LIMICS at ArchEHR-QA 2025: Prompting LLMs Beats Fine-Tuned Embeddings

Adam Remaki[1] Armand Violle[1] Vikram Natraj[1] Étienne Guével [2] Akram Redjdal[1,3]

[1]Sorbonne Université, Inserm, Université Sorbonne Paris-Nord, Laboratoire d'Informatique Médicale et d'Ingénierie des Connaissances en e-Santé, Paris, France

[2]Sorbonne Cluster for Artificial Intelligence, Paris, France

[3]Univ Gustave Eiffel, Aix-Marseille Univ, LBA, F-13016 Marseille, France

Correspondence: adam.remaki@etu.sorbonne-universite.fr, armand.violle@sorbonne-universite.fr, natrajvikram.sivabalasubramanian@sorbonne-universite.fr, etienne.guevel@sorbonne-universite.fr, akram.redjdal@esiee.fr,

Abstract

In this paper, we investigated two approaches to clinical question-answering based on patient-formulated questions, supported by their narratives and brief medical records. The first approach leverages zero- and few-shot prompt engineering techniques with GPT-based Large Language Models (LLMs), incorporating strategies such as prompt chaining and chain-of-thought reasoning to guide the models in generating answers. The second approach adopts a two-steps structure: first, a text-classification stage uses embedding-based models (e.g., BERT variants) to identify sentences within the medical record that are most relevant to the given question; then, we prompt an LLM to paraphrase them into an answer so that it is generated exclusively from these selected sentences. Our empirical results demonstrate that the first approach outperforms the classification-guided pipeline, achieving the highest score on the development set and the test set using prompt chaining. Code: `github.com/armandviolle/BioNLP-2025`

1 Introduction

The ArchEHR-QA 2025 shared task (Soni and Demner-Fushman, 2025b) focused on grounded electronic health record question answering. The goal was to design a system that could answer patients' questions based on sentences from the patient's medical notes providing evidence supporting the answer's statements.

A development dataset (Soni and Demner-Fushman, 2025a) of 20 cases was at our disposal, structured as follows in `XML` format: a *patient narrative* ($P^{narrative}$), where the patient states their situation and asks their question(s); the original patient question ($Q^{patient}$); its clinician reformulation ($Q^{clinician}$); and a *medical note* summarizing the patient's history, presented both as a whole and sentence-by-sentence. Additionally, a `JSON` file was provided which contained a label for each sentence ("essential," "supplementary," or "not relevant") with respect to the questions.

Three guidelines were set for the generated answers: 1. a maximal length of 75 words, 2. one sentence per line with, at the end of each line, the cited `id` attribute(s) of the supporting medical note sentence(s) and 3. avoiding using external data or knowledge (relaxed later on).

The answers went through a two-step evaluation based on their *factuality*, i.e. the effective citation of "essential" sentences in the answers, and their *relevance*, i.e the semantic similarity with the inputs. Consequently, we tried to design systems suiting this layered structure with a *classification* step of the medical note sentences' relevance and a *summarization* step rephrasing relevant sentences into an answer to the $Q^{patient}$. We confronted these approaches to Large Language Models (LLMs) prompting strategies which we considered as baselines.

2 Methods

2.1 Sentence relevance classification

In this section, we present a method to identify question-relevant sentences using Sentence-BERT's bi-encoder and cross-encoder architectures (Reimers and Gurevych, 2019), enabling the LLM to generate answers grounded solely in the extracted content.

2.1.1 Single-sentence classification using short-context embeddings

First, we evaluated each clinical sentence individually against $P^{narrative}$ using pairwise comparisons.

We employed the pretrained cross-encoder ms-marco-MiniLM-L12-v2, which was originally trained on the MS MARCO dataset (Bajaj et al., 2016), a large corpus of query-document

BioNLP 2025 Shared Tasks, pages 150–159

pairs ranked by relevance, and then fine-tuned on 15 cases (5 for validation) from ArchEHR dataset (Soni and Demner-Fushman, 2025a).

We also evaluated a bi-encoder model Jina-embedding v3 (Sturua et al., 2024). Sentences with cosine similarity score ≥ 0.5 to the query were considered essential, using 0.5 as a midpoint heuristic within the range of [0, 1]. eFigure 1 in the Appendix shows the distribution of similarity scores across label categories.

