MedRAG-2

Ahmad Sait
KAUST
ahmed.sait@kaust.edu.sa

Wenxuan Zhu
KAUST
wenxuan.zhu@kaust.edu.sa

Aznaur Aliev
KAUST
aznaur.aliev@kaust.edu.sa

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in medical question answering, but their deployment remains hindered by hallucinations, unstructured outputs, and reliance on static knowledge bases. In this work, we present MedRAG-2, an enhanced Retrieval-Augmented Generation pipeline that addresses these challenges through prompt redesign enforcing strict grounding and fallback logic, structured output validation using Pydantic schemas, and cross-encoder based re-ranking for improved retrieval precision. Evaluations across multiple medical QA benchmarks using LLaMA 3-8B and Qwen 3-8B show that MedRAG-2 significantly improves answer accuracy, traceability, and reliability over prior methods. Additionally, we propose a search-augmented QA system that leverages real-time web evidence to further enhance factuality and timeliness. Our framework offers a scalable solution for deploying LLMs in clinical settings with improved trustworthiness and performance.

1 Introduction

Recent studies have shown that Large Language Models (LLMs), when used in medical settings such as diagnosis or clinical decision support, often exhibit a critical limitation: they hallucinate. That is, they generate confident but factually incorrect or misleading outputs, especially when faced with queries outside the scope of their training data. This is particularly dangerous in medicine, where incorrect information can have serious consequences.

The issue is exacerbated in clinical domains where data is sparse (rare diseases) and temporally evolving (new diseases are introduced) - conditions under which LLMs struggle due to the lack of grounding in real-time or structured domain knowledge (e.g., from EHRs or clinical databases). As a result, there is an urgent need for methods that enhance the factual accuracy of LLMs in medical tasks by grounding their outputs in trusted sources.

While recent work has attempted to address these limitations through Retrieval-Augmented Generation (RAG) systems that integrate biomedical corpora or knowledge graphs, existing approaches remain inefficient, brittle, and prone to hallucinations due to poor prompt design, redundant retrieval, and unstructured outputs. In this work, we propose **MedRAG-2**, a more robust medical RAG pipeline that improves grounding and accuracy through prompt enforcement, structured output with Pydantic, cross-encoder re-ranking, and real-time search-augmented QA.

Code for MedRAG-2 are publicly available at: `https://github.com/AhmadSait/MedRAG-2`

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

2 Background

To address hallucinations and lack of factual grounding in LLMs, several recent methods have emerged in the medical Retrieval-Augmented Generation (RAG) domain. These approaches integrate structured biomedical knowledge, either as knowledge graphs or as curated corpora and snippet-level evidence retrieved from biomedical literature, clinical textbooks, and decision-support tools - as seen in systems like MedGraphRAG [1] and MedRAG [2] - into the language model pipeline to improve accuracy and trustworthiness.

2.1 MedGraphRAG

MedGraphRAG introduces a triple-linked graph structure that connects user data, trusted clinical knowledge, and controlled vocabularies. This structure supports evidence-based medical reasoning by reducing hallucination risks during response generation.

For MedGraphRAG's implementation, we have found major inefficiencies while reproducing their code:

Excessive LLM API Calls

`agentic_chunker.py` repeatedly calls an LLM to summarize each chunk (`_get_new_chunk_summary`), to title each chunk, to update the chunk metadata with each new proposition (e.g., "The patient was administered 500mg of acetaminophen."), and to decide whether a proposition fits an existing chunk (`_find_relevant_chunk`). This results in many redundant LLM calls per document, leading to high token usage and increased API costs.

Double LLM Passes for Proposition Extraction

`data_chunk.py` extracts propositions by first using a LangChain hub chain (`wfh/proposal-indexing`), then piping that output into the LLM again for sentence-level extraction, followed by a separate structured parser. This leads to multiple GPT-4 calls per paragraph, compounding token usage and increasing cost.

Redundant Graph Construction Calls

`creat_graph.py` builds a Neo4j graph by calling the LLM twice through `KnowledgeGraphAgent.run()` (once for freeform output, once for structured), then embedding each node individually via OpenAI and assigning IDs. This creates a new graph for every chunk, with repeated LLM and embedding API calls that significantly increase runtime and cost.

