UNIBUC-SD at ArchEHR-QA 2025: Prompting Our Way to Clinical QA with Multi-Model Ensembling

Dragoș-Dumitru Ghinea and Ștefania Rîncu
University of Bucharest
{dragos-dumitru.ghinea, stefania.rincu}@s.unibuc.ro

Abstract

In response to the ArchEHR-QA 2025 shared task, we present an efficient approach to patient question answering using small, pre-trained models that are widely available to the research community. Our method employs multi-prompt ensembling with models such as Gemma and Mistral, generating binary relevance judgments for clinical evidence extracted from electronic health records (EHRs). We use two distinct prompts (A and B) to assess the relevance of paragraphs to a patient's question and aggregate the model outputs via a majority vote ensemble. The relevant passages are then summarized using a third prompt (C) with Gemma. By leveraging off-the-shelf models and consumer-grade hardware (1x RTX 5090), we demonstrate that it is possible to improve performance without relying on resource-intensive fine-tuning or training. Additionally, we explore the impact of Chain-of-Thought (CoT) prompting and compare the performance of specialized versus general-purpose models, showing that significant improvements can be achieved through effective use of existing models.

1 Introduction

Responding to patient inquiries via patient portals is a major contributor to clinician workload, and automating this process using electronic health records (EHRs) could significantly reduce that burden. The ArchEHR-QA shared task (Soni and Demner-Fushman, 2025b) challenges participants to generate answers grounded in clinical evidence extracted from EHRs, requiring both accurate relevance detection and effective summarization. Submissions are evaluated on Factuality (how well answers cite annotated evidence sentences) and Relevance (how well they align with gold-standard 'essential' notes) using metrics such as Citation F1, BLEU, ROUGE, SARI, BERTScore, AlignScore, and MEDCON. The final score averages strict Factuality and aggregated Relevance metrics. Baseline scores on Codabench were generated using LLaMA 3.3 70B in a zero-shot setting, prompted to produce citation-containing answers; invalid responses (e.g., overly long or missing citations) were regenerated up to five times.

We address this challenge by breaking the task into two main subtasks: paragraph relevance classification and summarization. First, we classify the relevance of clinical paragraphs to the patient's question using a multi-model approach. Then, we generate a summary of the most relevant paragraphs to answer the query. Our method employs two pre-trained models, Gemma 3 27B (Gemma Team, 2025) and Mistral 3.1 Small 24B (Mistral AI, 2025b), to assess relevance and summarize the relevant passages. We experimented with a variety of models and prompting strategies, including Chain-of-Thought (CoT), and explored different configurations of the majority vote ensemble to optimize model performance.

A key aspect of our approach is the use of off-the-shelf models and consumer-grade hardware (1× RTX 5090), avoiding the need for fine-tuning or training from scratch. By focusing on smaller, more accessible models, we show that strong performance can be achieved with minimal computational overhead. Our experiments further demonstrate that carefully designed prompt engineering and ensemble methods can enhance performance effectively, making the approach both practical and scalable. While larger models may yield further gains in real-world deployments, our work highlights the untapped potential of lightweight setups for clinical question answering.

2 Related Work

Prompt-based methods have shown that LLMs can achieve strong performance across diverse tasks without fine-tuning. Techniques such as CoT

BioNLP 2025 Shared Tasks, pages 11–21

prompting improve performance on a range of tasks (Wei et al., 2022; Kojima et al., 2022). Further strategies like self-consistency decoding improve CoT outputs by aggregating multiple reasoning paths (Wang et al., 2023, 2024). Prompt ensembling, using diverse prompts with majority voting, has been shown to improve reliability (Yang et al., 2023), and further analysis on compound systems has been made (Chen et al., 2024).

In the medical domain, recent studies show that general purpose LLMs can be competitive with specialized models when guided by carefully designed prompts (Nachane et al., 2024; Russe et al., 2024; Sivarajkumar et al., 2024).

Our system builds on this line of work by using a multi-prompt ensemble strategy with Gemma 3 (27B) and Mistral 3.1 Small (24B), relying solely on prompt engineering and avoiding fine-tuning.

3 Method

3.1 Paragraph Relevance Assessment

Each instance from the dataset (Soni and Demner-Fushman, 2025a) provides a clinical note both as continuous text and as a list of indexed paragraphs, which must be cited in the final answer; in our experiments, we use only the paragraph-indexed format. Additionally, each instance includes a patient-authored question and a corresponding clinician-formulated question. We found that using the patient question for paragraph relevance classification introduces more false positives, so we rely exclusively on the clinician question, a refined and focused version of the original, for relevance assessment. The patient question is instead used during the summarization subtask to better reflect natural inquiry phrasing.