2.1.2 Multi-sentence classification using long-context embeddings

In our second approach, we utilized Jina-embedding v3's 8k-token capacity to process multiple sentences in context. Unlike the single-sentence setup, each example consists of a concatenated input of the $P^{narrative}$ and candidate sentences, formatted as `[Question] </s> [Sentence 1] </s> ... [Sentence N]`. The model outputs binary labels indicating whether each sentence is *Essential* or not (*Supplementary/Not Relevant*).

2.1.3 Data augmentation for robust classification

To address data scarcity, we generated 748 synthetic question-answer pairs from i2b2 (Uzuner et al., 2011), emrQA (Pampari et al., 2018), and MIMIC-III (Johnson et al., 2016)) clinical corpora. Each instance contained: (i) a question (generated via OpenAI's gpt-o4-mini with manual prompt tuning), (ii) clinical note excerpts, and (iii) binary relevance labels. For sentence selection, we embedded text using `text-embedding-ada-002`, retrieved top-k matches via FAISS, and assigned labels (*Essential/Supplementary/Not relevant*) based on ranking position. We evaluated augmentation effectiveness by fine-tuning both a ms-marco-MiniLM-L12-v2 cross-encoder (short-context) and a Jina Embedding v3 classifier (long-context). Details on the training are available in the section A of the Appendix.

2.2 Prompting LLMs for answer generation

In this section, we present an end-to-end method that generates the answer using LLMs. To evaluate different prompting strategies, we used the OpenAI API with data sharing explicitly disabled, ensuring that no inputs, outputs, were used to train or improve OpenAI models.

2.2.1 Zero-shot prompting

Zero-shot prompting was our first approach to generate the answer, specifically to understand how effectively LLMs could tackle both classification and paraphrasing sub-tasks at once. We adapted the prompt's *instructions* and format according to the observed output and best practices found in the literature, as well as diverse combinations of input data. We tested GPT-4.1-mini (OpenAI, 2025) and Mistral Large (AI, 2024). More details on the prompts can be found in eFigure 2 and eFigure 3 of the Appendix.

2.2.2 Prompting reasoning steps with chain-of-thought

As chain-of-thought (CoT) has proven to be an efficient prompting strategy to increase model reasoning abilities, we decomposed the task in a sequence of distinct steps to help the model tackle the task. We incorporated these *reasoning steps* into the system prompt and fed it to a GPT-4.1-mini (OpenAI, 2025) model, mostly to control outputs' format, trying to force the model to autonomously check and adapt its answer to the expected format. Prompt is presented in eFigure 4 of the Appendix.

2.2.3 Few-shot prompting

In few-shot prompting, we created pairs of question-answers to add as examples in our prompts. To generate the "gold standard" answers, we prompted GPT-4.1-mini (OpenAI, 2025) to paraphrase essential sentences from the medical note, based on the available labels in the dataset, into an answer to the $Q^{patient}$. Then, for each case, we sampled randomly a subset of pairs among the other available cases that were included in the prompt as examples, before the inference case's input. Prompts are presented in eFigure 5 and eFigure 6 of the Appendix.

2.2.4 Prompt chaining: `divide-and-conquer`

We adopted a prompt chaining approach based on the divide-and-conquer principle, breaking down the overall task into a structured sequence of smaller, interdependent subtasks. Each subtask is addressed by a language model, and the intermediate outputs are passed as inputs to subsequent stages. An overview of the full pipeline is shown in Figure 1.

This pipeline comprised five steps:

(i) Free answer generation. We prompted `o4-mini-2025-04-16` to generate a detailed and