We made several attempts to reproduce MedGraphRAG's code. Despite numerous bug fixes and lack of open-source corpora, we estimated that it would approximately take 42 days to run their code on a single A100 with 10 workers for API parallelization, processing 48000 MIMIC files. Table 1 shows preliminary results that were obtained from running their system on only 1000 files (less than 5% of the original MIMIC dataset), which took a very time-demanding time of 62 hours. This experiment used a LLaMA-3.3-70B-Instruct-AWQ-INT4 as the knowledge graph agent and LLaMA-3-8B for evaluation.

Table 1: Comparison Between Our MedGraphRAG Attempt and LLM-only Accuracy in %. The bottom row shows the absolute improvement of LLM-only over our approach.

	MMLU	**MedQA**	**MedMCQA**	**PubMedQA**	**BioASQ**
Our attempt	64.37	56.17	47.62	30.20	51.46
LLM-only Accuracy	**71.90**	**59.94**	**58.79**	**58.60**	**75.89**
(Ours - LLM)	-7.53	-3.77	-11.17	-28.40	-24.43

2.2 MedRAG

MedRAG combines multiple domain-specific retrievers (e.g., BM25, MedCPT) and corpora (e.g., PubMed, StatPearls, MedCorp) to enhance medical QA through comprehensive retrieval-augmented generation pipelines.

Despite MedRAG's robust retrieval setup, our implementation revealed several key limitations that hinder reliability and accuracy.

Ineffective Use of Large General Corpora

Table 7 in MedRAG's paper shows that Wikipedia, despite its massive size (29.9M snippets), consistently underperforms across nearly all MIRAGE tasks. This confirms that bigger isn't better if domain relevance is weak. Their choice to include Wikipedia dilutes the relevance of retrieved contexts, adding noise and harming precision.

Prompt Limitations and Output Failures

Despite operating in a high-stakes medical domain, MedRAG employs a loosely defined prompt that merely instructs the model to "Think step-by-step and generate the output in JSON." However, this minimal instruction lacks constraints on evidence citation, structured format, or fallback behavior. Prior research, including the StructuredRAG [3] benchmark (Shorten et al., 2024), has shown that such prompt-only approaches are insufficient to guarantee parseable or logically sound output. These prompt limitations manifest in several ways:

Hallucinations and Redundancy. In our experiments, the model frequently hallucinated citations or produced verbose, redundant outputs. One common failure mode involved the model repeating the same document citation or reasoning statement indefinitely, preventing it from completing a response.

Example 1: Infinite citation loop

> "Document [1] lists the clinical consequences of lesions in the cavernous sinus, including facial paralysis, which may be caused by compression of the facial nerve at the stylomastoid foramen."
>
> (This sentence was repeated indefinitely; the model failed to progress to the next question.)

Example 2: Redundant reasoning

> "The patient's T-score is below -2.5, which indicates osteoporosis. Therefore, pharmacotherapy is indicated."
>
> (This rationale was repeated with minor wording changes across multiple steps.)
>
> "According to the guidelines… patients be considered for treatment when BMD is >2.5 SD below the mean…"
>
> (The same guideline was reused repeatedly with slight rephrasing.)

Unstructured and Unreliable Output. Even when prompted for structured output, the model frequently generated malformed JSON, ambiguous final answers, or inconsistent key formats—e.g., responses like "The correct answer is C" embedded in free-form text. These outputs often broke downstream parsers or required manual post-processing.

Example 3: Ambiguous free-form answer

> "The salivary glands are derived from ectomesenchyme. But the sweat glands are not. So if the options include sweat glands, then D is not correct. So the answer would be C. Melanocytes are derived from the neural crest, which is ectomesenchyme. Therefore, the correct answer is C."

Together, these failures - repetitive hallucination, unstructured output, and lack of fallback handling -significantly degrade MedRAG's reliability, particularly in clinical QA settings where traceability and precision are essential.

