Multi-System Biomedical QA with Post-Hoc Sentence Attribution for TREC BioGen

Harikrishnan Gurushankar Saisudha, Sabine Bergler
h_gurush@cse.concordia.ca, sabine.bergler@concordia.ca

Computational Linguistics at Concordia (CLaC),
Concordia University, Montreal, Canada

Abstract

This paper presents six systems developed for the TREC 2024 Biomedical Reference Attribution task, covering both Task A (grounding answers) and Task B (reference attribution). For Task A, two LLM-based citation attribution systems were designed to identify supporting and contradicting evidence from PubMed, using a combination of sparse retrieval, dense reranking, and optional NLI-based contradiction filtering. For Task B, four systems were developed: two SimpleQA pipelines emphasizing lightweight, narrative-aligned answer generation, and two MedHopQA-based pipelines adapted from a multi-hop biomedical QA framework. Across systems, we evaluate the impact of retrieval strategies, question decomposition, and reranking on evidence selection and answer quality. Our analysis highlights key challenges in grounding citations, particularly for generic statements and contradiction detection, and identifies opportunities to improve LLM-based attribution through refined retrieval strategies and specialized fine-tuning. Together, these systems provide a comprehensive exploration of reference attribution in biomedical QA and illustrate the trade-offs between retrieval coverage, narrative alignment, and attribution accuracy.

1 Introduction

Large language models (LLMs) have been successful in many domains, especially in the biomedical domain, such as biomedical question answering and literature and clinical note summarization. However, hallucinations and confabulations remain one of the key challenges when using LLMs in the biomedical domain. Inaccuracies in biomedical applications may be harmful in high-risk situations, such as clinical decisions or biomedical research. To address this, TREC (Text Retrieval Conference) introduces a task of reference attribution to mitigate the generation of false statements by LLM in biomedical question answering.

TREC BioGen consists of two tasks: *grounding answer* and *reference attribution*. Grounding answer requires using a triple of *given question*, *answer*, and *existing citations* to find relevant supporting and contradicting documents for each answer sentence from PubMed and return their PMIDs.

Reference attribution requires using the given *question*, *topic*, and *narrative* to answer the question aligned with the topic and narrative, using documents from PubMed. The answer must

contain citations from PubMed for each sentence.

The rest of the paper is organized as follows: Section 2 covers two systems developed for Task A, Section 3 describes a simple QA system developed for Task B and Section 3 describes our MedHopQA non-agentic pipeline system used for this task. The systems developed for Task A primarily compare the impact of NLI-based filtering usage for contradicting citations, and the systems developed for Task B compare the retrieval strategies and the pipeline itself.

2 System 1 & 2 - LLM-based citation attribution (Task A)

2.1 Motivation

The underlying motivation for both System 1 and System 2 is the same: to identify citations that reliably support or contradict a given answer sentence. In both systems, supporting citations are identified by first retrieving documents using a sparse retriever and then reranking them with a dense reranker using the answer sentence as the query. The assumption is that sentence-level retrieval will surface documents that are semantically closer to the target sentence, thereby increasing the likelihood of identifying relevant supporting evidence.

For contradicting citations, System 1 is built on the premise that question-based retrieval may yield documents containing information that contradicts the candidate answer sentences. Similar to supporting citation retrieval, the system applies a sparse retrieval stage followed by dense reranking to isolate question-related documents. These documents are then fed into a RAG (Retrieval-Augmented Generation) setup, where an LLM is tasked with detecting contradictions.

While System 1 relies solely on question-based retrieval to surface contradictory citations, a more selective process may better identify contradictory evidence. Therefore, System 2 introduces an NLI-based filtering stage into the pipeline. Natural Language Inference (NLI) involves determining whether a hypothesis sentence is entailed by, contradicts, or is neutral with respect to a given premise sentence. For this experiment, the *SciFive-large-PubMed_PMC-MedNLI* [9] model is used, as it is fine-tuned on the MedNLI [10] dataset. In this setup, each retrieved document is split into sentences, and every sentence is compared against each answer sentence using the NLI model. Documents containing one or more sentences classified as contradictory to the answer are then retained and passed to the LLM prompt.