3.1.1 Individual Model Performance

We first prompted models to classify the relevance of the entire list of paragraphs in a single pass. While their explanations were often coherent, the final outputs frequently included incorrect indices or mismatched list lengths that didn't align with their own reasoning. To address this, we reformulated the task as a binary classification problem: the model is given the full list and asked whether a specific paragraph (identified by index) is relevant to the clinician's question. This approach significantly improved both consistency and interpretability.

Prompt selection followed two strategies: (1) manual trial-and-error and (2) suggestions from Gemini 2.5 Pro, chosen for its availability via Google AI Studio and generous usage limits that enabled extensive testing. Table 5 highlights some of our strongest prompt engineering results. Alongside prompt design, we also varied sampling temperature (ranging from 0.1 to 1.0) based on guidance from model developers and the open-source community (DeepSeek AI, 2024; Unsloth Team, 2025; Mistral AI, 2025a).

Model	Quantization	Overall Factuality	Prompt Used
gemma3-27b-it	Q6	56.04	A
mistral-small-3.1-24b	Q8	54.08	A
gemma3-12b-it	Q8	52.03	C
gemma2-9b-it	Q8	52.00	D
phi4	Q8	51.08	D
phi4-o1	i1 Q6_K	50.00	A
deepseek-llama-8b	F16	44.44	D
phi4-QwQ	Q8	40.89	D
phi4-mini-it	Q8	40.26	A
deepseek-qwen-32b	Q6_K	30.24	D
*all-relevant**	–	48.76	–
baseline (LLaMA 3.3 70B)	–	43.10	–

Table 1: Best overall factuality scores on the **dev dataset**. Prompt labels (A-D) refer to variants described in the Appendix A. The *all-relevant* baseline assumes all paragraphs are relevant. A list of more detailed scores is available in Appendix C.

The variation in temperature settings may affect the reproducibility of certain scores. To address this, in subsequent experiments we fix the temperature at 0.1, a value that provides stable and reproducible outputs while maintaining performance comparable to the best results observed.

We did not observe significant performance gains from models fine-tuned on medical data (WhyHow.AI Team, 2024; mradermacher, 2025a, 2024a), with most yielding only marginal improvements over the baseline (Table 2). Additional experiments with models such as `OpenBioLLM-Llama3-70B (i1-IQ3_XXS)` (mradermacher, 2024b), `Med-Chatbot-R1-Qwen-7B (F16)` (mradermacher, 2025c), and `ClinicalGPT-R1-Qwen-7B-EN-preview (F16)` (mradermacher, 2025b) were similarly unpromising. These models often failed to follow the required Yes/No output format, even without chain-of-thought prompting, making rule-based evaluation via regex unreliable. While output post-processing with another LLM is a possible workaround, we deemed it unnecessarily complex for the scope of this task.

3.1.2 Impact of Chain-of-Thought Prompting

We observe that chain-of-thought (CoT) prompting generally improves performance, as shown in Table 3. However, its effectiveness varies depending

Model	Quantization	Overall Factuality
PatientSeek	Q4_K_M	45.48
BioMistral-MedMNX	F16	42.96
DeepSeek-R1-Distill-Llama-8B-Medical-Expert	F16	44.39

Table 2: Overall factuality scores on the **dev dataset** for a few medical finetuned models using prompt A.

on the model. In our implementation, we introduce CoT reasoning using the following instruction: *"Create a chain of thought to determine if the paragraph is relevant to answering the question. Put your reasoning between <think> and </think> tags."*

While larger models such as `gemma3-27b-it` (Gemma Team, 2025) and `mistral-small-3.1-24b` (Mistral AI, 2025b) benefit significantly from CoT prompting, smaller models, particularly those in the `phi` family (Abdin et al., 2024; LM Studio Community, 2025), sometimes fail to complete the task reliably. In many cases, these models generate only reasoning text within the '<think>' tags without producing a final answer. We observed similar behavior in other small variants of LLaMA and Mistral, suggesting that limited context handling or weaker instruction-following may hinder CoT execution in compact models. To maintain consistency in automatic evaluation, if a model output could not be parsed using regular expressions to extract a valid binary answer, we defaulted to treating the paragraph as not relevant.

Model	Quantization	No CoT	CoT
gemma3-27b-it	Q6	45.48	**48.67**
mistral-small-3.1-24b	Q8	41.05	**52.22**
phi4	Q8	51.08	**51.32**
phi4-mini-it	Q8	29.49	**38.69**

Table 3: Comparison of factuality scores on the **dev dataset** with (prompt A) and without (prompt D) chain-of-thought (CoT) prompting.

3.1.3 Ensembling Model Performance

To improve robustness, we ensemble predictions using simple majority voting, selecting the most frequent Yes/No label per paragraph. To balance performance and efficiency, we limit ensembles to three models and treat different prompt configurations for the same model as distinct components. We explored various model-prompt combinations (Appendix B), with our best-performing ensemble shown in Figure 1. While overall factuality scores are useful, we prioritized confusion matrices when selecting ensembles, as they better reveal false positive and false negative trade-offs.

