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

Anonymous ACL submission

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 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 accurate answers often require synthesizing multiple medical concepts across diverse documents. 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 require access to the entire offline knowledge corpus (Edge et al., 2024; Guo et al., 2024b; Wu et al., 2024) or 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 more comprehensive reasoning in Biomedical QA*.

To remedy this, we propose utilizing the construction of a knowledge graph to underlay layerwise document summarization as an alternative via **CLAIMS** (Connected Layered Analysis of Information through Multi-level Summarization). Propositional claims are utilized to represent information and facilitate handling conflicting and noisy claims extracted from retrieved unstructured documents. The knowledge graph 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 a novel approach of structuring information from retrieved documents as propositional claims in local knowledge graphs to capture cross-document relationships.
- We introduce utilizing layerwise topological graph summaries of key claims in this local knowledge graph as context for LLM QA tasks.
- We evaluate CLAIMS on a comprehensive set of benchmarks, including testing the properties of the intermediate components of the approach, its impact on LLM reasoning, 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 large external corpora into pre-trained language models' context windows. The initial naive RAG approach utilized a trained retriever and a seq2seq model to capture knowledge from retrieved documents (Lewis et al., 2020), and has since been followed by many follow-up refinements (Gao et al., 2023b). A number of 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 the retriever for improved retrieval of medical documents (Li et al., 2024b), and Self-BioRAG which uses on-demand retrieval and reflection tokens to select the best evidence (Yu et al., 2023), among many others (Liu et al., 2024; Zhou et al., 2023), which tend to take the strategies used in general domain RAG and adapt them to the biomedical domain. While these works provide benefits for QA tasks, they fall short in capturing all of the relevant multi-document relationships in retrieved documents.

2.2 Summarization

Summarization can condense input documents into relevant information while using less input tokens, and is one method by which retrieved documents can be processed to better suit downstream tasks. RAPTOR (Sarthi et al., 2024) uses hierarchical summarization of input 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 as a way to capture complex relationships. An extensive line of prior work exists due to the widespread usage of external knowledge graphs as a data structure. Common RAG methods involving them 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 a line of work performing community-based summarization on generated knowledge graphs. They partition the knowledge graph into modular parts, either via communities as with Graph Rag (Edge et al., 2024), or into hierarchical tags as in MedGraphRAG (Wu et al., 2024). While these methods are able to capture more multi-document relationships, they perform their method on the entire offline retrieval corpus

