

Retrieval-Augmented Generation for Telemedicine: A Privacy-Preserving AI Assistant for Healthcare

Dmitriy An¹, Andrew Paice¹, Christiane Brockes², Ali Sigaroudi²,
and Mareike Brockes²

¹iHomeLab, Lucerne University of Applied Sciences and Arts, Horw,
Switzerland, {dmitriy.an, andrew.paice}@hslu.ch

²alcare AG, Wil, Switzerland, christiane.brockes@alcare.ch
{christiane.brockes, ali.sigaroudi,
mareike.brockes}@alcare.ch

September 11, 2025

Abstract

The growing demand for telemedicine has intensified challenges such as doctor shortages and data privacy concerns. Generating telemedicine responses requires accuracy, clarity, and coherence, making it a time-consuming task for medical professionals. To address this, we developed a modular Retrieval-Augmented Generation system that leverages a Large Language Model to reduce doctors' workload. The system operates on-premise, eliminating the need for an internet connection and ensuring data privacy. It integrates a database of 10 million verified PubMed articles, encoded as sparse and dense embeddings, enabling a hybrid vector search to retrieve relevant information based on patient input. Retrieved articles undergo reranking to further refine results, minimizing LLM hallucinations. The system generates telemedicine responses, reducing doctors' average processing time per patient form from 13.46 to 2.82 minutes – an efficiency gain of nearly 80%. This solution demonstrates significant potential in enhancing telemedicine efficiency and alleviating medical professionals' workload. Future optimizations, such as integrating additional data sources, offer further opportunities for improvement.

1 Introduction

Telemedicine enables remote medical consultations, a demand that surged during COVID-19 [20]. However, written responses are time-intensive, requiring doctors to ensure clarity, accuracy, and coherence.

In the current workflow, patients submit forms describing their medical problem, often supplemented by relevant background information such as age and gender. At alcare AG, responding to each patient form takes on average nearly 15 minutes. To reduce this effort, alcare AG aims to semi-automate the process: instead of drafting full responses, physicians would receive AI-generated drafts that they only need to review and, if necessary, correct. This approach preserves clinical oversight while substantially reducing the time spent on documentation.

This paper explores optimizing written telemedicine using Large Language Models (LLMs). While LLMs can efficiently generate clear responses, they present challenges such as hallucinations and data privacy risks due to reliance on cloud-based models.

To address these issues, we propose a Retrieval-Augmented Generation (RAG) system that minimizes hallucinations by using only verified information. The system operates entirely on-premise, ensuring data privacy while reducing doctors' workload.

The main contributions of this work are as follows: (1) we present a privacy-preserving Retrieval-Augmented Generation (RAG) architecture for written telemedicine that runs fully on-premise and fits on a single GPU, enabling secure deployment without reliance on cloud services; and (2) we demonstrate a substantial efficiency gain of nearly 80% in physician response time when using AI-assisted draft generation. In addition, we provide a detailed evaluation of retrieval performance, including comparisons of similarity metrics, the impact of the number of retrieved articles on recall, and the influence of sparse weighting in hybrid search. Finally, the system's output was reviewed by medical professionals to assess clinical usefulness and safety.

Figure 1 illustrates the system's architecture, comprising offline and online components. The offline phase initializes the RAG system, while the online phase processes patient forms.

The paper is structured as follows: Section 2 reviews related work, Section 3 outlines the methodology and system architecture, Section 4 evaluates performance, Section 5 discusses findings and limitations, and Section 6 concludes the study.

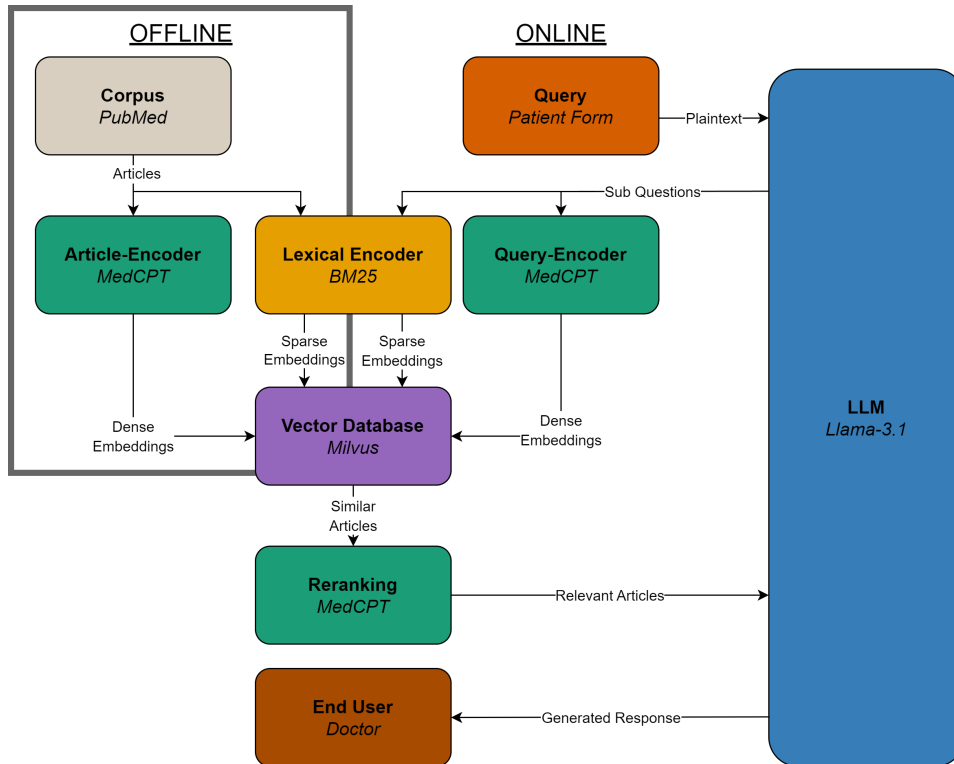


Figure 1: System architecture of the telemedicine RAG pipeline. The offline phase encodes and stores PubMed articles in a vector database, while the online phase processes patient forms, retrieves relevant articles, and generates responses using an LLM.

2 Literature Review

LLMs are increasingly used in medicine, primarily as chatbots and support tools [10]. More broadly, AI enhances telemedicine by improving patient monitoring, data management, and workflow efficiency [22]. However, it also introduces challenges, including security risks, regulatory compliance, and user acceptance [24].

Despite their potential, LLMs in telemedicine face specific challenges:

- Limited contextual understanding of medical concepts.
- Poor interpretability of reasoning behind responses.
- Bias from training data.

- Ethical concerns regarding data privacy.
- Trust bias affecting healthcare professionals.
- Lack of validation studies [26].

A key limitation is hallucination - generating misleading information. While it cannot be eliminated, methods like RAG mitigate it by integrating verified external sources, including proprietary medical data [29, 27].

Few studies have applied RAG in telemedicine. A thesis developed a RAG-powered veterinary chatbot with high accuracy and user satisfaction [9], and another study proposed a cloud-based RAG system for summarization and diagnostics, which outperformed models like GPT 4 [15]. These works highlight RAG’s potential but underscore the need for further research, particularly in human telemedicine and privacy-focused solutions.

Addressing these challenges can enhance AI-driven telemedicine, improving reliability while reducing doctors’ workload and maintaining high-quality patient care.

2.1 Comparison of ICL, Fine-Tuning, and RAG

LLMs can be specialized using three main approaches:

- **In-Context Learning (ICL)** - Guides responses by providing examples within the prompt.
- **Fine-Tuning** - Updates model parameters using additional training data.
- **RAG** - Integrates external information through database retrieval.

ICL and Fine-Tuning offer comparable improvements, but ICL is limited by context length and increased inference time. However, it requires no ML expertise and remains highly flexible [17].

Fine-tuning demands extensive labeled data and computational resources. While pre-trained models like PMC-LLaMA [28] and Medicine-Llama3-8B [2] provide domain-specific capabilities, they may lag behind state-of-the-art models.

RAG tends to outperform both In-Context Learning and Fine-Tuning in applications that require access to extensive factual or domain-specific knowledge not contained in the model’s parameters — such as question answering, summarization, and information retrieval in open-domain or knowledge-intensive settings. However, combining RAG with fine-tuned models does not always enhance performance and can sometimes degrade it [21].

Table 1 summarizes the key differences.

Criteria	ICL	Fine-Tuning	RAG
Definition	Examples in prompts	Adjust model	Uses external data
Data Needs	Few examples	Large dataset	Live database
Compute	Low (slow inference)	High (training)	Variable
Flexibility	High	Task-specific	Adaptable
Accuracy	Variable	High	Very high
Complexity	Low	High (ML expertise)	Medium (database integration)
Challenges	Context limits	Overfitting, data	Latency, database updates

Table 1: Comparison of In-Context Learning, Fine-Tuning, and Retrieval-Augmented Generation, highlighting their advantages, limitations, and suitability for different applications.

