semantic information. New datasets, including QASA and SPIQA, have been introduced to push the boundaries of QA tasks further. QASA (Lee et al., 2023) emphasizes understanding the full-text content, creating QA pairs through manually reading complete articles, and covering more document details. PeerQA (Baumgärtner et al., 2025) and ScienceQA(Saikh et al., 2022) also focus on academic QA but mostly remain single-document. SPIQA (Pramanick et al., 2025) introduces multimodal information processing, using Visual Large Language Models (VLLMs) to generate high-quality QA pairs based on figures and corresponding citations in articles. However, these datasets focus on deep exploration within a single document. This constraint restricts broader comparative or synthetic queries that require elements drawn from multiple articles. Similar to our work, a recent proposed dataset SciDQA (Singh et al., 2024) leverages peer-review questions and author-provided answers for a deep understanding of scientific papers. Only 11% of the questions need multiple documents to answer. In contrast, our MDA-QA dataset relies on an automated pipeline, generating multi-document QA at a larger scale and with strictly enforced multi-document requirements.

2.2 Multi-Document QA Datasets

Beyond single-document QA, several datasets explicitly address multi-document reasoning. WikiHowQA (Bolotova-Baranova et al., 2023) provides around 12K how-to questions grounded in 75K passages from WikiHow, focusing on procedural rather than factoid content. ELI5 (Fan et al., 2019) contains 25K open-ended Reddit questions with web evidence, but without enforced multi-document synthesis. LFRQA (Han et al., 2024) emphasizes long-form answers that integrate multiple documents, while Loong (Wang et al., 2024) evaluates long-context LLMs on multi-document QA across domains such as finance, law, and science, with around 11 supporting documents per instance. M3SciQA (Li et al., 2024) constructs 1.5K questions over clusters of an anchor paper and all its cited papers. Nevertheless, none of these resources strictly requires synthesis across multiple documents. In contrast, our MDA-QA leverages citation networks to construct thematically coherent communities, ensuring that each question strictly requires synthesizing evidence across multiple scientific articles.

2.3 Complex QA Challenges

Several studies have addressed textual comprehension methods and knowledge-based reasoning to increase the complexity of questions. From traditional single-document QA datasets like SQuAD (Rajpurkar et al., 2016) and Natural Question (Kwiatkowski et al., 2019) to multi-hop reasoning tasks such as HotpotQA (Yang et al., 2018) or hybrid datasets incorporating structured data like Hybrid QA (Chen et al., 2020b) and OTT-QA (Chen et al., 2020a), there has been a steady movement toward more diverse and complex QA settings. However, these datasets remain confined to single-document scenarios or carefully chosen multi-hop passages, which do not cover the real multi-document scenarios. Our work seeks to accommodate QA scenarios requiring the synthesis of distinct, complementary pieces of information from multiple documents. This differs from these datasets because the final answer is not fully contained in a single source and cannot be obtained simply by “hopping” within one text or among multiple documents.

3 Proposed method

In this section, we present our pipeline for generating QA pairs that draws on multiple sources from the scientific literature. As shown in Figure 1, this process includes three key modules: Community Building (Section 3.1), QA Generation (Sec-

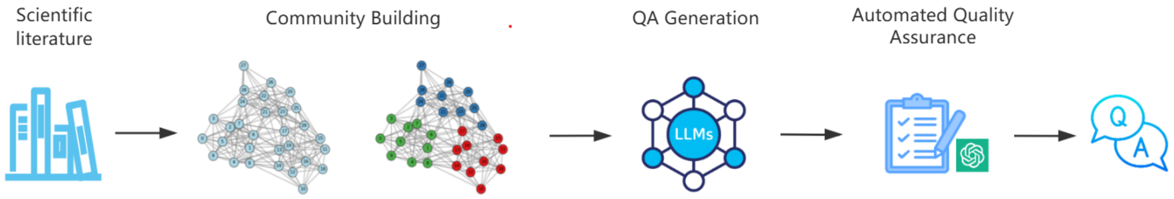


Figure 1: Our proposed process for generating QA pairs from scientific literature.

tion 3.2), and Automated Quality Assurance (Section 3.3).

3.1 Community Building

In scientific literature, articles often have complex and diverse relational connections, such as citations, methodological similarities, thematic similarities, keyword overlaps, or co-author networks (Qiu et al., 2014). We focus on citation links, representing each paper as a node in a directed graph, where each edge indicates a citation from one paper to another. To detect communities, we use the Speaker-Listener Label Propagation (SLPA) algorithm (Kuzmin et al., 2013), which iterates over nodes to exchange labels, allowing nodes to accumulate multiple labels and thus belong to multiple communities.

Our choice of SLPA is based on its capacity to handle overlapping communities and produce relatively small, thematically coherent communities. Preliminary tests with Louvain (Blondel et al., 2008) or Leiden (Traag et al., 2019) that we made resulted in one large component (An et al., 2004), making them less suitable for multi-document QA generation. We decided specifically to focus on smaller communities so that during the QA generation process, the LLM can effectively process and integrate each article's key points, providing more precise and feasible multi-document questions. Each subset of articles exhibits thematic or methodological coherence, serving as a candidate group for generating multi-document QA pairs.

3.2 QA Generation

Once the communities have formed, we employ structured prompting (Schulhoff et al., 2024) with step-by-step instructions (Appendix A presents the prompt for QA generation). Specifically, we provide the LLM with (i) all the context within the community, (ii) instructions to cross-reference at least two documents, and (iii) guidance to generate a single-sentence question and a concise, multi-evidence answer. This prompt encourages the model to examine overlaps across documents systematically. For instance, this prompt may ask the model to compare results reported in different articles, identify conflicting conclusions, or combine an idea from one article with data from another.