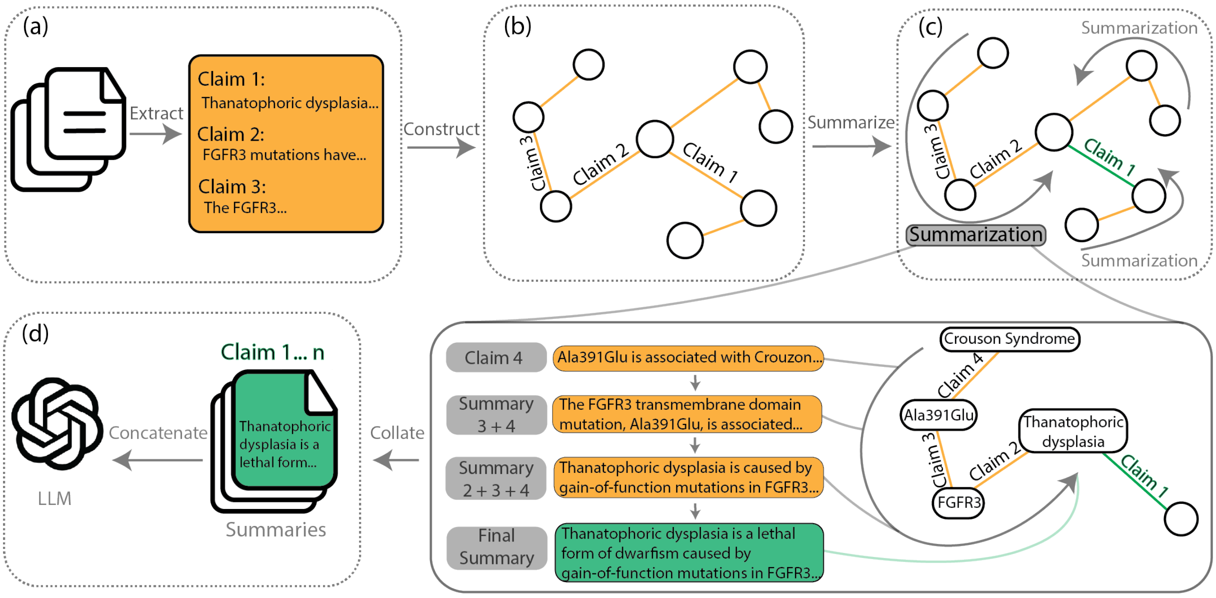

Figure 1: Overview of the proposed CLAIMS framework. **(a) Relation extraction:** load in documents with a retriever relevant to an input question, break documents into claims, break claims into triples. **(b) Graph construction:** build local graph with claims and triples. **(c) Graph summarization:** 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.

rather than dynamically retrieved online input documents. This requires a high upfront cost and a different level of granularity compared to our method, while also requiring additional effort when updating their graph summaries with new information.

Alternatively, retrieved documents can be turned into a graph structure for additional processing. Several works have opted for this method, 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., 2024c). Our method utilizes the 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 retrieved documents (Zhang et al., 2025b), but does not use propositional claims as their chunking modality and summarizes the content for 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 1). It proceeds by first extracting decontextualized claims from each $d \in D$ (Section 3.1), using the entities in these claims to build a graph (Section 3.2), before summarizing the content in the graph into several key claims that are provided to L to solve the question (Section 3.3).

3.1 Relation extraction

The relation extraction step transforms retrieved unstructured documents into propositional claims and associated RDF triples. This turns complex technical documents into atomic pieces of information that can be reliably connected and analyzed.

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 I.

Claim extraction: To connect information across documents, documents are broken down into concise and independent pieces. From the retrieved text chunks $d \in D$, the model L extracts propositional claims $C = \{c_1, c_2, ..., c_n\}$. These propositional 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 propositional claims, with each one having 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 graph construction step processes the RDF triples and claims from Section 3.1 into a local graph structure that captures the relationships between pieces of information. This is crucial for identifying multi-document interactions that are not apparent from individual claims.

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 in the graph is an entity from the RDF triples $(subj,\ pred,\ obj)$, one of the $subj$ or obj entities. Each edge $e \in G$ 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 of the edges are treated as undirected in further processing, and allow for multiple edges between two entities.

3.3 Graph summarization

The final graph summarization stage of 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 large number of 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.

It proceeds to determine each claim of interest's potential to produce meaningful summaries for our later layerwise summarization. Since claims closer to these entry points will be given more weight in the final summaries, each claim of interest's 1-hop neighboring claims are examined. These neighboring claims are used as context to generate *test summaries* that approximate the final summaries, and the relevance of these test summaries are used to again rerank the claims of interest.

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.

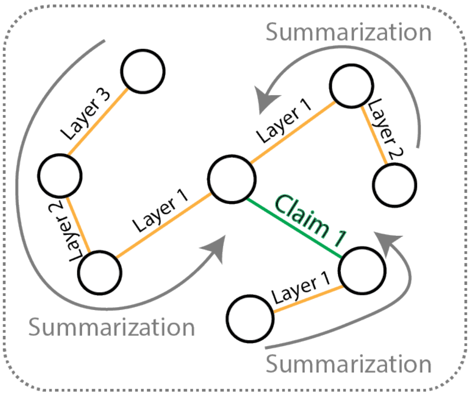

Figure 2: Layerwise summarization approach overview. For a given claim of interest (Claim 1), the graph is organized into layers based on the distance of each connected claim from it. The summarization begins from the furthest layer, moving inwards. For each layer claims are summarized using the previously generated summaries of connected claims in lower layers. This process ensures that path information and multi-document relationships are preserved while filtering out irrelevant information in the final summaries.

