Biomedical Question Answering via Multi-Level Summarization on a Local Knowledge Graph

Lingxiao Guan[1], Yuanhao Huang[2], Jie Liu[1,2]

[1]Department of Computer Science and Engineering, University of Michigan
[2]Department of Computational Medicine and Bioinformatics, University of Michigan
{lxguan, hyhao, drjieliu}@umich.edu

Abstract

In Question Answering (QA), Retrieval Augmented Generation (RAG) has revolutionized performance in various domains. However, how to effectively capture multi-document relationships remains an open question. This is particularly critical for biomedical tasks due to their reliance on information spread across multiple documents. In this work, we propose a novel method CLAIMS, which utilizes propositional claims to construct a local knowledge graph from retrieved documents. Summaries are then derived via layerwise summarization from the knowledge graph to contextualize a small language model to perform QA. The structured summaries effectively capture explicit and implicit relationships between entities in the documents, thus having a more comprehensive context to provide to LLMs. CLAIMS achieved comparable or superior performance over RAG baselines on several biomedical QA benchmarks. We also evaluated its generalizability and each individual step of our approach with a targeted set of metrics, demonstrating its effectiveness.

1 Introduction

Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) has shown promise in augmenting Large Language Models (LLMs) with documents retrieved from established corpora. The process uses these documents to ground LLM outputs, reducing hallucinations and improving the contextual relevance of generated responses. For a typical Question Answering (QA) task, RAG tends to retrieve multiple documents relevant to an input question. However, recognizing and leveraging the multi-document relationships across these documents remains an underexplored challenge. Relying on a single LLM call to integrate all of these relationships tends to prove inadequate, especially in Biomedical QA where even seemingly straightforward questions like determining which muscle fiber types fatigue first during sprinting can require connecting properties described in one document with mechanisms explained in another (Figure 1). Existing work has introduced targeted techniques to mitigate this problem, such as hierarchical summarization of semantically related chunks (Sarthi et al., 2024; Tang et al., 2024) or integrating Knowledge Graphs (KGs) to represent explicit connections in retrieved text. Yet reliance on semantically related chunks can miss documents that share topics but differ in semantic focus, and works that utilize KGs can face different limitations: some require access to the entire offline knowledge corpus (Guo et al., 2024b; Wu et al., 2025) and operate through hierarchical community-based partitioning of such large corpora (Edge et al., 2024), while others suffer from explicit information loss during graph traversal for retrieval (Wang et al., 2024; Guo et al., 2024a). Therefore, there is a need for a method that *effectively represents and utilizes relevant multi-document relationships from dynamically updated knowledge bases, enabling systems to trace cross-document connections and achieve more comprehensive reasoning in Biomedical QA.*

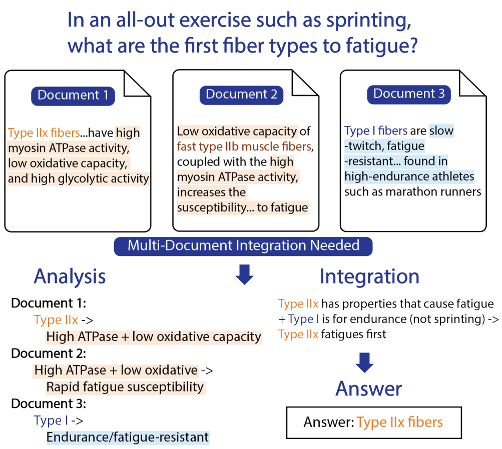

Figure 1: Multi-document reasoning example in biomedical QA. Requires connecting Type IIx fiber properties across documents to infer fatigue susceptibility.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16338–16364
July 2-7, 2026 ©2026 Association for Computational Linguistics

To remedy this, we propose utilizing the construction of a knowledge graph to underlay layerwise document summarization via **CLAIMS** (Connected Layered Analysis of Information through Multi-level Summarization). Propositional claims (Chen et al., 2024b) are utilized to represent information and facilitate handling conflicting and noisy claims extracted from retrieved documents. A knowledge graph dynamically constructed from these propositional claims captures relationships beyond semantic similarity. Finally, our approach performs layerwise graph summarization around several key claims of interest to comprehensively capture and filter multi-document relations and fit them into a limited context window.

CLAIMS utilizes the properties of decontextualized claims in the knowledge graph structure and layerwise topological summarization to capture explicit and implicit relationships between entities in the documents, thus having a more comprehensive context to provide to LLMs. We evaluate each part of our methodology, and compare CLAIMS to traditional RAG retrieval baselines on several biomedical QA datasets, achieving comparable or superior performance over all baselines.

Our approach makes three main contributions.

- We introduce dynamic local knowledge graph construction with propositional claims, enabling capture of implicit cross-document relationships without requiring prebuilt knowledge graphs.
- We introduce layerwise topological graph summaries of key claims in this local knowledge graph as context for LLM QA tasks, utilizing progressive summarization to filter irrelevant and conflicting information from retrieved documents.
- We evaluate CLAIMS on a comprehensive set of benchmarks, including testing its intermediate components, its impact on LLM reasoning, generalizability across domain-specific models, and the final accuracy on several datasets.

2 Related Work

We review relevant work in RAG, summarization techniques, and knowledge graph applications for Biomedical QA. Current approaches face challenges in effectively capturing cross-document relationships. CLAIMS builds upon these foundations while addressing their limitations through the novel combination of propositional claims, local knowledge graphs, and layerwise summarization.

2.1 Retrieval Augmented Generation

Information Retrieval methods have been used for general QA tasks, including biomedical QA (Jin et al., 2022). RAG extends these methods for use with LLMs, allowing for the integration of external corpora into pre-trained LLMs' context windows. The initial naive RAG approach utilized a retriever and a seq2seq model to capture knowledge from documents (Lewis et al., 2020), and has since been followed by many follow-up refinements (Gao et al., 2023b). Many works have been conducted on the application of RAG in biomedical QA, such as MedRAG which retrieves documents from a variety of corpora (Xiong et al., 2024), BioMedRAG which trains a retriever for improved retrieval of medical documents (Li et al., 2024b), and Self-BioRAG which uses on-demand retrieval and reflection tokens to select evidence (Yu et al., 2023), among others (Liu et al., 2024; Zhou et al., 2023), which tend to take strategies from general domain RAG and adapt them to the biomedical domain. While they provide benefits for QA tasks, they fall short in capturing all relevant multi-document relationships in retrieved documents.

2.2 Summarization

Summarization can condense documents into relevant information while using less tokens, and is one method by which retrieved documents can be processed. RAPTOR (Sarthi et al., 2024) uses hierarchical summarization of documents to capture both locally relevant information and distant interdependencies. However, its reliance on semantic similarity means that it may miss explicit, non-semantic connections. Long-context summarization methods like MemTree (Rezazadeh et al., 2024) or iterative hierarchical summarization methods like ILM-TR (Tang et al., 2024) also use embedding similarity to group contextual information, and thus suffer from the same problem of missing explicit connections. SiReRAG extends RAPTOR with an additional hierarchical summarization of propositional claims (Zhang et al., 2025a), but while this does capture relationships between shared entities it still misses explicit multi-hop connections.

2.3 RAG with knowledge graphs

Graph based RAG is an alternative to semantic similarity for capturing complex relationships. Extensive prior work leverages knowledge graphs as data structures, with recent advances in KG construc-

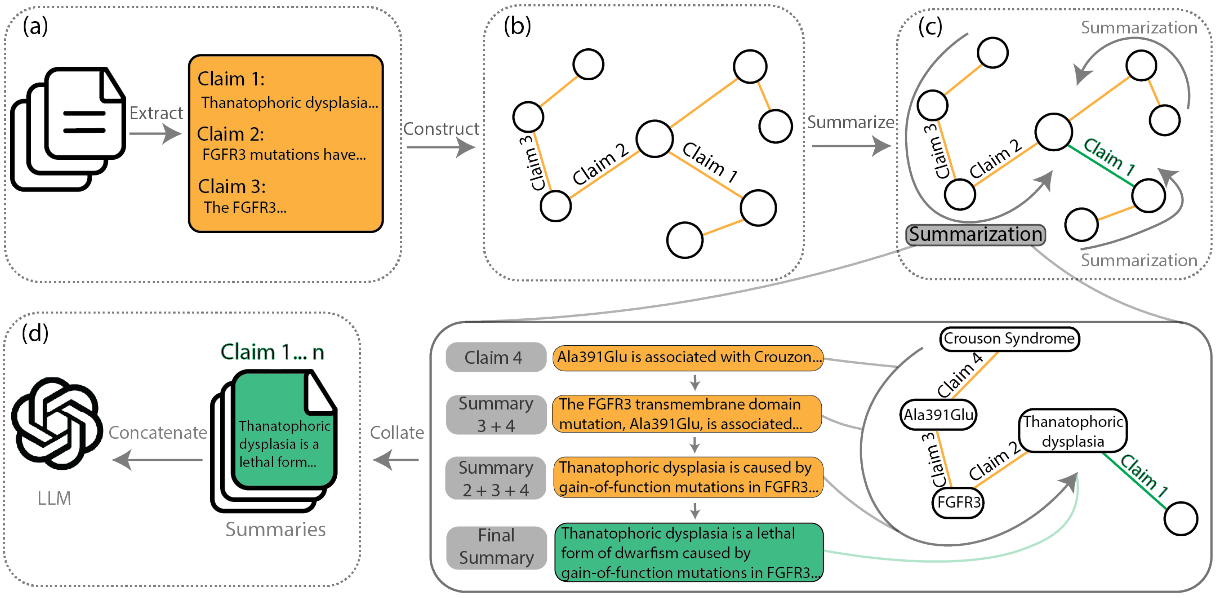

Figure 2: Overview of the proposed CLAIMS framework. **(a) Relation extraction:** load in documents with a retriever relevant to an input question, break documents into claims, break claims into triples. **(b) Graph construction:** build local graph with claims and triples. **(c) Graph summarization:** summarize the graph layerwise with the top re-ranked claims as the roots. **(d) QA with LLM:** the final summaries for each top-ranked claim are collated and provided to a model as context for downstream QA tasks.

tion (Yang et al., 2025; Mo et al., 2025). Common RAG methods include directly retrieving relevant triples from the graph (Baek et al., 2023), subgraph extraction (Gutiérrez et al., 2025; Sarmah et al., 2024; Li et al., 2024a), or path based retrieval of relevant documents (Chen et al., 2024a; Luo et al., 2024; Jiang et al., 2024b; Ma et al., 2025). These methods may miss out on information outside of the explicit subgraphs or paths that are retrieved.

More recently, there has been work performing hierarchical summarization on generated knowledge graphs using different organizational strategies. Graph RAG (Edge et al., 2024) employs community-based summarization, partitioning entire offline knowledge graphs into communities and creating hierarchical summaries at multiple granularities. Similarly, MedGraphRAG (Wu et al., 2025) uses hierarchical tagging approaches. While these community-detection methods can capture broad multi-document relationships, they operate on static, large-scale corpora and require global community identification. In contrast, CLAIMS processes relationships hop-by-hop around specific claims of interest, enabling summarization without upfront community detection. Additionally, these approaches require high upfront costs and additional effort to update their graph summaries with new information, making them less suitable for dynamically retrieved documents.

Alternatively, retrieved documents can be turned into a graph structure for additional processing. Several works have opted for this, with many using semantic similarity of text chunks in combination with structural information to construct the graph. Even with explicit connections formed by structural relationships, the retrieval uses agents (Wang et al., 2024; Guo et al., 2024a) that can miss information outside of returned paths or requires a trained GNN (Li et al., 2025). CLAIMS utilizes explicit connections from knowledge graph Resource Description Framework (RDF) formats and does layerwise summarization to capture these connections with off-the-shelf LLMs. Another work generates minigraphs from documents (Zhang et al., 2025b), but does not use propositional claims as their chunking modality and performs literature review creation instead of QA.