2.2 Methodology

Systems 1 and 2 are designed to identify both supporting and contradicting citations for each sentence in a given answer. Originally developed for Task A, the same pipelines were used in Task B for identifying supporting citations. Both systems rely on the Mistral 7B model [5], configured with bits-and-bytes [1] optimization to efficiently handle large-scale document processing.

The retrieval component in both systems combines a sparse BM25 retriever with a dense ColBERT reranker [6]. System 2 extends this pipeline with an additional Natural Language Inference (NLI) [2] stage using the SciFive model [9], selected for its MedNLI fine-tuning and suitability for biomedical contradiction detection. While the overall structure of the pipelines is similar, the NLI stage in System 2 provides finer-grained filtering of contradictory evidence.

To locate supporting citations, given a question and its answer, the pipeline processes each answer

[1] https://huggingface.co/docs/transformers/en/quantization/bitsandbytes

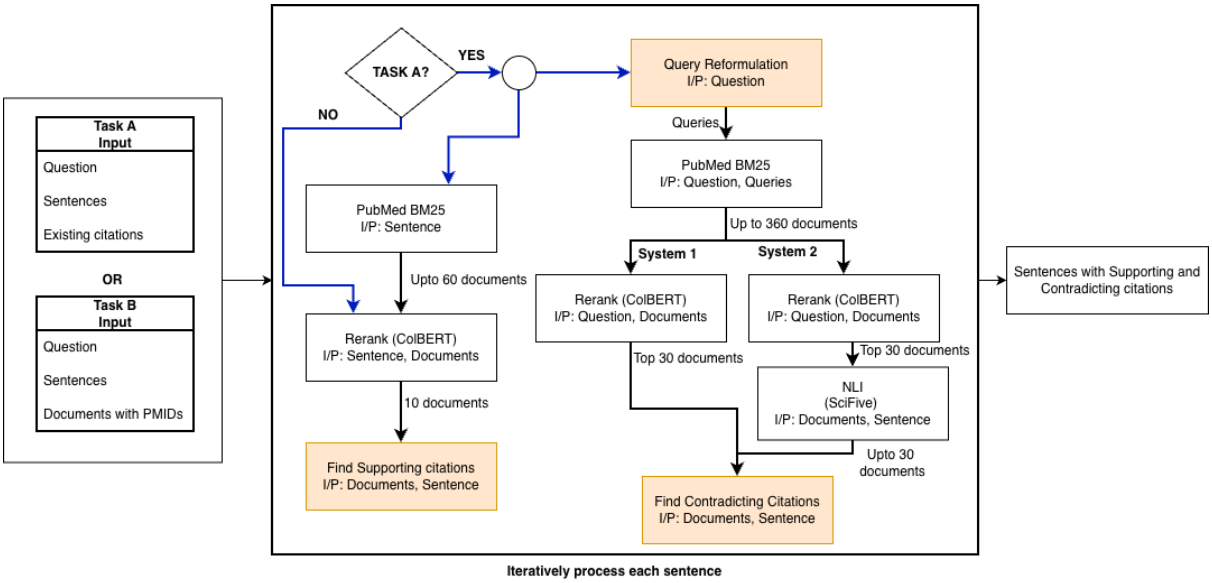

Figure 1: System 1 & 2 Pipeline (Orange boxes indicate an LLM step)

sentence independently. First, the top 60 documents are retrieved from the BM25 PubMed [8] index using the answer sentence as the query, ensuring broad coverage and capturing a diverse pool of candidate evidence. These documents are then reranked using ColBERT, which enables more accurate semantic matching between the query sentence and retrieved passages [7]. The top 10 reranked documents are supplied to the LLM along with a prompt (Appendix A) instructing it to identify up to three supporting citations.