Further, to ensure the QA truly requires multiple documents, we integrate an automatic cross-check process, where the LLM compares its multi-document answer with single-document attempts. We only retain the cases that strictly demand more than one article.

3.3 Automated Quality Assurance

Although LLMs can generate complex questions from multiple documents, some answers can still be found in a single document. To address this, we design an automated multi-level quality assurance strategy to ensure that the generated QA pairs require evidence from multiple documents.

As shown in Algorithm 1, for each QA pair (q_i, a_i) and its corresponding support document set S, we individually provide the model with a document $s_j \in S$, allowing the model to answer q_i, and record this answer as a_i^j. We then let the LLM compare a_i and a_i^j to determine which answer is more accurate and complete. Only if a_i is better than a_i^j for all documents in S, the QA pair is retained, eliminating questions answerable by a single document. Appendix B presents the prompts used for the `LLMAnswer` step and `Compare` step. We further conducted a sanity check, complementing Algorithm 1. Three domain experts reviewed 30 randomly sampled QA pairs. The result shows that our automated filter reliably removes single-document cases while retaining questions that require cross-document synthesis. The user interface (UI) and detailed guidelines are presented in Appendix C.2

Our automated quality assurance process removes QA pairs that appear complex but can be solved by a single document in large-scale data

Algorithm 1 QA Filtering

```
Require: A set of QA pairs ${(q_i, a_i)}$; For each $(q_i, a_i)$, a corresponding set of supporting documents $S = \{s_1, s_2, \ldots, s_n\}$.
Ensure: A filtered set of QA requiring multiple documents to answer.
for each QA pair $(q_i, a_i)$ do
    Initialize $valid \leftarrow True$
    for each document $s_j$ in $S$ do
        $a_i^j \leftarrow$ LLMAnswer$(q_i, s_j)$ {Model attempts to answer $q_i$ using only $s_j$}
        $comparison \leftarrow$ Compare$(q_i, a_i, a_i^j, S)$ {Check completeness/accuracy}
        if $comparison \neq$ "$a_i$ is strictly better" then
            $valid \leftarrow False$
            break {If a single document answer is comparable or better, mark invalid}
        end if
    end for
    if $valid$ then
        Retain $(q_i, a_i)$ {Only keep QA that demands multiple documents}
    else
        Discard $(q_i, a_i)$
    end if
end for
```

generation. Consequently, it reinforces the multi-document reliance for each question.

4 Dataset Generation and Analysis

In this section, we detail our MDA-QA dataset generation with the pipeline in Section 3. Then, we introduce the statistics of MDA-QA.

4.1 Data Acquisition and Community-Driven Document Aggregation

Citation Graph Construction We base our study on open-access academic publications collected by SPIQA (Pramanick et al., 2025). SPIQA collected 25,859 peer-reviewed articles from 19 top-tier machine learning conferences between 2018 and 2023, focusing on the publicly accessible PDF files and corresponding TeX source files to extract original high-quality article texts. All articles in this collection are indexed in Semantic Scholar, making it easier to retrieve relational information across papers and to build the citation network. In addition, we exclude 1,977 documents from the original SPIQA set due to unreadability and inconsistencies caused by missing content, formatting anomalies, or unprocessable TeX segments. We use the Semantic Scholar API[2] to obtain citation information and build a citation graph with the public Neo4j graph database[3], treating each document as a node and each citation as an edge.

Community Detection As mentioned in Section 3.1, we adopt the SLPA (Speaker-Listener Label Propagation) algorithm to detect communities within the directed citation graph. SLPA propagates labels over multiple iterations and filters them based on frequency thresholds, allowing each document to belong to multiple communities. While SLPA supports both directed and undirected graphs, our pilot test shows that using the directed graph yields smaller, more thematically coherent communities. We set the iteration count to 50, controlling the number of times labels are exchanged, and the filtering parameter to 0.1, specifying the minimum frequency below which labels are discarded. In these settings, we initially obtained 11,373 communities.

To ensure the subsequent QA tasks reflect meaningful multi-document scenarios with manageable context for the LLM, we refine the initially detected communities based on their size and connectivity. Specifically, we discard 6,787 single-node communities arising from isolated citations or newer uncited papers. We also remove 414 large-scale communities exceeding 13 documents, which risk surpassing input token limits for QA generation. This filtering procedure yields 4,172 small-to medium-sized communities, covering 14,698 papers overall. In the future, we will consider hierarchical or sub-community approaches to preserve the information of large communities.

4.2 Dataset Construction

Next, we apply the QA Generation (Section 3.2) and Automated Quality Assurance (Section 3.3) steps to produce QA pairs grounded in multiple documents. Since large language models may differ in their capacity to integrate multi-document context, we conducted a pilot study with GPT-4o (Hurst et al., 2024), Claude-3.5-sonnet (Anthropic, 2024), and Gemini-1.5-Pro (Gemini Team, 2024) on 50 randomly selected communities. The findings suggest that Claude-3.5 tends to generate multi-document questions with more precise

[2] https://www.semanticscholar.org/product/api

[3] https://neo4j.com/

answers. Details of this study and sample outputs from different models can be found in Appendix C.1. We then use Claude first to create 8,573 multi-document question-answer pairs and apply our filtering procedure, which yields a final set of 6,804 QA pairs.