3 Methods

Approach overview: CLAIMS handles the problem of processing and connecting distributed evidence from multiple retrieved documents to solve biomedical questions. At its core, our method takes in a biomedical question, a set of retrieved documents, and possible multiple choice answers before using a language model to process the documents and determine the correct answer. More formally, given an input biomedical question q, a set of answer options A, and a corpus of dynamically updated unstructured documents D, a language model L is used to generate the correct answer $a \in A$. The output should satisfy three requirements:

1. Comprehensively identify and connect multi-document relations.
2. Efficiently use the limited context window of L.
3. Reduce noise and preserve relevant information.

CLAIMS improves the extraction and presentation of relevant information and multi-document relations from unstructured documents by the addition of layerwise graph summarization (Figure 2). It proceeds by first extracting decontextualized claims from each $d \in D$ (Section 3.1), assembling these into a local knowledge graph where shared entities reveal cross-document connections (Section 3.2), before performing layerwise summarization around key claims that progressively integrates information along graph paths while filtering noise (Section 3.3). This approach enables systematic identification of multi-hop relationships that would be difficult to discover from the documents alone.

3.1 Relation extraction

CLAIMS first transforms retrieved unstructured documents into propositional claims and associated RDF triples.

Retrieval: To accurately answer biomedical questions, CLAIMS gathers relevant information from several knowledge bases. For a given input question q, it is first preprocessed into a better suited retrieval query to retrieve relevant documents $d \in D$ via question rewriting (Ma et al., 2023) and HyDE candidate answer generation (Gao et al., 2023a).

The final query with the rewritten question, answer options, and candidate answer is used to retrieve text chunks $d \in D$. Further details on the retrieval corpora and the retrieval process can be found in Appendix A.1.

Claim extraction: To connect information across documents, documents are broken down into concise, independent pieces. From the retrieved text chunks $d \in D$, the model L extracts propositional claims $C = \{c_1, c_2, ..., c_n\}$. These claims must be

- Atomic: includes only a single statement that cannot be broken down, and
- Decontextualized: fully understandable on its own with no unresolved entity references.

This chunking strategy improves the retriever's performance (Chen et al., 2024b) and is especially important in CLAIMS for later reranking and summarization.

Triple extraction: Once a claim $c \in C$ is extracted, it is prepared for addition to the local graph G. We assume that the claim extraction process has given us atomic claims with only one key relation. This step involves extracting a single RDF triple $(subj,\ pred,\ obj)$ from each claim c. This triple format captures the relationship $pred$ between the two entities $subj$ and obj, with the extraction being based on the LLM's best judgment.

3.2 Graph construction

The RDF triples and claims from Section 3.1 are processed into a local graph structure that captures the relationships between concepts.

Deduplication: While our claim extraction phase (Section 3.1) resolves coreferences to the same entities, the entities in each RDF triple can still have multiple possible representations. Deduplication of entities in the RDF triples is performed to ensure that all references to the same concept point towards the same node in the graph. Specifically, embeddings are placed into the same cluster using a similarity threshold of 0.8 with Unweighted Average Linkage Clustering (UPGMA) (Sokal and Michener, 1958).

Graph structure: After deduplication, the processed RDF triples and claims are used to construct the graph G. Each node $n \in G$ is an entity from the RDF triples. Each edge $e \in G$ between nodes includes the representative claim c the entities were extracted from and relevancy score s. The scores are calculated using a reranker R according to the edge claim's relevance to the input question q. All edges are treated as undirected in further processing, and allow for multiple edges between nodes.

3.3 Graph summarization

CLAIMS condenses the content in G into several claims of interest to capture the most relevant information for answering the input question.

Obtaining claims of interest: Due to the many documents under consideration, our method selects several key claims of interest K from G, which provides a diverse set of entry points into the graph.

CLAIMS starts with the top 10 ranked claims in the graph based on their relevance scores s.

To evaluate which claims are most suitable for summarization, we assess the informativeness of their local neighborhoods. For each candidate claim of interest, we gather its 1-hop neighboring claims, concatenate them as context, and generate a preliminary *test summary*. These summaries are scored for relevance using reranker R, with the scores used to rerank the claims of interest, prioritizing those likely to yield high-quality summaries.

As adjacent claims should produce similar summaries, we remove all claims that are 1-hop neighbors of higher ranked claims in K. This returns a more focused list, improving efficiency while retaining coverage of relevant information.

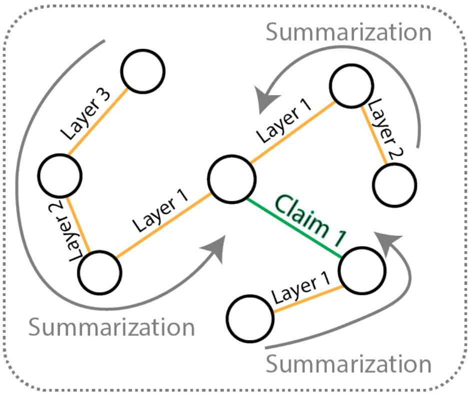

Figure 3: Layerwise summarization overview. For a claim of interest (Claim 1), the graph is organized into layers based on each connected claim's distance from it. The summarization begins from the furthest layer, moving inwards. For each layer, claims are summarized using previously generated summaries of connected claims in lower layers. This process ensures that path content and multi-document relationships are preserved while filtering out irrelevant information.

Layerwise summarization: Layerwise summarization for each claim of interest involves organizing its connected component in G into layers based on each claim's distance (Figure 3).

Definition 1 (Layer)**.** Given a claim of interest k in graph G, the ith layer consists of all claims that are exactly i-hop away from k in G.

The summarization process starts from the outermost layer and proceeds inwards. For each claim in the current layer, our method considers the summaries of connected claims one layer below. These summaries from connected claims are again summarized to create the current claim's own summary. Each claim is processed only once and uses summaries from already processed claims, ensuring that there are no cycles. This occurs layer by layer until the claim of interest is reached.

Summary generation: The final summary for each claim of interest captures information from its entire connected component in G, focused around the central claim. Although the claims of interest share common topics due to their high relevance to the input question, each final summary differs as they emphasize their local relationships. The final output concatenates the summaries in the order of their relevance rankings. This set of summaries is provided as context for an LLM to perform QA. Additional method details are provided in Appendix A.

4 Experiments

Our experiments assess both CLAIMS' overall QA performance and the effectiveness of its individual components. We evaluate on multiple benchmarks (Section 4.1), comparing against standard RAG baselines (Section 4.3) and domain-specific models (Section 4.4). We employ masking tests to evaluate CLAIMS' ability to improve LLM reasoning capabilities (Section 4.5). Additionally, each part of CLAIMS was assessed individually (Section 4.6) and as a whole (Section 4.7) to test its robustness.

4.1 Evaluation datasets

We evaluate CLAIMS on established biomedical QA benchmarks. We use the test sets of PubMedQA (Jin et al., 2019), MedQA (Jin et al., 2021), and the MMLU clinical topics datasets (Hendrycks et al., 2021) (Anatomy, Clinical Knowledge, College Biology, Professional Medicine, College Medicine, and Medical Genetics). For validation, a combination of the validation sets of the MMLU datasets is used, termed MMLU validation. More information is provided in Appendix G.

4.2 Experimental setup

Model configuration: All QA experiments were run with greedy decoding to ensure reproducible results. Results were averaged across 5 runs (1 non-shuffled and 4 shuffled answer options) to control for positional bias. For CLAIMS contexts generation and QA benchmarking we used Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) based on preliminary experiments demonstrating that domain-specific models had degraded performance on auxiliary tasks such as entity extraction. More details are in Appendix E.

Approach	MMLU-A*	MMLU-CM	MMLU-CB	MMLU-PM	MMLU-MG	MMLU-CK	MMLU-V	PMQA	MedQA	Overall SE
MedRAG	0.57	0.50	0.51	0.54	0.69	0.54	0.57	0.60	0.44	0.0027
KGP	0.48	0.46	0.52	0.52	0.61	0.54	0.55	0.54	0.44	0.0033
SiReRAG	0.58	0.53	0.58	0.62	0.70	0.59	0.65	0.55	0.51	0.0009
RAPTOR	0.56	0.57	0.62	0.61	0.72	0.64	0.66	**0.68**	0.50	0.0034
CLAIMS	**0.62**	**0.60**	**0.64**	**0.64**	**0.74**	**0.66**	**0.70**	0.61	**0.52**	0.0021

*MMLU prefixes denote: V-Validation, A-Anatomy, CB-College Biology, CM-College Medicine, PM-Professional Medicine, MG-Medical Genetics, CK-Clinical Knowledge

Table 1: Accuracy scores across RAG approaches with Overall SE representing the standard error of overall performance (average across all datasets) over 5 runs. CLAIMS shows consistent improvements over baselines.

Approach	MMLU-A*	MMLU-CM	MMLU-CB	MMLU-PM	MMLU-MG	MMLU-CK	MMLU-V	PMQA	MedQA	Overall SE
MMed	0.61	0.54	0.65	0.67	0.69	0.66	0.69	0.57	0.53	0.0032
BioMistral	0.51	0.52	0.56	0.57	0.62	0.60	0.57	0.38	0.46	0.0049
Meditron 3	0.61	0.66	0.76	0.79	0.76	0.75	0.77	0.58	0.61	0.0030
Med42	0.70	0.67	0.79	0.77	0.77	0.76	0.80	0.53	0.62	0.0045
MMed Ctxs	0.65	0.56	0.65	0.69	0.76	0.67	0.73	0.65	0.57	0.0034
BioMistral Ctxs	0.58	0.56	0.60	0.65	0.76	0.64	0.69	0.57	0.51	0.0040
Meditron 3 Ctxs	0.71	0.68	0.76	0.82	0.81	0.77	0.81	0.70	0.66	0.0038
Med42 Ctxs	0.72	0.68	0.80	0.81	0.80	0.76	0.81	0.65	0.65	0.0039

*MMLU prefixes denote: V-Validation, A-Anatomy, CB-College Biology, CM-College Medicine, PM-Professional Medicine, MG-Medical Genetics, CK-Clinical Knowledge

Table 2: Accuracy scores across various domain specific LLMs, with and without the contexts produced by our CLAIMS method. Overall SE represents the standard error of overall performance (average across all datasets) over 5 runs. Augmentation with CLAIMS shows consistent improvements over the baseline domain-specific models.

Answer extraction protocol: Due to the variability in LLM outputs, we implemented a standardized answer extraction via Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) to parse the model output and extract the selected option into a JSON object. JSON format is enforced via lm-format-enforcer[1].

4.3 Comparison with RAG baselines

We compared CLAIMS with several representative RAG approaches: RAPTOR for hierarchical summarization (Sarthi et al., 2024), KGP for dynamic knowledge graph generation (Wang et al., 2024), SiReRAG for claim summarization (Zhang et al., 2025a), and MedRAG for direct retrieval (Xiong et al., 2024) with our retrieval datasets.

Experimental details: We used these approaches' provided code and prompts, with minor modifications for compatibility with our retrieval corpora and evaluation framework. All approaches were subject to truncation to model max lengths. For MedRAG, we utilized retrieval from our corpora using only embedding similarity for document scores.

Results: CLAIMS achieved comparable or superior performance to all baseline methods (Table 1), with improvements ranging from 1.46% (RAPTOR) to 11.64% (KGP) on non-validation datasets. While CLAIMS showed a decrease compared to RAPTOR on PubMedQA, it outperformed RAPTOR and the other methods on all other benchmarks, demonstrating that structured knowledge representation and layerwise summarization provide meaningful advantages over both direct graph traversal and semantic clustering approaches.

4.4 Generalizability across biomedical models

To evaluate the generalizability of CLAIMS' contexts, we tested its ability to improve the performance of models pretrained on biomedical texts. This demonstrates that CLAIMS provides genuinely useful information rather than simply compensating for missing domain knowledge, and validates that our approach works across different model architectures and training paradigms.