3 Our System

We propose several solutions to solve MedRAG's limitations:

3.1 Corpus Selection

To address the limitations observed in MedRAG's use of large, general-purpose corpora like Wikipedia, our system deliberately avoids aggregating broad sources with low domain specificity. Instead, we prioritize targeted medical datasets such as MedCorp, which is a combination of three datasets - PubMed, StatPearls, and medical textbooks - which offer higher semantic relevance and factual density to biomedicine. This improves retrieval precision and reduces the noise introduced by irrelevant or loosely related documents like Wikipedia.

3.2 Prompt Redesign to Prevent Hallucinations

We observed frequent hallucination patterns in MedRAG, particularly repetitive loops where the model recycled the same set of document citations without reaching a definitive answer. This behavior stems from shortcomings in MedRAG's prompt, which offers no constraints on evidence citation, factual grounding, or fallback logic.

MedRAG's prompt suffers from three critical limitations: (1) it provides no rules for how retrieved documents should be cited, leading to repeated or ambiguous references; (2) it lacks a fallback mechanism when no supporting evidence is found, causing the model to guess or stall; and (3) it loosely encourages a "definite answer" without enforcing factual accuracy, allowing for speculation or fabricated content. Our redesigned prompt directly addresses these gaps. It enforces a one-to-one grounding policy, introduces explicit fallback logic, and prohibits hallucinations through the constraint: "Do not invent new facts." These changes promote factual consistency, prevent citation inflation, and ensure traceable, grounded reasoning.

Figure 1 shows MedRAG's original prompt, which exhibits these shortcomings in structure, grounding, and citation policy.

To address these issues, we designed a new prompt enforcing three key constraints:

- **One-to-one grounding**: Each reasoning step must cite a distinct document or fallback to prior knowledge.
- **Fallback logic**: If no document supports the answer, the model must explicitly state this and stop citing.
- **Anti-hallucination policy**: The prompt includes a hard constraint: *"Do not invent new facts."*

These changes ensure factual consistency, prevent citation inflation, and enforce clear attribution throughout the reasoning process. Figure 2 presents our redesigned prompt.

To evaluate its impact, we tested our prompt under both LLaMA 3-8B and Qwen 3-8B configurations. As shown in Table 2, LLaMA gains up to +4% accuracy on MedMCQA and 9% on PubMedQA benchmarks. Table 3 shows consistent improvements on Qwen 3-8B, particularly in datasets prone to citation ambiguity, like PubMedQA and BioASQ.

Table 2: Effect of Our Prompt on LLaMA 3-8B (Accuracy in %). Structured prompt improves performance, particularly in MedMCQA and PubMed QA.

k	Dataset	Enhanced	CoT/MedRAG	MMLU	MedQA	MedMCQA	PubMed QA	BioASQ	Avg
N/A	N/A	No	CoT	**66.20**	55.02	47.90	42.07	61.50	54.52
16	MedCorp	No	MedRAG	65.01	54.05	47.26	40.20	60.84	53.47
16	MedCorp	Yes	MedRAG	64.74	**55.38**	**51.33**	**49.60**	**67.96**	**57.80**
			(Enhanced - Baseline)	-0.27	+1.33	+4.07	+9.40	+7.12	+4.33

These results confirm that enforcing grounding, fallback, and anti-hallucination constraints at the prompt level leads to more reliable and accurate responses across multiple medical benchmarks.

```
general_medrag_system = '''
You are a helpful medical expert,
and your task is to answer a multi-choice medical question using the relevant documents.
Please first think step-by-step and then choose the answer from the provided options.
Organize your output in a json formatted as Dict{"step_by_step_thinking": Str(explanation),
"answer_choice": Str{A/B/C/...}}.
Your responses will be used for research purposes only, so please have a definite answer.
'''

general_medrag = Template('''
Here are the relevant documents:
{{context}}

Here is the question:
{{question}}

Here are the potential choices:
{{options}}

Please think step-by-step and generate your output in json:
''')
```

Figure 1: MedRAG's original prompt

Table 3: Effect of Our Prompt on Qwen3-8B (Accuracy in %). Structured prompt improves performance, particularly in MedMCQA and PubMed QA.

