Follow-up Question Generation For Enhanced Patient-Provider Conversations

Joseph Gatto[1], Parker Seegmiller[1], Timothy Burdick[2,4],
Inas S. Khayal[1,4], Sarah DeLozier[3], Sarah M. Preum[1]
[1] Department of Computer Science, Dartmouth College
[2] Department of Community and Family Medicine, Dartmouth Health
[3] Department of Medicine, Dartmouth Health
[4] The Dartmouth Institute, Dartmouth College
{joseph.m.gatto.gr, sarah.masud.preum}@dartmouth.edu

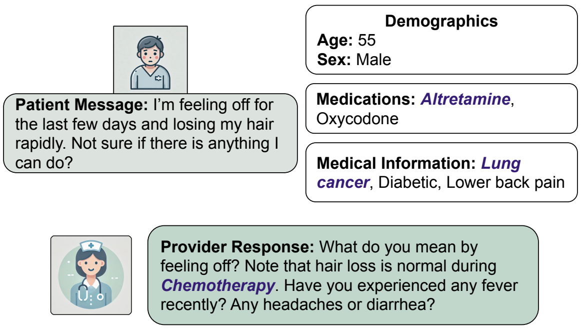
Abstract

Follow-up question generation is an essential feature of dialogue systems as it can reduce conversational ambiguity and enhance modeling complex interactions. Conversational contexts often pose core NLP challenges such as (i) extracting relevant information buried in fragmented data sources, and (ii) modeling parallel thought processes. These two challenges occur frequently in medical dialogue as a doctor asks questions based not only on patient utterances but also their prior EHR data and current diagnostic hypotheses. Asking medical questions in *asynchronous conversations* compounds these issues as doctors can only rely on static EHR information to motivate follow-up questions.

To address these challenges, we introduce **FollowupQ**, a novel framework for enhancing asynchronous medical conversation. FollowupQ is a multi-agent framework that processes patient messages and EHR data to generate personalized follow-up questions, clarifying patient-reported medical conditions. FollowupQ reduces requisite provider follow-up communications by 34%. It also improves performance by 17% and 5% on real and synthetic data, respectively. We also release the first public dataset of asynchronous medical messages with linked EHR data alongside 2,300 follow-up questions written by clinical experts for the wider NLP research community. [1]

1 Introduction

Asking relevant, useful follow-up questions while conversing fosters deeper understanding, and ensures meaningful and productive conversations. Thus neural dialogue systems may benefit from the ability to generate good follow-up questions, eliciting required information and reducing conversational ambiguity in real-time (Yi et al., 2024). Follow-up Question Generation has been studied in domains such as social media (Meng et al., 2023; Liu et al., 2025), healthcare (Li et al., 2024), document understanding (Ko et al., 2020), and conversational surveys (Ge et al., 2023).



Demographics
Age: 55
Sex: Male
Medications: Altretamine, Oxycodone
Medical Information: Lung cancer, Diabetic, Lower back pain

Patient Message: I'm feeling off for the last few days and losing my hair rapidly. Not sure if there is anything I can do?

Provider Response: What do you mean by feeling off? Note that hair loss is normal during *Chemotherapy*. Have you experienced any fever recently? Any headaches or diarrhea?

Figure 1: We demonstrate the core challenges of asynchronous follow-up question generation. Providers need to consider complex, fragmented data sources to generate *multiple* follow-up questions reflecting parallel thought processes.

Generating relevant and meaningful follow-up questions is a nontrivial NLP task, as it often involves collecting relevant information that is fragmented across multiple sources (Chen et al., 2021) or requires complex parallel thought processes (Lee et al., 2024; Cao, 2024). For example, in patient-provider communication, providers must (i) consider patient utterances while attending to information scattered throughout the patient's electronic health record (EHR) (Tu et al., 2024), and (ii) consider numerous different thought processes while formulating a patient's diagnosis (McDuff et al., 2023; Huynh et al., 2023). A similar challenge arises in customer service interactions, where support agents must (i) integrate information from a customer's message while referencing their past interactions, purchase history or account details (Xu et al., 2024; Misischia et al., 2022), and (ii) simultaneously consider multiple resolution strategies, such as troubleshooting steps, redirection to other support or refund policies (Pi et al., 2024).