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 2).

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, but is focused around the central claim. Although these claims of interests share common topics due to their high relevance to the input question, each final summary should differ because they emphasize their local relationships. The final output is a concatenation of the summaries in the order of their relevance rankings. This set of summaries is provided as contexts for an LLM to perform QA.

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), and compare against standard RAG baselines (Section 4.2). Each part of CLAIMS was also individually assessed to test its robustness (Section 4.3). Additionally, we employ entity masking tests to evaluate CLAIMS' ability to improve LLM reasoning capabilities independent of parametric knowledge (Section 4.4).

4.1 Evaluation datasets

We use the test sets of PubMedQA (Jin et al., 2019), MedQA (Jin et al., 2020), and the MMLU clinical topics datasets (Hendrycks et al., 2021) (Anatomy, Clinical Knowledge, College Biology, Professional Medicine, College Medicine, and Medical Genetics). For validation and ablation tests, a combination of the validation sets of the MMLU datasets is used, termed MMLU validation.

4.2 QA baselines

We compared the QA accuracy of CLAIMS with four alternative measures.

- 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.
- RAPTOR (Sarthi et al., 2024): We use the HyDE query generation method to retrieve documents. The RAPTOR process[1] is used to produce a context for each question for QA.

4.3 Component level analysis

We evaluated the capabilities of the core components in CLAIMS over our MMLU Validation dataset. These included the modules of relation extraction, graph construction, and graph summarization as can be seen from Figure 1.

[1] https://llamahub.ai/l/llama-packs/llama-index-packs-raptor

Relation extraction: The goal of the relation extraction phase is to turn the retrieved documents into decontextualized claims with associated RDF triples. These claims should be self-contained and should retain the meaning of the source documents. Thus, for relation extraction, we evaluated the method's ability on three key criteria, namely.

- Decontextualization: fraction of explicit entity references over all entity references extracted with SpaCy from each claim.
- Preservation of semantic meaning: semantic similarity between the input document and the concatenated form of all of the extracted claims.
- Key claim extraction: the fraction of key claims extracted from the retrieved documents using a judge LLM, that were extracted with the method under evaluation.

To assess our method, it is compared with several alternatives.

- Single-stage (Our Method): Extracts the claims from documents and decontextualizes them in a single prompt.
- Two-stage: Performs the extraction and decontextualization separately, potentially improves 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.
- Pairs relations: Extracts the entities first before extracting the relations between entities, a more traditional KG creation method.

Graph construction: The goal of the graph construction phase is to have the communities in the graph make sense upon consideration of their relevance to the input question. Thus, for graph construction, the method's ability to *have high quality graph communities centered around key claims* was tested.

We compared the summaries produced from subgraphs and semantic communities around the claims of interests from graph summarization (Section 3.3).

- Subgraph communities: All 1-hop connections around the entities in the claims of interests are considered, using the claims on these connections to produce summaries for each claim of interest.
- Semantic communities: All claims that have a similarity above the cosine similarity threshold of 0.8 with the claims of interests are retrieved, and use these claims to produce summaries.

A method's score for an index is calculated by obtaining the relevance score relative to the input question of the concatenation of all produced summaries. Which of the two methods had a higher score for each index is recorded.

Graph summarization: The goal of graph summarization is to ensure that the summaries produced are of high quality. The requirements for these summaries are that the content should *have little hallucinations*, be *relevant*, and *integrate information from various sources*.

Thus, for graph summarization, we further test three different metrics:

- Faithfulness (hallucination rate): fraction of claims in the output summaries that are supported by the input documents.
- Answer relevance: fraction of claims relevant to the input question in the output summaries.
- Score diversity: fraction of input documents that have their content included in the final summaries.

We compared CLAIMS with the summaries produced from the subgraph and semantic communities around the claims of interests. These are the same summaries used in the graph construction component analysis.