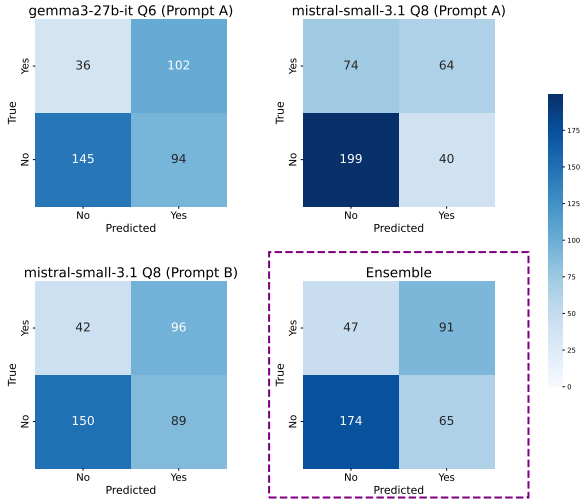

Figure 1: Confusion matrices for the best-performing ensemble and its constituent models.

3.2 Summarization

The summarization subtask requires (1) generating a coherent, concise answer from selected paragraphs and (2) citing the source paragraph(s) for each sentence. Instead of decoupling these, we adopt a unified approach where the model generates citations inline, avoiding the need for external alignment.

In our experiments, `Gemma` (Gemma Team, 2025) and `Mistral` (Mistral AI, 2025b) were the most reliable at handling complex summarization prompts. `Gemma` stood out for its consistent adherence to the required format, or outputs that were easily corrected via postprocessing, making it our primary choice. Using the same models for both relevance classification and summarization also helped maximize GPU parallelism on our RTX 5090 by reducing the number of concurrently loaded models.

We tested `Mistral` (Mistral AI, 2025b) as a correction layer for `Gemma`'s (Gemma Team, 2025) outputs, but observed only a minor readability improvement (+0.02), with inconsistent results. Moreover, `Mistral` frequently exceeded the 75-word constraint and resisted shortening even in multi-turn settings.

A consistent challenge for `Gemma` (Gemma Team, 2025) was maintaining citation coverage under tight length limits. To comply, it sometimes dropped relevant content, typically omitting one relevant paragraph for every 3-5 irrelevant ones removed.

To better enforce the 75-word limit, we implemented a multi-turn prompting strategy: after an initial response (often 80+ words), we re-prompted with explicit instructions to shorten. Curiously, instead of trimming toward the limit, the model often produced much shorter summaries (typically 40-50 words), indicating that it is capable of brevity but defaults to verbosity unless guided. We allowed up to three retries, though this cap was never reached in evaluation.

When relevance classification yielded fewer than three paragraphs, we used the full paragraph set to ensure sufficient context.

To evaluate the impact of relevance filtering, we conducted an ablation study using the best-performing summarization prompt. Summaries generated from the full paragraph set scored **45.6**, while those using only the filtered paragraphs reached **48.1**, confirming that relevance assessment contributes positively to final summarization quality.

3.3 Postprocessing

Instead of enforcing strict formatting in the prompt, we correct inconsistencies through regex-based postprocessing. `Gemma` (Gemma Team, 2025), for instance, often misplaces citations, adds extra spaces, or cites the question itself. To let the model focus on content, we fix these issues afterward by grouping adjacent citation markers, removing internal whitespace, and relocating misplaced citations to sentence ends. This lightweight pipeline improves formatting while preserving the summary's meaning.

3.4 Final Scores

Method	Dev		Test	
	Factuality	Relevance	Factuality	Relevance
our approach	60.4	35.8	53.8	32.7
baseline	43.1	28.7	33.6	27.8

Table 4: Scores overview on both the **dev and test datasets**.

Our approach substantially improves both factuality and relevance over the baseline (Table 4). On the test set, we observe a +20.2 Factuality and +4.9 Relevance gain, demonstrating the robustness and generalization of our method beyond the dev set. The code used to generate the test submission can be found on GitHub. [1]

[1] Test Submission Generation Source Code.

3.5 Error Analysis

We identified eight sentences misclassified by all models, revealing unanimous relevance assessment failures, six false positives, where irrelevant content was marked as relevant, and two false negatives, where relevant sentences were wrongly dismissed.

In the false-negative cases, models typically judged that the sentences lacked a clear link to the question or were too general. For example, in Case 6 ("Why did they find out later that he had fungal pneumonia?"), the statement "Initially in the 160s, but has improved with fluids" was dismissed for lacking an explicit connection. In Case 16 ("Could her back pain and dizziness be concerning for a stroke?"), the suggestion "You can take the oxycodone for a short time and follow up with Dr. ___" was seen as generic advice rather than a direct answer.