Experimental details: We evaluated four representative biomedical models: BioMistral-7B (Labrak et al., 2024), Meditron3-8B (Chen et al., 2023), Llama3-Med42-8B (Christophe et al., 2024), and MMed-Llama-3-8B (Qiu et al., 2024). Each model was tested with baseline (parametric knowledge only) and augmented (with CLAIMS-generated contexts) conditions on the evaluation datasets.

Results: CLAIMS was able to improve the performance across all models, showcasing its effectiveness (Table 2). The improvements ranged from 3.27% (Med42) to 8.26% (BioMistral) on

[1]https://github.com/noamgat/lm-format-enforcer

Approach	MMLU-A*	MMLU-CM	MMLU-CB	MMLU-PM	MMLU-MG	MMLU-CK	MMLU-V	PMQA	MedQA	Overall SE
HyDE	0.25	0.24	**0.27**	0.27	0.31	0.27	0.25	0.38	0.26	0.0011
CLAIMS	**0.26**	**0.30**	0.26	**0.30**	**0.35**	**0.32**	**0.32**	**0.42**	**0.28**	0.0016

*MMLU prefixes denote: V-Validation, A-Anatomy, CB-College Biology, CM-College Medicine, PM-Professional Medicine, MG-Medical Genetics, CK-Clinical Knowledge

Table 3: Accuracy scores with masked retrieved documents, input questions, and answer options. Overall SE represents the standard error of overall performance (average across all datasets) over 5 runs. Our CLAIMS approach achieved higher scores on 7 of 8 non-validation datasets.

the non-validation datasets. The consistent improvements across models with different training backgrounds demonstrate that CLAIMS provides structured knowledge that complements rather than duplicates existing model capabilities.

4.5 Entity masking evaluation

Experimental details: To evaluate whether CLAIMS improves LLM reasoning beyond simply leveraging memorized biomedical facts, we masked biomedical entities. This removes the model's ability to rely on entity associations learned during pretraining while preserving the logical structure of the reasoning task.

We prompted the Llama-3.3-70B-Instruct model (Dubey et al., 2024) to identify and mask key biomedical entities into one of 13 categories as generic labels. We compared CLAIMS under this circumstance against directly adding masked documents to the LLM's context window (HyDE in the table). More details can be found in Appendix I.

Entity masking results: Notably, the accuracy scores of the masked configuration are significantly lower than their unmasked variants, suggesting that the masking of entities has disrupted entity-specific knowledge that had been learned during pretraining. CLAIMS had superior performance on 7 of 8 non-validation datasets, achieving an average improvement of 3% on them (Table 3). This demonstrates that CLAIMS improves the reasoning ability of LLMs independent of parametric knowledge, rather than only utilizing entity patterns learned during pretraining.

4.6 Component level analysis

To demonstrate the effectiveness of each step of CLAIMS, we obtained evaluation results for each of CLAIMS' three core components: relation extraction, graph construction, and graph summarization on our MMLU validation dataset. All results were obtained using Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) with greedy decoding once.

Approach	Ref Score	Sem. Sim.	Claim Ret.
single-stage	0.94	0.90	1.00
two-stage	0.95	0.90	1.00
direct-triples	0.97	0.87	1.00
pairs-relations	0.99	0.82	1.00

Table 4: Relation extraction evaluation with Ref. Score (decontextualization ability), Sem. Sim. (original meaning preservation), and Claim Ret. (key information preservation). Results demonstrate the trade-off between entity and claim based approaches, with our single-stage method achieving a balanced performance while maintaining good computational efficiency.

Relation extraction: We evaluated our single-stage claim extraction to determine whether it effectively balances decontextualization with semantic preservation. We compared four methods across Ref Score (ratio of explicit to total entity references), Sem. Similarity (embedding similarity between documents and extracted claims), and Claim Ret. (fraction of important claims preserved) (Table 4). Of the methods, our single-stage method extracts and decontextualizes claims in one step, compared to performing them separately (two-stage), directly extracting RDF triples (direct-triples), and extracting entities before relations (pairs-relations).

Entity-based approaches achieved higher Ref scores (direct-triples: 0.97, pairs-relations: 0.99) compared to claim-based ones (single-stage: 0.94, two-stage: 0.95) due to extracting explicit entities which naturally avoids leaving unresolved references. However, the claim-based methods achieved strong Sem. Similarity performance (0.90 and 0.90 vs 0.87 and 0.82). This advantage suggests that retaining the sentence structure of claims results in lower loss of semantic meaning. All methods achieved a perfect Claim Ret. score.

The results support our usage of the single-stage approach with its comparable Ref score and superior Sem. Similarity to the entity extraction approaches, and almost identical performance to two-stage at a fraction of the computational cost. This ensures reliable extraction that preserves semantic meaning for downstream graph creation and QA.

Approach	Summary Score Wins
Graph Communities	59.35%
Semantic Communities	40.65%

Table 5: Relevance scores between graph and semantic-based summarization. Results show percentage of times each method produced higher relevance score summaries, and demonstrate the graph community summary's superior ability to capture relevant information.

Graph construction: We evaluated whether graph-based communities contain more relevant information to the input question than semantic communities by comparing their produced summaries' relevance scores. Graph communities were formed from 1-hop neighbors around claims of interest, while semantic communities used claims with >0.8 cosine similarity to claims of interest.

Graph community summaries had higher relevance scores compared to semantic communities 59.35% of the time (Table 5). This demonstrates that while semantic communities are limited to capturing relationships based on semantic similarity, our graph construction identifies connections that may be relevant topically yet semantically dissimilar, explaining CLAIMS' advantages over semantic clustering approaches in the main QA experiments.

Approach	Faithfulness	Relevancy	Source Div.
CLAIMS	0.96	0.84	0.96
Semantic	0.97	0.86	0.92
Subgraph	0.95	0.79	0.94

Table 6: Three summarization approaches across faithfulness (hallucination), relevancy (relevance to input question), and source diversity (multi-document relations) metrics. Results demonstrate CLAIMS' ability to maintain a high faithfulness and relevancy while achieving superior source diversity.

Graph summarization: We assessed whether CLAIMS successfully integrates multi-document information while maintaining quality, comparing CLAIMS against the graph and semantic communities from the graph construction component level analysis. Their output summaries' claims were evaluated on faithfulness (hallucination rate), relevancy (relevance to the input question), and source diversity (fraction of documents integrated).

CLAIMS achieved comparable faithfulness (0.96) and relevancy scores (0.84) while having superior source diversity (0.96) (Table 6). CLAIMS had a slightly lower faithfulness score (0.96) compared to semantic communities (0.97), likely due to the multiple LLM calls in its pipeline increasing the potential for hallucination. It also had a slightly lower relevancy (0.84) than semantic clustering (0.86), reflecting the inclusion of individually irrelevant bridging information that connects relevant statements across documents. However, the relatively small differences suggest these are acceptable trade-offs for improved reasoning capabilities, and this balance between direct relevance and comprehensive multi-document integration explains CLAIMS' effectiveness across diverse QA benchmarks where multi-document reasoning is essential. The results of our evaluations are discussed in more detail in Appendix K, with additional QA accuracy ablations in Appendix H.

4.7 Human Evaluations

Human evaluation of MMLU validation graphs showed high core entity extraction (> 92%) and claims accuracy (> 91%) with substantial inter-rater agreement (>72%) (Appendix B). Case study analysis revealed that successful cases demonstrated effective discrimination between similar concepts and mechanistic pathway reasoning. Failed cases revealed two modes: sparse graph construction, and knowledge gaps in generated contexts. Detailed information provided in Appendix C and D.

5 Conclusion

We introduce CLAIMS, a novel method for retrieval-based biomedical QA tasks, targeting the key challenge of recognizing and leveraging multi-document relationships. It utilizes propositional claims to construct a local knowledge graph from retrieved documents, before constructing summaries derived via layerwise summarization from the graph. These summaries contextualize a small language model to produce the final QA decisions. Our comprehensive evaluation demonstrates CLAIMS' effectiveness across multiple dimensions. CLAIMS achieved consistent improvements over established RAG baselines (1.46-11.64% average gains) and successfully augmented domain-specific models with its produced contexts (3.27% to 8.26% improvements), demonstrating broad generalizability in enabling even a small model to effectively synthesize complex multi-document information. Entity masking experiments revealed that CLAIMS provides reasoning benefits beyond leveraging entity relationships, while component level evaluations showed the robustness of each pipeline stage. Together, these results establish CLAIMS as an effective approach for biomedical QA tasks.

6 Limitations

Denoising: Our approach currently relies on the summarization's inherent denoising ability to remove irrelevant information from the constructed graph. This was done in lieu of entirely removing irrelevant claims in an attempt to retain connections that were individually irrelevant yet important to connect relevant content together for the summaries. Future work will target methods to limit the effects of these irrelevant claims and improve detection and removal of conflicting information.

Model use: We currently only test on Mistral-7B-Instruct-v0.1 (Jiang et al., 2023) for the main model. We chose this model due to its balance of performance and computational accessibility, allowing our method to be implemented with more modest hardware requirements compared to larger models. In future work, we plan on testing on other newer, more advanced models as well as a more diverse set of retrieval datasets and evaluation benchmarks.

Claim extraction efficiency: CLAIMS uses LLM-based extraction to ensure high-quality claim decontextualization and relationship identification. Although this provides superior semantic understanding compared to rule-based approaches, it involves computational overhead proportional to document volume. Future work could explore hybrid approaches combining LLMs with more efficient preprocessing.

Data leakage: Like other RAG approaches, CLAIMS operates on retrieved documents that may overlap with evaluation data. We follow standard evaluation practices and our method's improvements over established baselines suggest genuine methodological contributions rather than data artifacts. Future work will incorporate additional data provenance tracking.

Free-form reasoning: Our evaluation focuses on multiple-choice QA, which enables standardized assessment but may not fully capture real-world biomedical reasoning complexity. While CLAIMS effectively captures cross-document relationships for structured questions, extending to free-form reasoning would require adapting our summarization approach for open-ended answer generation and developing appropriate evaluation metrics. Future work will investigate CLAIMS' performance on free-form biomedical reasoning tasks.

Source attribution: While CLAIMS effectively synthesizes information across multiple documents, the multi-level aggregation process makes it challenging to trace specific claims in the final summary back to exact source sentences. To address this, we have implemented a document reranking approach (Appendix J) that retrieves and prioritizes source documents most likely to contain supporting evidence for the generated summaries. However, this provides document-level rather than sentence-level attribution, representing an inherent trade-off of hierarchical summarization methods that prioritize comprehensive cross-document reasoning over granular source traceability.

7 Ethical Considerations

Our system, while demonstrating improved QA Accuracy on biomedical QA benchmarks, inherits the fundamental limitations of LLM-based approaches in healthcare contexts. We caution against using CLAIMS or similar systems for medical diagnosis or treatment decisions without expert oversight. The knowledge graphs constructed reflect the information and potential biases in retrieved source documents, so verification of model outputs is essential. This tool is not intended to replace clinical expertise, and implementations should include clear limitation disclaimers and verification mechanisms.

8 Code Availability

Our code is available at `https://github.com/lxguan1/local_graph_clean`.

9 Acknowledgments

The authors would like to acknowledge the support from NIH awards U24DK138515 and OT2OD038003.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*.

Ruirui Chen, Weifeng Jiang, Chengwei Qin, Ishaan Singh Rawal, Cheston Tan, Dongkyu Choi, Bo Xiong, and Bo Ai. 2024a. Llm-based multi-hop question answering with knowledge graph integration in evolving environments. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 14438–14451. Association for Computational Linguistics.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024b. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 15159–15177. Association for Computational Linguistics.

Zeming Chen, Alejandro Hern'andez Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Kopf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. 2023. Meditron-70b: Scaling medical pretraining for large language models. *ArXiv*, abs/2311.16079.

Clément Christophe, Praveen K Kanithi, Tathagata Raha, Shadab Khan, and Marco AF Pimentel. 2024. Med42-v2: A suite of clinical llms.