[1] Paper resources can be found at https://tinyurl.com/Followup-Question-Generation

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 25222–25240
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

Previous works have studied the information seeking capacity of dialogue systems, including LLM-based systems in the context of *synchronous* medical conversation (Zeng et al., 2020; Varshney et al., 2022) — where the provider and patient engage in multiturn real-time conversation (Li et al., 2024; Wang et al., 2024; Winston et al., 2024). However, to our knowledge, there has been limited exploration of LLMs as information-seeking agents in the context of *asynchronous* medical dialogue — where two parties exchange online messages at their convenience. Asynchronous medical conversations are getting increasingly popular with telehealth services (Shaver, 2022) and increased use of online patient portals (Hansen et al., 2023), and have unique traits which demand targeted NLP solutions.

First, when one messages their doctor asynchronously, it is common for them to assume the reader of their message has significant knowledge of their personal and medical background (Gatto et al., 2024). This data characteristic requires NLP systems to rely heavily on fragmented data in the EHR to fill in missing context, as there is no real-time access to the patient for information solicitation. This is crucial as solely relying on patients for self-reported medical context can be risky (see Appendix G for an example). Second, when a provider solicits additional information from a patient in this setting, they need to generate a ***list*** of follow-up questions to rule out a broad range of possible diagnoses all at once to reduce the need for additional follow-up communication (e.g., calling the patient). This contrasts with the synchronous communication paradigm, where patients are usually asked one question at a time.

An NLP system capable of automatically generating relevant follow-up questions when patients report their medical symptoms has significant real-world benefits. It can reduce the volume of follow-up communications required from providers, thereby alleviating a key contributor to provider burnout — the burden of asynchronous messaging (Stillman, 2023; Budd, 2023). However, this task remains largely unexplored in the NLP community due to the absence of publicly available datasets and effective evaluation methods.

In this study, we solve these two problems by introducing and formalizing a new task, **Follow-up Question Generation** for Asynchronous Patient-Provider Conversations. We release **Followup-Bench**, the first public dataset of 250 semi-synthetic patient data samples, containing patient messages and EHR data. This dataset contains over **2,300 follow-up questions** composed by a team of 9 primary care providers (including doctors, nurses, and medical residents) at a regional university medical center. Our study additionally explores follow-up question generation on a set of 150 real patient messages with ground truth questions extracted from responses written by real providers during their normal workflow. We introduce a novel evaluation metric, Requested Information Match, which formalizes the task of comparing predicted vs ground-truth question sets in the context of asynchronous medical conversations using LLM-as-Judge (Zheng et al., 2024).

However, our experiments demonstrate that *off-the-shelf LLMs struggle to generate follow-up questions* that align with ground-truth questions written by real providers. We address this limitation in this setting by introducing **FollowupQ**, a multi-agent framework for personalized follow-up question generation in patient messages. FollowupQ utilizes clinical priors in an agentic divide-and-conquer strategy to alleviate the complexity of mapping multiple complex data sources to a set of follow-up questions. FollowupQ agents, developed in collaboration with a team of clinical experts, collaborate to emulate complex clinical thought processes spanning a broad range of clinical inquiry. The contributions of this work can be summarized as follows:

1. We introduce FollowupQ: A novel multi-agent framework for follow-up question generation for patient message enhancement. FollowupQ outperforms baselines by 17% and 5% on real and synthetic datasets, respectively.

2. We release the first asynchronous medical messaging dataset with both patient messages and EHR data — mimicking a real-world environment. Our open-source dataset contains 2,300 questions from real providers covering a wide variety of medical conditions.

3. We introduce a novel evaluation metric, Requested Information Match (RIM), which uses LLM-as-Judge to help quantify the potential reduction in provider workload.

2 Related Work

2.1 Follow-Up Question Generation

The importance of generating high-quality follow-up question generations has been studied throughout conversational NLP. For example, there have been a variety of studies on information-seeking systems for social media conversations (Meng et al., 2023; Liu et al., 2025), and conversational surveys (Ge et al., 2023). However, unlike FollowupQ these works focus on generating one question at a time. Asking follow-up questions in medical conversations has been explored in the related context of synchronous dialogue generation. For example, Winston et al. (2024) explores the ability of LLMs to ask patients questions towards eliciting pertinent background details about their visit. Wang et al. (2024) transform medical dialogue datasets into a format suitable for medical question generation evaluation. Li et al. (2024) transform Medical-QA datasets such as Med-QA (Jin et al., 2021) to explore the capacity of LLMs to ask follow-up questions that lead to a medical diagnosis.