Conversely, the six false-positive cases involved irrelevant sentences incorrectly identified as relevant. These errors often stemmed from chain-of-thought reasoning, in which the models associated the current sentence with earlier contextual information. The presence of medical terminology or explanatory language appeared to bias the models toward overestimating relevance. These findings suggest that the models may overly rely on surface-level cues such as technical vocabulary or narrative structure when determining relevance.

4 Conclusion

Our contribution to the ArchEHR-QA 2025 shared task presents a resource-efficient approach to clinical question answering from EHR data, showing that strong performance is achievable without fine-tuning or specialized hardware. Using multi-prompt ensembling across pre-trained models like Gemma and Mistral on consumer-grade GPUs, we improved the robustness and accuracy of paragraph relevance identification over individual models. Our modular two-stage pipeline (filtering relevant evidence before summarization) proved effective, with relevance assessment clearly improving final answer quality. The approach relies on careful prompt engineering, combining Chain-of-Thought reasoning and majority vote aggregation. While there is still room for improvement, our results demonstrate the promise of prompt-based methods with accessible LLMs as a scalable, cost-effective solution for clinical QA, especially in resource-limited settings.

Limitations

While our approach demonstrates promising results, several limitations should be acknowledged.

Firstly, our study primarily focused on relatively small, accessible models (up to 27B parameters) due to our emphasis on resource efficiency and consumer-grade hardware (1x RTX 5090). Although we show strong performance is achievable under these constraints, it is likely that larger, state-of-the-art models could yield further improvements, albeit at significantly higher computational cost. The use of quantized models, necessary for fitting them onto our hardware, might also introduce a minor performance degradation compared to full-precision versions.

Secondly, the performance of our system relies heavily on prompt engineering. While effective, identifying optimal prompts required considerable experimentation (manual trial-and-error and assistance from Gemini 2.5 Pro). The sensitivity to prompt wording means that adapting the system to different QA formats or clinical contexts might require further prompt tuning. Additionally, Chain-of-Thought prompting, while beneficial for the larger models tested, proved less reliable for smaller models, indicating limitations in their reasoning capabilities or instruction following.

Thirdly, the evaluation was conducted on the specific dataset provided for the ArchEHR-QA 2025 task. The generalization performance of our prompts and ensemble strategy on different EHR datasets or in real-world clinical deployment remains to be validated.

Fourthly, while our summarization component using Gemma generally adhered to formatting requirements, it sometimes struggled with strict length constraints and citation completeness on the first pass, necessitating multi-turn prompting and postprocessing steps. This indicates potential brittleness in complex instruction following for the summarization task.

Finally, our internal development and optimization efforts disproportionately focused on maximizing the automatic factuality score. While ensuring factual grounding is critical, this narrow focus meant that other important automatic metrics provided by the shared task (BLEU, ROUGE, BERTScore, AlginScore, MEDCON), received comparatively less attention during model and prompt selection. Consequently, the system's performance across these diverse dimensions of quality may be underdeveloped relative to its factuality performance. Furthermore, a complete assessment of the system's clinical utility, relevance, and overall correctness ultimately requires evaluation by domain experts in a practical setting.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Lingjiao Chen, Jared Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. 2024. Are more llm calls all you need? towards the scaling properties of compound ai systems. In *Advances in Neural Information Processing Systems*, volume 37, pages 45767–45790. Curran Associates, Inc.

DeepSeek AI. 2024. Deepseek - usage recommendations. https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-32B#usage-recommendations. Accessed: 2025-05-03.

Gemma Team. 2025. Gemma 3.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc.

LM Studio Community. 2025. Phi-4-mini-instruct-gguf. https://huggingface.co/lmstudio-community/Phi-4-mini-instruct-GGUF. Accessed: 2025-05-03.

Mistral AI. 2025a. Mistral ai api. https://docs.mistral.ai/api/. Accessed: 2025-05-03.

Mistral AI. 2025b. Mistral-small-3.1-24b-instruct-2503. https://huggingface.co/mistralai/Mistral-Small-3.1-24B-Instruct-2503. Accessed: 2025-05-03.

mradermacher. 2024a. Deepseek-r1-distill-llama-8b-medical-expert-ggufmradermacher/deepseek-r1-distill-llama-8b-medical-expert-gguf. https://huggingface.co/mradermacher/DeepSeek-R1-Distill-Llama-8B-Medical-Expert-GGUF. Accessed: 2025-05-03.

mradermacher. 2024b. Openbiollm-llama3-70b-i1-gguf. https://huggingface.co/mradermacher/OpenBioLLM-Llama3-70B-i1-GGUF. Accessed: 2025-05-03.

mradermacher. 2025a. Biomistral-medmnx-gguf. https://huggingface.co/mradermacher/BioMistral-MedMNX-GGUF. Accessed: 2025-05-03.

mradermacher. 2025b. Clinicalgpt-r1-qwen-7b-en-preview-gguf. https://huggingface.co/mradermacher/ClinicalGPT-R1-Qwen-7B-EN-preview-GGUF. Accessed: 2025-05-03.

mradermacher. 2025c. Med-chatbot-r1-qwen-7b-gguf. https://huggingface.co/mradermacher/Med-Chatbot-R1-Qwen-7B-GGUF. Accessed: 2025-05-03.