4.4 Entity masking

In order to evaluate the effect of CLAIMS on LLM reasoning beyond the parametric knowledge of the models, we masked the entities in the retrieved documents, questions, and answer options. The masking was performed via prompting the Llama-3.3-70B-Instruct model (Dubey et al., 2024) to identify and mask key biomedical entities into one of 13 categories. These entities will be replaced with a generic label, and the generic label masks used for each entity were aligned across all documents, answer options, and the question for each index. This allowed us to evaluate whether our approach was able to improve the model's performance in the absence of any prior knowledge of how the entities were related to each other. We compared CLAIMS under this circumstance against the HyDE baseline

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
RAPTOR	0.63	0.54	0.63	0.55	0.60	0.75	0.63	**0.66**	0.50
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 1: Accuracy scores across various BioMedical QA approaches. Results show the performance on MMLU Clinical Topics, PubMedQA, and MedQA benchmarks. CLAIMS shows consistent improvements over baseline methods, with comparable or superior performance across the non-validation datasets. The MMLU prefixes denote different subject areas, as noted under the table.

from the QA Accuracy evaluation. More information about the masking procedure can be found in Appendix K.

5 Results

5.1 QA accuracy

The largest average improvement of our method is over the Rewrite method at 14.63% and the smallest over RAPTOR at 2.00% on all of the non-validation datasets (Table 1). Other than the PubMedQA dataset where it obtained a 59% accuracy, CLAIMS has comparable or improved performance over the baselines on all datasets. For PubMedQA, we believe that the slight drop in performance is due to insufficient denoising in the created graph, which we plan on addressing in future work. In all, these results imply that our method has allowed the model to more thoroughly analyze cross-document relationships in its limited context window, therefore more effectively synthesizing information from the retrieved documents.

Approach	Ref Score	Sem. Sim.	Claim Ret.
single_stage	0.941	0.901	1.0
two_stage	0.946	0.903	1.0
direct_triples	0.971	0.865	1.0
pairs_relations	0.994	0.815	1.0

Table 2: Relation extraction methods across three metrics. Ref. Score measures decontextualization ability, Sem. Sim. measures preservation of original meaning, and Claim Ret. measures preservation of key information. Scores range from 0-1.0. Results demonstrate the trade-off between entity-based and claim-based approaches, with our single stage method achieving a balanced performance while maintaining good computational efficiency.

5.2 Component level analysis results

We obtained evaluation results for each of CLAIMS' three core components, namely relation extraction, graph construction, and graph summarization. Our relation extraction evaluation compared four methods across three metrics: decontextualization quality (Ref Score), semantic preservation of original documents' meanings (Sem. Similarity), and key claim retention (Claim Ret.) (Table 2). The entity-based claim extraction approaches (direct_triples and pairs_relations) achieved higher reference tracking scores (0.994 and 0.971) compared to claim-based methods (single_stage 0.941, two_stage 0.946) due to their focus on extracting explicit entities which naturally avoids leaving unresolved references. However, the claim-based methods achieved strong semantic preservation performance (0.901 and 0.903 vs 0.865 and 0.815). This advantage suggests that retaining the sentence structure of the claims results in lower information loss of semantic meaning. All of the methods tested achieved a perfect key claim retention score. These results support our usage of the single stage approach with its comparable decontextualization and superior semantic preservation scores compared to the entity extraction approaches, and it achieves almost identical performance to the two stage approach at a fraction of the computational cost.

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

Table 3: Relevance scores between graph and semantic-based summarization. Results show the percentage of times each method produced summaries with a higher relevance score, and demonstrate the graph community summary's superior ability to capture relevant information from the input documents.