To further validate the reliability of these QA, we randomly sampled 200 QA pairs together with their supporting documents. Each case was manually evaluated by reviewers with expertise in machine learning against three criteria: (i) assess the quality and clarity of the question–answer pair; (ii) verify that a correct response necessitates synthesizing information from all listed support documents rather than any single paper; and (iii) check the factual correctness of the reference answer against the sources. Of these 200 instances, 198 satisfied all criteria. The remaining 2 were flagged for unclear question wording. We did not observe cases where the answer could be correctly obtained from a single document alone. This evaluation verifies the reliability of the resulting QA pairs and shows that the automatic generation and filtering pipeline produces questions that require cross-document synthesis.

4.3 Statistics of MDA-QA

The MDA-QA dataset is based on 14,698 high-level machine learning articles, constructing 4,172 academic communities ranging in size from 2 to 13 documents, each containing an average of 4,043 words. We generate high-quality 6,804 QA, focusing on multi-reference articles. On average, the questions and answers contain 18 and 59 words, respectively. Notably, the variance in question and answer lengths differs significantly, 2.96 for questions and 12.86 for answers. This indicates that the distribution of question lengths is relatively uniform. In contrast, answer lengths show more significant variability, suggesting that the content needed for answers in a multi-document integration context is more diverse.

5 Experiments on QA

In this section, we evaluate whether existing retrieval methods can successfully identify the multiple documents required by our MDA-QA dataset. We first describe how we set up the retrieval task and the methods tested, then analyze retrieval performance. Finally, we explore how the similarity among documents in a community affects retrieval results.

5.1 Experimental Setup

Task Formulation. Given a question q_i supported by multiple documents $\{d_1, d_2, \ldots\}$, the goal is to retrieve the ground-truth support set among a pool of candidate documents. We measure performance via Recall@k, which reveals how many of the gold support documents appear within the top-k retrieved results. We also report Exact Match, EM@k, which requires that *all* gold documents must appear within the top-k results.

Dataset and ground truth. We use the 6,804 multi-document QA pairs in MDA-QA for testing retrieval. 2-3 gold support documents accompany each QA pair; none of the questions can be fully answered without referring to *all* of these documents.

Comparison Methods. We compare classic lexical retrieval and several semantic embedding–based methods:

1. **BM25** (Robertson et al., 2009): A strong lexical baseline that scores query-document matches based on term frequencies.
2. **ColBERT** (Khattab and Zaharia, 2020): A neural ranking model using BERT encodings; we adopt its default configuration for passage-level retrieval.
3. **BGE** (Xiao et al., 2023): A recent open-source general embedding model that encodes queries and documents into a shared semantic space.
4. **MPNET** (Song et al., 2020): A widely used sentence embedding model leveraging masked and permuted language modeling.

We evaluate each model in two retrieval paradigms:

1. **Chunk-based Retrieval (Chunk).** Each document is segmented into smaller chunks containing 400 tokens, and embeddings (or BM25 indexing) are computed at the chunk level. The question is treated as one embedding/query, and top-k chunks are retrieved based on the cosine similarity between the query and the chunk. We then merge chunk-level retrievals by their parent document to see if any chunk from a gold document was retrieved.

Method	Recall@10	Recall@20	Recall@50	EM@10	EM@20	EM@50
Chunk						
BM25	0.35 ± 0.34	0.42 ± 0.36	0.51 ± 0.37	0.12	0.19	0.28
Colbert	0.44 ± 0.32	0.52 ± 0.33	0.62 ± 0.33	0.16	0.25	0.36
BGE	0.49 ± 0.34	0.57 ± 0.35	0.66 ± 0.35	0.22	0.33	0.46
MPNET	0.41 ± 0.35	0.50 ± 0.37	0.60 ± 0.37	0.18	0.27	0.39
Doc						
BM25	0.44 ± 0.36	0.51 ± 0.37	0.60 ± 0.37	0.21	0.28	0.37
BGE	0.51 ± 0.36	0.58 ± 0.36	0.68 ± 0.35	0.27	0.36	0.48
MPNET	0.45 ± 0.37	0.54 ± 0.37	0.64 ± 0.37	0.24	0.32	0.45
BGE-neighbor	0.55 ± 0.40	0.63 ± 0.39	0.73 ± 0.36	0.37	0.46	0.57
MPNET-neighbor	0.48 ± 0.40	0.57 ± 0.40	0.68 ± 0.38	0.31	0.40	0.53

Table 3: Retrieval performance on our multi-document QA dataset.

2. **Document-level Retrieval (Doc).** Each document is represented by a single vector, either by averaging all its chunk-level embeddings or by indexing each full text as a single BM25 unit.

Finally, we adopt a simple **neighbor** approach to exploit structural information in the document-level retrieval setting. We incorporate the neighbor relationship from the citation graph built in Section 4.1 for comparison. Given a citation graph G whose nodes represent documents, we update each node embedding $\mathrm{Emb}(v)$ in a neighborhood-averaging manner:

$$\begin{aligned}\mathrm{Emb}'(v) &= \alpha \times \mathrm{Emb}(v) \\ &\quad + (1-\alpha) \times \left(\frac{1}{|N(v)|} \sum_{u \in N(v)} \mathrm{Emb}(u) \right), \quad (1)\end{aligned}$$

where $N(v)$ is the set of in-neighbors of v in the citation graph, and α is a hyperparameter set to 0.5. We detail the selection of α in Appendix C.3. We label the neighbor-augmented methods *BGE-neighbor* and *MPNET-neighbor*.

5.2 Results on Multi-Document Retrieval

Table 3 summarizes the retrieval performance, evaluated by Recall@k and EM@k for each approach. We report (mean ± standard deviation) for recall@k across all QA pairs.