2.2 Embedding Models

Embedding models in RAG architectures transform modalities into vector representations, enabling efficient vector search. In text-based RAG applications, embeddings are commonly categorized as:

- **Dense embeddings** - Capture semantic similarity, ideal for contextual understanding.
- **Sparse embeddings** - Focus on keyword matching, prioritizing direct term overlap.

2.2.1 Dense Embedding Models

Medical-specific embedding models, such as PubMedBERT, BioBERT, SPECTER, SciNCL, and MedCPT, enhance text representation for biomedical applications[6].

Among these, **MedCPT** achieves superior performance by employing distinct embedding models for queries and documents while aligning them in the same vector space, reducing the semantic gap [6].

2.2.2 Sparse Embedding Methods

Sparse embeddings rely on term-based models, with the two most common approaches being:

- **BM25** - Scores text relevance using term frequency (TF) and inverse document frequency (IDF).
- **SPLADE** - A BERT-based model leveraging semantic information, but slower and less interpretable than BM25 with minimal performance gains [12].

2.3 Data Sources

High-quality medical datasets are essential for effective RAG systems in telemedicine, ensuring accurate and relevant information retrieval. The ideal dataset should meet the following criteria:

- Relevance - Aligns with patient queries.
- Coverage - Offers extensive and diverse medical knowledge.
- Clinical Credibility - Provides reliable, peer-reviewed information.

Among the available datasets, PubMed is the most suitable for a written telemedicine RAG system, as it offers relevant abstracts with clinical insights, ensuring credible access to peer-reviewed research across diverse medical fields, with over 37 million biomedical citations [18].

Other datasets offer specialized information beneficial for medical retrieval:

- MIMIC - Electronic health records from critical care patients [7].
- CTTI - Metadata on clinical studies and trial outcomes[3].
- BioASQ - Biomedical data sources used for NLP applications like question answering and document retrieval [11].

While these datasets serve specific purposes, PubMed remains the primary choice due to its relevance, broad coverage, and credibility, making it the most effective resource for retrieving accurate medical information.

2.4 Vector Databases

Vector databases enable efficient similarity search, making them crucial for RAG systems. Key selection criteria for this use case include on-premise deployment and Python support. Table 2 compares identified relevant vector databases, which were found in the literature review conducted in 2024.

Database	Scalability	Metadata Filtering	Hybrid Search	Multi Vector
Milvus [16]	20B	x	x	x
Faiss [8]	1B			
pgvector [23]	Resource-dependent	x		x
ChromaDB [4]	1M	x		x
ScaNN [25]	Resource-dependent			
NucliaDB [19]	100M	x		

Table 2: Comparison of vector databases based on key criteria for RAG systems.

Milvus emerges as the optimal choice, meeting all requirements while supporting up to 20 billion vectors, ensuring scalability for large-scale retrieval tasks.

2.5 Database Information Retrieval and Reranking

Optimizing search strategies is crucial for accurate medical information retrieval. **Hybrid search**, which combines lexical and semantic methods, achieves the best performance, especially when paired with Hypothetical Document Embeddings (HyDE) [27]. HyDE generates a pseudo-document from the query, improving semantic search accuracy [5]. Studies indicate that weighting lexical search at 0.3 provides the best balance of precision and recall [27].

Reranking further enhances retrieval by directly evaluating document relevance to queries. The only reranking model specialized in medical data, MedCPT’s cross encoder, has shown to increase performance [6].

Strategically positioning retrieved documents within the LLM’s context also improves RAG performance. Experiments show that **reverse rele-**

vance ordering - placing the most relevant information at the end - yields the best results [27].

MedRAG-Toolkit refines retrieval strategies, demonstrating up to 18% accuracy improvement. It suggests an optimal chunk count of 32 [27], while Anthropic AI’s findings point to 20 chunks [1]. Thus, the ideal chunk count for RAG applications likely falls between 20 and 32.

3 Methodology

3.1 System Architecture and Design

3.1.1 Offline Part

A corpus of 10 million PubMed articles is processed, generating dense embeddings with the MedCPT article encoder and sparse embeddings with BM25. Since most abstracts fit within MedCPT’s 512-token limit, chunking is unnecessary. The embeddings are stored in a Milvus vector database for efficient retrieval. This initialization takes several hours but is required only at setup or when new data is added.

3.1.2 Online Part

Upon receiving a patient form, the system uses a local LLM to generate sub-questions. These are encoded into dense and sparse embeddings using the MedCPT query encoder and BM25. The query vectors are then sent to Milvus, where hybrid search retrieves the most relevant articles.

To improve relevance, the retrieved articles are reranked using the MedCPT cross-encoder. The highest-ranked articles, along with the patient form, are provided to the LLM to generate a telemedicine response. The system also records PMIDs (PubMed ID) of relevant articles for further reference for medical professionals.

3.2 Components

3.2.1 Corpus

PubMed was selected for its extensive collection of peer-reviewed articles. The system uses the preprocessed MedRAG dataset, which includes titles, abstracts, and PMIDs from 24 million articles. To refine the dataset, abstracts with fewer than eight words were removed, and only the 10 million most recent articles (1992-2022) were retained for relevance.

3.2.2 MedCPT

MedCPT consists of three specialized BERT-based models fine-tuned on PubMed data, ensuring compatibility with the corpus while representing the state-of-the-art in open-source medical embeddings:

- **Article Encoder** - Generates 768-dimensional dense embeddings from PubMed titles and abstracts.
- **Query Encoder** - Maps queries into the same embedding space as articles, reducing the semantic gap.
- **Cross Encoder** - Directly assesses article-query relevance without embedding loss and further improves accuracy by jointly processing both inputs during reranking.

3.2.3 BM25

BM25 is a resource-efficient ranking algorithm, implemented in Milvus using the NLTK library to process titles and abstracts by removing stopwords and stemming terms.

The output is a sparse vector — a dictionary-like structure — that assigns BM25 weights to individual tokens. This representation enables efficient similarity comparison using BM25 scoring logic during retrieval.

3.2.4 Vector Database

Milvus Standalone was chosen for its scalability (up to 100 million vectors on a single machine) and hybrid search capabilities.

During initialization, hyperparameter tuning to balance speed and accuracy was omitted since real-time responses were not required. Hybrid search retrieves relevant articles by combining:

- **Dense embeddings** from MedCPT (semantic search)
- **Sparse embeddings** from BM25 (lexical search)

Results are ranked using the formula:

$$s_h = s_s + w \cdot s_l \tag{1}$$

Where s_h is the hybrid score, s_s is the semantic score, and s_l is the lexical score, weighted by w .

Hyperparameters such as retrieved article count of the vector search and reranking, and lexical weighting are analyzed in the results section.

3.2.5 LLM

LLM selection was based on LM Arena rankings [14] with the following constraints :

- Must fit on a single user GPU (Nvidia RTX 4090, 24GB VRAM).
- Must support at least 32 average-length articles, aligning with MedRAG recommendations [27].

Meta-Llama-3.1-8B-Instruct ranked 50th in LM Arena, was chosen. It could process over 100 articles on average, whereas Gemma-2-9b-it was limited to about 12 before encountering memory errors.

The LLM performs two key steps:

1. **Sub-question Generation** – The patient form is decomposed into structured sub-questions using a local LLM. This process improves retrieval precision by aligning queries with the MedCPT query encoder’s training format and PubMed’s English-language corpus. Sub-questions are generated in English regardless of the original language, as PubMed articles are exclusively in English.

This approach allows the system to support patient queries in any language the LLM understands. Since the final response is generated by the same multilingual LLM that processes the input, it is returned in the original language of the patient form. This ensures usability in multilingual settings—such as Switzerland—despite relying on an English-only retrieval corpus.

The LLM is not constrained to produce a fixed number of sub-questions. Instead, it autonomously determines how many are necessary to adequately cover the patient’s case. This flexibility enables fine-grained vector search, where each sub-question retrieves targeted information for a specific medical aspect. In contrast, querying the full patient form directly would return only documents addressing all concerns simultaneously — often limiting retrieval performance.

For example:

Patient form (German):

Lässt sich die Vasektomie rückgängig machen?