For the graph construction component, the summaries produced by the graph communities had a higher relevance score to the input question compared to the summaries produced by the semantic

Approach	MMLU-V*	MMLU-A	MMLU-CB	MMLU-CM	MMLU-PM	MMLU-MG	MMLU-CK	PMQA	MedQA
HyDE	0.15	0.18	0.28	0.23	0.22	0.36	0.23	**0.56**	0.28
CLAIMS	**0.26**	**0.22**	**0.28**	**0.28**	**0.26**	**0.43**	**0.34**	0.48	**0.30**

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

Table 4: Accuracy scores across various BioMedical QA approaches, with masked retrieved documents, input questions, and answer options. Our CLAIMS approach achieved higher scores on all datasets other than PubMedQA. The MMLU prefixes denote different subject areas, as noted under the table.

communities 59.35% of the time (Table 3). While semantic communities are limited to capturing relationships based on pure semantic similarity, our graph construction identifies connections that may be relevant topically yet semantically dissimilar.

For the graph summarization component, CLAIMS achieved comparable faithfulness (0.9569) and relevancy scores (0.8414) compared to the alternative approaches while having superior source diversity (0.9647) (Table 5). The slightly lower relevancy score of our CLAIMS method (0.8414) compared to semantic clustering (0.8604) stems from the inclusion of information in the summaries that is not directly relevant to the question but is useful for connecting relevant statements. 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 consistently high faithfulness values (>0.94) for all three alternative methods confirms that none of them suffer from significant hallucinations. Our method achieving a strong faithfulness (0.9569) balanced with superior source diversity, meaning that it can integrate information from many of the retrieved documents with little hallucination in the produced summaries. The results of our evaluations are discussed in more detail in Appendix L.

Approach	Faithfulness	Relevancy	Source Div.
CLAIMS	0.9569	0.8414	0.9647
Semantic	0.9706	0.8604	0.9170
Subgraph	0.9453	0.7938	0.9356

Table 5: Three summarization approaches across faithfulness (hallucination), relevancy (relevance to input question), and source diversity (multi-document relations) metrics. Scores range from 0-1.0. Results demonstrate CLAIMS' ability to maintain a high faithfulness and relevancy while achieving superior source diversity.

5.3 Entity masking results

Notably, the accuracy scores of the masked configuration are significantly lower than their unmasked variants, suggesting that the masking of the entities has broken many of the connections between them that had been learned during pretraining. Looking at the results, other than on PubMedQA, our CLAIMS approach had a comparable or higher accuracy score than the HyDE baseline, achieving an average improvement of 3.13% on all of the non-validation datasets (Table 4). This suggests that our approach is capable of representing information in a manner that fundamentally improves the reasoning ability of the LLM, instead of only utilizing heuristic patterns between entities learned during pretraining.

6 Conclusion

We introduce a novel method called CLAIMS 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 were used to contextualize a small language model to produce the final QA decisions. CLAIMS achieved comparable or superior performance over RAG baselines on several biomedical benchmarks, with average improvements ranging from 2.00% to 14.63%, demonstrating its effectiveness in enabling even a small model to effectively synthesize complex multi-document information. Additional experiments covering the intermediate stages of our pipeline and its effects on LLM reasoning showed the robustness of each part of our approach. The results reveal that outside of improvements on traditional benchmarks, CLAIMS provides benefits on QA tasks even when existing connections between entities are masked.

7 Limitations

Denoising: Our approach currently relies on the summarization's inherent denoising ability to remove irrelevant information from the constructured 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: Our current claim and triple extraction steps all require LLM generation for each claim/triple, which can become expensive depending on the number of retrieved documents. We plan on looking into non-LLM approaches to do the extractions to improve the method's efficiency.

8 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.

References

Jinheon Baek, Alham Fikri Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. *CoRR*, abs/2306.04136.

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.

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. 2020. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *arXiv preprint arXiv:2009.13081*.

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).

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, Hua Xu, and Rui Zhang. 2024b. Biomedrag: A retrieval augmented large language model for biomedicine. *arXiv preprint arXiv:2405.00465*.

Zijian Li, Qingyan Guo, Jiawei Shao, Lei Song, Jiang Bian, Jun Zhang, and Rui Wang. 2024c. Graph neural network enhanced retrieval for question answering of llms. *ArXiv*, abs/2406.06572.