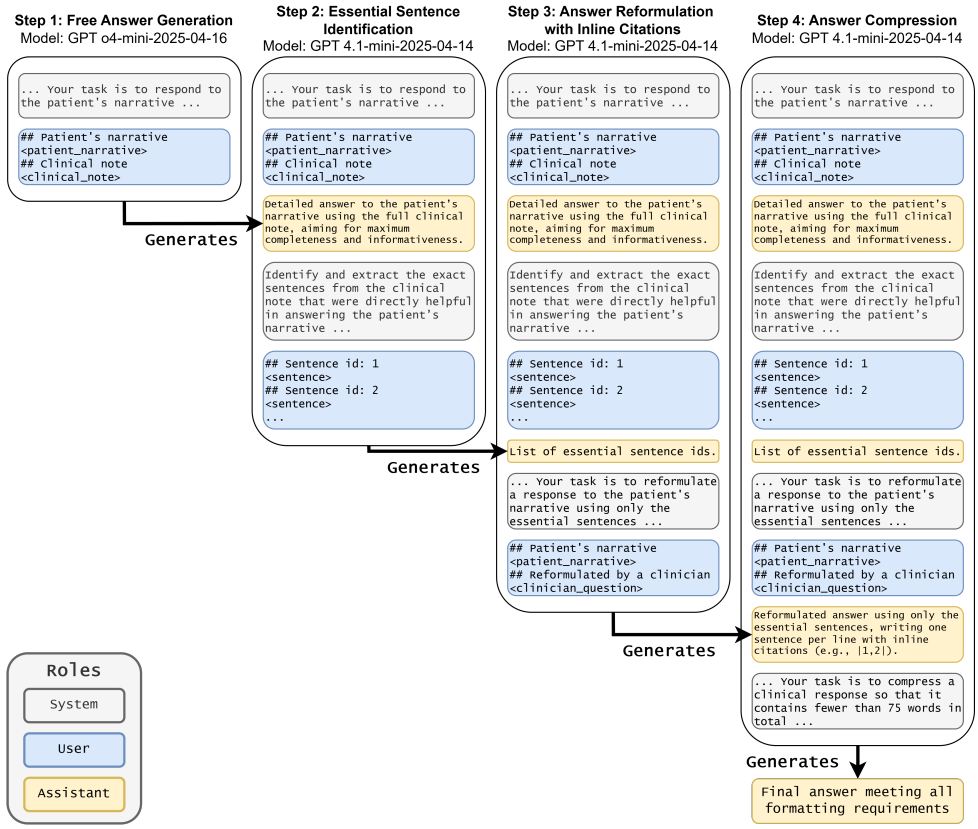


Figure 1: Overview of the prompt chaining workflow: each step refines the answer, improves grounding, and enforces formatting.

informative answer, given the $P^{narrative}$ and the associated full medical note. The prompt was designed to encourage completeness, with no formatting constraints, in order to generate as many relevant elements of the medical note as possible.

(ii) Essential sentence identification. The output from Step 1, along with the medical note (provided as a list of markdown-formatted sentences), is passed to `gpt-4.1-mini-2025-04-14`. The model was prompted to identify the minimal subset of sentences that directly support the answer.

(iii) Answer reformulation with inline citations. Using only the essential sentences from Step 2, the same model was prompted to reformulate the answer in a structured format. Each sentence appears on a new line and includes inline citations (e.g., `|3,7|`) referencing the supporting sentence IDs.

(iv) Answer compression. We prompted the same model to compress the reformulated answer into a concise version constrained to 75 words, while preserving the same inline citations.

(v) Strict answer compression (optional). If the compressed answer still exceeds 75 words, we prompt the same model again using the same compression rules, but presented in a more structured and imperative format. We allow up to three retries. If the constraint remains unmet, we restart the entire pipeline with a new seed.

The system prompts used in the pipeline are provided in eFigure 7 in the Appendix. One may note that only prompt chaining and CoT consistently produced answers within the 75-word limit. Other methods required post-processing compression, as described in Section E of the Appendix.

3 Results

3.1 Sentence relevance classification

Table 1 reports the performance of various embedding-based models in identifying essential sentences. We present precision, recall, and F1-score for each model configuration including the pretrained cross-encoder ms-marco-MiniLM-L12-v2 (with 33.4 million parameters), Jina Embedding v3 (Sturua et al., 2024) (with 572 million parameters) evaluated in both a bi-encoder (single-sentence) and a multi-sentence classification setting. The second and third columns indicate fine-tuning on the ArchEHR sample and the augmented dataset, respectively.

Model	ArchEHR FT*	Augmented FT*	Precision	Recall	F1-score
ms-marco-MiniLM			0.24 (0.20-0.28)	0.51 (0.42-0.60)	0.29 (0.25-0.34)
ms-marco-MiniLM	✓		0.37 (0.34-0.41)	0.28 (0.24-0.33)	0.29 (0.26-0.32)
ms-marco-MiniLM		✓	0.36 (0.35-0.37)	**0.90 (0.88-0.92)**	0.51 (0.50-0.52)
Jina (single-sentence)			**0.49 (0.41-0.61)**	0.55 (0.43-0.66)	**0.52 (0.44-0.59)**
Jina (multi-sentence)		✓	0.39 (0.33-0.46)	0.70 (0.59-0.82)	0.50 (0.44-0.56)

Table 1: Performance of embedding-based models for essential sentence classification on the development set. Metrics are reported as mean (95% confidence interval). *FT: fine-tuned.