Unlike lexical-based BM25, neural embedding methods (ColBERT, BGE, MPNET) consistently yield higher recall, indicating that purely lexical matching struggles to capture the multi-document context. Among semantic methods, *BGE* leads under chunk-based retrieval, while *BGE-neighbor* attains the highest recall when using document-level embeddings augmented with local citation neighbors. Note that these results apply to both recall@k and EM@k metrics. These observations confirm that (1) capturing finer-grained semantics is crucial, and (2) leveraging the citation structure yields noticeable gains, thus demonstrating that MDA-QA remains challenging to naive or solely lexical and semantical approaches.

Despite these improvements over BM25, the recall values are still relatively low (often below 0.7 even at 50 documents retrieved), underscoring the difficulty of retrieving multiple relevant documents that collectively answer the question. This shortfall exemplifies the complex multi-document nature of MDA-QA.

5.3 Influence of Document Similarity in a Community

We now analyze whether more homogeneous communities (i.e., documents within them are highly similar) are easier or harder to retrieve. Let $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ be a set of communities. Recall from Section 3 that each of our QA questions is anchored in a small community $C_k = \{d_1, d_2, \ldots, d_{n_k}\}$.

For each community C_k, define:

$$S(C_k) = \frac{1}{\binom{n_k}{2}} \sum_{1 \le i < j \le n_k} \mathrm{sim}\big(d_i,\, d_j\big), \quad (2)$$

where $n_k = |C_k|$ and $\mathrm{sim}(\cdot, \cdot)$ denotes the cosine similarity between two documents. We use MPNET doc embeddings for simplicity.

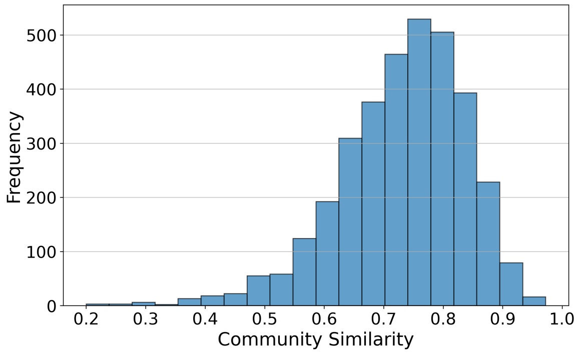


Figure 2: Distribution of community similarity.

Figure 2 shows a histogram of community similarity values. Communities tend to form a strong mode peaking around 0.75–0.85, though there is also a broad tail of less-similar document pairs.

To further quantify how similarity impacts retrieval, Figure 3 plots the average Recall@10 for questions grouped by quantiles of community similarity. We see a slight upward trend: questions from more-similar communities can be somewhat easier to retrieve, presumably because the shared terminology or themes enable a single embedding query to retrieve *all* relevant documents more consistently. Nonetheless, the variance is relatively large, suggesting that multi-document retrieval remains nontrivial, even when the supporting documents are semantically similar.

5.4 RAG Baseline

Beyond pure retrieval metrics, we run a preliminary end-to-end Retrieval Augmented Generation (RAG) (Lewis et al., 2020) experiment. Specifically, we retrieve the top-10 documents via BGE or BGE-neighbor, then feed those documents together with the question into GPT-4o to generate the final answer. We also include a zero-shot GPT-4o setup for comparison.

To evaluate the free-form answers, we follow the previous work (Pramanick et al., 2025; Singh et al., 2024; Lee et al., 2023) under four evaluation metrics - METEOR (Banerjee and Lavie, 2005), CIDEr (Vedantam et al., 2015), ROUGE-L (Lin, 2004), BERTScore F1 (Zhang et al., 2019).

Method	**M**	**R-L**	**C**	**B-F1**
GPT-4o	0.16	0.19	0.54	0.18
w/ BGE	0.23	0.19	0.54	0.17
w/ BGE-neighbor	0.24	0.20	0.50	0.20

Table 4: End-to-end RAG results with GPT-4o on multi-document QA. M: METEOR, R-L: ROUGE-L, C: CIDEr, B-F1: BERTScore F1

As Table 4 shows, using BGE-neighbor yields higher METEOR, ROUGE-L, and BERTScore than the zero-shot GPT-4o and the standard BGE approach, indicating that integrating citation structure helps locate more relevant evidence, thus allowing the LLM to generate the answer. However, the CIDEr score drops slightly for BGE-neighbor. One likely reason is that while the generated answers align better semantically, the specific word overlap with reference answers may decrease. Overall, these findings suggest that incorporating structured retrieval can improve certain aspects of answer quality, but gaps in retrieving all necessary documents still limit overall performance.

5.5 Discussion

Our experiments demonstrate that current retrieval methods, whether lexical BM25 or dense embedding models (BGE, MPNET, ColBERT), struggle to retrieve *all* required documents to answer a question in MDA-QA. Even when increasing the top-k to larger values (e.g., 50), the average Recall remains below 0.70, indicating that current methods still fail to retrieve all relevant documents consistently. Augmenting document embeddings with local graph structure (“neighbor”) helps, but there remains a large gap between these results and the performance one might desire for multi-document question answering. Moreover, the relative similarity of documents within a community affects retrieval to some extent, yet higher similarity does not eliminate the challenge of multi-document integration.

These findings highlight the relevance of our dataset for multi-document retrieval and QA. Further progress may require more advanced techniques that jointly consider document semantics, citation networks, and question structure, rather than relying on independent indexes or simple embedding averages.