Donald C. Comeau, Rezarta Islamaj Dogan, Paolo Ciccarese, Kevin Bretonnel Cohen, Martin Krallinger, Florian Leitner, Zhiyong Lu, Yifan Peng, Fabio Rinaldi, Manabu Torii, Alfonso Valencia, Karin M. Verspoor, Thomas C. Wiegers, Cathy H. Wu, and John Wilbur. 2013. Bioc: a minimalist approach to interoperability for biomedical text processing. *Database: The Journal of Biological Databases and Curation*, 2013.

Gordon V. Cormack, Charles L. A. Clarke, and Stefan Büttcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, pages 758–759. ACM.

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

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. 2024. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130.

Wikimedia Foundation. Wikimedia downloads.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023a. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada. Association for Computational Linguistics.

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

Tiezheng Guo, Chen Wang, Yanyi Liu, Jiawei Tang, Pan Li, Sai Xu, Qingwen Yang, Xianlin Gao, Zhi Li, and Yingyou Wen. 2024a. Leveraging inter-chunk interactions for enhanced retrieval in large language model-based question answering. *CoRR*, abs/2408.02907.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. 2024b. Lightrag: Simple and fast retrieval-augmented generation. *CoRR*, abs/2410.05779.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. From RAG to memory: Non-parametric continual learning for large language models. *CoRR*, abs/2502.14802.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yuanhao Huang, Zhaowei Han, Xin Luo, Xuteng Luo, Yijia Gao, Meiqi Zhao, Feitong Tang, Yiqun Wang, Jiyu Chen, Chengfan Li, et al. 2024. Building a literature knowledge base towards transparent biomedical ai. *bioRxiv*, pages 2024–09.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *CoRR*, abs/2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024a. Mixtral of experts. *CoRR*, abs/2401.04088.

Boran Jiang, Yuqi Wang, Yi Luo, Dawei He, Peng Cheng, and Liangcai Gao. 2024b. Reasoning on efficient knowledge paths: Knowledge graph guides large language model for domain question answering. In *IEEE International Conference on Knowledge Graph, ICKG 2023, Shanghai, China, December 1-2, 2023*, pages 142–149. IEEE.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14).

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)*, pages 2567–2577, Hong Kong, China. Association for Computational Linguistics.

Qiao Jin, Zheng Yuan, Guangzhi Xiong, Qianlan Yu, Huaiyuan Ying, Chuanqi Tan, Mosha Chen, Songfang Huang, Xiaozhong Liu, and Sheng Yu. 2022. Biomedical question answering: A survey of approaches and challenges. *ACM Comput. Surv.*, 55(2).

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. In *Annual Meeting of the Association for Computational Linguistics*.

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

Chaofan Li, Zheng Liu, Shitao Xiao, and Yingxia Shao. 2023. Making large language models a better foundation for dense retrieval. *Preprint*, arXiv:2312.15503.

Dawei Li, Shu Yang, Zhen Tan, Jae Young Baik, Sukwon Yun, Joseph Lee, Aaron Chacko, Bojian Hou, Duy Duong-Tran, Ying Ding, Huan Liu, Li Shen, and Tianlong Chen. 2024a. DALK: dynamic co-augmentation of llms and KG to answer alzheimer's disease questions with scientific literature. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 2187–2205. Association for Computational Linguistics.

Mingchen Li, Halil Kilicoglu, Hualei Xu, and Rui Zhang. 2024b. Biomedrag: A retrieval augmented large language model for biomedicine. *Journal of biomedical informatics*, page 104769.

Zijian Li, Qingyan Guo, Jiawei Shao, Lei Song, Jiang Bian, Jun Zhang, and Rui Wang. 2025. Graph neural network enhanced retrieval for question answering of large language models. 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 6612–6633, Albuquerque, New Mexico. Association for Computational Linguistics.

Lei Liu, Xiaoyan Yang, Junchi Lei, Xiaoyang Liu, Yue Shen, Zhiqiang Zhang, Peng Wei, Jinjie Gu, Zhixuan Chu, Zhan Qin, and Kui Ren. 2024. A survey on medical large language models: Technology, application, trustworthiness, and future directions. *ArXiv*, abs/2406.03712.

Linhao Luo, Yuan-Fang Li, Gholamreza Haffari, and Shirui Pan. 2024. Reasoning on graphs: Faithful and interpretable large language model reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Shengjie Ma, Chengjin Xu, Xuhui Jiang, Muzhi Li, Huaren Qu, Cehao Yang, Jiaxin Mao, and Jian Guo. 2025. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of* *the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Belinda Mo, Kyssen Yu, Joshua Kazdan, Proud Mpala, Lisa Yu, Chris Cundy, Charilaos I. Kanatsoulis, and Oluwasanmi Koyejo. 2025. Kggen: Extracting knowledge graphs from plain text with language models. *ArXiv*, abs/2502.09956.

Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy. Association for Computational Linguistics.

Pengcheng Qiu, Chaoyi Wu, Xiaoman Zhang, Weixiong Lin, Haicheng Wang, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2024. Towards building multilingual language model for medicine. *Preprint*, arXiv:2402.13963.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. 2024. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *CoRR*, abs/2410.14052.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance, ICAIF 2024, Brooklyn, NY, USA, November 14-17, 2024*, pages 608–616. ACM.

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

Robert R Sokal and Charles D Michener. 1958. A statistical method for evaluating systematic relationships.

Yimin Tang, Yurong Xu, Ning Yan, and Masood S. Mortazavi. 2024. Enhancing long context performance in llms through inner loop query mechanism. *CoRR*, abs/2410.12859.

Yu Wang, Nedim Lipka, Ryan A. Rossi, Alexa F. Siu, Ruiyi Zhang, and Tyler Derr. 2024. Knowledge graph prompting for multi-document question answering. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19206–19214. AAAI Press.

Junde Wu, Jiayuan Zhu, Yunli Qi, Jingkun Chen, Min Xu, Filippo Menolascina, Yueming Jin, and Vicente Grau. 2025. Medical graph RAG: Evidence-based medical large language model via graph retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28443–28467, Vienna, Austria. Association for Computational Linguistics.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. 2024. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 6233–6251. Association for Computational Linguistics.

Rui Yang, Boming Yang, Xinjie Zhao, Fan Gao, Aosong Feng, Sixun Ouyang, Moritz Blum, Tianwei She, Yuang Jiang, Freddy Lecue, Jinghui Lu, and Irene Li. 2025. Graphusion: A rag framework for scientific knowledge graph construction with a global perspective. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 2579–2588, New York, NY, USA. Association for Computing Machinery.

Han Yu, Peikun Guo, and Akane Sano. 2023. Zero-shot ecg diagnosis with large language models and retrieval-augmented generation. In *Proceedings of the 3rd Machine Learning for Health Symposium*, volume 225 of *Proceedings of Machine Learning Research*, pages 650–663. PMLR.

Nan Zhang, Prafulla Kumar Choubey, Alexander R. Fabbri, Gabriel Bernadett-Shapiro, Rui Zhang, Prasenjit Mitra, Caiming Xiong, and Chien-Sheng Wu. 2025a. Sirerag: Indexing similar and related information for multihop reasoning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Zhi Zhang, Yan Liu, Sheng-hua Zhong, Gong Chen, Yu Yang, and Jiannong Cao. 2025b. Mixture of knowledge minigraph agents for literature review generation. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 26012–26020. AAAI Press.

Hongjian Zhou, Boyang Gu, Xinyu Zou, Yiru Li, Sam S. Chen, Peilin Zhou, Junling Liu, Yining Hua, Chengfeng Mao, Xian Wu, Zheng Li, and Fenglin Liu. 2023. A survey of large language models in medicine: Progress, application, and challenge. *ArXiv*, abs/2311.05112.

Appendix

A Methods Details

A.1 RAG retrieval

The retrieval corpora include Simple Wikipedia (CC-BY-SA) (Foundation), medical textbooks from MedQA (MIT License) (Jin et al., 2021), PubMed abstracts and fulltext articles taken from GLKB (CC BY-NC-ND 4.0) (Huang et al., 2024), and StatPearls articles[2] (CC BY-NC-ND 4.0). Simple Wikipedia provides accessible explanations of basic scientific concepts using simplified language to aid in cross-document entity linking, medical textbooks provide authoritative structured academic knowledge with standardized medical terminology and established medical principles, StatPearls documents provide comprehensive clinically-focused reference information bridging academic knowledge with practical medical application, and PubMed abstracts/fulltext articles capture the latest research findings and evidence-based clinical data. This combination ensures comprehensive coverage across different knowledge domains and complexity levels, inspired by MedRAG (Xiong et al., 2024). For Simple Wikipedia and medical textbooks, we chunk them into chunks of 1000 tokens via LlamaIndex's SentenceSplitter, with 200 token overlaps. PubMedCentral full text articles are chunked using semantic chunking sentence by sentence with a breakpoint threshold of 0.95 to ensure we have relevant chunks. For StatPearls, we use MedRAG's scripts to chunk them hierarchically.

We retrieve from each corpus with a variety of methods. From Simple Wikipedia, we use the BM25 Retriever (Robertson and Zaragoza, 2009) to retrieve relevant articles due to the size of the corpora and the retrieval process's speed. From the medical textbooks and StatPearls, we use both BM25 and dense vector retrieval to include semantic meanings that might be missed from pure BM25 retrieval. Reciprocal Rank Fusion (Cormack et al., 2009) is used to combine the results of the two retrieval methods. We use LlamaIndex's implementations of BM25 Retriever and Vector Index Retriever to implement these retrieval processes. GLKB retrieval of abstracts is conducted via dense vector retrieval through the GLKB API (Huang et al., 2024). For each of the retrieved abstracts, we consider their pubid. If these articles are part of the PubMedCentral corpus, we extract and chunk their fulltext articles via the BioC API (Comeau et al., 2013). We retrieve 3 documents from each of these retrieval sources. Before we do further processing, we first perform an additional chunking of all inputs to be within 1024 tokens via the LlamaIndex SentenceSplitter to ensure model context window limits are not exceeded, as well as remove special characters to ensure smooth handling of the texts. All data was used in accordance with their license agreements.

[2] https://www.statpearls.com/

```
Extract ALL claims from this medical
    text as independent statements. For
    each claim:

1. Make it atomic - break apart any
    grouped findings/measurements into
    individual claims, even if they were
     presented together in the text (e.g
    ., if text says 'measurements A, B
    and C showed improvement', create
    three separate claims)
2. Make it standalone by including ALL
    necessary context within each claim:

- Study type and time period
- Population characteristics and sample
    size
- Study setting
- Relevant conditions
- Statistical Significance if noted

3. Structure each claim as a complete
    sentence that:

- Avoids phrases like 'the study found'
    or 'results showed'
- Includes full technical terms with
    abbreviations
- Could be understood without any other
    context
- Contains all qualifying information

Example transformations:
BAD claims (missing context, ambiguous
    source, or incomplete):
CLAIM: Blood pressure and heart rate
    improved.
CLAIM: The study found improvements in
    vital signs.
CLAIM: 30% of patients showed positive
    outcomes.
CLAIM: This randomized trial
    demonstrated efficacy.
CLAIM: In this study, BMI decreased
    significantly.
CLAIM: The present analysis showed
    improved outcomes..

GOOD claims:
CLAIM: The 2010-2015 Mayo Clinic
    randomized controlled trial of 100
    hypertensive patients aged 45-65
    demonstrated systolic blood pressure
     decreases of 15mmHg (95% CI: 10-20
    mmHg, p<0.001) after 6 weeks of
    treatment.
CLAIM: The 2010-2015 Mayo Clinic
    randomized controlled trial of 100
    hypertensive patients aged 45-65
    showed resting heart rate decreases
    of 8 bpm (95% CI: 5-11 bpm, p<0.001)
     after 6 weeks of treatment.
CLAIM: The 2018-2020 Cleveland Clinic
```

```
    prospective cohort study of 250
    diabetic patients aged 30-50
    demonstrated hemoglobin A1C level
    decreases of 1.2% (95% CI: 0.8-1.6%,
     p<0.01) in the intervention group
    receiving intensive lifestyle
    modification.
CLAIM: The 2015-2017 Johns Hopkins
    Hospital double-blind placebo-
    controlled trial of 180 arthritis
    patients aged 50-75 showed morning
    stiffness duration decreases of 45
    minutes (95% CI: 30-60 minutes, p
    <0.005) in patients receiving the
    experimental treatment.
CLAIM: The 2012-2014 Stanford Medical
    Center retrospective analysis of 300
     obesity clinic patients aged 18-40
    demonstrated body mass index
    decreases of 2.5 kg/m   (95% CI:
    1.8-3.2 kg/m  , p<0.001) after 12
    months of structured weight
    management.

Format each claim starting with 'CLAIM:'
     on a new line. Include every
    finding mentioned in the text, no
    matter how minor.

Text:   {text}
```