k	Dataset	Enhanced	CoT or MedRAG	MMLU	MedQA	MedMCQA	PubMed QA	BioASQ	Avg
N/A	N/A	No	CoT	**82.80**	**67.46**	55.67	36.60	70.44	62.59
16	MedCorp	No	MedRAG	82.34	67.10	55.61	35.40	70.56	62.20
16	MedCorp	Yes	MedRAG	77.09	64.63	**56.79**	**41.08**	**74.12**	**62.74**
			(Enhanced - Baseline)	-5.25	-2.47	+1.18	+5.68	+3.56	+0.54

3.3 Structured Output with Pydantic

By integrating Pydantic, we enforced strict adherence to a predefined schema, ensuring that each model output was both syntactically valid and semantically unambiguous. This substantially improved evaluation reliability and simplified result aggregation. Figure 3 illustrates our Pydantic integration for structured response enforcement.

As shown in Table 4, incorporating structured output led to consistent performance improvements across all benchmarks for the LLaMA 3-8B model, with gains of up to almost 11 percentage points on PubMedQA, almost 6 points on BioASQ, around 5% on MedMCQA, and an overall average increase of 4.7 percentage points from MedRAG's implementation.

We replicated this experiment using Qwen 3-8B to test generalizability. As shown in Table 5, structured output again resulted in higher accuracy, particularly in datasets with longer, more complex answers like MedMCQA and BioASQ.

````python
general_medrag_system = '''
You are a meticulous, evidence-based medical question-answering assistant.
Answer using **only** the provided documents—or exactly one standard fact if none suffice.

Follow these rules **without exception**:

1. **Grounding**
   - Cite each document **once** and only once.
   - As soon as you’ve used “Document X,” you may never cite it again.

2. **Fallback**
   - If no document gives the answer, preface with
     “Applying standard knowledge: [fact]”
   - Then stop using documents entirely.

3. **Step-By-Step (Max 3 Steps)**
   - Use **at most three** numbered steps.
   - Each step must reference a **different** document or the one standard fact.
   - **Immediately** after your final step, stop reasoning.

4. **Immediate JSON Output**
   - Without any extra text, output **only** this JSON:
     ```json
     {
       "step_by_step_thinking": "up to 3 numbered steps…",
       "answer_choice": "<one of 'A','B','C','D'>"
     }
     ```
   - Any deviation (extra keys, additional sentences, repeats) → your answer is dropped.

5. **No Hallucinations**
   - Do not invent new facts.
   - Do not exceed the 3-step limit.
'''

general_medrag = Template('''
Here are the relevant documents:
{{context}}

Question:
{{question}}

Choices:
{{options}}

Please follow the system instructions exactly:
1. Ground each step in “Document X” or in one standard fact.
2. Number your reasoning steps clearly; avoid repetition.
3. Do not speculate or invent new details.
4. Output **only** this JSON object:

{
  "step_by_step_thinking": "...",
  "answer_choice": "<one of 'A','B','C','D'>"
}
''')
````

Figure 2: Our redesigned prompt with grounding constraints and fallback logic

Table 4: Effect of Structured Output on LLaMA 3-8B Performance (Accuracy in %)

k	Dataset	Structured	CoT/MedRAG	MMLU	MedQA	MedMCQA	PubMed QA	BioASQ	Avg
N/A	N/A	No	CoT	66.20	**55.02**	47.90	42.07	61.50	54.52
N/A	N/A	Yes	CoT	66.10	54.91	51.87	51.03	**67.50**	**58.28**
16	MedCorp	No	MedRAG	65.01	54.05	47.26	40.20	60.84	53.47
16	MedCorp	Yes	MedRAG	**66.82**	53.71	**52.34**	**51.13**	66.75	58.15
			(Structured - Baseline)	+1.81	-0.34	+5.08	+10.93	+5.91	+4.68

Together, these results highlight the robustness and portability of schema-constrained output generation, improving both evaluation stability and overall accuracy in retrieval-augmented medical question answering.