3.2 Prompting LLMs for answer generation

Table 2 reports the performance of various prompting methods using large language models. The first column lists the prompting strategies. The second column presents the *factuality score*, measured as the F1-score on the essential sentence identification task. The third column shows the *relevance score*, computed as the average of several semantic similarity metrics (bleu, rouge, medcon, alignescore, bertscore, and sari) between the generated answer and the concatenation of the essential sentences, the $P^{narrative}$, and the $Q^{clinician}$.

Development Set		
Method	**Factuality**	**Relevance**
Zero-shot Mistral	51.1 (2.6)	31.1 (0.7)
Zero-shot GPT	56.6 (2.1)	32.5 (0.6)
Chain-of-thought	52.4 (1.9)	33.2 (0.5)
Few-shot	54.5 (1.9)	32.5 (0.5)
Prompt chaining	**59.3** (0.2)	**37.9** (0.3)
Test Set		
Method	**Factuality**	**Relevance**
Prompt chaining	**54.2**	**35.5**

Table 2: Comparison of methods on factuality and relevance score for the development and test sets. Results are reported as mean (standard deviation) over 10 random seeds for the development set. Test result is shown for the best-performing method.

4 Discussion

Our findings highlight several important insights regarding the classification of essential sentences in clinical narratives. First, fine-tuning on the ArchEHR dataset alone did not yield consistent performance gains. We attribute this to the dataset's limited size (only 20 annotated cases), which is insufficient for effective adaptation. Moreover, the augmented dataset significantly improved the performance of the cross-encoder model. It not only boosted F1-scores but also reduced variance across runs, suggesting that the model benefited from the synthetic data. However, fine-tuning the Jina-Embedding v3 model with augmented data and multi-sentence input did not improve performance. This may be due to the LoRA adapters being poorly suited for this fine-tuning setup, or because the model's initial performance left little room for improvement. Further investigation is needed to understand the cause.

Despite extensive experimentation with embedding-based approaches, including both single and multi-sentence configurations, we observed that LLMs outperformed them on the sentence classification task. Nevertheless, it is noteworthy that a relatively small 33M-parameter BERT cross-encoder achieved the same F1-score of 0.51 as the much larger 123B-parameter Mistral large model, highlighting a meaningful tradeoff between performance and computational cost.

Results indicate that prompting strategies isolating subtasks through sequential prompt chaining led to more accurate sentence classification, improved answer relevance, and reduced variability, with standard deviation nearly ten times smaller for the factuality score. Interestingly, zero-shot prompting outperformed both few-shot and CoT approaches. While the reason remains unclear, this may suggest that overly long system prompts were less effective for this task.

5 Conclusion

This study addressed the ArchEHR-QA challenge, where the goal is to answer patient-specific clinical questions by identifying and citing essential sentences from clinical notes. For sentence classification, augmenting the dataset with synthetic QA pairs improved performance and reduced variation. While embedding models such as bi-encoders and cross-encoders produced solid results, LLMs consistently outperformed them. For this task, prompt chaining, which isolates subtasks, gave the best result.

Limitations

The first limitations to mention are related to the LLMs we used for prompting strategies. Indeed OpenAI's GPT models and Mistral AI's models are proprietary and thus lack transparency on their training process (e.g data corpora used) and some functionality (e.g "determinism not guaranteed" when fixing seed parameter). In research, it is a major drawback as it is hard to truthfully build upon undisclosed features. Moreover, these models are pay-as-you-go, so we stuck to smaller, cheaper models that enabled us to run multiple experiments (we spent almost 100$ worth of OpenAI tokens for the challenge). Scaling up to models such as GPT-4.5, o1 or o3 may have improved performances-but it comes at a cost.

One limitation of our synthetic dataset is that the complexity of the sentence classification task often requires domain-specific medical knowledge. As a result, the generated data may not fully capture the nuances present in real clinical scenarios. Incorporating validation and annotation by medical experts could help ensure the reliability and clinical relevance of the synthetic data, thereby increasing its impact for downstream tasks.

To conclude, we reflect on the evaluation methodology, particularly the suitability of BLEU (Papineni et al., 2002) for assessing the *relevance* metric. BLEU includes a brevity factor that can disproportionately penalize predicted answers that differ in length from the reference. In our case, relatively short predicted answers (with a maximum expected length of 75 words) were evaluated against much longer references composed of concatenated $P^{narrative}$, $Q^{clinician}$, and essential sentences. This mismatch in length likely contributed to the uniformly low BLEU scores observed across the leader board.