To locate contradictory evidence, the question is decomposed into multiple sub-queries using the LLM (Appendix B). For each sub-query and for the original question, the top 60 BM25 documents are retrieved. These documents are then reranked using ColBERT with respect to the original question, and the top 30 passages are selected.

In System 2, an additional NLI-based filtering step is applied to these passages. Each document is segmented into sentences, and the SciFive MedNLI model compares every sentence with the corresponding answer sentence to determine whether it contradicts it. Only documents containing at least one contradictory sentence are retained. Thus, System 1 always passes 30 reranked documents to the LLM, while System 2 passes up to 30 documents after NLI filtering. The retained documents are then provided to the LLM with instructions to extract contradicting citations (Appendix A) .

3 System 3 & 4 - SimpleQA system (Task B)

3.1 Motivation

System 3 is motivated by the need to build a lightweight yet effective question-answering pipeline that produces concise, accurate answers from biomedical literature. Many biomedical questions are multi-faceted, and retrieving relevant evidence with a single query often misses important aspects

of the information need. To address this, System 3 decomposes the original question into several focused sub-queries, improving retrieval coverage while keeping computational overhead low. A reranker then prioritizes the retrieved documents using the topic, question, and narrative, ensuring that the most relevant evidence is used for downstream answer generation.

System 4 is motivated by a complementary challenge: single-mode retrieval, whether sparse or dense, may fail to capture the full spectrum of relevant biomedical documents. Sparse retrieval excels at lexical overlap, while dense retrieval captures semantic similarity [7]. System 4 integrates both signals through a hybrid dense + sparse retrieval strategy, increasing document diversity and recall. By combining lexical and semantic evidence before reranking, System 4 aims to build a more robust system capable of retrieving articles that might be missed by either approach alone.

3.2 Methodology

Both System 3 and System 4 follow a similar multi-stage pipeline that integrates question decomposition (Appendix B), document retrieval, reranking, answer generation, and citation extraction, with the primary distinction between the systems occurring in the retrieval stage. The pipeline begins with decomposing the original question into several focused queries. This step is used in both systems and helps capture different aspects of complex biomedical questions, improving the likelihood of retrieving relevant evidence. These queries, together with the topic and narrative, and the original question are used to initiate document retrieval.

In System 3, retrieval is performed using a BM25 sparse retriever, which retrieves twenty documents per query, topic, question and narrative. This produces a lexically grounded document pool optimized for precision. System 4 extends this retrieval stage by combining BM25 with a dense FAISS retriever [1]. The hybrid approach allows System 4 to capture both lexical matches and deeper semantic relationships, improving recall and bringing forward documents that sparse retrieval alone might miss. After retrieval, both systems apply the same reranking mechanism. All retrieved documents, whether from the BM25-only pipeline of System 3 or the hybrid pipeline of System 4, are passed through a ColBERT reranker. ColBERT's late-interaction architecture allows it to compute fine-grained relevance scores conditioned on the original question, narrative, and topic. The top 25-ranked documents form the final evidence set that will be fed into the LLM.

For answer generation, both systems use the same approach. The top documents are combined with a structured prompt template (Appendix C) and passed to the Qwen3-8B-AWQ [12] model, which synthesizes a concise answer grounded in the supplied evidence. Then it is passed on to the answer condensation stage (Appendix C) to reduce the answer to under 250 words. The output at this stage is an answer without citations.

To attach citations, the systems rely on the citation-generation pipeline built for System 1, with only minimal adaptation. The adaptation involves bypassing the retrieval stage of the System 1 citation pipeline, since both System 3 and System 4 already provide the full set of documents used to generate the answer. The subsequent steps, evidence scoring, span extraction, and citation selection remain unchanged. This produces a final answer supplemented with validated PubMed citations, completing the pipeline for both systems.