Saeel Sandeep Nachane, Ojas Gramopadhye, Prateek Chanda, Ganesh Ramakrishnan, Kshitij Sharad Jadhav, Yatin Nandwani, Dinesh Raghu, and Sachindra Joshi. 2024. Few shot chain-of-thought driven reasoning to prompt LLMs for open-ended medical question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 542–573, Miami, Florida, USA. Association for Computational Linguistics.

Maximilian Frederik Russe, Marco Reisert, Fabian Bamberg, and Alexander Rau. 2024. Improving the use of llms in radiology through prompt engineering: from precision prompts to zero-shot learning. *RoFo: Fortschritte auf dem Gebiete der Röntgenstrahlen und der Nuklearmedizin*, 196(11):1166–1170. Epub 2024 Feb 26.

Sonish Sivarajkumar, Mark Kelley, Alyssa Samolyk-Mazzanti, Shyam Visweswaran, and Yanshan Wang. 2024. An empirical evaluation of prompting strategies for large language models in zero-shot clinical natural language processing: Algorithm development and validation study. *JMIR Med Inform*, 12:e55318.

Sarvesh Soni and Dina Demner-Fushman. 2025a. A dataset for addressing patient's information needs related to clinical course of hospitalization. *arXiv preprint*.

Sarvesh Soni and Dina Demner-Fushman. 2025b. Overview of the archehr-qa 2025 shared task on grounded question answering from electronic health records. In *The 24th Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, Vienna, Austria. Association for Computational Linguistics.

Unsloth Team. 2025. How to run gemma 3 effectively with our ggufs on llama.cpp, ollama, open webui and how to fine-tune with unsloth! https://docs.unsloth.ai/basics/gemma-3-how-to-run-and-fine-tune. Accessed: 2025-05-03.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2024. Soft self-consistency improves language models agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 287–301, Bangkok, Thailand. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

WhyHow.AI Team. 2024. Patientseek. https://huggingface.co/whyhow-ai/PatientSeek. Accessed: 2025-05-03.

Han Yang, Mingchen Li, Huixue Zhou, Yongkang Xiao, Qian Fang, and Rui Zhang. 2023. One llm is not enough: Harnessing the power of ensemble learning for medical question answering. *medRxiv*.

A Prompts

This section outlines the prompt structure used during our experiments. All prompts follow the OpenAI API format, with each prompt represented as a list of message objects.

For paragraph relevance assessment, we typically separate the main instruction, the question, and the list of paragraphs into distinct message objects, as shown below.

```
messages = [
    {
        "role": "user",
        "content": f"{prompt}"
    },
    {
        "role": "user",
        "content": f"Question: {q}"
    },
    {
        "role": "user",
        "content": f"List of paragraphs: {l}"
    }
]
```

The `prompt` variable is the component we vary most frequently across experiments. The `q` and `l` variables consistently represent the clinician-derived question and the list of paragraphs, respectively. Each paragraph in the list is formatted as: `#{i} - "{s}"`, where `i` is the paragraph ID and `s` is the paragraph content.

For summarization, we use a single message object with the role `"user"`, applying a prompt, referred to as Prompt Z, crafted through a combination of manual trial and error and refinements

suggested by Gemini 2.5 Pro. The variables enclosed in curly braces within Prompt Z are substituted with their respective values, as done for the paragraph relevance assessment prompts. The full content of Prompt Z is included in the list of prompts below.

A.1 Prompt A

This prompt was selected after several manual iterations, relying on intuitively designed instructions.

You will receive a clinical question interpreted from a patient's question and a list of paragraphs extracted from a clinical note.

The questions are asked through the patient portal by patients.

Create a chain of thought to determine if the paragraph is relevant to answering the question.
Put your reasoning between <think> and </think> tags.

Is the paragraph number %paragraph-number% (indexed from 0) relevant to answering the question?
The paragraph does not need to give a full answer, but should be relevant in formulating the answer.
Give a Yes or No answer.

A.2 Prompt B

This prompt uses a different message structure than the one previously described and can be found in the GitHub source code. It was generated by Gemini 2.5 Pro when asked to refine an arbitrary prompt (e.g., Prompt A) with the goal of maximizing performance.