Prompt 1: Claim Extraction Prompt

```
Given these existing claims, find any
    ADDITIONAL claims from the text that
     weren't already captured.
Do NOT modify or restate the existing
    claims - only add new ones.
If all claims have already been captured
    , respond with 'NO_ADDITIONAL_CLAIMS
    '.

Each new claim must be self-contained
    and decontextualized with:
- All relevant entities and background
    information
- Study conditions, populations, and
    timeframes
- Statistical significance where
    mentioned
- All context needed for independent
    understanding
- Clear, single statements (not
    paragraphs)
- Be a standalone, self-contained
    statement that does not reference or
     depend on any other claims, the
    original text, or any external
    context

Existing claims:
{claims}

Text: {text}

List only NEW claims, starting each with
     'CLAIM:' (or respond with '
    NO_ADDITIONAL_CLAIMS')
```

Prompt 2: Claim Extraction Verification Prompt

A.2 Claim extraction prompts

For claim extraction (Section 3.1), we do the process in two gleanings. The first one can be seen with Prompt 1. The second one takes the extracted claims from the first pass, and asks the model to extract claims it missed from the documents as shown in Prompt 2. This is to ensure that we don't miss any important information while keeping efficiency at a reasonable level. We deduplicate all of the extracted claims in each document to prevent repeats from occurring.

```
Consider the following question and
    answer options. Choose the correct
    response and explain your decision.
Question: {question}
Answer Options: {answer_options}
Answer:
```

Prompt 3: HyDE Candidate Answer Prompt

HyDE queries: In HyDE query generation (Section 3.1), as the answer options are multiple choice for the benchmarks we are considering, we prompt the model to generate an accompanying explanation for the selected answer choice. This ensures we are taking advantage of the parametric knowledge inside of the model using this explanation to find associated documents, and are not stuck with only a simple multiple choice selection in the HyDE query. The prompt for creating the accompanying explanation can be seen in Prompt 3.

```
Question: {question}
Main Claim: {claim}
Related Claims from Local Community:
{unique_contexts}

Please provide a comprehensive analysis
    of how the main claim relates to the
     question, considering the context
    from related claims.
```

Prompt 4: Claim of Interest Prompt

Claim of Interest prompts: In the claims of interest summarization prompts (Section 3.3), we emphasize the central claim of interest when contextualizing it with the surrounding contexts. This is to ensure that the central claim is not overwhelmed by the surrounding contexts. The output of this procedure is a test summary that is used to rerank the claims of interest. This can be seen in Prompt 4.

```
You are tasked with enriching and
    contextualizing claims using related
     information. Your goal is to create
     a comprehensive summary that:
1. Preserves ALL important information
    from the original claims
2. Integrates relevant context from
    related claims
3. Makes implicit relationships explicit
4. Filters out redundant or irrelevant
    information

The following summaries provide relevant
     context. Each represents a claim
    that leads to or supports the above
    claims:
{context_summaries}

The claims to contextualize are:
{claims}

Produce a summary that:
- MUST preserve the complete meaning and
     all key details of the original
    claims
- Incorporate relevant context that
    helps understand or validate the
    claims
- Make implicit connections explicit (e.
    g., if context suggests a cause-
    effect relationship not directly
    stated)
- Filter out redundant or tangential
    information from the context
- Use clear, precise language
- Maintain factual accuracy without
    speculation

Focus on enriching the claims while
    ensuring NO important information is
     lost. When in doubt, include
    information rather than exclude it.

Summary:
```

Prompt 5: Layerwise Summarization Summary Prompt

Layerwise summarization prompts: In the layerwise summarization prompts (Section 3.3), we emphasize several key points. These include preserving all important medical knowledge, integrating information together to capture multi-hop relations, capturing implicit relationships that are not explicitly mentioned, and filtering out redundant or irrelevant information. To ensure that information important for multi document relations are retained even when they are not apparent, we ask in the prompt to preserve information if possible, as long as it does not conflict with the removal of noise. This can be seen in Prompt 5.

```
The following will be several examples
    of claims, and their extraction into
     subject - predicate - object
    triples.
Extract only the single most important
    relationship from each claim. For
    research results, focus on the main
    finding. For factual claims, focus
    on the central relationship.

Claim: A correlation exists between
    histologic chorioamnionitis and the
    usage of antibiotics.
SUBJECT: histologic chorioamnionitis
PREDICATE: correlation
OBJECT: usage of antibiotics

Claim: Early cast-related complaints
    predicted the development of complex
     regional pain syndrome.
SUBJECT: early cast-related complaints
PREDICATE: predict
OBJECT: development of complex regional
    pain syndrome

Given the following claim, identify the
    single most important relationship.
List exactly one triple using "SUBJECT",
     "PREDICATE", and "OBJECT" on
    separate lines.
All fields must contain content from the
     claim.
Claim:
```

Prompt 6: RDF Triple Extraction Prompt

A.3 Triple extraction fallbacks

For RDF triple extraction (Section 3.1), we begin with Prompt 6. Occasionally, the model has the tendency to leave an entity field or the relation field empty when extracting RDF triples from the propositional claims. In those cases, we have several fallbacks which we sequentially attempt when the previous one fails.

Triple extraction fallback: The first is to provide the previous faulty output of the RDF triple extraction to the model, mention that there is a missing/malformed output, and prompting the model to provide the correctly formatted output.

Entity extraction fallback: The second is to fall back to extracting two key entities and the relation, with one prompt extracting the two entities. The first two listed entities are used if there are more than two entities in the outputs. The relation between entities is extracted with another prompt.

SpaCy extraction fallback: If this still fails due to malformed outputs, we use SciSpaCy (Neumann et al., 2019) to extract two entities from the claim, and use the "associated" relation to describe their relation.

A.4 Deduplication of numerical entities

Due to the free-form entity extraction process (Section 3.2), sometimes numerical items are used as entity nodes. We have empirically found that the embeddings of these numerical items can receive high semantic similarities between each other, resulting in nodes being placed in the same cluster that are completely unrelated from our entity deduplication. To combat this special case, we check the contents of each entity node, and if over half of the characters are numeric, we treat them as numeric nodes and don't allow them to be placed in other clusters.

In addition, we don't use character-based Levenshtein distance because medical entities that have only minor character differences can have entirely different meanings.

```
Contexts: {context_claims}
Question: {question}
Answer Options: {answer_choices}
```

Prompt 7: Model Generation Prompt

A.5 Summary generation

Throughout our layerwise summarization method (Section 3.3), we need to ensure that combining summaries does not exceed the model's context window. When the combined tokenization length of the connected summaries exceeds a predefined token limit (2k tokens for our testing), semantic clustering based compression is used to cut down on the size while preserving key information. After first determining a rough number of clusters from the total length of the input summaries, summaries are placed into the same cluster using KMeans with their individual embeddings. Each cluster is summarized, and if the combined resulting clusters are still too long, they are recursively summarized. The final summaries of the resulting clusters are returned to continue the layerwise summarization.

The layerwise summarization process is used because it has three key benefits. First, it is capable of capturing all the information in the local connected component, including both the direct content and path-based information. This is important for understanding multi-document relations between different medical concepts. Second, our layerwise processing of claims will inherently filter out irrelevant content. Finally, this method places emphasis on claims closer in G to the claims of interest, which naturally prioritizes more topically relevant information in the final summaries. The summaries produced by layerwise summarization are concatenated together and fed to the LLM via Prompt 7.

Component	Time Taken (seconds)
Claim Extraction	121.75
Triple Extraction	116.82
Graph Construction	5.14
Graph Summarization	39.98
Final Generation	1.62

Table 7: Average time taken for each step in CLAIMS over all of the questions in all evaluated QA benchmarks.

Method	Average Time (seconds)
MedRAG	8.45
KGP	46.39
RAPTOR	66.00
CLAIMS (Sequential)	285.29
CLAIMS (Parallelized)	80.55
SiReRAG	1139.42

Table 8: Average inference time per question for different methods.

A.6 Computational efficiency

We recorded the overall efficiency of our approach, for each of the claim extraction, triple extraction, graph construction, layerwise summarization, and final generation steps of CLAIMS. The average times needed for each of the components over every question in each of the benchmarks is shown in Table 7. The most computationally expensive part of CLAIMS is triple extraction, while the least expensive is the final generation.

Comparison with baselines: We compared the computational efficiency of CLAIMS against baseline approaches. Table 8 shows the average inference time per question across all benchmarks. While CLAIMS requires more computation time than simpler retrieval methods like MedRAG (8.45 seconds) and KGP (46.39 seconds), it is comparable to RAPTOR (66.0 seconds) and provides substantially better accuracy on multi-document reasoning tasks. SiReRAG requires significantly more computation time (1139.42 seconds). This overhead stems from SiReRAG's dependence on strictly formatted outputs at multiple pipeline stages. We used the same non-fine-tuned Mistral-7B-Instruct-v0.1 model (Jiang et al., 2023) for all methods to ensure fair comparison. Even when using tools like lm-format-enforcer[3] to constrain JSON generation, smaller non-finetuned models

[3] https://github.com/noamgat/lm-format-enforcer

frequently produce improperly formatted or truncated outputs that necessitate re-runs. These formatting failures compound across SiReRAG's pipeline, substantially increasing total inference time. While fine-tuned models specifically trained for structured output generation might improve SiReRAG's efficiency, this reliance on specialized models reduces practical accessibility compared to methods that work reliably with off-the-shelf models.

CLAIMS parallelization: To improve CLAIMS' computational efficiency in practice, one approach is to parallelize the processing of individual documents during the relation extraction phase. While we currently lack the infrastructure to implement full parallelization, we approximated its potential impact by identifying the longest processing time for each step across all documents for each question in the MMLU validation dataset. The results show that when processing documents in parallel, the bottleneck would be determined by the slowest documents: 9.17 seconds for the first pass of claim extraction, 8.32 seconds for the second pass of claim verification, and 16.32 seconds for triple extraction. Compared to the sequential processing times in Table 7 (121.75 seconds for claim extraction and 116.82 seconds for triple extraction), parallelization could provide substantial speedup, reducing the total time to approximately 80.55 seconds. This would make CLAIMS comparable to RAPTOR while maintaining superior performance on multi-document reasoning tasks. This represents a promising direction for improving CLAIMS' practical deployment. Additionally, we have implemented continuous batching using vLLM, which on the MMLU validation dataset reduced sequential processing time from 245.31 to 118.69 seconds (a greater than 2× speedup), and we are actively exploring how these approaches can be combined.

A.7 Hyperparameter selection

We initially selected the deduplication threshold of 0.8 and 10 claims of interest based on small-scale pilot experiments during method development. To validate these choices, we conducted comprehensive sensitivity analysis on the full MMLU validation set, testing deduplication thresholds of 0.5, 0.6, 0.7, 0.8, and 0.9 (with 10 claims of interest held constant), as well as 5 and 15 claims of interest (with 0.8 threshold) with their average scores over 5 rounds of answer shuffling taken (Table 9). On the validation set, only 0.7 threshold achieved marginally better performance (0.71 vs 0.70 for 0.8). However, when evaluated across all benchmarks, 0.8 performed better (Table 10), suggesting 0.7 may have overfit to the validation set characteristics. All other configurations showed minimal performance variation, confirming the method's robustness to these hyperparameters. We attribute this robustness to our filtering of connected claims (Section 3.3), which naturally constrains the final claim set regardless of initial selection parameters.