3.4 Re-ranking

To further improve the precision of document retrieval in our medical RAG pipeline, we incorporate a cross-encoder-based re-ranking stage, following the retrieval pipeline design explored in recent dense and hybrid retrieval systems [5, 6]. While bi-encoder retrievers such as MedCPT and SPECTER provide efficient initial filtering, they rank documents based on independent query and document

```python
from pydantic import BaseModel
from enum import Enum

class AnswerChoice(str, Enum):
    A = "A"
    B = "B"
    C = "C"
    D = "D"

class MedRAgOutput(BaseModel):
    step_by_step_thinking: str
    answer_choice: AnswerChoice

json_schema = MedRAgOutput.model_json_schema()

def openai_client(messages, **kwargs):
    completion = oai_client.chat.completions.create(
        messages=messages,
        seed=42,
        timeout=150,
        max_tokens=4096,
        extra_body={"guided_json": json_schema},
        **kwargs
    )
    choice = completion.choices[0].message
    return choice.reasoning_content
```

Figure 3: Integration of Pydantic for Reliable Structured Output

Table 5: Effect of Structured Output on Qwen 3-8B Performance (Accuracy in %)

k	Dataset	Structured	CoT/MedRAG	MMLU	MedQA	MedMCQA	PubMed QA	BioASQ	Avg
N/A	N/A	No	CoT	**82.80**	**67.46**	55.67	36.60	70.44	62.59
N/A	N/A	Yes	CoT	79.69	64.88	58.10	**42.78**	**74.52**	**63.99**
16	MedCorp	No	MedRAG	82.34	67.10	55.61	35.40	70.56	62.20
16	MedCorp	Yes	MedRAG	80.32	63.27	**58.45**	42.73	73.47	63.65
			(Structured - Baseline)	-2.02	-3.83	+2.84	+7.33	+2.91	+1.45

embeddings. This approach struggles with nuanced semantic dependencies such as negation, causality, or contradiction that are prevalent in medical literature. Cross-encoders address this by jointly encoding the `(question, document)` pair, enabling fine-grained, context-aware relevance scoring.

The pipeline proceeds in three stages:

- The query is encoded using a bi-encoder and compared against a FAISS index of document embeddings. Multiple retrievers (e.g., BM25 and MedCPT) can be fused using Reciprocal Rank Fusion (RRF) to return top-k candidate snippets.
- Each retrieved document is paired with the original query and passed through a cross-encoder model, which outputs a relevance score. The documents are then sorted by this score, yielding a more semantically aligned top-k set.

- The reranked snippets are used for downstream answer generation. Since these documents are better aligned with the intent of the question, they improve the grounding and accuracy of the language model's responses.

This method allows the model to disambiguate subtle medical nuances. For instance, given the question "What hormone lowers blood glucose?", bi-encoders might surface documents about both insulin and glucagon due to lexical overlap. However, the cross-encoder correctly elevates insulin, the only truly relevant answer, by jointly reasoning over the full statement.

As shown in Table 6, integrating re-ranking into our Qwen 3-8B pipeline led to consistent improvements in key QA benchmarks, demonstrating the benefit of context-sensitive scoring.

Table 6: Effect of Re-ranking on Qwen 3-8B with Our Prompt and Structured Output (Accuracy in %). Experiments use MedCorp (k=16) under the MedRAG setting.

Enhanced	Structured	Re-ranking	MMLU	MedQA	MedMCQA	PubMedQA	BioASQ	Avg
No	No	No	82.34	67.10	55.61	35.40	70.56	62.20
Yes	Yes	No	82.25	**64.60**	**59.88**	44.05	74.17	64.99
Yes	Yes	Yes	**82.81**	64.41	59.59	**44.43**	**74.69**	**65.19**
		(Re-ranked - Baseline)	+0.47	-2.69	+3.98	+9.03	+4.13	+2.99

These results confirm that even modest reordering of retrieved documents using semantic relevance contributes to improved grounding and accuracy in complex medical QA tasks.