6 Conclusion

This study proposes a pipeline for constructing datasets tailored to multi-document academic QA tasks based on the SPIQA original data and citation network structure. We generate MDA-QA of 6,804 high-quality multi-document QA pairs and conduct initial retrieval experiments on it. The result indicates that in multi-document scenarios, straightforward keyword matching or semantic embedding

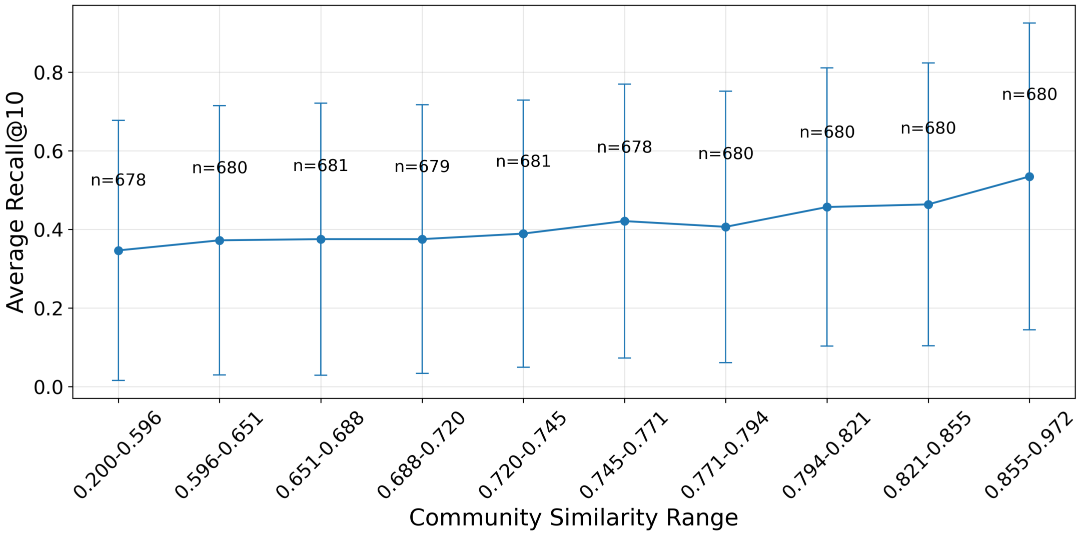


Figure 3: Average Recall@10 across quantiles of community similarity. n indicate the number of questions in each bin. Error bars indicate variability among questions in each bin.

methods remain insufficient for capturing complementary information and latent relationships across documents. Future work may explore graph-based methods, contextual reasoning, and knowledge integration for more effective document merging at the retrieval stage, and introduce advanced multi-document reasoning models in the QA stage, addressing both complex academic inquiries and real-world industrial use cases.

7 Limitation

Although MDA-QA offers a new direction for multi-document scientific QA, several limitations remain. First, our dataset primarily covers the papers for the specific machine learning domain, which does not fully capture broader scientific domains. Extending the approach to diverse fields could yield more comprehensive benchmarks. Secondly, our QA generation and filtering processes lean heavily on LLM prompts and responses. Although we have conducted a small-scale expert review for the automated quality control process, large models can produce hallucinations or overlook complex multi-document reasoning. Finally, this pipeline requires extensive text parsing and repeats LLM calls, imposing resource costs and potentially limiting scalability. Adopting a more efficient model interaction process could reduce resource demands and support larger-scale deployment.

8 Acknowledgments

This project was provided with computing AI and storage resources by GENCI at IDRIS thanks to the grant 2024-AD011014704R1 on the supercomputer Jean Zay's V100 partition. We also thank Worldline for its support, which enabled us to use internal resources to carry out our experiments.

References

Yuan An, Jeannette Janssen, and Evangelos E Milios. 2004. Characterizing and mining the citation graph of the computer science literature. *Knowledge and Information Systems*, 6:664–678.

Anthropic. 2024. Claude 3.5 sonnet model card addendum. Available online: https://www-cdn.anthropic.com/fed9cc193a14b84131812372d8d5857f8f304c52/Model_Card_Claude_3_Addendum.pdf (accessed on April 30, 2025).

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Tim Baumgärtner, Ted Briscoe, and Iryna Gurevych. 2025. Peerqa: A scientific question answering dataset from peer reviews. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 508–544.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

Valeriia Bolotova-Baranova, Vladislav Blinov, Sofya Filippova, Falk Scholer, and Mark Sanderson. 2023. Wikihowqa: A comprehensive benchmark for multi-document non-factoid question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5291–5314.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W Cohen. 2020a. Open question answering over tables and text. In *International Conference on Learning Representations*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036.

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

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Google Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context (2024). Available online at: https://storage.googleapis.com/deepmind-media/gemini/gemini_v1_5_report (accessed on April 30, 2025).

Rujun Han, Yuhao Zhang, Peng Qi, Yumo Xu, Jenyuan Wang, Lan Liu, William Yang Wang, Bonan Min, and Vittorio Castelli. 2024. Rag-qa arena: Evaluating domain robustness for long-form retrieval augmented question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4354–4374.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William Cohen, and Xinghua Lu. 2019. Pubmedqa: A dataset for biomedical research question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

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

Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. 2023. Bioasq-qa: A manually curated corpus for biomedical question answering. *Scientific Data*, 10(1):170.