Ich wollte fragen, ob es grundsätzlich möglich ist, eine Vasektomie rückgängig zu machen. Ich habe vor ca. 10 Jahren meinen Samenleiter

abtrennen lassen. Jetzt möchte ich mit meiner neuen Partnerin noch einmal ein Kind zeugen. Wie hoch sind die Erfolgschancen und mit welchen Risiken und allfälligen Komplikation sollte man bei solchen Eingriffen rechnen? Was würde das Ganze ungefähr kosten? (Alter = 49)

Generated sub-questions:

- Is vasectomy reversal possible after many years?
- How successful is vasectomy reversals at an older age?
- At what cost can I expect to pay for a vasectomy reverse procedure?
- Which risks and complications may occur during a vasectomy reversal surgery?
- Will insurance typically cover the costs associated with reversing a vasectomy?
- Do men over 40 have lower chances of success from a vasectomy reversal operation?
- What steps do you need to follow before undergoing a vasecomy reversal treatment?

2. **Response Generation** - Reranked articles for all sub-questions are merged using reversed interleaved merging, placing the most relevant information closest to the original patient form at the bottom. The LLM then generates a telemedicine response using one-shot learning for consistency.

The used prompts to generate the sub-questions and telemedicine responses are detailed in the Appendix A and Appendix B.

4 Results

4.1 Recall@K

To evaluate retrieval accuracy, 100 random PubMed articles were selected, each paired with a generated sub-question. The goal was to assess how effectively the system could retrieve the original article.

4.1.1 Dense and Sparse Retrieval Methods

BM25 (lexical search) and MedCPT (semantic search) were compared using cosine distance (COSINE), Euclidean distance (L2), and inner product (IP) as similarity metrics.

$$\text{COSINE}(a, b) = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|} \quad (2)$$

$$\text{L2}(a, b) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (3)$$

$$\text{IP}(a, b) = \sum_{i=1}^n a_i b_i \quad (4)$$

Figure 2 shows that semantic search with the IP outperforms other methods. Recall follows a logarithmic trend, improving as more results are retrieved, suggesting that maximizing the number of retrieved results, at least up to 1024, is always beneficial.

4.1.2 Sparse Weight

Hybrid search was analyzed to determine the optimal balance between lexical and semantic search. While Recall@K remained identical to semantic search alone, weighting still influenced ranking positions.

As shown in Figure 3, the optimal sparse weight for IP similarity is 0.1, providing the best overall ranking performance.

MedCPT’s cross-encoder can process approximately 1400 article-query pairs per forward pass with 24GB VRAM. Therefore, 1024 articles are retrieved from the hybrid search to optimize reranking efficiency.

4.1.3 Reranking

The 1024 retrieved articles from hybrid search were reranked using the MedCPT cross-encoder. Figure 4 shows that reranking improves retrieval performance, with recall increasing log-linearly up to $K = 16$ before plateauing. Across all K values, IP similarity remains the most effective metric.

Determining the optimal number of articles per sub-question must account for the LLM’s lost-in-the-middle issue [13], balancing context length and relevance. In this context, a *chunk* refers to a full PubMed abstract, which is treated as a single retrieval unit. Literature suggests that providing

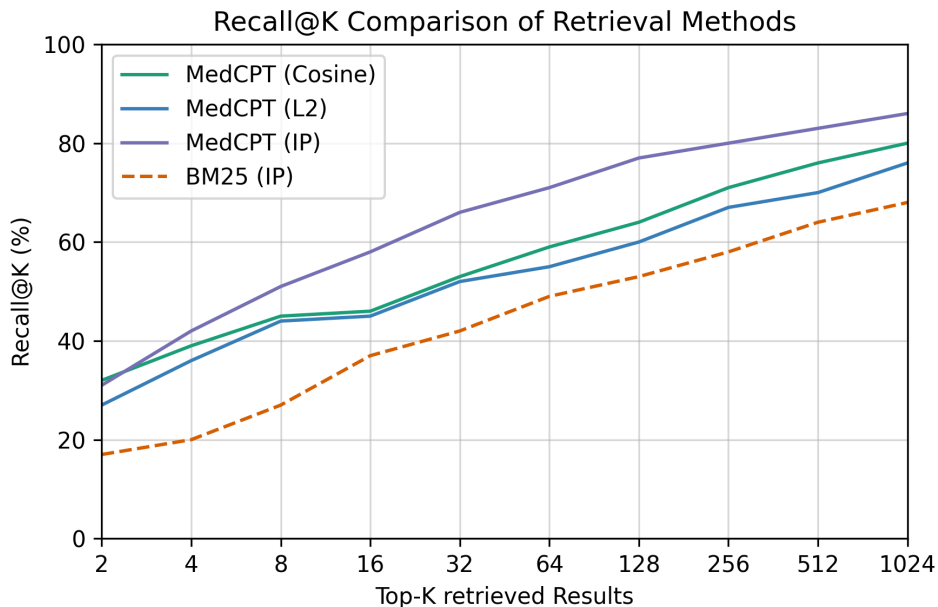


Figure 2: Recall@K comparison of dense (MedCPT) and sparse (BM25) retrieval methods. The x-axis (Top-K retrieved results) is logarithmic.

20 to 32 such chunks in the LLM context achieves the best trade-off between informativeness and model attention.

Given that the LLM generates 5 to 9 sub-questions, assigning 4 abstracts per sub-question aligns with this range. Using inner product similarity, a sparse weight of 0.1, and reranking 1024 retrieved results from the hybrid search, the system achieves a Recall@4 of 54% within a 10-million-article corpus.

4.2 Quality of Generated Responses

Evaluating telemedicine responses is challenging due to their open-ended nature. Instead of standard benchmarks, an telemedicine company with 20 years of experience assessed 39 LLM-generated responses based on real patient forms. The cases covered STDs (11 forms), benign prostate enlargement (12 forms), fertility and contraception (15 forms), and one general inquiry.

4.2.1 Workload Reduction

Telemedicine professionals compared the time required to write responses manually versus modifying LLM-generated responses. Writing from scratch took an average of 13.46 minutes, while adjustments to generated responses required only 2.82 minutes - a reduction of 10.64 minutes per form, equating to a 79.05% efficiency gain.

Figure 5 shows the workload distributions, highlighting that the RAG system significantly reduces time per patient form. Most responses required little to no modification, whereas manual writing ranged from 10 to 20 minutes.

4.2.2 Ratings

Evaluators assessed the accuracy and safety of generated responses. Of the 39 responses, 36 were deemed acceptable, while 3 contained misinformation - 2 of which were classified as potentially dangerous. Figure 6 presents the categorical breakdown.

Professionals noted that these errors were easily detectable and unlikely to go unnoticed in a production setting.

Example of professional feedback (dangerous response): One case involved a 37-year-old reporting a persistent tension-like sensation in the lower leg, without swelling or nocturnal symptoms. The generated response discussed muscular causes and lifestyle factors but failed to mention deep vein thrombosis (DVT) — a critical differential diagnosis.

A physician commented:

“The most dangerous diagnosis — deep vein thrombosis — is not mentioned at all in this response.”

Such feedback illustrates the importance of medical review, especially in cases with ambiguous symptoms where safety-critical omissions can occur.

4.3 Time Measurements

The offline phase requires several hours to initialize, with dense embedding generation accounting for 70-75% of the total time.

The online phase takes at least one minute per request, with model and BM25 loading comprising nearly half of this time. While real-time responses are impractical, immediate processing upon receiving patient forms ensures response readiness without delays.

The system specifications are listed in Appendix C, with detailed time measurements and scalability tests in Appendix D.

5 Discussion

5.1 Interpretation of Findings

The results confirm that hybrid vector search with reranking effectively retrieves relevant documents, achieving a Recall@4 of 54% in a 10-million-article corpus. Given that multiple articles may be equally or more relevant than the original, the retrieval approach ensures high-quality references for generating telemedicine responses.

While hybrid search did not increase the number of correctly retrieved articles, it improved the similarity of retrieved results, potentially enhancing response reliability in different scenarios.

The selection of **Meta-Llama-3.1-8B-Instruct** ensures efficient performance on a single GPU (24GB VRAM). The model can be scaled down for resource-constrained environments or upgraded for improved performance. Given the rapid advancements in LLMs, periodic evaluations are recommended to assess potential upgrades.

The proposed RAG system significantly reduces telemedicine professionals' workload, cutting response time by **79.05%**. This is particularly beneficial amid medical staff shortages. Professionals noted that the system's occasional misinformative responses were easily identifiable, suggesting that **trust bias** is not yet a major concern but requires ongoing monitoring.

While system initialization requires several hours, processing each patient form takes only 1-2 minutes, making this approach a highly efficient solution for telemedicine applications.

5.2 Comparison with Prior RAG Applications in Medicine

RAG frameworks have been applied in areas such as veterinary telemedicine and automated medical report summarization. However, no prior work has proposed a privacy-preserving RAG system specifically for written telemedicine targeting patients or quantified its impact on response time reduction. In our setup, privacy is preserved by running the entire pipeline — including retrieval and the Large Language Model — locally on-premise. No patient data is transmitted to external servers or third-party APIs, thereby mitigating data exposure risks commonly associated with cloud-based LLMs.