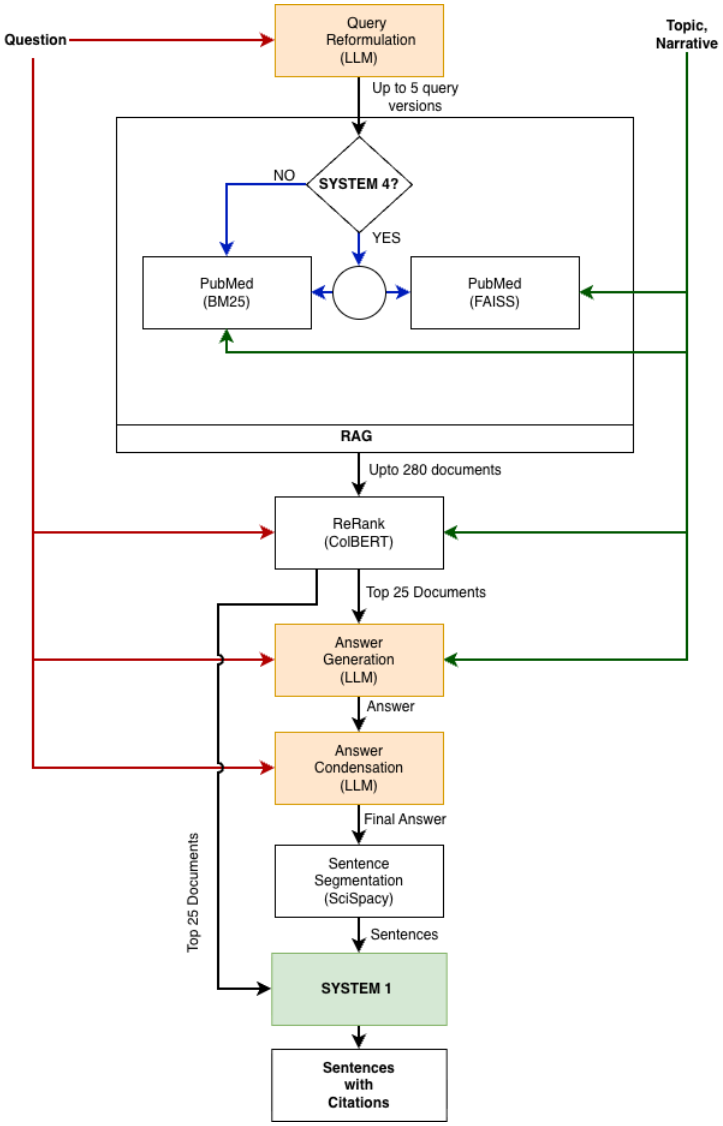

Figure 2: System 3 & 4 Pipeline, the post-hoc sentence attribution is available in the Figure 1 (Green box indicates another system invocation)

4 System 5 & 6 - MedHopQA System (Task B)

4.1 Motivation

The goal of Systems 5 and 6 is to evaluate how effectively our MedHopQA system [11], originally developed for the BioCreative XI MedHopQA shared task [4], transfers to the TREC biomedical QA setting. This cross-task evaluation allows us to assess the system's robustness and generalization

capabilities across related but distinct biomedical question answering benchmarks. To support this comparison, we reuse the non-agentic MedHopQA pipeline while integrating different retrieval configurations suited to the TREC task: System 5 employs a PubMed BM25 retriever, whereas System 6 incorporates a hybrid dense (FAISS) + sparse (BM25) retrieval strategy.

4.2 Methodology

The MedHopQA non-agentic pipeline functions as an iterative multi-hop question answering system. It begins by generating a focused sub-question derived from an important phrase in the original question. Retrieved evidence for this sub-question is then used to iteratively produce additional sub-questions, enabling multi-hop reasoning across multiple document sets.

However, most of the TREC questions are more direct and do not require multi-hop decomposition. To better align with this structure, we adapt the pipeline to bypass sub-question generation for questions that are not complex. This modification reduces computational cost while preserving answer quality for the majority of TREC questions. After this adjustment, the system retrieves 60 documents from PubMed using a BM25 retriever, and all subsequent stages of the original MedHopQA pipeline remain unchanged; full details are provided in the MedHopQA paper [11].