Konstantin Kuzmin, S Yousaf Shah, and Boleslaw K Szymanski. 2013. Parallel overlapping community detection with slpa. In *2013 International Conference on Social Computing*, pages 204–212. IEEE.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, and 1 others. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. 2023. Qasa: advanced question answering on scientific articles. In *International Conference on Machine Learning*, pages 19036–19052. PMLR.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Chuhan Li, Ziyao Shangguan, Yilun Zhao, Deyuan Li, Yixin Liu, and Arman Cohan. 2024. M3sciqa: A multi-modal multi-document scientific qa benchmark for evaluating foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15419–15446.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2025. Spiqa: A dataset for multimodal question answering on scientific papers. *Advances in Neural Information Processing Systems*, 37:118807–118833.

Jun-Ping Qiu, Ke Dong, and Hou-Qiang Yu. 2014. Comparative study on structure and correlation among author co-occurrence networks in bibliometrics. *Scientometrics*, 101:1345–1360.

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

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Scienceqa: a novel resource for question answering on scholarly articles: Scienceqa: a novel resource for question answering on scholarly articles. *International Journal on Digital Libraries*, 23(3):289–301.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, and 1 others. 2024. The prompt report: A systematic survey of prompting techniques. *CoRR*.

Shruti Singh, Nandan Sarkar, and Arman Cohan. 2024. Scidqa: A deep reading comprehension dataset over scientific papers. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20908–20923.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *Advances in neural information processing systems*, 33:16857–16867.

Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. 2019. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):1–12.

George Tsatsaronis, Georgios Balikas, Prodromos Malakasiotis, Ioannis Partalas, Matthias Zschunke, Michael R Alvers, Dirk Weissenborn, Anastasia Krithara, Sergios Petridis, Dimitris Polychronopoulos, and 1 others. 2015. An overview of the bioasq large-scale biomedical semantic indexing and question answering competition. *BMC bioinformatics*, 16:1–28.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Minzheng Wang, Longze Chen, Fu Cheng, Shengyi Liao, Xinghua Zhang, Bingli Wu, Haiyang Yu, Nan Xu, Lei Zhang, Run Luo, and 1 others. 2024. Leave no document behind: Benchmarking long-context llms with extended multi-doc qa. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5627–5646.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Andrew Zhu, Alyssa Hwang, Liam Dugan, and Chris Callison-Burch. 2024. Fanoutqa: A multi-hop, multi-document question answering benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 18–37.

A Prompts for QA generation

The prompt we used for QA Generation is presented in Figure 6. It contains instructions for cross-document references and filtering steps.

B Prompts for Automated Quality Assurance

LLMAnswer

```
Given the following question and one scientific paper, generate an answer based on the given paper. If the question cannot be answered by the provided contents, please provide a reason why. Directly respond to the question, don’t provide any additional information. You must provide a concise answer in one sentence only.
Question: <question>
Paper: <paper>
```

Compare

```
Given the following question and a set
of corresponding scientific articles:
Question: <question>
Articles: <all_articles>
Compare the model answer with the
grand truth answer based on the
provided articles:
Model answer: <LLMAnswer>
Ground truth answer: <ground_truth>
Determine which answer is better and
why.  choose between [model_answer,
ground truth answer, neither, equal]
```

C Experiment Details

C.1 Comparison of LLM Outputs

In our pilot study, we randomly selected 50 academic communities and applied the same prompt to three different large language models (Claude 3.5-Sonnet, Gemini-1.5-Pro, and GPT-4o) to generate multi-document QA pairs. We present an example in Table 5.

We manually verified the quality of the generated QA by each model, focusing on three aspects: (i) Whether the model understands the prompt's requirement to "generate QA based on multiple documents"; (ii) Whether the questions are neither too trivial nor too specific for any single supporting document; (iii) Whether the answer requires integrating all supporting documents to be complete and accurate.

While each model was able to produce questions referencing more than one document, we found that Claude 3.5 generally provided more coherent, concise, and contextually relevant questions and answers than the other two. In contrast, the other models occasionally generate questions that are either too general or include unnecessary details.

C.2 Expert Sanity Check of the Automated Filter

This study is a lightweight sanity check designed to validate the reliability of our large-scale automated QA filtering (Algorithm 1). The UI used is shown in Fig 5.

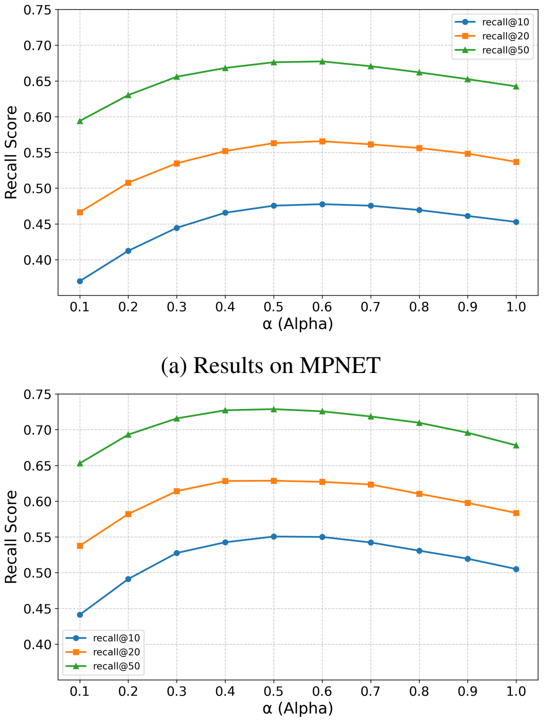
We randomly sampled 30 QA pairs produced by our pipeline and applied Algorithm 1 to them. During the **Compare** step, the LLM was additionally required to provide a brief rationale explaining why the multi-document answer is strictly better than any single-document answer.

Three machine learning experts independently examined each sampled QA pair and assessed: (i) the overall quality and clarity of the question–answer pair; (ii) the correctness of the LLM's **Compare** step reasoning; and (iii) whether answering the question truly requires at least two documents.