Config	Accuracy
0.5 similarity threshold	0.67
0.6 similarity threshold	0.69
0.7 similarity threshold	0.71
0.8 similarity threshold	0.70
0.9 similarity threshold	0.68
5 claims of interest	0.67
15 claims of interest	0.70

Table 9: Performance of different hyperparameter configurations on MMLU validation. Similarity thresholds are thresholds for deduplication, and number of claims are the number of claims we consider for claims of interest before claim filtering. The similarity threshold configurations all use 10 initial claims of interest, and the number of claims configurations all use a threshold of 0.8.

A.8 Retrieval Source Ablation

To better understand the impact of each retrieval source on CLAIMS performance, we conducted an ablation study testing different combinations of our retrieval corpora. This analysis ensures that each source contributes positively to overall performance rather than introducing noise or redundancy.

We evaluated CLAIMS on the MMLU validation dataset using five progressive configurations: (1) no retrieval sources (baseline), (2) Simple Wikipedia only, (3) Wikipedia + medical textbooks, (4) Wikipedia + medical textbooks + StatPearls, and (5) all sources including PubMed abstracts and fulltext articles. The ordering progresses from general to domain-specific knowledge, with Wikipedia providing general domain information, medical textbooks offering foundational medical knowledge, StatPearls contributing specific clinical information, and PubMed articles providing recent research findings. Each configuration was run 5 times with answer option shuffling to control for positional bias using the Mistral-7B-Instruct-v0.1 model.

The results demonstrate that all retrieval sources contribute positively to performance (Table 11). Both medical textbooks and PubMed abstracts/full-

Similarity Threshold	MMLU-A*	MMLU-CM	MMLU-CB	MMLU-PM	MMLU-MG	MMLU-CK	MMLU-V	PMQA	MedQA	Overall SE
0.7	**0.63**	0.58	0.63	0.62	0.71	0.65	**0.71**	0.60	0.51	0.0033
0.8	0.62	**0.60**	**0.64**	**0.64**	**0.74**	**0.66**	0.70	**0.61**	**0.52**	0.0021

*MMLU prefixes denote: V-Validation, A-Anatomy, CB-College Biology, CM-College Medicine, PM-Professional Medicine, MG-Medical Genetics, CK-Clinical Knowledge

Table 10: Comparison of accuracy scores between 0.7 and 0.8 deduplication similarity threshold over 5 runs. The 0.8 threshold achieved comparable or higher scores on all datasets. The MMLU prefixes denote different subject areas, as noted under the table.

Configuration	Average Acc	Overall SE
Baseline	0.54	0.0129
W	0.57	0.0095
W + T	0.63	0.0093
W + T + S	0.64	0.0075
W + T + S + P	0.70	0.0107

Table 11: Retrieval Source Ablation with different retrieval corpora combinations on the MMLU validation dataset. Overall SE represents the standard error of overall performance over 5 runs. W=Wikipedia, T=Medical Textbooks, S=StatPearls, P=PubMed Abstracts/Fulltext Articles.

text articles provided substantial 6% improvements, though at different stages of the ablation. Medical textbooks increased performance from 57% to 63% when added after Wikipedia, while PubMed abstracts and fulltext articles later increased performance from 64% to 70% as the final addition. The addition of Simple Wikipedia alone improved accuracy by 3% over the no-retrieval baseline (from 54% to 57%), while StatPearls provided a 1% improvement (from 63% to 64%). This progressive improvement pattern validates our multi-source retrieval strategy and confirms that the breadth of knowledge sources enhances CLAIMS' ability to answer complex biomedical questions.

B Human Evaluations

Metric	**Rater 1 Mean**	**Rater 2 Mean**	**Agreement ($\leq$ 0.10)**
Graph Comp	0.95	0.92	74%
Claim Acc	0.91	0.95	72%

Table 12: Inter-rater reliability and performance metrics for human evaluation of CLAIMS outputs on MMLU validation set graphs.
Note: Agreement reported as percent of graphs where raters differed by $\leq$ 0.10.

To evaluate CLAIMS' performance, we conducted a two-phase human evaluation. First, we performed a systematic quantitative assessment of 123 graphs from the MMLU validation dataset, measuring Graph Comprehensiveness and Claim Accuracy across all samples. This large-scale evaluation established the overall reliability and quality of the system's outputs. Second, we conducted detailed qualitative case studies on selected indices to examine the system's reasoning process, error modes, and specific strengths in handling complex biomedical queries (Appendix D). Together, these complementary approaches provide both breadth of evaluation coverage and depth of understanding of CLAIMS' capabilities.

Human evaluations were conducted by a CS PhD student and a Bioinformatics postdoctoral researcher who independently assessed 123 graphs from the MMLU validation dataset across two metrics.

Graph Comprehensiveness evaluated whether entity extraction captured the core concepts needed for the final summaries. Raters first examined output summaries to identify central concept entities, then determined how many of these concepts were represented in the entities composing the subgraph around claims of interest.

Claim Accuracy assessed the faithfulness of extracted claims. Raters examined each claim in the subgraph and its source document, determining whether the claim's information could plausibly have been derived from the source.

Inter-rater reliability between human raters was 74% agreement (within 0.10) for Graph Comprehensiveness and 72% for Claim Accuracy (Table 12). Both metrics showed high mean scores (Comprehensiveness: R1 M=0.95, R2 M=0.92; Accuracy: R1 M=0.91, R2 M=0.95), indicating that CLAIMS successfully extracted comprehensive entities and accurate claims from biomedical documents. The high scores across both metrics demonstrate the system's ability to perform reliable knowledge graph construction for biomedical QA tasks.

C Failure Mode Analysis

To understand what distinguishes successful from failed reasoning, we analyzed both structural (graph-theoretic) and content-based features across correct and incorrect outputs. Graph topology metrics of the subgraphs around claims of interest, including node count, edge count, density, degree distribution, clustering coefficient, diameter, and radius, exhibited minimal predictive power, with only the number of disconnected components showing weak significance (p=0.050, Cohen's d=-0.34). In contrast, a content-based semantic scoring metric (reranker relevance scores of concatenated summaries) demonstrated statistically significant differentiation between correct and incorrect answers (Mann-Whitney U test: p=0.0097, Cohen's d=0.46), with correct outputs scoring substantially higher (mean: 8.46 vs 7.48). This small-to-medium effect size contrasts sharply with topological features, showing stronger significance and 35% larger effect size than the sole marginally significant structural metric. These results indicate that reasoning correctness is primarily determined by semantic content quality rather than graph structure, supporting the two failure modes found in our case studies (Appendix D): (1) sparse or disconnected graphs that fail to retrieve key entities, preventing relevant content from entering the graph, and (2) semantic gaps where retrieved information lacks critical details needed for accurate reasoning, despite adequate graph connectivity.

D Case Studies

To provide deeper insights into CLAIMS' behavior, we performed several case studies of representative correct and incorrect answers. We analyzed the constructed graph and the CLAIMS generated contexts to better understand their contribution to the model output. All examples are taken from questions in the MMLU validation dataset without answer shuffling.

D.1 Correct example 1

Question: In all-out exercise such as sprinting the first fibre types to fatigue are the:
Answer Choices:
(A) Type I fibres.
(B) Type Ia fibres.
(C) Type IIa fibres.
(D) Type IIX fibres.
CLAIMS Answer: (D) Type IIx fibres.

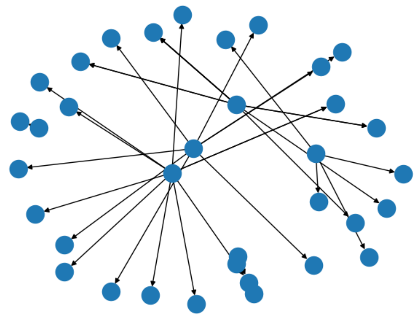

Figure 4: Graph structure of Correct example 1, focused on the subgraphs around the claims of interest. Nodes are entities, edges are claims extracted from retrieved documents that are related to the two entities.

Challenge: Distinguishing between multiple fiber types (I, Ia, IIa, IIx).

Due to the lengths of the summaries, we present relevant snippets.
Key information in CLAIMS generated contexts:

- Type IIx fibers rapid fatigue: "...Type IIx fibers are recruited for high-intensity, short-duration exercises such as full-effort sprints. These fibers rapidly fatigue due to their low oxidative capacity and high myosin ATPase activity, which makes them more prone to fatigue..."
- Type I fibers are slow-twitch, fatigue-resistant: "...Type I fibers are slow-twitch, fatigue-resistant motor units that generate less tension than other fibers. These fibers are found in high-endurance athletes such as marathon runners, where they are responsible for generating endurance and are characterized by their ability to contract slowly and continuously over long periods of time. The high percentage of Type I fibers in the muscles of these athletes is what allows them to sustain their performance for extended periods..."
- Type IIb fibers fatigue rapidly: "... type IIb fibers have a low level of oxidative enzymes but exhibit high anaerobic enzyme activity and store a considerable amount of glycogen. As a result, they fatigue rapidly as a result of production of lactic acid..."

CLAIMS Success: Successfully identified Type IIx as specifically associated with sprinting context, avoiding confusion with Type IIb (also rapidly fatiguing) and correctly rejecting Type I (fatigue-resistant). Although Type IIa was not mentioned, it allows the LLM to make an informed decision on the remaining information. A visualization of the graph is available in Figure 4.

D.2 Correct example 2

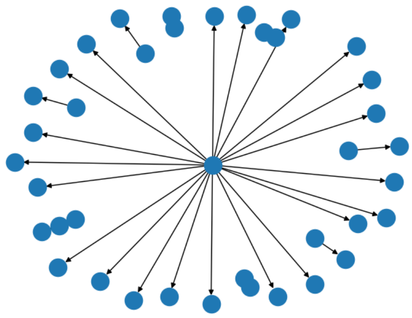

Figure 5: Graph structure of Correct example 2, focused on the subgraphs around the claims of interest. Nodes are entities, edges are claims extracted from retrieved documents that are related to the two entities.

Question: During exercise, adrenaline secretion from the adrenal glands is stimulated by:
Answer Choices:
(A) increased plasma glucose.
(B) increased plasma fatty acids.
(C) increased plasma ACTH.
(D) increased sympathetic nerve activity.
CLAIMS Answer: (D) increased sympathetic nerve activity.

Challenge: Identifying the direct mechanism that stimulates adrenaline secretion during exercise.

Due to the lengths of the summaries, we present relevant snippets.
Key information in CLAIMS generated contexts:

- Direct regulation mechanism: "...The secretion of epinephrine and norepinephrine from the adrenal medulla is regulated primarily by descending sympathetic signals in response to various forms of stress, including exercise, hypoglycemia, and hemorrhagic hypovolemia...."
- Chemical pathway: "...The chemical signal for secretion of catecholamine from the adrenal medulla is acetylcholine (ACh), which is secreted from preganglionic sympathetic neurons and binds to nicotinic receptors on chromaffin cells..."
- Exercise-specific connection: "...Sweating responds to emotional state, leading to an increase in sympathetic nerve activity and epinephrine secretion from the adrenal gland..."

CLAIMS Success: Successfully identified the mechanistic pathway where sympathetic nerve activity directly stimulates adrenaline secretion during exercise. The contexts provided both the general mechanism (sympathetic signals regulate adrenaline) and the specific chemical pathway (ACh from sympathetic neurons), allowing the LLM to correctly choose sympathetic nerve activity over the other choices. A visualization of the graph is available in Figure 5.