*Role:** You are an expert clinical information analyst specializing in evaluating text relevance using Mistral models.

Context: You will be provided with a clinical question (derived from a patient's query via a patient portal) and a list of numbered paragraphs extracted from a clinical note.

Goal: Determine if a *specific* paragraph from the list is relevant to answering the provided clinical question.

Definition of Relevance: A paragraph is relevant if it contains information that helps answer, contributes to answering, or is directly related to the topic of the question. It does *not* need to provide the complete answer on its own.

Task Instructions:
1. You will receive the Clinical Question and the full List of Paragraphs first.
2. Then, you will be asked to evaluate a *specific* paragraph, identified by its number (0-indexed).
3. Focus your analysis *exclusively* on the content of the specified paragraph number. Do not base your relevance decision on other paragraphs in the list.
4. Generate a step-by-step Chain of Thought (CoT) reasoning process to justify your decision. Clearly explain *why* the specified paragraph is or is not relevant based on the question's topic and the paragraph's content.
5. Enclose your entire Chain of Thought reasoning securely within '<think>' and '</think>' tags.
6. Immediately following the closing '</think>' tag, provide your final answer as *only* "Yes" or "No".

Output Format:
<think>
[Your detailed step-by-step reasoning comparing the specific paragraph's content to the question's requirements, focusing only on the specified paragraph.]
</think>
[Yes or No]

A.3 Prompt C

Same as Prompt A but we don't specify the source of the questions.

You will receive a clinical question interpreted from a patient's question and a list of paragraphs extracted from a clinical note.

Create a chain of thought simulating a doctor's (that needs to provide a response) thinking to determine if the paragraph is relevant to answering the question.
Put your reasoning between <think> and </think> tags.

Is the paragraph number %sentence-number% (indexed from 0) relevant to answering the

question?
The paragraph does not need to give a full answer, but should be relevant in formulating the answer.
Give a Yes or No answer.

A.4 Prompt D

Same as Prompt A, but without chain of thought.

You will receive a clinical question interpreted from a patient's question and a list of paragraphs extracted from a clinical note.

The questions are asked through the patient portal by patients.

Is the paragraph number %paragraph-number% (indexed from 0) relevant to answering the question?
The paragraph does not need to give a full answer, but should be relevant in formulating the answer.
Give a Yes or No answer.

A.5 Prompt E

Another recommendation from Gemini 2.5 Pro.

*Role:** You are an expert clinical information analyst. Your purpose is to evaluate the relevance of clinical note paragraphs to patient questions.

Context: You will receive:
1. A **Clinical Question** from a patient.
2. A **List of Paragraphs** (0-indexed) from a clinical note.
3. A specific **Paragraph Number** to evaluate.

Goal: Determine if the *specified paragraph* (identified by its number) is relevant for answering the Clinical Question.

Definition of Relevance:
* A paragraph is relevant if its content *directly addresses, contributes to answering, or is topically related* to the Clinical Question.
* When evaluating the specified paragraph, consider its *intrinsic content* primarily.
* Also, consider its *contextual value*: Does it provide essential background for another relevant paragraph? Is it the *most* relevant piece of information available, even if only weakly related, especially if other paragraphs are irrelevant?

Instructions:
1. **Analyze the Request:** Understand the Clinical Question and review all provided paragraphs to grasp the overall context.
2. **Focus on the Target:** Concentrate your relevance analysis on the *specific* paragraph number provided in the final user request.
3. **Perform Chain-of-Thought (CoT) Reasoning:** Generate a step-by-step reasoning process detailing your evaluation.
* Start by stating the paragraph number being evaluated.
* Summarize the core information in the specified paragraph.
* Compare this information directly against the Clinical Question.
* Explicitly discuss *how* or *why* it is (or isn't) relevant.
* If applicable, briefly mention its contextual role (e.g., "This paragraph provides context for paragraph X," or "While weakly related, it's the only paragraph mentioning Y topic").
* Conclude your reasoning with a clear statement about the relevance of the *specified paragraph*.
4. **Enclose Reasoning:** Place your *entire* step-by-step reasoning within ‘<think>‘ and ‘</think>‘ tags. **Crucially, there should be NO text before the opening ‘<think>‘ tag and NO text between the closing ‘</think>‘ tag and the final Yes/No answer.**
5. **Provide Final Answer:** Immediately following the closing ‘</think>‘ tag, output *only* the word "Yes" or "No" indicating the relevance of the *specified paragraph*.

Output Format:
<think>
[Step-by-step reasoning analyzing the specified paragraph's relevance to the question, considering context as defined above.]
</think>
[Yes or No]

A.6 Prompt Z

Goal: Create a concise (70 words) answer for the **Clinical Question** using *only* the information present in the **Relevant paragraphs**.

The questions are asked through a patient portal.