The result shows that our LLM-based filter's reasoning is consistent with the experts' assessment, indicating that the automated process successfully flags single-document questions while preserving those that require cross-document synthesis.

C.3 Alpha in Neighbor Method

We systematically evaluated α from 0.1 to 1.0 in increments of 0.1, measuring how Recall@10, Recall@20, and Recall@50 would vary. We conducted this test on MPNET and BGE. Figure 4 presents the resulting curves.



(a) Results on MPNET

(b) Results on BGE

Figure 4: Effect of α Parameter on Recall@k Metrics. In these two figures, the green, orange, and blue curves represent Recall@10, Recall@20, and Recall@50, respectively. All three curves peak or plateau around $\alpha = 0.5$ for both MPNET and BGE.

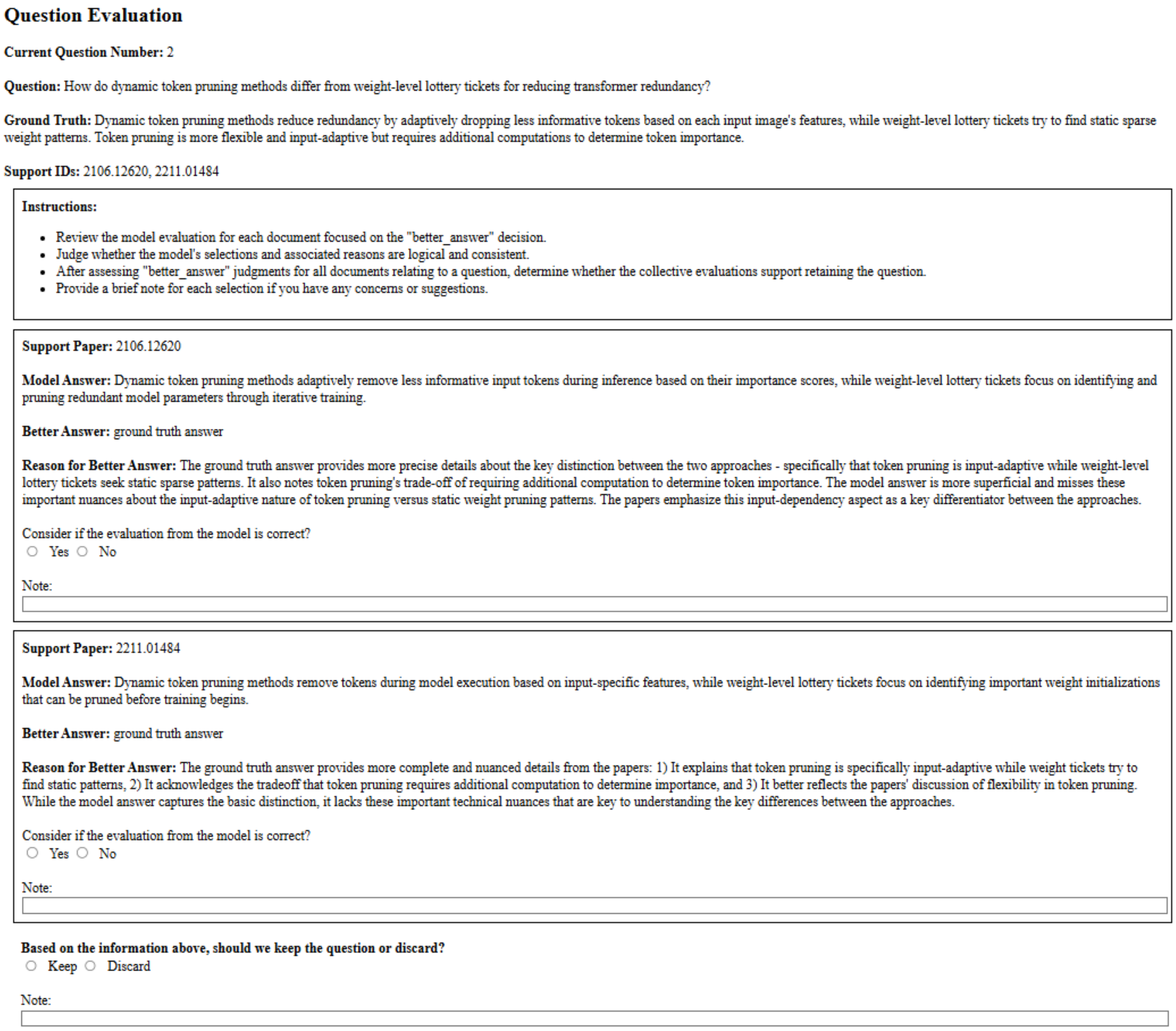
As α increases from 0.1, all three Recall metrics initially rise, indicating that incorporating a