The system produces both short and long answers, but for the TREC evaluation, we use the long answer as the final output. Supporting citations are appended using the System 1 citation pipeline, which operates on the document set retrieved by the MedHopQA module.

System 6 follows the same pipeline but differs in its retrieval strategy. Instead of relying solely on BM25, it applies a hybrid dense (FAISS) + sparse (BM25) retriever, retrieving 30 documents from each. This hybrid approach increases evidence diversity while maintaining the overall workflow and objectives established in System 5.

5 Evaluation and Results

Task A systems were automatically evaluated using the Llama-3-70B model, reporting precision, recall, and F1-score for supported and contradicted citations.

Task B systems were automatically evaluated using the BioACE framework [3], reporting precision, recall, completeness, correctness for answer quality, and citation coverage, support rate, and contradict rate. The citation contradict rate is a penalizing score.

run_id	supported_precision	supported_recall	supported_f1	contradicted_precision	contradicted_recall	contradicted_f1
LLM_NLI_BM25	**67.18**	**74.36**	**67.74**	3.61	**7.73**	4.57
LLM_BM25	66.75	67.46	64.1	**3.95**	7.6	**4.77**

Table 1: Automatic evaluation results of Task A systems

run_id	Strict Support		Strict Contradict		Relaxed Support		Relaxed Contradict	
	Prec.	Recall	Prec.	Recall	Prec.	Recall	Prec.	Recall
LLM_BM25	41.67	43.33	0	0	68.33	70	0	0

Table 2: Human evaluation result of Task A Priority system

Run ID	Precision	Recall	Completeness	Correctness
MedhopQA_FAISS	86.30	**37.56**	78.06	67.45
simpleQA_BM25	91.23	35.59	75.97	66.42
MedHopQA_BM25	87.91	37.02	76.07	**68.39**
simpleQA_Hybrid	**92.15**	35.16	**82.67**	67.50

Table 3: Automatic evaluation results for Answer quality of Task B systems

Run ID	Citation Coverage	Citation Support Rate	Citation Contradict Rate
MedhopQA_FAISS	93.30	91.99	0.59
simpleQA_BM25	91.92	80.00	1.79
MedHopQA_BM25	93.21	**93.88**	**0.19**
simpleQA_Hybrid	**96.10**	82.81	2.26

Table 4: Automatic evaluation results for citation quality of Task B systems

6 Analysis

A surface-level comparison shows that the MedHopQA systems generally produced longer answers than the SimpleQA systems. This is expected, as the MedHopQA pipeline must generate a detailed long-form explanation to justify each short answer. However, the responses were often presented at a clinical, rather than patient-friendly, level. They also tended to be poorly aligned with the narrative, likely because the MedHopQA pipeline uses only the question and does not incorporate narrative or topic information. The evaluation of the completeness score penalizes information overload and irrelevance of the answer [3], which was observed in the MedHopQA systems (Table 3) due to the lack of narrative and topic information, and the generation of long, detailed answers.

In contrast, the SimpleQA systems produced answers that were more narrative-aligned, demonstrating the benefit of using the narrative during retrieval, reranking, and generation. However, some SimpleQA answers advised the user to consult a healthcare professional, while the task says that the patient had already contacted a provider, or where the provider themselves was asking the question. This inconsistency indicates that future prompts may require further fine-tuning.

The type of retriever used influenced answer quality, with the hybrid retriever approach consistently improving completeness and achieving competitive correctness scores compared to BM25.

Additionally, the SimpleQA system, which incorporates a reranker, achieved the highest precision scores among all systems.This suggests that reranking improves the quality of retrieved documents by prioritizing the most relevant and contextually aligned evidence. By providing higher-quality inputs to the LLM, reranking helps reduce unsupported or incorrect information in the generated answers, thereby improving answer precision in the RAG pipeline.