None of these studies explore asynchronous conversations such as those that occur in online patient portals and telehealth services. Additionally, our work instead focuses on generation of *sets* of questions, enabling comparison to ground-truth questions from real portal message interactions. To the best of our knowledge, the only other work to explore follow-up question generation in a similar setting is Liu et al. (2024) — but at an extremely small scale (n=7 samples) and without any utilization of linked EHR data.

2.2 Multi-Agent Systems in Healthcare

Recently, LLM-based Multi-Agent systems have been shown to provide significant performance increases across a broad range of tasks and domains, including healthcare (Guo et al., 2024). Frameworks such as MedAgents (Tang et al., 2024), RareAgents (Chen et al., 2024), MDAgents (Kim et al., 2024), and TriageAgent (Lu et al., 2024) – all leverage collaborative multi-round discussion between multiple LLM agents to perform medical decision making. Our framework, FollowupQ, is inspired by prior works, as we also employ multiple agents for follow-up question generation. However, these prior systems are designed for medical decision-making, not information seeking, making them non-trivial to apply to our dataset. We further highlight the unique evaluation challenges addressed by FollowupQ in Section 3.

3 Follow-Up Question Generation

3.1 Problem Formalization

Consider a patient message T, their corresponding EHR C, and a text generator f (e.g., an LLM or LLM-based framework).

$$f(T, C) = \hat{Q} = \{\hat{q_1}, \hat{q_2}, \dots, \hat{q_n}\}$$

The goal of the text generation system f is to produce a set $\hat{Q}$ where each $\hat{q}_i \in \hat{Q}$ is a follow-up question to the patient's message. Crucially, f exists in an asynchronous environment without real-time access to the patient and must generate all pertinent follow-up questions as a list.

We specifically define a patient's EHR record as $C = \{A, H, M\}$. This includes a patient's demographics A (e.g., age and gender), medical history H (e.g., problem list and recent medical encounters), and medication list M. Each component of C is represented as a string in our framework. An example message with EHR data and provider response are shown in Appendix H.

3.2 Evaluation Strategy

In the context of the real-world use cases of asynchronous medical dialogue, the quality of the set of questions $\hat{Q} = \{\hat{q_1}, \dots, \hat{q_n}\}$ produced by f is primarily determined by *the reduction in dialogue turns required by the provider to make a diagnosis or recommendation.* To make their decision, providers must first have all the necessary information from the patient. To reduce provider workload, patient responses to questions in $\hat{Q}$ must contain at least the information requested in the ground truth question set Q. Thus, we design metrics for comparing generated questions $\hat{Q}$ against ground truth questions Q, to identify how well the information requested in Q is covered by the information requested in $\hat{Q}$.

Importantly, a generated question $\hat{q}_i \in \hat{Q}$ is considered as matching a ground truth question $q_j \in Q$ when they *request the same information* (i.e., invoke similar responses)— in addition to when they match exactly. For example, an LLM asking "Do you have a cough or fever?" will elicit a similar response to the doctor's question "Have you been coughing?" and thus should be considered a match. We employ LLM-as-judge (Zheng et al., 2024) to do this semantic matching. We present the validation of the LLM-as-judge process in Section 5.3.