Acknowledgments

This work was performed using HPC resources from GENCI–IDRIS (Grant 2025-AD011015342R1). Some computations were also performed using the GPU cluster resources of the Sorbonne Center for Artificial Intelligence (SCAI) at Sorbonne University. The authors thank Xavier Tannier and Stéphane Dohayon for their valuable advice on the design.

References

Mistral AI. 2024. Au Large.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, and 1 others. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*.

Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards general text embeddings with multi-stage contrastive learning. *arXiv preprint arXiv:2308.03281*.

OpenAI. 2025. Introducing GPT-4.1 in the API.

Anand Pampari, Pradeep Raghavan, Jinfeng Liang, and Jian Peng. 2018. emrqa: A large corpus for question answering on electronic medical records. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2357–2368.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

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.

Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Andreas Koukounas, Nan Wang, and Han Xiao. 2024. jina-embeddings-v3: Multilingual embeddings with task lora. *Preprint*, arXiv:2409.10173.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Appendix

A Training details for sentence classification

A.1 Cross-encoder finetuning

We trained the cross-encoder on both the augmented dataset and archEHR sample using identical hyperparameters: binary cross-entropy loss, AdamW optimizer (learning rate 2×10^{-5}), and batch size of 64. Training proceeded for 10 epochs.

A.2 Multi-sentence classification

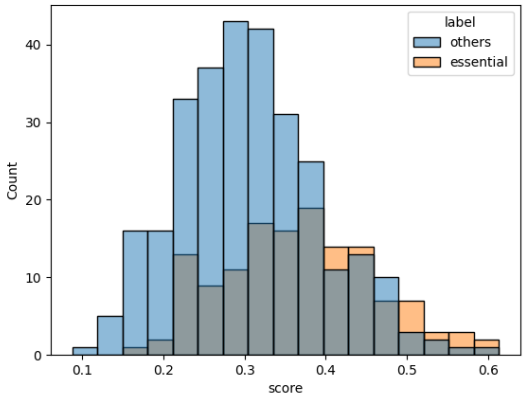
We fine-tuned the LoRA adapter specifically designed for classification in the Jina-Embedding v3 model, which includes five task-specific LoRA adapters in total. These adapters are integrated into the embedding and linear layers of the multi-head attention mechanism, with a rank of 4 and $\alpha = 1$. We fine-tuned the classification adapter on our synthetic QA dataset for 5 epochs using the AdamW optimizer (learning rate: 2×10^{-5}). Due to the long input sequences, we used a batch size of 1. Class imbalance was addressed using a weighted `BCEWithLogitsLoss`, and mixed-precision training (`bfloat16`) was enabled via `torch.cuda.amp`. Inputs followed the format: `[Question] </s> [Sentence 1] </s> ... [Sentence N]` The final prediction was produced by a linear head applied to sentence embeddings extracted at the `</s>` token positions.

B Bi-encoder classification

The bi-encoder approach using Jina Embedding v3 demonstrated significantly higher cosine similarity scores between patient questions and sentences labeled as "Essential" (mean = 0.62) compared to other categories (mean = 0.41, t-test $p < 1 \times 10^{-10}$). eFigure 1 shows the distribution of similarity scores across label categories, revealing clear separation between essential and non-essential phrases.

C Classification with a large encoder

Here we report an evaluation of gte-Qwen2-7B-instruct (Li et al., 2023). When using the prompt presented in eFigure 8, the model ended up overfitting on the training set while failing to generalize the information on the validation set . For the accuracy it reached 0.98 and the f1 0.98 in training, while in validation the best metrics were : f1 0.33, recall 0.28, precision 0.41.



eFigure 1: Distribution of cosine similarity between question and sentence

D CoT and Few-Shot implementation details

We used the three roles offered by the `chat.completions.create` of the OpenAI API client: `system` to describe the general beahior of the model, `user` to input data and additional information helping the model to respond such as reasoning steps or examples, and `assistant` to input example responses for few-shot prompting. The `system` prompt and `user` prompts were very similar in few-shot (see Figure 6) and CoT (see Figure 4). For the `user` inference prompt, we just concatenated the selected data consisting in $P^{narrative}$, $Q^{patient}$, $Q^{clinician}$ and the sentence-by-sentence medical note excerpt in few-shot, while in CoT we first prompted the reasoning steps and then the same inputs.