The citation evaluation results show that both retrieval strategy and reranking influence evidence grounding quality (Table 4). In the MedHopQA setup, BM25 achieved the highest citation support rate and lowest contradiction rate, indicating more precise alignment between claims and retrieved documents, likely due to its strict lexical matching and direct use of retrieved evidence. In contrast, hybrid retrieval achieved higher citation coverage, reflecting broader evidence retrieval. However, in the simpleQA setup, where reranking is also applied before answer generation, BM25 did not

achieve the highest support rate, suggesting that reranking alters evidence selection and citation attribution. These results highlight the interaction between retriever precision and reranker-based refinement in determining citation quality in RAG systems.

For task A, all the submitted systems performed poorly in identifying contradicting citations, as shown in Table 1. Notably, the NLI-based system performed worse than the non-NLI system. This is unexpected, as the additional NLI-based filtering step was intended to improve citation verification before the LLM stage. Instead, the observed decline in performance suggests that the NLI filtering component may not be sufficiently reliable for detecting contradictions, highlighting the need for a more robust pipeline for contradicting citation identification.

7 Future Work

All Task B systems in this paper rely on post-hoc sentence attribution. However, developing a system that generates answers and performs attribution in a single step would allow for a more direct evaluation of the robustness of the post-hoc approach. Since citation attribution by the LLM may result in suboptimal performance, a LoRA-based fine-tuning strategy using citation attribution datasets, including last year's TREC 2024 BioGen submissions, might help the LLM identify more accurate citations.

8 Conclusion

Across the systems developed for Task B, we observed clear differences in answer quality, narrative alignment, and evidence grounding driven by retrieval strategy and reranking. The SimpleQA systems produced more narrative-aligned and precise responses, demonstrating the benefit of conditioning retrieval, reranking, and generation on narrative and topic information. In contrast, the MedHopQA systems generated longer, clinically detailed answers that were often misaligned with the narrative and penalized in completeness due to information overload and lack of narrative conditioning. Notably, although MedHopQA was originally designed for a different shared task focused on multi-hop biomedical reasoning, it achieved competitive and, in some cases, superior performance across several answer quality and citation reliability metrics, highlighting its robustness and effective adaptation to the TREC BioGen cross-evaluation setting.

Retriever choice significantly influenced answer completeness and citation behavior. Hybrid retrieval consistently improved completeness and citation coverage, reflecting its ability to retrieve a broader range of relevant evidence. In contrast, BM25 achieved higher citation support rates and lower contradiction rates in the MedHopQA setup, indicating more precise alignment between claims and retrieved documents due to strict lexical matching. The use of reranking in the SimpleQA systems improved answer precision and altered citation attribution, highlighting the interaction between retriever coverage and reranker-based refinement in determining answer accuracy and evidence grounding.

For Task A, all systems struggled with contradiction attribution, and the NLI-based filtering approach did not improve performance over the non-NLI system. This suggests that contradiction detection remains a challenging problem and requires more robust retrieval, filtering, and attribution strategies. Overall, our findings demonstrate that narrative-aware retrieval and reranking improve answer relevance and precision, while retrieval precision and semantic filtering remain critical for reliable citation grounding in biomedical RAG systems.

Acknowledgments

We would like to express our sincere gratitude to Narjes Tahaei[2] and Aleksandr Vinokhodov[3] for their invaluable assistance throughout this project. Their support and insightful feedback greatly contributed to the quality and completion of this work.

References

[1] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. 2024.

[2] Petros Eleftheriadis, Isidoros Perikos, and Ioannis Hatzilygeroudis. Evaluating deep learning techniques for natural language inference. *Applied Sciences*, 13(4):2577, 2023.

[3] Deepak Gupta, Davis Bartels, and Dina Demner-Fushman. Bioace: An automated framework for biomedical answer and citation evaluations. 2026.