The choice of **MedCPT** over models like BioBERT and PubMedBERT was based on its superior retrieval performance. By aligning articles and queries within the same vector space, MedCPT effectively bridges the semantic gap, enhancing retrieval accuracy.

Although semantic search with MedCPT outperforms lexical search via BM25, hybrid search remains beneficial by improving similarity rankings, as demonstrated in previous studies. Since vector search accounts for only a small portion of total processing time, integrating hybrid search introduces no significant efficiency trade-offs.

5.3 Limitations and Challenges

While the RAG system reduces hallucinations, LLMs remain prone to generating incorrect information. In the evaluation, two of 39 responses contained potentially dangerous diagnoses, and one included misinformation, underscoring the need for medical professionals to review all generated responses for accuracy and safety.

Another concern is **trust bias** - the risk of doctors over-relying on AI-generated responses. If professionals accept outputs without verification, errors could go unnoticed. Ensuring awareness and careful review is essential to mitigate this risk.

Finally, restricting the data source to PubMed limits available information to titles and abstracts, which may lack sufficient detail in certain cases.

5.4 Multilingual Support and Future Opportunities

A key strength of the proposed system is its ability to handle multilingual patient inputs and outputs. Thanks to the capabilities of the LLM, patient forms can be submitted in any supported language, and the final response is generated in that same language, ensuring accessibility across diverse populations — including multilingual regions like Switzerland.

While the retrieval step relies on an English-language corpus (PubMed), the system bridges this gap by generating intermediate sub-questions in English. This language-agnostic retrieval architecture enables high-quality, factual grounding without requiring patients to submit queries in English.

Future enhancements could include multilingual medical corpora or machine translation pipelines to retrieve from non-English datasets, further improving coverage and localization.

5.5 Implications of RAG on Telemedicine Workflow

Integrating a RAG system into telemedicine significantly reduces doctors' workload, potentially shortening patient response times, while preserving data privacy. Since medical professionals still review and validate AI-generated responses, the risks of incorrect medical advice are mitigated, ensuring reliability in clinical decision-making.

5.6 Future Work

Enhancing the RAG system with more reliable data sources would have the greatest impact, as data quality directly influences LLM-generated responses. Expanding the dataset to include additional medically verified documents or doctor-reviewed telemedicine Q&A pairs could improve retrieval relevance and response quality. Advanced filtering techniques may further refine data selection.

While improving embedding and LLM models could enhance performance, this is a secondary priority due to limited availability of high-quality medical embedding and reranking models. Fine-tuning and LLM would also require extensive telemedicine Q&A datasets and substantial computational resources. However, since the architecture is modular, components such as embeddings, rerankers, or LLMs can be easily swapped, allowing for future improvements without significant structural changes.

Future research could explore recent advancements in efficient reasoning LLMs, such as distilled versions of DeepSeek-R1, benchmarking their performance against conventional models for telemedicine applications.

6 Conclusion

This work presents a modular and privacy-preserving RAG system for telemedicine, leveraging hybrid vector search and reranking for document retrieval. An local LLM formulates responses based on retrieved medical documents.

The system achieved a Recall@4 of 54% on a 10-million-article corpus, effectively retrieving relevant PubMed articles. It also significantly reduced doctor workload by nearly 80%, addressing medical staff shortages.

Despite these benefits, doctors must remain vigilant against LLM hallucinations and trust bias, ensuring careful review of AI-generated responses.

Future research should enhance retrieval relevance while maintaining a balance between automation and human oversight.

This study lays the groundwork for ethical and reliable AI-driven decision support in telemedicine.

References

- [1] Anthropic. Introducing contextual retrieval, 2024. Accessed: 2025-02-21.
- [2] Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. Instruction pre-training: Language models are supervised multitask learners. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2550, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [3] Clinical Trials Transformation Initiative. Aggregate analysis of clinical-trials.gov (aact) database, n.d. Accessed: 2025-02-21.
- [4] Fendy Feng. Comparing vector databases: Milvus vs. chroma db, 2024. Accessed: 2025-02-21.
- [5] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [6] Qiao Jin, Won Kim, Qingyu Chen, Donald C Comeau, Lana Yeganova, W John Wilbur, and Zhiyong Lu. Medcpt: Contrastive pre-trained transformers with large-scale pubmed search logs for zero-shot biomedical information retrieval. *Bioinformatics*, 39(11):btad651, 11 2023.
- [7] Alistair E W Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li-Wei H Lehman, Leo A Celi, and Roger G Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Sci Data*, 10(1):1, January 2023.
- [8] Hervé Jégou, Matthijs Douze, and Jeff Johnson. Faiss: A library for efficient similarity search, 2017. Accessed: 2025-02-21.

- [9] Amirhossein Kargar Khabbazi Sardroud. Enhancing veterinary medicine with rag-based ai chatbots: Improving accessibility and quality of care. Master’s thesis, Università degli Studi di Padova, Padova, Italy, 2023.
- [10] Jin K. Kim, Michael Chua, Mandy Rickard, and Armando Lorenzo. Chatgpt and large language model (llm) chatbots: The current state of acceptability and a proposal for guidelines on utilization in academic medicine. *Journal of Pediatric Urology*, 19(5):598–604, 2023.
- [11] Anastasia Krithara, Anastasios Nentidis, Konstantinos Bougiatiotis, and Georgios Paliouras. BioASQ-QA: A manually curated corpus for biomedical question answering. *Sci Data*, 10(1):170, March 2023.
- [12] Carlos Lassance, Hervé Déjean, Thibault Formal, and Stéphane Clinchant. Splade-v3: New baselines for splade, 2024.
- [13] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 02 2024.
- [14] LM Arena. LM Arena Leaderboard. <https://lmarena.ai>, 2024. Accessed: 2024-09-03.
- [15] M. Mahalakshmi, Shardul Bharadwaj, and Aklanta Niraz Bhuyan. A real-time medical report analysis and ai-powered diagnosis: A cloud-based solution for improved patient care. In *2024 Second International Conference on Advances in Information Technology (ICAIT)*, volume 1, pages 1–6, 2024.
- [16] Milvus Contributors. Milvus: High-performance vector database built for scale, 2025. Accessed: 2025-02-21.
- [17] Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12284–12314, Toronto, Canada, July 2023. Association for Computational Linguistics.
- [18] National Center for Biotechnology Information. Pubmed, 2025. Accessed: 2025-02-21.

- [19] Javier Torres Niño. NucliaDB Index: Powering Fast, Smart, and Scalable RAG, 2024. Accessed: 2025-02-21.
- [20] Stefano Omboni, Raj S Padwal, Tourkiah Alessa, Béla Benczúr, Beverly B Green, Ilona Hubbard, Kazuomi Kario, Nadia A Khan, Alexandra Konradi, Alexander G Logan, Yuan Lu, Maurice Mars, Richard J McManus, Sarah Melville, Claas L Neumann, Gianfranco Parati, Nicolas F Renna, Philippe Ryvlin, Hugo Saner, Aletta E Schutte, and Jiguang Wang. The worldwide impact of telemedicine during COVID-19: current evidence and recommendations for the future. *Connect Health*, 1:7–35, January 2022.
- [21] Oded Ovadia, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Fine-tuning or retrieval? comparing knowledge injection in LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 237–250, Miami, Florida, USA, November 2024. Association for Computational Linguistics.
- [22] Danica Mitch M. Pacis, Jr. Subido, Edwin D. C., and Nilo T. Bugtai. Trends in telemedicine utilizing artificial intelligence. *AIP Conference Proceedings*, 1933(1):040009, 02 2018.
- [23] pgvector Contributors. pgvector: Open-source vector similarity search for postgres, 2025. Accessed: 2025-02-21.
- [24] Sachin Sharma, Raj Rawal, and Dharmesh Shah. Addressing the challenges of AI-based telemedicine: Best practices and lessons learned. *J Educ Health Promot*, 12:338, September 2023.
- [25] Philip Sun. Announcing scann: Efficient vector similarity search, 2020. Accessed: 2025-02-21.
- [26] Ehsan Ullah, Anil Parwani, Mirza Mansoor Baig, and Rajendra Singh. Challenges and barriers of using large language models (LLM) such as ChatGPT for diagnostic medicine with a focus on digital pathology - a recent scoping review. *Diagn Pathol*, 19(1):43, February 2024.
- [27] Xiaohua Wang, Zhenghua Wang, Xuan Gao, Feiran Zhang, Yixin Wu, Zhibo Xu, Tianyuan Shi, Zhengyuan Wang, Shizheng Li, Qi Qian, Ruicheng Yin, Changze Lv, Xiaoqing Zheng, and Xuanjing Huang. Searching for best practices in retrieval-augmented generation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings*

of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 17716–17736, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

- [28] Chaoyi Wu, Weixiong Lin, Xiaoman Zhang, Ya Zhang, Weidi Xie, and Yanfeng Wang. PMC-LLaMA: toward building open-source language models for medicine. *J Am Med Inform Assoc*, 31(9):1833–1843, September 2024.
- [29] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models, 2025.