For CoT, we created examples using the first 2 cases by prompting successively the reasoning steps and input data in ChatGPT. We then used the final answers as “gold standard” to provide an example for each case before inference, which resulted in the following (considering a single case):

1. We prompt the `system` role (see Figure 4 for detailed prompts).

2. We prompt the `user` role with the reasoning steps and the input data of an example case.

3. We prompt the `assistant` role with the final answer obtained with ChatGPT.

4. Finally, we prompt the actual inference case to the `user` role.

In few-shot, before generating the answers, we zero-shot a summarization task with the `system` prompt on Figure 5 and a `user` prompt containing essential sentences only and $Q^{patient}$. We used them in the few-shot strategy to provide as follows, considering a single case:

1. We randomly sampled 5 cases among the 19 other available to serve as examples.
2. We prompt the `system` role.
3. We prompt 5 times `user-assistant` roles successively, `user` prompts being the sampled cases formatted as inference prompts, and `assistant` prompts being the corresponding sampled cases' previously generated paraphrase.
4. Finally, we prompt the actual inference case to the `user` role.

For CoT, we created examples using the first 2 cases by prompting successively the reasoning steps and input data in ChatGPT. We then used the final answers as "gold standard" to provide an example for each case before inference, which resulted in the following (considering a single case):

1. We prompt the `system` role (see Figure 4 for detailed prompts).
2. We prompt the `user` role with the reasoning steps and the input data of an example case.
3. We prompt the `assistant` role with the final answer obtained with ChatGPT.
4. Finally, we prompt the actual inference case to the `user` role.

E Answer post-processing to enforce word limit

To enforce the 75-word limit required by the evaluation protocol, we apply a post-processing script to the model-generated answers. Although the summarization prompt explicitly specifies this limit, responses occasionally exceed it. The cleanup process ensures validity and evaluation compatibility through the following steps:

- **Grouped summarization:** Consecutive sentences with identical citations are grouped and summarized using `GPT-4.1-mini`, with a dynamic word limit to ensure the final output stays within the 75-word constraint.
- **Citation preservation:** Citations from the original outputs are preserved and reattached to the corresponding summarized segments to maintain factual alignment.
- **Fallback handling:** If summarization fails or exceeds the limit, a generic sentence is inserted: *"Additional supporting evidence."* with the missing citations appended.
- **Format compliance:** The evaluation script requires at least one citation line in the format `Sentence or summary. |citation_id(s)|`, but not necessarily one for every sentence.

This method prioritizes factual consistency and strict format adherence, and was found to be effective when used with a controlled summarization model such as `GPT-4.1-mini`.

F System prompts

```
You are a clinical assistant. Carefully review the patient narrative,
clinician question, and the provided clinical note sentences. Provide a
medically accurate and detailed answer to the clinician's question.

Example:

Patient Narrative:
"I had difficulty breathing and fever, and was hospitalized."

Clinical Note Sentences:
[0] Patient admitted on Wednesday evening.
[1] Patient complained of difficulty breathing.
[2] Chest X-ray showed clear infiltrates in lower lobes.
[3] White blood cell count significantly elevated, indicative of infection.
[4] Patient was discharged after five days.

Correct JSON Response:
{
  "answer": "Yes, the patient has clinical evidence of pneumonia |2,3|,
supported by X-ray infiltrates and elevated white blood cell count |2|."
}

Important Instructions:

Include ALL sentences that could partially or fully support answering
the clinician's question by mentionning them in |sentenceIDs|.
If uncertain, lean towards including the sentence.
Prioritize recall and completeness of supporting evidence.

Patient Narrative:
{patient_narrative}


Clinical Note Sentences:
{formatted_sentences}


Respond strictly in the JSON format:
{{
  "answer": "your detailed answer here cite sentences IDs between ||"
}}
```

eFigure 2: Mistral large zero-shot prompt

```
You are a clinical assistant. Your goal is to answer the patient's question
using only the sentences provided below.
- Every sentence used must be cited at the end using |sentence_id|.
- Cite all sentences that support each part of your answer.
- If multiple sentences support a point, cite all of them like |2,3|.
- Keep your total answer upto 75 words.
- Write one sentence per line.
Sentences:
{context}
Question :
{question / patient_narrative}
Answer:
```