3.5 Search-Augmented Medical Question Answering

We propose a system that enhances Large Language Models (LLMs) for medical question answering by combining strategic tool use with search-augmented generation. Rather than relying solely on the model's internal knowledge, our approach leverages web search to inject up-to-date and contextually relevant medical information into the answering pipeline. Central to our method is the generation of optimized search queries that are more effective than directly using the original question.

In traditional retrieval-augmented medical QA, using the raw question as a search query often leads to suboptimal retrieval. It may return copies of the question itself, overlook critical medical terminology, or miss relevant answer choice concepts. To address this, we prompt the Qwen2.5-7B-Instruct model to reformulate the input question into a search-optimized query. The prompt, seen in 4 guides the model to:

- Extract core medical entities and concepts
- Identify clinically relevant relationships
- Eliminate question-specific or redundant phrasing
- Preserve domain-specific terminology
- Generate concise and targeted queries suitable for retrieval

This reformulated query is then used to retrieve external evidence, which is fed back into the LLM for answer generation. Our experimental results show that this query optimization and retrieval-aware prompting pipeline leads to measurable gains in accuracy and reliability on medical QA benchmarks.

We assess the effectiveness of this query optimization pipeline in the Evaluation section, where we show that search-augmented prompting improves answer accuracy across multiple datasets.

4 Evaluation

4.1 MedRAG vs MedRAG-2

We first assess the combined impact of our redesigned prompt and structured output enforcement. Table 7 presents the accuracy of LLaMA 3-8B and Qwen 3-8B across five medical QA benchmarks. We compare the baseline chain-of-thought (CoT) prompting, the MedRAG baseline, and our proposed enhancements—applied individually and jointly.

```
prompt = f"""
You are an expert medical search query generator. Your task is to analyze medical multiple-choice
questions and transform them into effective search queries that will retrieve relevant information.

INSTRUCTIONS:
1. Analyze the medical question and all answer choices provided.
2. Extract key medical concepts, conditions, treatments, or relationships mentioned in both the
   question and answer choices.
3. Formulate a search query that:
    - Includes the core medical concept from the question
    - Incorporates important medical terms from the answer choices
    - Uses proper medical terminology
    - Avoids simply repeating the question verbatim
    - Is specific enough to return relevant medical information
    - Is between 5-15 words in length
4. For questions about relationships (e.g., "Which medication is contraindicated for condition X?"),
   make sure to include both elements in your search query.
5. Remove question-specific language like "which of the following" or "what is the best" from your
   search query.
6. Format your output as a concise search query ready to be sent to Tavily Search.
7. Here is the format of the output, return the JSON response, no additional text:
{{
    'search_query': 'xxxx',
}}

EXAMPLE INPUT:
Question: What is the first-line treatment for uncomplicated urinary tract infection in non-pregnant
women?
Options:
A. Nitrofurantoin
B. Ciprofloxacin
C. Amoxicillin
D. Trimethoprim-sulfamethoxazole

EXAMPLE OUTPUT:
{{
'search_query': 'first-line treatment uncomplicated UTI non-pregnant women nitrofurantoin ciprofloxacin
amoxicillin trimethoprim-sulfamethoxazole antibiotics',
}}

Now, please generate the search query for the following question:
Question: {question}
Options: {options}
"""
```

Figure 4: Prompt to reformulate the input question into a search-optimized query for input to Tavily-Search.

Table 7: Ablation Study: Accuracy (%) of LLaMA 3-8B and Qwen 3-8B on MedCorp (k=16) using the MedRAG setup. We evaluate the effect of enhanced prompting, structured output, and cross-encoder re-ranking across five medical QA datasets.

LLM	Enhanced	Structured	Re-ranked	MMLU	MedQA	MedMCQA	PubMed QA	BioASQ	Avg
LLaMA 3-8B	No	No	No	65.01	54.05	47.26	40.20	60.84	53.47
LLaMA 3-8B	Yes	Yes	No	68.04	54.52	53.31	**52.40**	66.99	59.05
LLaMA 3-8B	Yes	No	No	64.74	55.38	51.33	49.60	67.96	57.80
LLaMA 3-8B	No	Yes	No	66.82	53.71	52.34	51.13	66.75	58.15
Qwen 3-8B	No	No	No	82.34	**67.10**	55.61	35.40	70.56	62.20
Qwen 3-8B	Yes	No	No	77.09	64.63	56.79	41.08	74.12	62.74
Qwen 3-8B	No	Yes	No	80.32	63.27	58.45	42.73	73.47	63.65
Qwen 3-8B	Yes	Yes	No	82.25	64.60	**59.88**	44.05	74.17	64.99
Qwen 3-8B	Yes	Yes	Yes	**82.81**	64.41	59.59	44.43	**74.69**	**65.19**