A Prompt: Sub-question Generation

The following prompt was used to generate English sub-questions from the patient form input:

```
Generate English sub-questions on the input query, regardless of
the query's original language. Ensure that:
- The sub-questions are clear, concise, and phrased as actual
  queries someone might ask.
- The sub-questions cover all relevant aspects of the input query
  comprehensively.
Only return a valid python list object without any explanation,
commentary, or metadata.
Return the output in the following format:
[
"<English Subquestion 1>",
"<English Subquestion 2>",
"<English Subquestion 3>",
...
]
```

Example Input and Output

Input Query:

```
Was ist die Alternative zu Latex-Kondomen?
Ich habe eine Latex-Allergie. Welche Kondome empfehlen Sie mir?
Sind diese Kondome so sicher wie die normalen Latex-Kondome?
Darf Gleitcreme verwendet werden?
```

Output:

```
[
"What are suitable condom alternatives for people with latex
allergies?",
"Are non-latex condoms as safe as latex condoms?",
"Can lubricants be used with non-latex condoms?",
"What materials are used in non-latex condoms?",
"What precautions should people with latex allergies take when
choosing condoms?"
]
```

B Prompt: Response Generation

The following prompt was used to generate medical responses based on retrieved articles:

You are a medical expert giving advice to patients (telemedicine).
Based on the articles provided, generate an answer that ensures:

- The answer is in the same language as the user input.
- The answer contains recommendations that are based on the articles.
- You will refuse to give harmful recommendations.

Example Input and Output

Input Query:

Was ist die Alternative zu Latex-Kondomen?
Ich habe eine Latex-Allergie. Welche Kondome empfehlen Sie mir?
Sind diese Kondome so sicher wie die normalen Latex-Kondome?
Darf Gleitcreme verwendet werden?

Output:

Mit dem etwas unpräzisen Begriff «Latex-Allergie» ist meistens eine allergische Kontaktreaktion auf bestimmte Proteine gemeint, welche spurenweise im aus Latex (Milchsaft des Kautschukbaums) hergestellten Naturkautschuk vorkommen. Die Kontaktstellen entzünden sich, es bilden sich Ekzeme. Seltener kann es auch zu gefährlichen Sofortreaktionen mit Atemnot, Blutdruckabfall, Krämpfen bis hin zur Bewusstlosigkeit oder zu nicht immunologisch bedingten Hautreaktionen kommen. Hinter einer vermeintlichen Latexallergie kann sich auch eine Allergie auf andere Inhaltsstoffe in Kondom oder Gleitmittel verbergen. Je nach Schweregrad einer Latex-Allergie kann es ausreichend sein, nur ungepuderte Latexkondome mit genügend Gleitmittel zu verwenden. Es sind auch hypoallergene Latexkondome erhältlich, aus denen die auslösenden Proteine weitgehend entfernt wurden, und «vegane» Kondome aus Naturkautschuk, bei deren Herstellung keine tierischen Produkte verwendet wurden und die daher zumindest keine tierischen Proteine enthalten. Muss ganz auf Latex verzichtet werden, dann können Kondome aus Polyurethan, Polyisopren oder dem Kunstharzderivat AT-10 verwendet werden. Alle bekannten Hersteller wie Ceylor, Durex, Manix und andere haben entsprechende Produkte im Angebot. Kondome aus Polyurethan (z.B. Ceylor Non-Latex Ultrathin) können besonders dünn hergestellt werden (0,02 mm) und werden daher als besonders gefühlsecht beworben. Allerdings sind sie nicht ganz so elastisch wie Kondome aus Latex. Polyisopren (verwandt mit dem Neopren der Taucheranzüge; z.B. Durex Latexfrei oder die Produkte der Skyn-Linie von Manix) ist eine Art synthetisch hergestelltes Latex ohne Eiweiss-Allergene. Solche Kondome sind unter den latexfreien am ehesten vergleichbar mit Kondomen aus Naturlatex. Produkte aus AT- 10 schliesslich (z.B. Pasante Unique) sind günstiger als ihre latexfreien Alternativen. Das wenig elastische Material kann aber bei Gebrauch etwas knistern und wird zuweilen als «künstlich» empfunden. Latexfreie Kondome sind ebenso sicher wie Latex-Kondome und bieten einen vergleichbar guten Schutz vor

HIV und anderen sexuell übertragbaren Infektionen. Bei Kondomen aus Polyurethan oder AT-10 können auch Gleitmittel auf Fett- oder Ölbasis verwendet werden, bei den Kondomen aus Polyisopren hingegen sollten - wie bei den Latex-Kondomen - nur Gleitmittel auf Wasser- oder Silikonbasis zur Anwendung kommen.

C System Specifications

C.1 Hardware Specifications

For the implementation, a High Performance Computer was used with the following hardware:

CPU (Central Processing Unit):

- Model: AMD EPYC 7302 16-Core Processor
- Cores / Threads: 16 / 32
- Clock Speed: 3.0 GHz

GPU (Graphics Processing Unit):

- Model: NVIDIA GeForce RTX 4090
- VRAM: 24 564 MiB
- Driver Version: 560.35.03
- CUDA Version: 12.6

RAM (Random-Access Memory):

- Total Memory: 251 GiB

Storage:

- Type: SSD
- Total Capacity: 5.4 TiB
- Available Storage: 906 GiB

C.2 Minimum Hardware Requirements

The following hardware requirements are recommended for running the RAG system:

- **GPU (Minimum for the same LLM):** 20 GB VRAM
- **CPU (Recommended):** 8 Cores
- **RAM (Recommended):** 32 GiB
- **Storage (Recommended):** 500 GB SSD

C.3 Software Specifications

This RAG system was implemented and executed in Docker on the following software environment:

Operating System:

- Distribution: Ubuntu 22.04.5 LTS (Jammy Jellyfish)

Kernel Information:

- Kernel Name: Linux
- Kernel Version: 5.15.0-124-generic
- Build Number: #134-Ubuntu
- Build Date: Fri Sep 27 20:20:17 UTC 2024
- Architecture: x86_64 (64-bit)

System Information:

- Hardware Platform: x84-64
- Operating System: GNU/Linux

D Time Measurements and Scalability

D.1 Offline Processing Time

To evaluate the scalability of the RAG system, time measurements were conducted for datasets containing 1 000, 10 000, 100 000, 1 000 000, and 10 000 000 articles. The processing time of individual steps in the offline phase for 10 million articles is shown in Figure 7.

D.1.1 Total Processing Time

The total offline processing time scales almost linearly but exhibits slight superlinear tendencies. Table 3 provides an overview of processing times across different dataset sizes.

Articles	1 000	10 000	100 000	1 000 000	10 000 000
Time	15.6 s	60.8 s	6.2 min	53.1 min	10.1 h

Table 3: Total offline processing times for different dataset sizes. For all dataset sizes, a pre-trained BM25 model was loaded.

D.1.2 Article-Dependent Steps

Most steps in the offline phase scale linearly or slightly superlinearly with the number of articles. Figure 8 visualizes this scalability.

D.1.3 Batch Size-Dependent Steps

Due to the large number of articles, dense vector conversion could not be performed simultaneously. Instead, articles were processed in batches, encoded into dense vectors and then inserted into the database. The scalability of these steps is shown in Figure 9.

D.1.4 Non-vectorized Steps

Sparse vector creation is non-vectorized, meaning it is independent of the number of articles or batch size. The encoding time for each article was measured separately and is 3.29 ± 25.36 ms. The high standard deviation is likely due to large variations in article length.

D.2 Online Processing Time

The time distribution for the online phase is shown in Figure 10.

D.2.1 Component Loading Times

Several components must be loaded before use in the online phase. How the database and BM25 are affected by it is shown in 11.

- **Vector Database:** Connecting to the pre-initialized database takes only a few milliseconds, except for 10 million articles, where a slight delay occurs.
- **LLM:**
 - Loaded twice: once for sub-question generation and once for response generation.
 - Average loading time: 8.98 ± 0.56 s
- **BM25:**
 - Sparse vector encoder loaded from a JSON file.
 - Loading time scales linearly with dataset size, averaging 20 seconds for 10 million articles.
- **Query Encoder (MedCPT):** 0.61 ± 0.12 s
- **Cross-Encoder (MedCPT):** 0.47 ± 0.1 s

D.2.2 Sub-Question Processing Steps

Figure 12 shows how encoding and creating the dense vectors scales with the number of sub-questions.