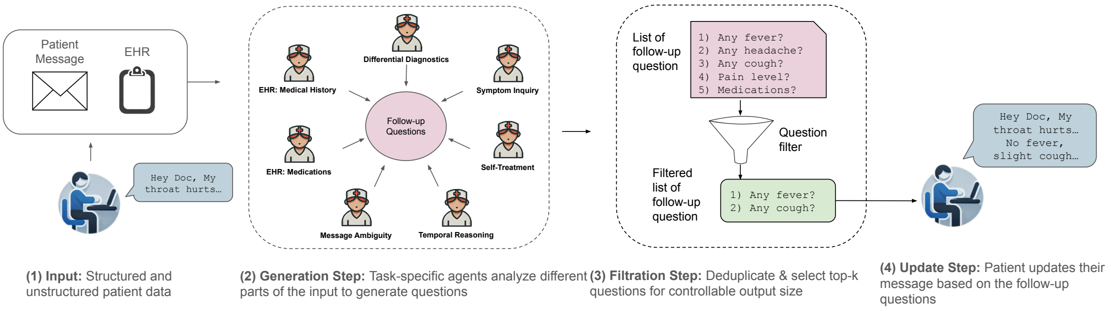

Patient Message
EHR
Hey Doc, My throat hurts…
Differential Diagnostics
EHR: Medical History
Symptom Inquiry
Follow-up Questions
EHR: Medications
Self-Treatment
Message Ambiguity
Temporal Reasoning
List of follow-up question
1) Any fever?
2) Any headache?
3) Any cough?
4) Pain level?
5) Medications?
Question filter
Filtered list of follow-up question
1) Any fever?
2) Any cough?
Hey Doc, My throat hurts… No fever, slight cough…
(1) Input: Structured and unstructured patient data
(2) Generation Step: Task-specific agents analyze different parts of the input to generate questions
(3) Filtration Step: Deduplicate & select top-k questions for controllable output size
(4) Update Step: Patient updates their message based on the follow-up questions

Figure 2: FollowupQ works by taking a patient message and a subset of their EHR and employing multiple LLM agents to explore diverse clinical thought processes — producing a pool of follow-up questions from different perspectives. If desired, FollowupQ can then filter the output to a controllable question set size.

First, we introduce the **Requested Information Match (RIM)** metric.

$$RIM(Q, \hat{Q}) = \frac{|Q \cap \hat{Q}|}{|Q|} \tag{1}$$

RIM measures the sample-wise percentage of provider-generated questions that are also generated by system f. RIM is a task-specific case of the Tversky Index (Tversky, 1977). RIM is notably distinct from other set comparison metrics, like Jaccard Similarity, as we do not penalize f based on the size of the set of generated questions, $|\hat{Q}|$. This design choice is informed by our prior work with providers, as *eliciting extra information helps providers*. For example, just because a provider only asked 3 questions to a patient, it does not mean there are no other useful inquiries to generate. Different providers might generate different list of questions based on their experience, preferences, and current mental models, resulting in subjectivity in the ground truth. Thus, requesting additional information helps a provider who may have forgotten to consider a certain outlying issue the patient may be facing, and address the subjectivity of their thought process.

Subsequently, we note that the first step towards solving asynchronous follow-up question generation is to optimize for coverage of ground truth questions (i.e., maximize RIM), with the secondary objective being controlling the size of the list of generated questions, $|\hat{Q}|$. In Section 4.3, we introduce a way to control $|\hat{Q}|$, i.e. the number of questions presented to a patient. However, as we discuss in Section 6, RIM maximization is a difficult challenge for LLMs with unconstrained $|\hat{Q}|$. Thus our core metric is the average RIM score across all samples in our dataset.

A RIM score of 1.0 denotes the case where system f requested all information actually requested by the patient's real doctor. For each output with an RIM score of 1.0, we argue that f has successfully reduced the number of clarification requests a provider needs to send by 1, assuming the patient responds to all questions. Conversely, while RIM scores below 1.0 are still suggestive of message improvement, they may not suggest any reduction in outgoing messages to be sent by a provider.

Hence, we introduce a second metric **Message Reduction % (MR%)** which measures the percentage of samples where the RIM score = 1.0. Models with a higher MR% have more real-world impact as they would lead to greater workload reductions. In the remainder of this section, we present the FollowupQ framework.

4 FollowupQ Framework for Generating Follow-Up Questions

4.1 Design Motivation

Our FollowupQ framework is summarized in Figure 2. Our methods were developed over a 12-month period in collaboration with a team of clinical experts who frequently communicate with patients in primary care. The experts include primary care physicians, medical residents, registered nurses, and triage nurses. Through a series of workshops and interviews with our team of experts, we discovered the following three thought processes essential to the asynchronous followup question generation process, which we embed into FollowupQ. (i) EHR Reasoning: where providers obtain contextual knowledge from fragmented EHR data to guide their inquiry. (ii) Differential Diagnostics: Where providers develop a mental list of potential diagnoses or explanations that describe the patient's symptoms — guiding their question formulation. (iii) Message Clarifications: Where providers ask a series of questions to fill in gaps in the patient's reported symptoms.