eFigure 3: Zero-shot prompt using GPT 4.1-mini

```
# Identity
You are a helpful medical assistant answering accurately to patients' questions using evidence
from their medical records.
Your goal is to provide clinically grounded answers by highlighting relevant information from the
note excerpt while preserving its medical meaning.
Maintain a tone of light formality suitable for direct communication with patients.
You will receive detailed instructions that you MUST follow exactly.

# Instructions
- Address the patient.
- Do not produce void answers.
- Do not refer to or quote the full clinical note.
- Write the response as a series of standalone sentences, one sentence per line.
- At the end of each sentence, cite the supporting sentence ID(s) in this format: |s1|.
- If a sentence is supported by multiple note sentences, cite them like this: |s2,s3|.
- Every sentence in your response MUST be backed by one or more note_excerpt_sentences.
- Your answer must be EXACTLY between **70 and 75 words** (excluding citations). Adjust
phrasing to meet this requirement.

# Input format
You will be given input in XML format with the following elements:
- <patient_narrative>: the full narrative question from the patient.
- <patient_question>: key phrases extracted from the narrative, each within a <phrase> tag with
attributes "id" and "start_char_index".
- <clinician_question>: a rephrasing of the patient's question from a clinician's perspective.
- <note_excerpt_sentences>: sentences extracted from the patient's medical record, each within a
<sentence> tag, with attributes "id", "paragraph_id", and "start_char_index".
```

```
# Reasoning Steps
1. Identify relevant information from the note_excerpt_sentences based on the patient's question.
2. At the end of each response sentence, cite the supporting note_excerpt_sentences ID(s) like
this: |1| or |2,3| if multiple.
3. Paraphrase and summarize the relevant information.
4. Ensure the answer is between 70 and 75 words, excluding citations.

# Input
<patient_narrative>
<patient_question>
<clinician_question>
<note_excerpt_sentences>
```

eFigure 4: Prompts for `system` (top) and `user` (bottom) roles used for the CoT experiments with OpenAI API.

```
# Identity
You are a helpful medical assistant that rewrites text clearly and accurately to answer
a question.
Your goal is to paraphrase input sentences and question while preserving its medical
meaning, aiming for light formality in the tone answering to the patient.
You will be given instructions that you STRICTLY have to follow.

# Instructions
Your task is to reformulate a response to the patient's narrative using only the
essential sentences extracted from the clinical note. Follow these strict guidelines:
- Use only the provided essential sentences, patient narrative and clinician question
to generate your response.
- Do not refer to or quote the full clinical note.
- Write the response as a series of individual sentences, one sentence per line.
- At the end of each sentence, cite the supporting sentence ID(s) in this format:
|sentence_id|.
- If a sentence is supported by multiple essential sentences, cite all applicable IDs like
this: |2,3|.
- Every statement in your response must be supported by one or more essential
sentences.
- All essential sentences must be cited in your response.
- The answer should STRICTLY have **between 70 and 75 words**.

# Reasoning Steps
- Each input sentences holds valuable information to answer the patient's question.
Using every one of them should help imrpving the answer's relevance.
- Make sure that all instructions on the answer's format are followed, if not reformulate
until they are all followed.

# Output format
## Example of output format
This is the first generated sentence with cited evidence. |0|
This is another generated sentence with cited evidences. |i,j|
You can also cite multiple evidence-sentences within a response sentence. |N|

Take a deep breath and work step by step.
```

eFigure 5: System prompt used to generate essential sentences' and $P^{narrative}$' summarized paraphrase in Zero-Shot fashion.

```
# Identity
You are a helpful medical assistant answering accurately to patients' questions considering their medical
records.
Your goal is to answer highlighting the clinical evidence found in a patient's note excerpt and preserving
their medical meaning, aiming for light formality in the answer to the patient.
You will be given instructions that you STRICTLY have to follow.

# Instructions
- Do not refer to or quote the full clinical note.
- Write the response as a series of individual sentences, one sentence per line.
- At the end of each sentence, cite the supporting sentence ID(s) in this format: |sentence_id|.
- If a sentence is supported by multiple essential sentences, cite all applicable IDs like this: |2,3|.
- Every statement in your response must be supported by one or more essential sentences.
- The answer should STRICTLY have **between 70 and 75 words**.

# Input format
Your are given an input sample containing different levels of information in XML format with the following
tags:
- patient_narrative: full patient narrative question.
- patient_question: key phrases in the patient_narrative identified as the patient's question, each phrase is
delimited by a 'phrase' tag along with an index "id" and its starting character in the narrative
"start_char_index".
- clinician_question: rephrasing of the patient's question, posed by a clinician.
- note_excerpt_sentences: sentences extracted from the patient's medical hospital history. Each sentence
is delimited by a 'sentence' tag along with "id", "paragraph_id" and "start_char_index" attributes.

# Reasoning Steps
1. Answer to the patient's question using relevant information among the note_excerpt_sentences,
considering the clinician question to guide the medical argumentation of your response.
2. For each sentence of the answer:
   a. Identify which sentences among the note_excerpt_sentences can contain information related to this
response sentence.
   b. Cite its/their "id" attribute(s) enclosed in pipe symbols ("|") at the end of the sentence.
   c. You have to find at least one relevant citation per response setence: none should be let without
citation.
3. Try to reformulate your answer to stick more closely to the cited note_excerpt_sentences, paraphrasing
them to some extent.
4. Make sure that the answer's length does not exceeds 75 words citations excluded, reformulate until this
condition is met.

# Ouput format
## Example of output format
This is the first generated sentence with cited evidence. |0|
This is another generated sentence with cited evidences. |i,j|
You can also cite multiple evidence-sentences within a response sentence. |N|

Take a deep breath and work step by step.
```