Patient Question:
{patient_question}

Clinical Question (Derived from patient question):
{question}

Relevant paragraphs (with 1-based indices):
{formatted_relevant_paragraphs}

Citation style: Paragraph's index between | | symbols. For example: |1| or |2,7| or |1,2,3,4| or |5,7,9|. Citations must be comma separated.

Output Detail:
- The sentences in the generated answer may be supported using one, multiple, or none (unsupported) of the paragraphs from the clinical note.
- The unsupported sentences in the answer may be ignored during the quantitative evaluation.
- The answers should be in the professional register to better match the contents of the clinical notes. Simplification of answers to lay language is assumed to be performed later and is not the focus of this task.
- The generated answer should be limited to 75 words, which roughly correspond to 5 sentences. This is based on our observations from the baseline experiments and existing literature supporting that a paragraph-long answer is preferred by users.
- There are no limitations to the number of note sentences cited.

You need to answer the patient's question, but do not take the information provided in it for granted and do not refer to it in your answer.
The answer should sound natural and be a coherent response to the question.
Do not add any additional information beside the answer such as "Your summary: ".
Your answer should be a summary of the information in the relevant paragraphs you found, with few sentences, and more like a long paragraph.

A.7 Other Prompts

We do not list all the prompts generated by Gemini, as they are largely similar, differing only in minor adjustments aimed at reducing false positives and/or false negatives to better balance the model's behavior.

B Ensembling

We try different ensembling strategies, justified by various scores obtained by individual models.

B.1 Candidate one

The best overall factuality ensemble overall, but with a bit worse confusion matrix than the chosen one. Excellent balance of top performance, high precision diversity, and balanced size/generation diversity.

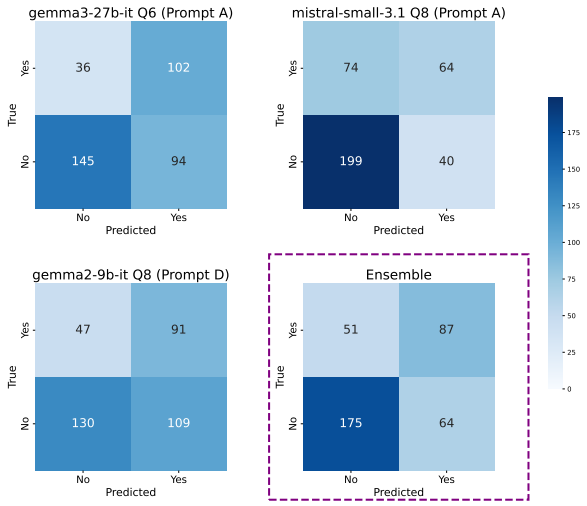

Figure 2: Ensemble candidate one.

B.2 Candidate two

Combines the best Gemma with both the high-precision and high-recall Mistral variants, maximizing Mistral architectural presence. Solid score with balanced stats.

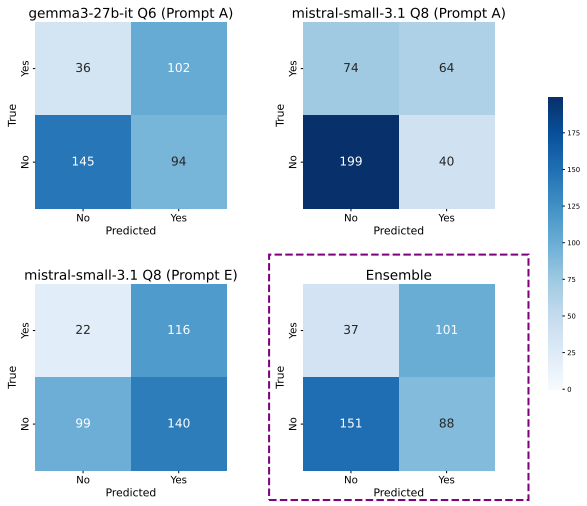

Figure 3: Ensemble candidate two.

B.3 Candidate three

An example of a high overall factuality score (56.359) with a slightly worse confusion matrix.

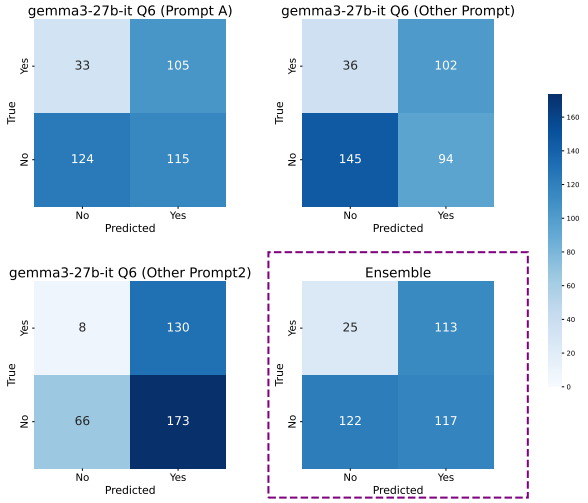

Figure 4: Ensemble candidate three.