D.2.3 Sub-Question and Response Generation

How the generation time of the sub-questions and the time it takes to perform hybrid search are affected by the number of generated sub-questions is shown in 13.

Sub-Question Generation:

- The number of sub-questions generated by the LLM typically ranges between 5 and 9.
- Vector search time scales almost linearly with the number of sub-questions.
- For 9 sub-questions, the search takes 2.55 seconds on average.

Answer Generation:

- More sub-questions result in more abstracts being included in the context.
- Context length does not significantly affect generation time.

D.2.4 Search Result Processing

For 10 sub-questions and 10 million articles, vector search time remains constant, regardless of the number of retrieved results. This is due to:

- The lexical and semantic searches always retrieve 1024 results.
- Hybrid search only combines and weights these results.

D.2.5 Semantic Search Metric Comparison

The choice of semantic similarity metric had minimal impact on retrieval speed. Tests were conducted for 100 sub-questions on the entire corpus of 10 million articles.

Search Metric	Avg. Time (s)	Std Dev (s)
Cosine Similarity	26.9	1.17
Euclidean Distance	25.4	1.12
Dot Product	25.7	1.08

Table 4: Search metric comparison for 100 queries. Even with 100 sub-questions, search time did not differ significantly between metrics.

D.2.6 Sparse Vector Processing

Sparse vector creation is significantly faster in the online phase than in the offline phase with 0.35 ± 1.1 ms, likely because sub-queries are shorter than full articles.

D.2.7 LLM Performance and Response Time

Response generation time scales linearly with the number of generated tokens. Table 5 summarizes key LLM performance metrics.

Metric	Sub-Question Generation	Response Generation
Avg. Token Count	155	451
Avg. Context Length	341	6079
Processing Time (s)	4.92	16.78
Speed (tokens/s)	31.4	26.96

Table 5: LLM performance metrics. Context length does not significantly impact generation time, and the speed difference may be due to GPU optimizations and caching behavior.

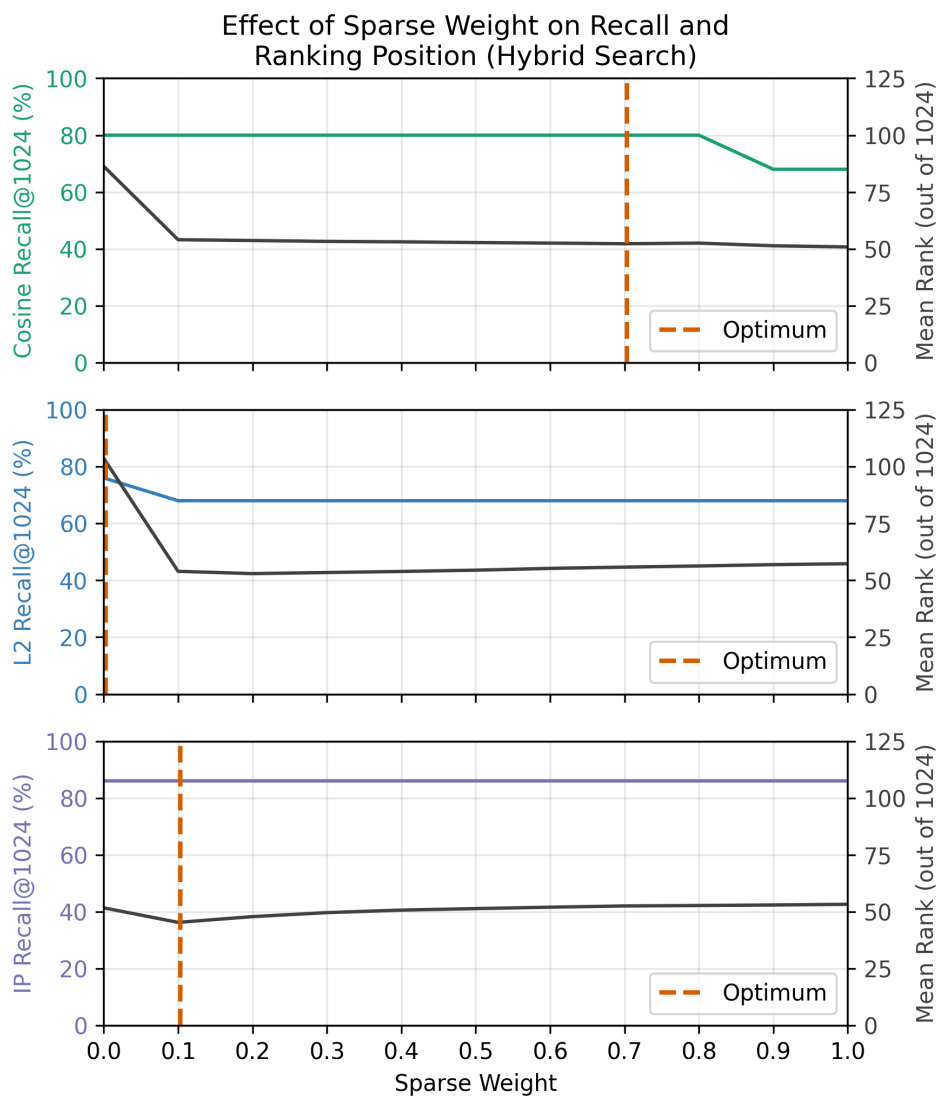


Figure 3: Impact of sparse weight on Recall@1024 and ranking position in hybrid search across different similarity metrics. The optimal weight (dashed orange line) is selected based on maximum Recall@1024, with the final selection minimizing mean rank.

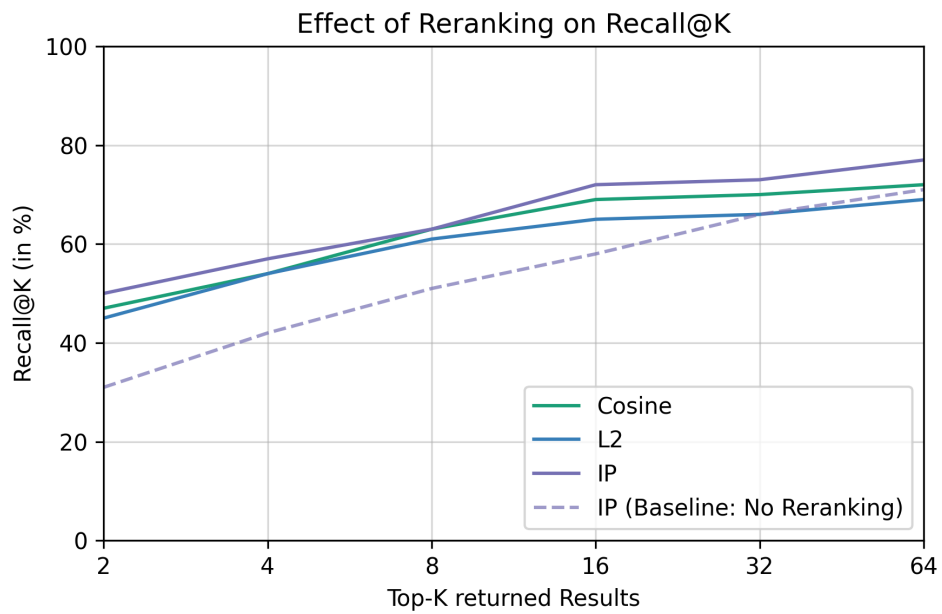


Figure 4: Impact of reranking on Recall@K. The x-axis (Top-K retrieved results) is logarithmic. The dashed line represents the baseline (IP without reranking from Figure 2), demonstrating improved retrieval effectiveness.

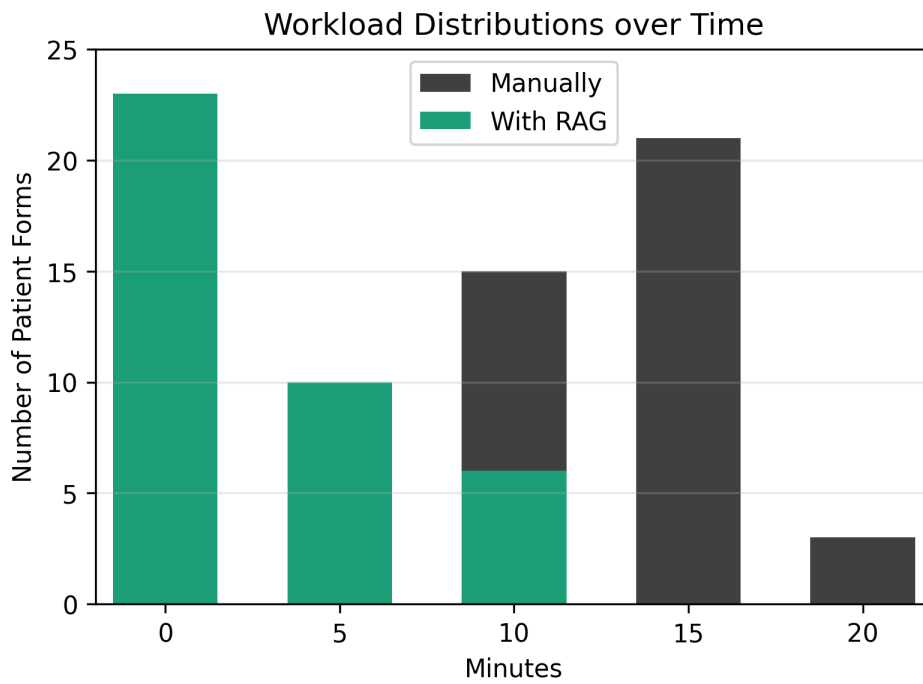


Figure 5: Workload distribution for manual vs. LLM-assisted telemedicine responses. Time per patient form (x-axis) vs. number of processed forms (y-axis).