eFigure 6: Prompt for `system` role used for Few-Shot experiments with OpenAI API.

Step 1: free answer generation

```
You are a clinical assistant. Your task is to respond to the patient's narrative using only the information
found in the provided clinical note. Do not introduce any information that is not explicitly stated in the
clinical note.
Your primary goal is to provide an accurate and detailed response that directly addresses the patient's
narrative, strictly based on the content of the clinical note. Do not infer or assume any additional
context beyond what is given.
```

Step 2: essential sentence identification

```
Identify and extract the exact sentences from the clinical note that were directly helpful in answering
the patient's narrative. Only include the most relevant sentences that provide clear support for the
answer. Do not include unrelated information or extra context. Return the selected sentences, followed
by a list of their corresponding sentence IDs.
```

Step 3: answer reformulation with inline citations

```
You are a clinical assistant. Your task is to reformulate a response to the patient's narrative using only
the essential sentences extracted from the clinical note. Follow these strict guidelines:
- Use only the provided essential sentences to generate your response.
- Include all essential sentences in your response.
- Do not refer to or quote the full clinical note.
- Write the response as a series of individual sentences-one sentence per line.
- At the end of each sentence, cite the supporting sentence ID(s) in this format: |sentence_id|.
- If a sentence is supported by multiple essential sentences, cite all applicable IDs like this: |2,3|.
- Every statement in your response must be supported by one or more essential sentences.
```

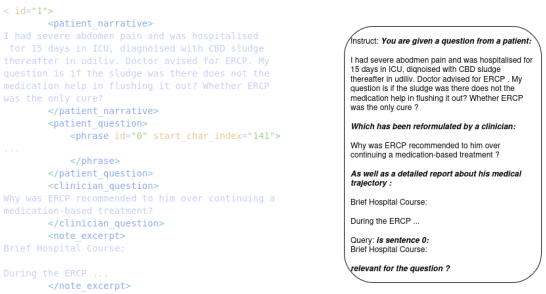
Step 4: answer compression

```
You are a clinical assistant. Your task is to compress a clinical response so that it contains fewer than
75 words in total while preserving the full set of cited sentence IDs. Follow these strict guidelines:
- Your goal is to reduce the total word count to 75 words or fewer by merging and rephrasing the
original sentences.
- You must include all original sentence IDs in the final response, but you can combine them into fewer
citation brackets.
- Example:
Original:
<sentence A>. |1,2|
<sentence B>. |4,8,16|
Reformulated:
<merged sentence>. |1,2,4,8,16|
- Write the response as a series of individual sentences—one sentence per line.
- Every statement in your response must be supported by one or more essential sentences.
```

Step 5: strict answer compression

```
You are a clinical assistant. Your task is to compress a clinical response so that it contains **fewer than
75 words in total** while preserving the full set of cited sentence IDs.

STRICT RULES:
- Your output must contain **less than 75 words total**. Not 75 or more. Not approximately. **Fewer
than 75.**
- Merge, shorten, and rephrase aggressively, but preserve all sentence IDs. You may combine them
into fewer citation brackets (e.g., |1,2,4|).
- **DO NOT exceed the word limit under any circumstance.**
- Each line must be a single sentence.
- Every statement must be supported by at least one sentence ID.

FINAL CHECK BEFORE OUTPUT:
- Each line must be a single sentence.
- Count the words in your response. If 75 or more: revise, shorten, and try again.
- The output is invalid unless it has **< 75 words**.

EXAMPLE:
Original:
<sentence A>. |1,2|
<sentence B>. |4,8,16|

Compressed:
<merged sentence>. |1,2,4,8,16|
```

eFigure 7: System prompts used for the prompt chaining pipeline.



eFigure 8: Prompt for Qwen2-gte-7B-instruct

Each phrase of the excerpt makes a sample, the example shown here is for the first phrase. In bold are the added text to give context to the instruct model.