Community: [1805.01089, 1906.05691, 1803.01465]	
Claude-3.5-Sonnet	**Q:** How does incorporating sentiment information into text summarization models help improve both tasks according to experimental results? **A:** Joint learning of sentiment and summarization leads to better performance in both tasks because sentiment classification provides supervision signals that guide summarization to capture sentiment tendencies, while summarization helps identify important sentiment-bearing content and removes noise, resulting in improved classification accuracy. This is demonstrated by experimental results showing significant improvements in both ROUGE scores for summarization and sentiment classification accuracy compared to independent models. **Support:** [1805.01089, 1906.05691]
Gemini-1.5-Pro	**Q:** What are the advantages and disadvantages of using word embeddings in the output layer of a neural paraphrase generation model compared to a traditional linear softmax operation? **A:** Using word embeddings in the output layer of a neural paraphrase generation model allows the model to capture semantic meaning by referring to the embedding of a word, while a traditional linear softmax operation learns the word itself rather than the relationship between words. However, word embeddings can increase the model's complexity and require careful consideration of how to effectively incorporate them. **Support:** [1803.01465, 1805.01089]
GPT-4o	**Q:** How do the proposed models in both papers address the challenge of improving sentence summarization while ensuring the generated summaries are coherent and relevant to the sentiment expressed in the reviews? **A:** Both models utilize an encoder-decoder framework where one enhances summarization by focusing on relevant aspects of sentiment within the reviews, while the other employs a retrieval style for word generation, capturing the semantic meaning through distributed representations, thereby improving coherence and relevance. **Support:** [1906.05691, 1805.01089]

Table 5: Comparison of multi-document QA generated by different LLMs in the same academic community based on the same Prompt.

greater proportion of the document's embedding is beneficial up to a certain point. However, when α exceeds approximately 0.5–0.6, performance begins to drop. We hypothesize that high α places too much emphasis on the document's representation, underutilizing the citation network context; conversely, low α dilutes the document's content features.

Hence, $\alpha = 0.5$ emerges as a sweet spot where retrieval performance across different recall metrics is either maximal or near-maximal, and remains relatively robust across various communities of differing sizes. Based on these findings, we adopt $\alpha = 0.5$ in our main experiments to have a balance between retaining each document's unique embedding and incorporating vital neighbor information.



Question Evaluation

Current Question Number: 2

Question: How do dynamic token pruning methods differ from weight-level lottery tickets for reducing transformer redundancy?

Ground Truth: Dynamic token pruning methods reduce redundancy by adaptively dropping less informative tokens based on each input image's features, while weight-level lottery tickets try to find static sparse weight patterns. Token pruning is more flexible and input-adaptive but requires additional computations to determine token importance.

Support IDs: 2106.12620, 2211.01484

Instructions:

- Review the model evaluation for each document focused on the "better_answer" decision.
- Judge whether the model's selections and associated reasons are logical and consistent.
- After assessing "better_answer" judgments for all documents relating to a question, determine whether the collective evaluations support retaining the question.
- Provide a brief note for each selection if you have any concerns or suggestions.

Support Paper: 2106.12620

Model Answer: Dynamic token pruning methods adaptively remove less informative input tokens during inference based on their importance scores, while weight-level lottery tickets focus on identifying and pruning redundant model parameters through iterative training.

Better Answer: ground truth answer

Reason for Better Answer: The ground truth answer provides more precise details about the key distinction between the two approaches - specifically that token pruning is input-adaptive while weight-level lottery tickets seek static sparse patterns. It also notes token pruning's trade-off of requiring additional computation to determine token importance. The model answer is more superficial and misses these important nuances about the input-adaptive nature of token pruning versus static weight pruning patterns. The papers emphasize this input-dependency aspect as a key differentiator between the approaches.

Consider if the evaluation from the model is correct?
○ Yes ○ No

Note:

Support Paper: 2211.01484

Model Answer: Dynamic token pruning methods remove tokens during model execution based on input-specific features, while weight-level lottery tickets focus on identifying important weight initializations that can be pruned before training begins.

Better Answer: ground truth answer

Reason for Better Answer: The ground truth answer provides more complete and nuanced details from the papers: 1) It explains that token pruning is specifically input-adaptive while weight tickets try to find static patterns, 2) It acknowledges the tradeoff that token pruning requires additional computation to determine importance, and 3) It better reflects the papers' discussion of flexibility in token pruning. While the model answer captures the basic distinction, it lacks these important technical nuances that are key to understanding the key differences between the approaches.

Consider if the evaluation from the model is correct?
○ Yes ○ No

Note:

Based on the information above, should we keep the question or discard?
○ Keep ○ Discard

Note:

Figure 5: **UI used for sanity check.** For each sampled QA, the interface displays the question, the multi-document ground-truth answer, per-support-paper model answers with the Compare-step rationale. We ask the experts to verify the model's judgment for each paper, add notes, and make a final keep/discard decision.

** Task Context **
Analyze scientific literature communities to generate complex questions requiring the synthesis of multiple documents. Each question must:

1. Involve at least 2 articles with deep interconnections

2. Cover question types including but not limited to:

    - Methodological comparison and critique
    - Contradictory conclusion analysis
    - Technological evolution tracing
    - Multidimensional evaluation
    - Hypothesis validation pathways

** Community Information **
<paper contents>

** Processing Pipeline **

1. Select at least two articles with deep interconnections.

2. Find the connection between these articles and identify Problem Spaces:

    - Method contrast: CNN in Paper X vs Transformer in Paper Y
    - Conclusion conflict: p<0.01 in Paper Z vs p>0.05 in Paper W
    - Technical progression: Baseline in Paper A → Optimized solution in Paper D

3. Based on the first two steps, generate questions following the rules:

    - Avoid asking simple or definitional questions.
    - Avoid asking questions like "How do different approaches...", "What are the key challenges...", "What are the key differences..."
    - The questions should be in one sentence only; they should not consist of more than one question.
    - The questions should not contain the titles or method names; don't use phrases like 'as discussed in the articles.'
    - Ensure that the answers are concise, accurate, and directly related to the corresponding question.
    - Do not generate any information that does not appear in the original documents, nor make unsupported inferences.

Repeat the process to generate questions as much as possible.

Figure 6: Prompt provided to Claude 3.5 during QA generation phase