D.3 Incorrect example 1

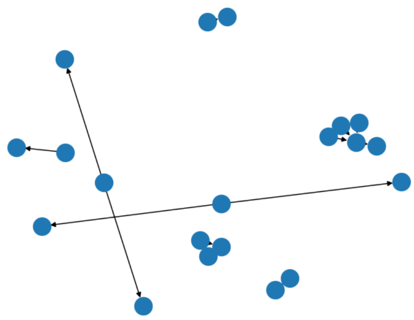

Figure 6: Graph structure of Incorrect example 1, focused on the subgraphs around the claims of interest. Nodes are entities, edges are claims extracted from retrieved documents that are related to the two entities.

Question: Which of the following physiological characteristics is not important for success in endurance events such as the marathon race?
Answer Choices:
(A) The ability to regulate body temperature.
(B) The ability to supply oxygen to the exercising muscles.
(C) The availability of body stores of carbohydrate.
(D) Muscle ATP and phosphocreatine content.
CLAIMS Answer: (A) The ability to regulate body temperature.
Correct Answer: (D) Muscle ATP and phosphocreatine content.

Challenge: Identifying which factor is NOT important for marathon success.

Due to the lengths of the summaries, we present relevant snippets.
Key information in CLAIMS generated contexts:

- Temperature regulation importance: "...Thermoregulatory efficiency was also found to play an important role in determining success in marathon running..."
- Oxygen supply importance: "...The oxygen cost of running (VO2 15) and fractional utilization of VO2 max at marathon race pace (%VO2 Ma X VO2 max-1) are additional factors that affect marathon performance..."
- Fuel/Carbohydrate importance: "...The choice of fuels for the exercising muscles is related to the %VO2 Ma X VO2 max-1, which is a major limiting factor to marathon performance. Therefore, optimizing fuel intake can significantly improve marathon performance..."
- No information about muscle ATP or phosphocreatine content.

CLAIMS Failure: The sparse graph construction prevented retrieval of information about muscle ATP and phosphocreatine content, leaving this option unaddressed in the final contexts. While the contexts did clearly establish that temperature regulation, oxygen supply, and carbohydrate stores are important for marathon success, the absence of information about ATP/phosphocreatine made it difficult for the model to properly evaluate option (D). A denser constructed graph with better entity connections during graph construction and deduplication might have better captured relevant information about ATP or phosphocreatine's limited role in endurance metabolism.

Potential improvements: Enhanced entity linking algorithms and more flexible similarity thresholds during deduplication could improve graph connectivity. Additionally, incorporating domain-specific entity recognition might capture more relevant connections between concepts. A visualization of the sparse graph is available in Figure 6.

D.4 Incorrect example 2

Question: Mr Wood has just returned from surgery and has severe internal bleeding. Which of the following observations would you NOT expect to find on undertaking post-operative observations?
Answer Choices:
(A) Hypotension.
(B) Bradycardia.
(C) Confusion.

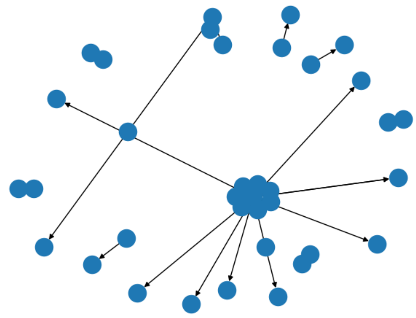

Figure 7: Graph structure of Incorrect example 2, focused on the subgraphs around the claims of interest. Nodes are entities, edges are claims extracted from retrieved documents that are related to the two entities.

(D) Tachypnoea.
CLAIMS Answer: (C) Confusion.
Correct Answer: (B) Bradycardia.

Challenge: Identifying which observation would NOT occur during severe internal bleeding.

Due to the lengths of the summaries, we present relevant snippets.
Key information in CLAIMS generated contexts:

- Hypotension mentioned: "...persistent hypotension despite restoration of intravascular volume necessitating vasopressor support..."
- General hemodynamic deterioration: "...Hemodynamic deterioration generally denotes ongoing bleeding for which some form of intervention (i.e., operation or interven-tional radiology) is required..."
- Cognitive effects suggested: "...inability to retain a span of digits... suggests that they may have cognitive impairments or other neurological issues that are contributing to their condition..."
- No specific information about heart rate responses (bradycardia) or breathing patterns (tachypnoea) in bleeding patients.

CLAIMS Failure: The contexts lacked specific information about cardiovascular responses to severe bleeding, particularly information about bradycardia, indicating a retrieval failure. While hypotension was mentioned and cognitive effects were suggested, the absence of detailed information about heart rate/respiratory responses prevented CLAIMS from identifying that bradycardia would not be expected.

Potential improvements: Expanding retrieval corpora to include more comprehensive medical texts could provide better coverage of expected physiological responses. Alternative query formulation strategies that specifically target clinical signs and symptoms might retrieve more relevant detailed information. A visualization of the graph is available in Figure 7.

```
You are an evaluation machine. Look at
    the following answer without
    considering the given explanation: [
    BEGIN PROVIDED ANSWER] {
    provided_answer} [END PROVIDED
    ANSWER] Looking at the answer, was
    the FINAL answer it gave {
    answer_choices}? Only give the final
     answer the answer explicitly
    returned in the provided answer text
    , do not do any additional reasoning
    . That is, at the very end of the
    answer text it should have
    explicitly mentioned that its final
    answer was one of the options {
    answer_choices}. Return that answer,
     and ignore all of the caveats the
    answer mentioned. Do not reason
    about the answer, simply return what
     the model explicitly put as its
    final answer. The answer should be
    in a json object, with only the
    letter corresponding to the answer
    under the key "answer", so if the
    answer as (A) the output should be
    {"answer" : "a"}
```

Prompt 8: Evaluation Output Extraction prompt

E Model Settings

We use the Mistral-7B-Instruct-v0.1 model (Apache 2.0) for both construction and summarization of the graph for all evaluations (Jiang et al., 2023), and run it without sampling. For experiments that involved LLM-as-a-judge capabilities, we used Mixtral-8x7B-Instruct-v0.1 (apache 2.0) (Jiang et al., 2024a). For the domain-specific model augmentation experiments, we used BioMistral-7B (apache 2.0) (Labrak et al., 2024), Meditron3-8B (Llama 3.1 License) [4], Llama3-Med42-8B (Llama 3 License) (Christophe et al., 2024), and MMed-Llama-3-8B (Llama 3 License) (Qiu et al., 2024). For the RAG methods experiments, we compared RAPTOR[5], SiReRAG[6], MedRAG[7], and KGP[8] with our approach. For Entity Masking, we use Llama-3.3-70B-Instruct (Llama 3.3 Community License Agreement) (Dubey et al., 2024). We applied each model's provided chat templates where available; otherwise we used jinja templates from MedRAG (Xiong et al., 2024). For Reranking, we used bge-reranker-v2-gemma (apache 2.0), and for embedding we used bge-large-en-v1.5 (MIT License) (Li et al., 2023). We use the en_core_sci_scibert spacy model (apache 2.0) (Neumann et al., 2019) due to its better performance on scientific tasks compared to general domain spacy models, and the neural entity recognition pipeline to extract entities. For answer extraction, we used lm-format-enforcer (MIT License) and Prompt 8. We run experiments on NVIDIA L40S and A40 GPUs, and H100s when possible. All of the experiments and benchmarks took approximately 1750 GPU hours to run once. All models were used only for academic research and did not violate their license agreements.

[4]https://huggingface.co/OpenMeditron/Meditron3-8B
[5]https://llamahub.ai/l/llama-packs/llama-index-packs-raptor
[6]https://github.com/SalesforceAIResearch/SiReRAG
[7]https://github.com/Teddy-XiongGZ/MedRAG

F Generative AI Use

In this work, we used Claude[9] to assist in generating code for some of the more tedious implementation components. This assistance was limited to routine programming tasks such as data processing functions, formatting conversions, etc. The core algorithmic approaches, system architecture design, and experimental methodology were conceived and developed by the authors. For writing this paper, generative AI use was limited to minor grammatical adjustments.

G Evaluation Datasets

For the datasets that we used, Table 14 lists the number of examples in each of them. We used MMLU Clinical Topics (MIT License) (Hendrycks et al., 2021), PubMedQA (MIT License) (Jin et al., 2019) and MedQA (MIT License) (Jin et al., 2021). The datasets were used in accordance with their license agreements.

H Ablation Test

We compared the QA accuracy of CLAIMS with four alternative methods, testing the importance of each individual component of our method to the final performance. We ran each method one time with greedy decoding.

[8]https://github.com/YuWVandy/KG-LLM-MDQA
[9]www.claude.ai

Approach	MMLU-V*	MMLU-A	MMLU-CB	MMLU-CM	MMLU-PM	MMLU-MG	MMLU-CK	PMQA	MedQA
Baseline	0.55	0.46	0.57	0.46	0.51	0.60	0.54	0.50	0.44
Rewrite	0.47	0.44	0.45	0.38	0.48	0.62	0.43	0.59	0.46
HyDE	0.55	0.47	0.47	0.45	0.57	0.65	0.46	**0.60**	0.50
Claim	0.55	0.50	0.47	0.42	0.54	0.69	0.45	0.58	0.48
CLAIMS	**0.69**	**0.59**	**0.67**	**0.58**	**0.61**	**0.78**	**0.68**	0.59	**0.52**

*MMLU prefixes denote: V-Validation, A-Anatomy, CB-College Biology, CM-College Medicine, PM-Professional Medicine, MG-Medical Genetics, CK-Clinical Knowledge

Table 13: Comparison of accuracy scores across various biomedical QA approaches, with Claim referring to the ablation configuration of only using the propositional claims without the final layerwise summarization. Our CLAIMS approach achieved comparable or higher scores on all datasets. The MMLU prefixes denote different subject areas, as noted under the table.

Dataset	Dataset Size
PubMedQA	500
MedQA	1273
MMLU Anatomy	135
MMLU College Biology	144
MMLU Professional Medicine	272
MMLU Clinical Knowledge	265
MMLU College Medicine	173
MMLU Medical Genetics	100
MMLU Validation	123

Table 14: Sizes of the evaluation datasets we used in this work.

- Baseline: Only includes the input question and answer options, relying on the model's parametric knowledge to answer the questions.
- Rewrite: Question rewriting is used to retrieve unstructured documents, added with reranking to the model's context window until the context limit is reached.
- HyDE (Gao et al., 2023a): The question, answer options, and candidate answer are used to retrieve unstructured documents. The retrieved documents are reranked and added to the model's context window up to the context limit.
- Claim: We use the HyDE query generation method, and chunk the documents into propositional claims. The claims are reranked and added to the model's context window up to the context limit.

Our final CLAIMS method achieved a comparable or higher score on all datasets. The largest average improvement of our method is over the Rewrite method at 14.63%. It had an average improvement of 11.13% over Claim over the non-validation datasets, which suggests that our graph construction and summarization had a significant improvement over just using propositional claims as a chunking modality (Table 13).

```
You are a biomedical NLP expert.
    Identify and extract key biomedical
    entities from the text. Categorize
    them into: Gene, Chemical, Disease,
    Phenotype, Policy,
    MedicalInterventions,
    ExperimentalTechnique, Examination,
    ComputationalMethod, Location,
    Population, Organism, or OtherEntity
    . Return the results in JSON format
    like: {"entities": [{"text": "entity
     text", "type": "entity type", "
    index": 1}]}. Return only the json
    object.
Text:
```

Prompt 9: Entity Extraction Prompt

```
You are a biomedical NLP expert. Your
    task is to:
1. Analyze the provided text and list of
     entities
2. For each entity, extract all its
    mentions in the text, skipping over
    mentions that are inside of other
    words
3. Return a JSON object with the
    following structure, ensuring that
    all fields are present:
{
    "entity_mentions": [
        {
            "entity_type": "type",
            "index": 1,
            "original_form": "main form
                ",
            "mentions": ["mention1", "
                mention2", "mention3", "
                mention4", "mention5"],
        }
    ]
}
Ensure consistent indexing for the same
    entity across all its forms. Each
    mention in "mentions" should be
    unique words, "mention1", "mention2"
     should not be the same.
You must output a single valid json
    object.
text: {text}
entities: {entities}
Return only the json object.
```

Prompt 10: Entity Mention Prompt

I Entity Masking

For the Entity Masking experiments, we masked the entities in the retrieved documents, questions, and answer options before providing them to the model. Llama-3.3-70B-Instruct (Dubey et al., 2024) classified key biomedical entities into one of 13 categories: Gene, Chemical, Disease, Phenotype, Policy, MedicalInterventions, ExperimentalTechnique, Examination, ComputationalMethod, Location, Population, Organism, or OtherEntity. The prompt to do so is in Prompt 9.