The best results occur when all enhancements—prompt redesign, structured output enforcement, and cross-encoder re-ranking—are applied, with Qwen 3-8B achieving an average accuracy of 65.19%.

4.2 MedRAG-2 vs Search-Augmented QA

Table 8 compares MedRAG-2, which incorporates our improvements and re-ranking, against our Search-Engine-Augmented QA approach. Both methods were evaluated on a randomly sampled subset of 500 questions, equally distributed across the five medical QA datasets.

Table 8: Comparing Accuracies (%) of Qwen2.5-7B with a MedRAG-2 and Search-Engine-Augmented QA setups across a random subset of 500 questions (100 questions from each medical QA dataset.)

k	Dataset	Enhanced	Structured	Re-ranked	MMLU	MedQA	MedMCQA	PubMedQA	BioASQ	Avg
16	MedCorp	Yes	Yes	Yes	68.00	**70.00**	57.00	**95.00**	85.00	74.80
N/A	N/A	N/A	N/A	N/A	**74.00**	66.00	**63.00**	89.00	**90.00**	**76.40**
	(Search-Engine-Augmented QA - MedRAG-2)				+6.00	-4.00	+6.00	-6.00	+5.00	+1.60

Our search-augmented approach outperforms MedRAG-2 on average by 1.60%, where real-time retrieval provides essential, up-to-date knowledge beyond static corpora.

5 Conclusion

This work presents MedRAG-2, a robust medical Retrieval-Augmented Generation system designed to address key limitations in existing pipelines, including hallucinations, unstructured outputs, and inefficient retrieval. By redesigning the prompt with strict grounding constraints, enforcing schema-based structured output via Pydantic, and integrating a cross-encoder re-ranking stage, MedRAG-2 achieves significant accuracy improvements across multiple challenging medical QA benchmarks.

Our experimental results demonstrate that these enhancements collectively improve model reliability, answer traceability, and semantic precision. Furthermore, we introduce a search-augmented QA pipeline that leverages real-time web evidence, further boosting performance by incorporating up-to-date medical knowledge.

MedRAG-2 offers a practical and scalable solution to improve factual grounding in LLM-driven clinical applications. Future work will focus on extending domain adaptation, optimizing computational efficiency, and enhancing interpretability to meet stringent clinical standards.

References

[1] Wu, J., Zhu, J., Qi, Y., Chen, J., Xu, M., Menolascina, F. & Grau, V. (2024) Medical Graph RAG: Towards safe medical large language models via graph retrieval-augmented generation. *arXiv preprint arXiv:2408.04187*. `https://arxiv.org/abs/2408.04187`

[2] Xiong, G., Jin, Q., Lu, Z., & Zhang, A. (2024) Benchmarking retrieval-augmented generation for medicine. Findings of the Association for Computational Linguistics: ACL 2024. `https://aclanthology.org/2024.findings-acl.372.pdf`

[3] Shorten, C., Pierse, C., Smith, T. B., Cardenas, E., Sharma, A., Trengrove, J., & van Luijt, B. (2024) StructuredRAG: Measuring structured response adherence in retrieval-augmented generation. *arXiv preprint arXiv:2408.11061*. `https://arxiv.org/abs/2408.11061`

[5] Khattab, O., & Zaharia, M. (2020). ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 39–48. `https://arxiv.org/abs/2004.12832`

[6] Tran, T. Q., Kang, M., & Kim, D. (2021). Dense Passage Retrieval with Progressive Training and Knowledge Transfer. arXiv preprint arXiv:2102.06328v2. `https://arxiv.org/abs/2102.06328v2`