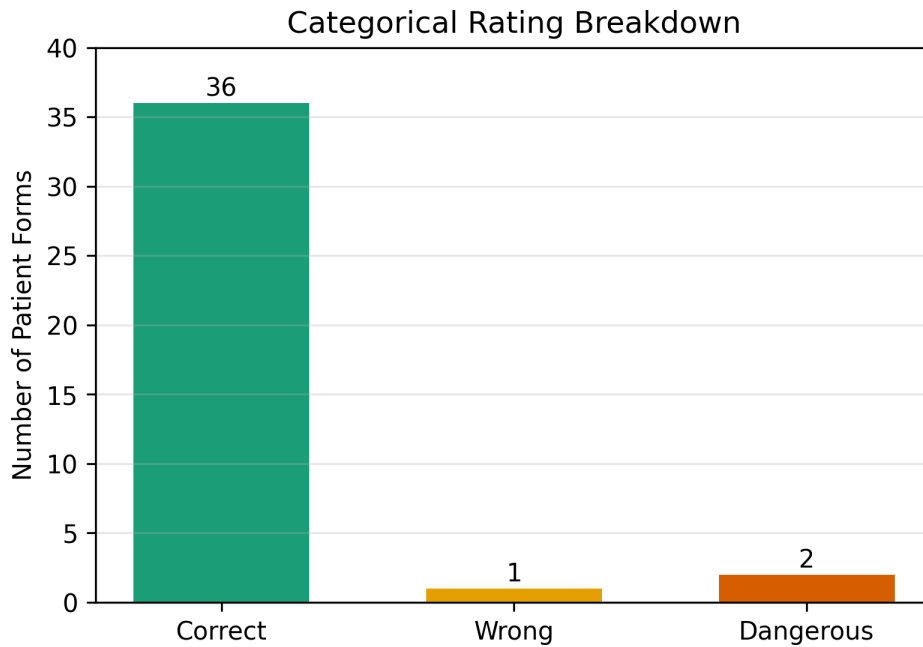


Figure 6: Categorical rating breakdown of LLM-generated telemedicine responses.

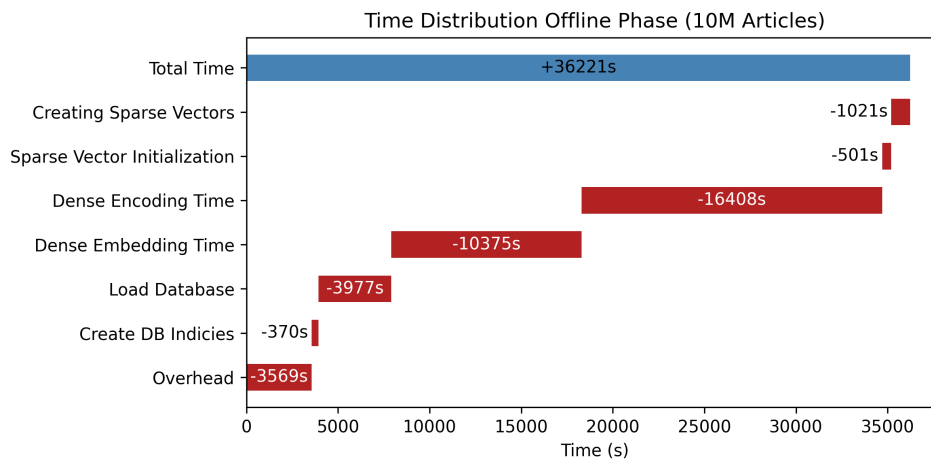


Figure 7: Time distribution in the offline phase. Steps contributing less than 1% of the total duration are grouped under "Overhead." A pre-initialized BM25 model was used, reducing total processing time by approximately 1 900 seconds.

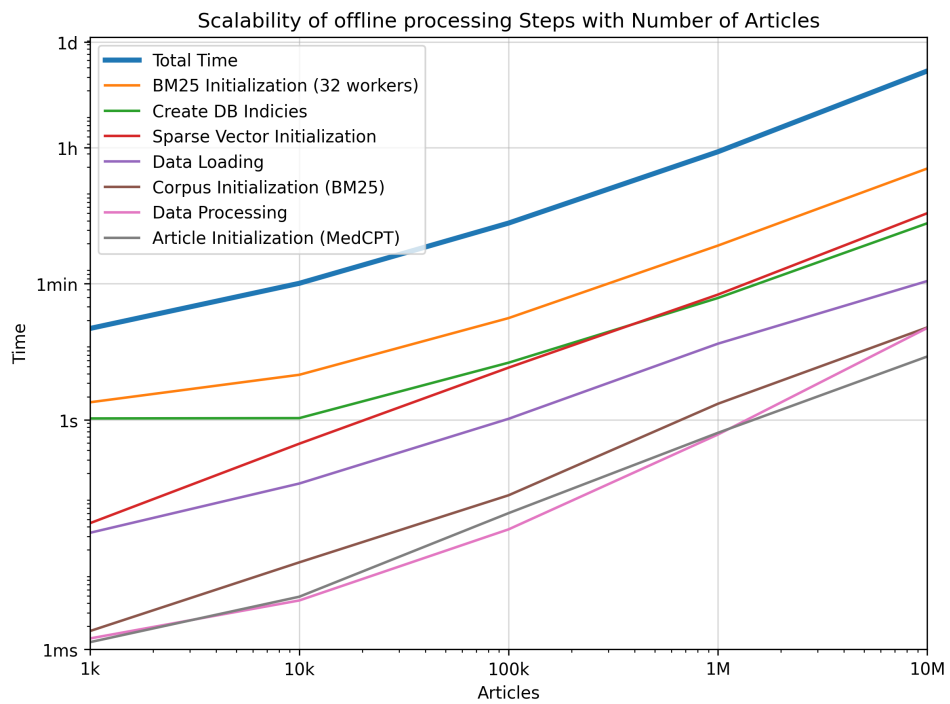


Figure 8: Scalability of offline processing steps with the number of articles. The total processing time is indicated at the top.

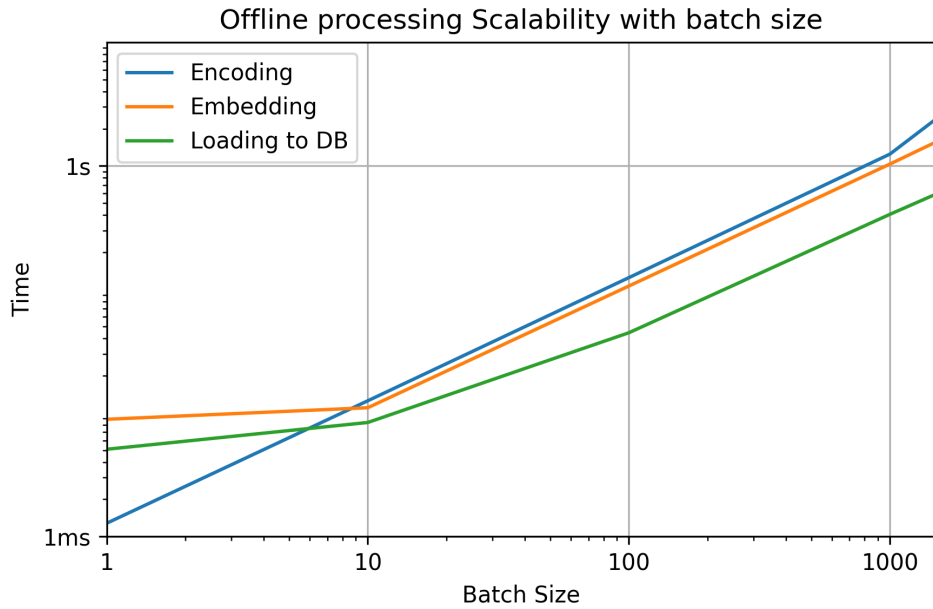


Figure 9: Offline processing scalability with batch size.