B.4 Candidate four

This is yet another example where ensembling contributes to more robust predictions, although it does not yield the strongest overall performance among our configurations.

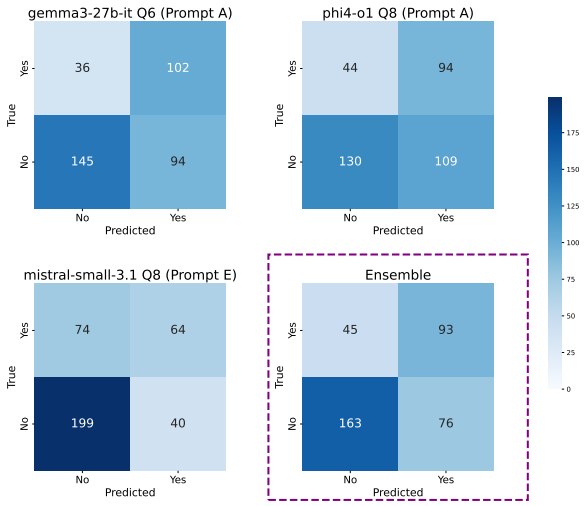

Figure 5: Ensemble candidate four.

B.5 Candidate five

Three models finetuned on clinical data.

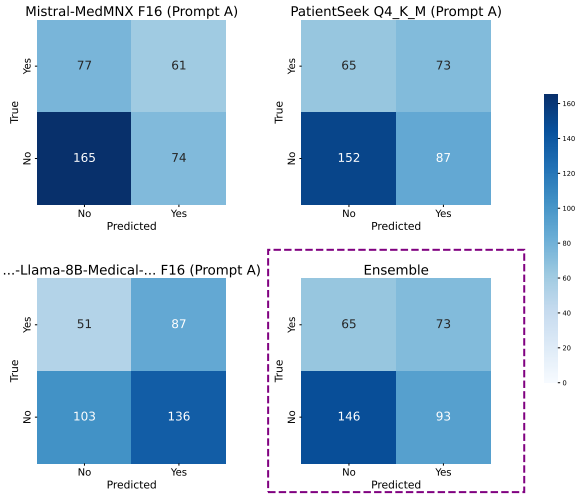

Figure 6: Ensemble candidate five.

C Detailed Factuality Scores

The detailed scores of the best overall factuality models.

		Lenient						Strict					
Model	Overall Factuality Score	Micro F1	Micro Recall	Micro Precision	Macro F1	Macro Recall	Macro Precision	Micro F1	Micro Recall	Micro Precision	Macro F1	Macro Recall	Macro Precision
gemma3-27b-it	56.044	63.614	69.841	58.407	61.791	70.745	63.075	56.044	73.913	45.133	55.379	74.500	49.222
mistral-small-3.1-24b	54.085	63.054	67.725	58.986	60.287	68.528	64.799	54.085	69.565	44.240	50.427	68.236	48.340
gemma3-12b-it	52.029	60.851	75.661	50.890	58.600	76.511	52.008	52.029	78.986	38.790	50.977	80.806	38.964
gemma2-9b-it	52.000	51.372	54.497	48.585	46.155	57.725	52.634	52.000	65.942	42.925	46.656	65.486	46.112
phi4	51.075	55.792	62.434	50.427	47.918	64.843	47.594	51.075	68.841	40.598	42.178	67.028	36.809
phi4-o1	50.000	60.422	68.254	54.202	56.766	68.274	57.243	50.000	68.116	39.496	47.773	67.972	43.843
deepseek-llama-8b	44.444	46.729	39.683	56.818	43.082	44.511	58.599	44.444	43.478	45.455	38.962	45.528	40.402
phi4-QwQ	40.892	42.500	35.979	51.908	35.067	34.888	44.154	40.892	39.855	41.985	34.895	37.778	37.061
phi4-mini-it	40.260	41.783	39.683	44.118	36.879	41.867	49.449	40.260	44.928	36.471	33.520	40.722	37.782
deepseek-qwen-32b	30.244	28.906	19.577	55.224	23.980	20.194	42.943	30.244	22.464	46.269	24.650	21.250	38.402
*all-relevant**	48.763	61.264	100.000	44.159	60.352	100.000	45.404	48.763	100.000	32.243	48.484	100.000	33.060
baseline (LLaMA 3.3 70B)	35.900	39.200	27.000	71.800	46.500	38.900	78.500	43.100	32.600	63.400	49.400	47.100	70.300

Table 5: Scores on the **dev dataset**. The *all-relevant* baseline assumes all paragraphs are relevant.