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 for retrieval-augmented large language models. *CoRR*, abs/2305.14283.

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.

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, and Yunli Qi. 2024. Medical graph RAG: towards safe medical large language model via graph retrieval-augmented generation. *CoRR*, abs/2408.04187.

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.

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

```
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

```
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 3: RDF Triple Extraction Prompt

A 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 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 4: 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 4.

```
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 5: Claim of Interest Prompt

```
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 6: Layerwise Summarization Summary Prompt

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

Prompt 7: Model Generation 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 interests. This can be seen in Prompt 5.

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 6.

B Triple Extraction Fallbacks

For RDF triple extraction (Section 3.1), we begin with Prompt 3. 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.

C 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.

D 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.

```
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 Output Evaluation

Due to the variability in LLM outputs, in order to extract the model's answer option from its outputs we utilized a subsequent extraction step. As seen in prompt 8, we take the model output, question, and answer options and ask the model to output a json object that captures the selected option. The outputs are forced to be json objects via lmformatenforcer (MIT License) [2]. We choose to do it in this manner compared to directly having the model output a json object when answering the question due to issues with invalid json objects and empirically noticing a drop in performance when doing so.

F Evaluation Datasets

For the datasets that we used, Table 6 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., 2020). The datasets were used in accordance with their license agreements.

G Generative AI Use

In this work, we used Claude[3] 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.

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

Table 6: Sizes of the evaluation datasets we used in this work.

H 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 Entity Masking, we use Llama-3.3-70B-Instrucz (Llama 3.3 Comunity License Agreement) (Dubey 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. We run experiments on NVIDIA L40S and A40 GPUs, and H100s when possible. All of the experiments and benchmarks took approximately 250 GPU hours to run once. All models were used only for academic research and did not violate their license agreements.

I RAG Retrieval

The retrieval corpora include Simple Wikipedia (CC-BY-SA) (Foundation), medical textbooks from MedQA (MIT License) (Jin et al., 2020), PubMed abstracts and fulltext articles taken from GLKB (CC BY-NC-ND 4.0) (Huang et al., 2024),

[2]https://github.com/noamgat/lm-format-enforcer
[3]www.claude.ai

and StatPearls articles[4] (CC BY-NC-ND 4.0). Simple Wikipedia provides general knowledge, medical textbooks provide foundational concepts, Statpearls documents provide detailed medical information, and PubMed abstracts/fulltext articles provide research findings. This combination is to improve the coverage of topics for which our method can retrieve relevant information, 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), and as GLKB returns various topics associated with the input query, we use one of these connected topics to retrieve an additional set of articles. 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, including the additional reference returned by GLKB. 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.

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

J Ablation test

We ran an ablation test of our approach to test whether the graph construction and summarization was necessary for the improved performance. We tested our approach against the alternative of:

- 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. 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 7).

K 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. These are the same documents that were retrieved for the datasets without masking to ensure we had a good set of documents to start out with. 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.

```
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
```

Approach	MMLU-V*	MMLU-A	MMLU-CB	MMLU-CM	MMLU-PM	MMLU-MG	MMLU-CK	PMQA	MedQA
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 7: 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.

```
    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

L 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 4.3 in more detail.

L.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

L.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

L.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.

L.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 2). 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 (0.994) followed by direct triples (0.971), while the two claim-based approaches scored slightly lower (0.941, 0.946). 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 (0.901) and the two stage (0.903) methods significantly outperformed the entity-based extraction methods, (0.865, 0.815). 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.

L.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 3). 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.

L.6 Graph summarization component results

Our CLAIMS method achieved comparable faithfulness (0.9569) and relevancy scores (0.8414) compared to the alternative approaches while having superior source diversity (0.9647) (Table 5). The higher source diversity score demonstrates our CLAIMS method's effectiveness at integrating multi-document relationships, surpassing the semantic (0.9170) and subgraph (0.9356) 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 (0.8414) compared to semantic clustering (0.8604) 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 (0.7938) 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 (0.9569) with superior source diversity. 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.