Then, the same model is used to identify all mentions of each entity. These mentions are all replaced with a generic label in format <Category + entity number>, such as <Gene1> or <Disease2> using Prompt 10. The generic label masks used were aligned in all documents, answer options, and the question for each index, ensuring that the 'entity number' used for each entity's mask is consistent across all of these mentions.

J Source Attribution

While CLAIMS effectively synthesizes information across multiple documents through layerwise summarization, this approach creates a trade-off between comprehensive information integration and precise source attribution. The multi-level aggregation process makes it challenging to trace specific claims in the final summary back to exact source sentences.

To support source verification and transparency, we implement a document reranking approach. After generating the final summary, we retrieve all documents that appear on relevant connected paths in the knowledge graph. These documents are then reranked based on their relevance to the generated summary content. This provides users with a prioritized list of source documents most likely to contain supporting evidence for the claims in the summary.

While this approach does not provide sentence-level attribution, it enables practical verification by directing users to the most relevant source materials. This trade-off is inherent to hierarchical summarization methods and represents a design choice that prioritizes comprehensive cross-document reasoning over granular source traceability.

K Component Level Analyses

We perform component level analyses to evaluate the effectiveness of each component in our approach. In relevant metrics that use the LLM-as-a-Judge methods, we use the token probabilities of 'Yes' vs. 'No' to determine the model's selection. The following sections discuss the analysis performed in Section 4.6 in more detail.

K.1 Relation extraction

The goal of the relation extraction phase is to turn the retrieved documents into decontextualized claims with associated RDF triples. The desired properties of these claims and triples are that each claim is self-contained and the meaning of the source documents are retained. In the case that the content in the documents are not exhaustively maintained, at least the key points must be. Thus, for relation extraction, we evaluated the method's ability on *three* key criteria, namely *decontextualization of entity references*, *preservation of semantic meaning* of the original documents, and *key claim extraction* from the original documents.

The **Reference tracker** evaluation tests the decontextualization. To do so, it uses SpaCy to extract both explicit entity mentions and all entity references in each claim. A claim's score is the number of explicit entity mentions over the total number of entity references. The score is aggregated over all claims that are extracted. A well-decontextualized set of claims would have a lower number of unresolved references and thus a higher score.

The **Semantic similarity** evaluation test assesses the method's ability to preserve the original document's meaning. The evaluation involves comparing the semantic similarity between the embedding of the input document and the concatenated form of all of the extracted claims. The score is averaged over all of the retrieved and chunked documents. The score of a set of extracted claims that preserve most of the original meaning would be high.

The **Key relation retention** evaluation test assesses the ability of the extraction to extract key claims. A larger judge LLM extracts important claims from the source documents, and is subsequently asked whether the claims retrieved from the document by the method under evaluation include the information from each of the key claims. The score is calculated by determining the fraction of key claims that are retained, averaging the scores over all of the source documents. The methods under evaluation must extract all relevant key claims to prevent unpredictable downstream behavior.

To assess our method, we compare it with several alternatives.

- Single stage (Our Method): Extracts the claims from the documents and decontextualizes them in a single prompt.
- Two stage: Performs the extraction and decontextualization separately, could potentially improve the performance of the decontextualization but has a drop in efficiency.
- Direct triples: Extracts RDF triples instead of claims, improves the efficiency of the overall pipeline due to skipping the claim extraction step.
- Pairs relations: Extracts the entities first before extracting the relations between entities, a more traditional KG creation method.

```
Summarize the following claims, focusing
     on how the additional claims
    provide context for the first claim:

MAIN CLAIM:
{claim}

CONTEXT CLAIMS:
{claims}
```

Prompt 11: Graph construction component level analysis subgraph and semantic summaries

K.2 Graph construction

The goal of the graph construction phase is to have the RDF triples that come out of the relation extraction phase connect related claims. The communities in the graph should make sense upon consideration of their relevance to the input question. Thus, for graph construction, we tested the method's ability to *have high quality graph communities centered around key claims.*

To evaluate the communities, we want communities that are effective at answering the input question and are centered at the claims of interest. We consider the summaries obtained from extracting a subgraph around the claims of interest that are the top 10 most relevant to the input question based on our reranker, filtered to those that are not within 1-hop of a higher ranked claim. This filtering is the same as that in our graph summarization procedure (Section 3.3). We compare our graph structure using subgraph retrieval with the alternative of retrieving semantically similar claims to the claims of interest. For the subgraph retrieval, we consider all 1-hop connections around the entities in the claims of interest. For semantic similarity, we retrieve all claims that have a similarity above the cosine similarity threshold of 0.8 with the claims of interest. The score for an index with either method is calculated by obtaining the relevance score of the concatenation of all produced summaries of that index via Prompt 11. As the actual relevance scores produced by rerankers are only useful to compare the two methods, we record which of the two methods had a higher score for each index.

```
We have extracted a claim from a summary
    . Was this claim derived from the
    below document?

SUMMARY: {summary}
CLAIM: {claim}
DOCUMENT: {doc}

Answer (Yes/No):
```

Prompt 12: Graph summarization component level analysis source diversity prompt

```
We have extracted a claim from a summary
    . Is this claim supported by this
    document?

SUMMARY: {summary}
CLAIM: {claim}
DOCUMENT: {source_doc}

Answer (Yes/No):
```

Prompt 13: Graph summarization component level analysis faithfulness prompt

```
We have extracted a claim from a summary
    . Is this claim relevant to
    answering the question in the
    context of the summary?

SUMMARY: {summary}
CLAIM: {claim}
QUESTION: {question}

Answer (Yes/No):
```

Prompt 14: Graph summarization component level analysis relevancy prompt

K.3 Graph summarization

The goal of graph summarization is to ensure that the summaries produced by the summarization method are useful for the input question. The requirements for these summaries are that the contents should be *relevant*, *have little hallucinations*, and *have information from various sources*. Thus, for graph summarization, we further test three different metrics: faithfulness, answer relevance, and source diversity.

We evaluate 3 different approaches,

- Our CLAIMS method,
- Subgraph retrieval, and

- Semantic similarity based extraction.

All metrics are tested on a subset of the top 10 ranked claims according to the input question, the claims of interest from Section 3.3. We first utilize our community ranking approach from our CLAIMS method to filter the top 10 claims, retaining the claims that are outside of other claims' 1-hop neighbors. For subgraph retrieval, we create summaries from the 1-hop neighbors of these claims of interest, while for the semantic similarity method we use all claims that have cosine similarity scores over 80% with the claims of interest. Each of the metrics obtain a score for each index, and the final score is the average score over all of the indices.

Answer relevance determines what fraction of the claims made in the output summary are relevant to answering the question. Using the output summary as context, we consider each of the claims we extract from the output summary one by one, and ask a Judge LLM whether it is relevant with Prompt 14. The percentage of relevant claims over all summaries in that index is used as the metric's performance. A higher score means that a higher proportion of claims in the summaries are relevant to the input question.

The **Source diversity** test tests the ability of each method to integrate information from a diverse number of source documents. For each claim extracted from the output summary, we ask the Judge LLM whether it could have come from any of the input source documents with Prompt 12. The score is the number of unique source documents over the total number of documents. The final score for each index is averaged over all of the indices for each individual summarization method. A higher score means that a larger number of multi-document relationships are present in the summaries.

The **Faithfulness** test ensures that each claim in the output summary is truthful based on whether it occurred in the input documents. For each extracted claim from the summaries, we consider each of its source documents from the source diversity test. For each possible source document, we ask the model whether the contexts fully support the accuracy of that claim with Prompt 13. The percentage of supported claims over all summaries in that index is used as the metric's performance. A higher score means less hallucination in the summaries.

K.4 Relation extraction component results

Our relation extraction evaluation compared four methods across three metrics: reference tracking (Ref Score), semantic preservation (Sem. Similarity), and key claim retention (Claim Ret.) (Table 4). The reference tracking scores show a clear pattern between the claim and entity-based approaches. The pairs relations method achieved the highest reference tracking score (0.99) followed by direct triples (0.97), while the two claim-based approaches scored slightly lower (0.94, 0.95). This difference is due to the inherent nature of direct entity extraction, which focuses on extracting explicit entities and thus naturally avoids leaving unresolved references. However, the claim-based methods still achieved strong scores above 0.94, indicating the effectiveness of the decontextualization while maintaining sentence structure.

In contrast, the semantic preservation performance of the two claim extraction methods are superior. Our single stage (0.90) and the two stage (0.90) methods significantly outperformed the entity-based extraction methods, (0.87, 0.82). This advantage suggests that retaining the sentence structure of the claims results in lower information loss of semantic meaning. All of our methods achieved a perfect key claim retention score, indicating that critical information was preserved regardless of which extraction approach was used.

These results support our usage of the single stage approach, as while it shows slightly lower reference tracking performance compared to the entity-based methods, it achieves essentially identical performance to the two stage approach while being more computationally efficient without the additional decontextualization step. The higher semantic similarity score suggests that the minor tradeoffs in the decontextualization performance are compensated by better preservation of the claims' original meanings. The perfect claim retention indicates that there is no loss of critical information. The balance of performance metrics and higher efficiency gives it an edge for extracting information from the retrieved documents.

K.5 Graph construction component results

The summaries produced by the graph communities had a higher relevance score compared to the summaries produced by the semantic communities 59.35% of the time (Table 5). This demonstrates that the summaries produced from our graph structure more effectively group relevant information for answering the input question. While semantic communities are limited to capturing relationships based on pure textual similarity, our graph construction identifies topical connections that may not be apparent from semantic similarity alone. This property allows for relevant topically related yet semantically dissimilar information to be added to the final summaries. Such connections might be missed by pure semantic grouping, contributing to our method producing more comprehensive relevant summaries for question answering.

K.6 Graph summarization component results

Our CLAIMS method achieved comparable faithfulness (0.96) and relevancy scores (0.84) compared to the alternative approaches while having superior source diversity (0.96) (Table 6). The higher source diversity score demonstrates our CLAIMS method's effectiveness at integrating multi-document relationships, surpassing the semantic (0.92) and subgraph (0.94) approaches. This implies that our layerwise processing has the advantage of incorporating information from a more diverse group of sources.

The slightly lower relevancy score of our CLAIMS method (0.84) compared to semantic clustering (0.86) stems from the nature of our graph structure, where information that is not directly relevant to the question but is useful for connecting relevant statements is included in the summaries. This design decision enables more comprehensive answers but lowers the total number of claims that are directly relevant to the input question in the summaries. The more significant drop in relevancy score for the subgraph method (0.79) demonstrates how our CLAIMS approach filters out irrelevant claims that subgraph extraction retains.

The consistently high faithfulness values (>0.94) for all three alternative methods confirms that none of them suffer from significant hallucinations, with our method achieving strong faithfulness (0.96) with superior source diversity. CLAIMS had a slightly lower faithfulness score (0.96) compared to semantic communities (0.97), likely due to the multiple LLM calls in its pipeline increasing the potential for hallucination. However, the relatively small difference suggest this is an acceptable trade-off for improved reasoning capabilities. This validates our CLAIMS approach's ability to maintain quality content while integrating information from more sources, therefore having a higher chance of combining relevant information that other methods would not have considered.