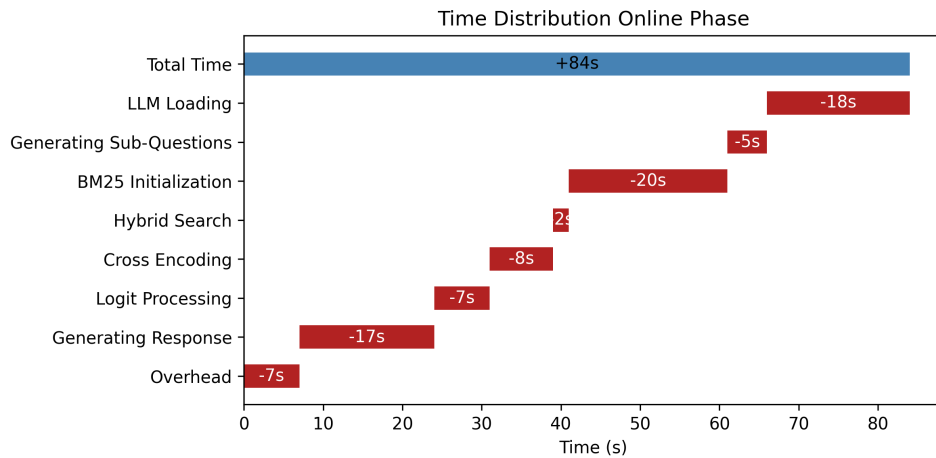


Figure 10: Time distribution in the online phase. Steps requiring less than one second are grouped under "Overhead."

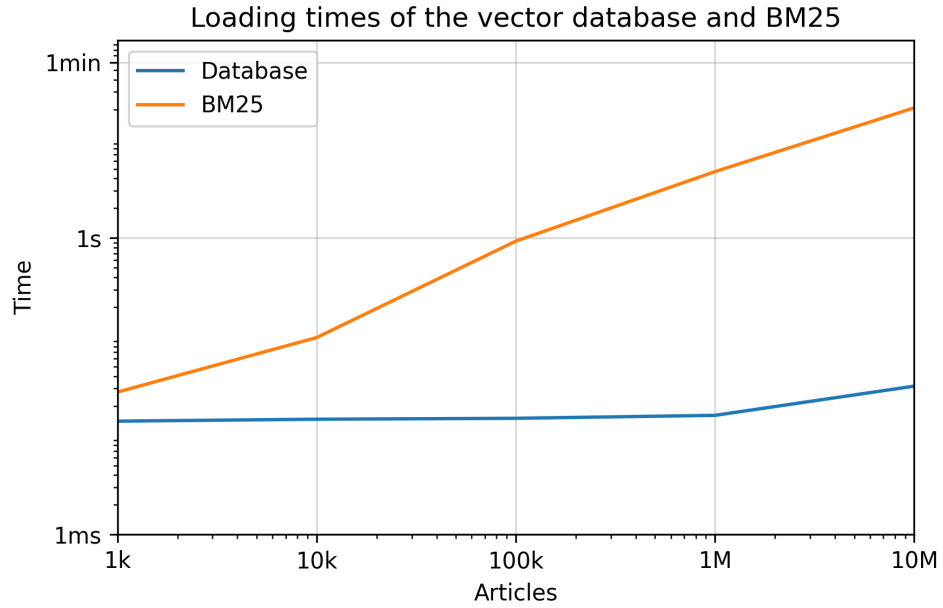


Figure 11: Loading times of the vector database and BM25.

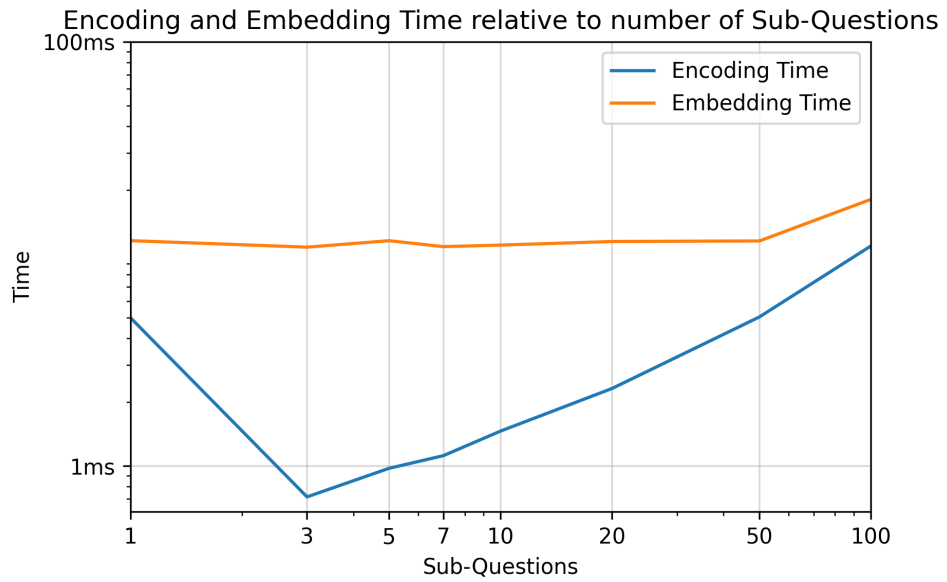


Figure 12: Online phase sub-question encoding and dense vector processing times.

Generation and Vector Search Time relative to the Number of Sub-Questions

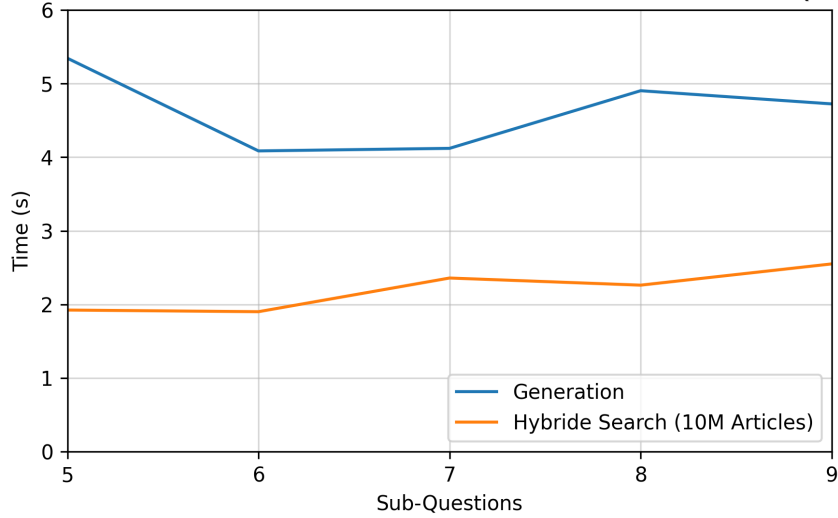


Figure 13: Processing times for vector search and response generation.

Times relative to the number of returned Results (10 Sub-Questions)

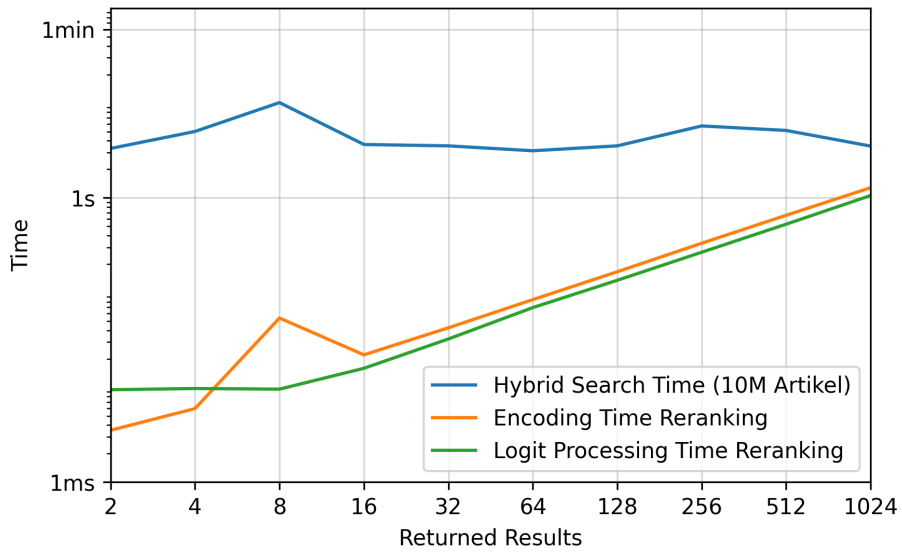


Figure 14: Vector search and reranking processing times.