[4] Rezarta Islamaj, Salvador Lima López, Dongfang Xu, Mhd Wesam Al-Nabki, Joey Chan, Martin Krallinger, Graciela Gonzalez Hernandez, and Zhiyong Lu. Proceedings of the biocreative ix challenge and workshop (bc9): Large language models for clinical and biomedical nlp at the international joint conference on artificial intelligence (ijcai). Zenodo, 2025.

[5] 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. Mistral 7b, 2023.

[6] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. *CoRR*, abs/2004.12832, 2020.

[7] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021.

[8] Joaquín Pérez-Iglesias, José R Pérez-Agüera, Víctor Fresno, and Yuval Z Feinstein. Integrating the probabilistic models bm25/bm25f into lucene. *arXiv preprint arXiv:0911.5046*, 2009.

[9] Long N Phan, James T Anibal, Hieu Tran, Shaurya Chanana, Erol Bahadroglu, Alec Peltekian, and Grégoire Altan-Bonnet. Scifive: a text-to-text transformer model for biomedical literature. *arXiv preprint arXiv:2106.03598*, 2021.

[10] Alexey Romanov and Chaitanya Shivade. Lessons from natural language inference in the clinical domain. *arXiv preprint arXiv:1808.06752*, 2018.

[11] Harikrishnan Gurushankar Saisudha, Ganesh Chandrasekar, and Sabine Bergler. Agentic and non-agentic multi-hop systems for medical question answering. Zenodo, 2025.

[2] narjesossadat.tahaei@mail.concordia.ca
[3] a_vinokh@live.concordia.ca

[12] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

A System 1 & 2 Prompt

Code Snippet 1: Supporting Citations Prompt

```
You are a medical citation expert. Your task is to provide supporting citations for a given
     sentence based on a provided list of PubMed documents.

---
**CORE INSTRUCTIONS:**
1.  **STRICTLY CITE A MAXIMUM OF 3 DOCUMENTS.** Do not, under any circumstances, provide
    more than three citations.
2.  Your final output must be in the format: 'Supporting Citations: [<PMID_1>, <PMID_2>]'.
    No other text, explanations, or formatting is allowed.

---
**PROCEDURE:**
1.  Carefully read the provided sentence and identify its factual claims. You ** MUST
    provide at least one citation **
2.  Review the content of each document to find supporting evidence for these claims.
3.  Select the **most relevant** documents. If more than three are relevant, prioritize the
     three that provide the strongest or most comprehensive support.
4.  Construct the final output based on the format specified above.


---
**EXAMPLE:**
* **Documents:**
    * PMID 33659106: [Content on lifestyle changes and OSA]
    * PMID 36617387: [Content on lifestyle changes and snoring/OSA]
    * PMID 32385263: [Content on osteoblast differentiation, irrelevant]
* **Sentence:** Lifestyle changes may prevent and treat sleep apnea.
* **Supporting Citations:** [33659106, 36617387]

---
**YOUR TASK:**

* **Documents:**
    {docs}

* **Sentence:**
    {sentence}
** Return only the list of supporting citations. DO NOT RETURN ANYTHING ELSE **
* **Output:**
```

Code Snippet 2: Contradicting Citations Prompt

```
You are a medical citation expert. Your task is to identify and cite documents that
    contradict a given sentence.

**Constraint Checklist & Instructions:**
```

```
1.  **Analyze the Sentence:** Carefully read the sentence and identify its primary factual
    claim.
2.  **Scan Documents:** Search the provided documents for information that directly refutes
    , opposes, or contradicts this claim.
3.  **Strict Citation Limit:** Your output must contain a **maximum of three** PMID
    citations.
4.  **No Contradictions Found:** If you cannot find any documents that contradict the
    sentence, your output must be an empty list: `[]`. Do not provide any other text or
    explanation.
5.  **Output Format:** Your final output must be a single line containing only the
    citations in a list format, for example: `["PMID_1", "PMID_2"]`. Do not include any
    other text, explanations, or formatting.
6.  **Prioritize Relevance:** If more than three documents contradict the sentence, select
    the three that provide the strongest and most direct refutation.

---

**Provided Information:**
* **Documents:**
    {docs}
* **Question:**
    {question}
* **Sentence:**
    {sentence}

**Your Task:**
Identify the PMIDs of up to three documents that contradict the given sentence. If no
    contradictions are found, return an empty list.

**Output:**
```

B Question Decomposition Prompt

Code Snippet 3: Question Decomposition Prompt

```
You are a biomedical queries extracting agent.

You are given a question and you need to identify all the possible queries that will be
    used to search PubMed via BM25.
Most of the question requires atleast finding one query.

Using the instruction below, generate a answer:

Step 1:
Identify the entities from the question. Form the queries using these enities.

Consider the following example for extracting queries:
Example:
Question: Why are transferrin and iron low in COVID patients, and ferritin high?
Possible Queries:  ["COVID-19 low transferrin low iron high ferritin", "COVID-19 ferritin",
     "COVID-19 iron metabolism","COVID-19 transferrin","SARS-CoV-2 iron","COVID-19
    ferritin"]

--------------------------------------------------------------------------------
Step 2:
Return the queries in the following format:
query: ["<query 1>","<query 2>","<query 3>","<query 4>","<query 5>"]
** Do not return anything other than the output in the specified format **


With the above instructions, think step-by-step and generate ** upto 5 keyword sepcific
    queries ** for the below question below and return the list.


Question: {question}


output:
```

C System 3 & 4 Prompts

Code Snippet 4: Answer Generation Prompt

```
You are a medical question-answering expert.

You are given relevant documents from PubMed, topic, narrative and a question.

Using the instruction below, generate an answer:

Step 1:
Find relevant information from the documents for the given question.

Step 2:
With all these relevant information, generate an answer containing upto * 250 words *.

Generate a Answer  a clear, medically grounded explanation with the relevant information
    from context. ** The Answer cannot be more than 250 words **


Here is an example for generating answers:

Example:

Question: Are there ways to prevent sleep apnea or treat it naturally?

Likely Answer: There are ways to prevent and treat sleep apnea naturally. Lifestyle changes
     may prevent and treat sleep apnea. Sleep apnea can be prevented by losing weight and
    keeping it down with diet and exercise; quitting alcohol and smoking; and changing
    sleep position. While Continuous Positive Airway Pressure (CPAP, a machine that uses
    mild air pressure to keep breathing airways open while you sleep) is the standard
    treatment for obstructive sleep apnea (OSA), Chinese massage Tui Na, dental treatments
     to change teeth and jaw position; and exercises for tongue and throat reduce snoring
    and apnea. Treatments with drugs, nerve stimulation and surgery were also suggested.



Step 3:
Return the answer in the following format, do not return anything else:
‘output: <answer>‘

With the above instructions, think step-by-step and generate an answer for the below
    question relavant to the ** narrative and topic ** using the PubMed documents given
    below.
** Do not exceed more than 250 words, don’t answer in points, keep it concise and brief
    under 250 words, produce only the answer **
** The documents are presented in order of relevance to the question. **

Documents:
{documents}
```

```
Topic: {topic}
Narrative: {narrative}
Question: {question}

** Return only the ouput in the specified format **
```

Code Snippet 5: Answer Condensation Prompt

```
You are a medical question-answering summarizing expert.

You are given long answer and a question.

Using the instruction below, generate upto 250 words summary:

Step 1:
Find relevant information from the long answer for the given question. If the ** answer is
    already less than 250 words **, return the ** original answer **

Step 2:
Use the relevant information from the long answer and produce a summary relevant to the
    question. This must integrate information seamlessly and generate a concise answer **
    less than 250 words ** adressing the input question with all the biomedical
    information required.

** The long answer will always have information related to the question **




Step 3:
Return the summary in the following format, do not return anything else:
‘output: <summary>‘

With the above instructions, generate a summary for the below question using the long
    answer given below.
Long Answer:
{long_answer}




Question: {question}

** Return only the output in the specified format **
```