4.2 Followup Question Generation

Recall from Section 3.1 that T and C denote a patient message and corresponding EHR record, respectively. The objective of the generation step is to create a set of questions $f(T, C) = \hat{Q}$ where $\hat{Q} = \{\hat{q_1} \dots \hat{q_n}\}$. We build $\hat{Q}$ by taking the union of question sets generated by m inquiry agents, i.e. $\hat{Q} = \hat{Q}_{agent_1} \cdots \cup \dots \hat{Q}_{agent_m}$. This strategy allows us to use multiple agents to generate question sets corresponding to the parallel clinical thought processes outlined in Section 4.1. $\hat{Q}$ is thus constructed using three high-level agents as described below: EHR reasoning agents, differential diagnostic agents, and message clarification agents.

EHR Reasoning Agents: EHR data is complex as the information related to a patient's current inquiry can be implicit or fragmented across different data tables, and fields. FollowupQ uses two EHR-specific reasoning agents to mitigate the challenges of generating questions from fragmented data sources. Specifically, for a given EHR record $C = \{A, H, M\}$ (see Section 3.1 for the definition of EHR elements), we define a medical history reasoning agent and a medication list reasoning agent. Each agent first extracts relevant pieces of EHR information concerning the patient's current inquiry T. This is critical as a patient's medical history and medication list can contain a lot of information that is not relevant to their current message.

$$\begin{aligned} I_{hist} &= f(A, H, P_{extract_H}, T) \\ I_{med} &= f(M, P_{extract_M}, T) \end{aligned} \tag{2}$$

Where I_{hist} and I_{med} are the elements from the patient's medical history and medication list most relevant to the patient's message. $P_{extract_H}$ and $P_{extract_M}$ are topic-specific information extraction prompts for history and medications. EHR reasoning agents then generate EHR-specific followup questions $\hat{Q}_{EHR} = \hat{Q}_{hist} \cup \hat{Q}_{med}$ as follows. Here, P_{hist} and P_{med} guide f on generating questions related to I_{hist} and I_{med}, respectively.

$$\begin{aligned} \hat{Q}_{hist} &= f(T, I_{hist}, P_{hist}, k) \\ \hat{Q}_{med} &= f(T, I_{med}, P_{med}, k) \end{aligned} \tag{3}$$

Differential Diagnostic Agents: Providers who read and respond to patient messages often mentally perform a differential diagnosis before coming up with the follow-up questions (Ferri, 2010). Specifically, providers will (i) decide what could be wrong with the patient and (ii) ask questions to rule out various diagnostic hypotheses. Inspired by this mental framework, FollowupQ uses differential diagnostic agents to generate a set of possible patient diagnoses D_{diff}, then generate follow-up questions based on these potential diagnoses, i.e. follow-up questions to rule out each diagnosis $d_i \in D_{diff}$.

Differential diagnostic agents first compute a pseudo-differential diagnosis by identifying the k best and worst-case diagnoses for a given patient, using prompts P_{best} and P_{worst} respectively.

$$D_{diff} = f(T, P_{best}, k) \cup f(T, P_{worst}, k) \tag{4}$$

Thus, D_{diff} is the union of the potential diagnoses produced by f under the assumptions of the best and worst case scenarios. This strategy is motivated by the following observation we made in our preliminary work: providers cast a wide net for gathering relevant information. Next, differential diagnostic agents iteratively build the question set $\hat{Q}_{D_{diff}} = \hat{Q}_{d_1} \cdots \cup \dots \hat{Q}_{d_{|D_{diff}|}}$, which has targeted questions to rule out each diagnosis $d_i \in D_{diff}$. For a given possible diagnosis d_i we compute $\hat{Q}_{d_i}$ as follows.

$$\hat{Q}_{d_i} = f(T, d_i, P_{rule-out}, k) \tag{5}$$

Where f outputs a question set of size k, using the prompt $P_{rule-out}$ to guide questions to see if the patient is suffering from potential diagnosis d_i. By taking the union of question sets about each potential diagnosis, differential diagnostic agents produce the set of questions $\hat{Q}_{D_{diff}}$.

Message Clarification Agents: FollowupQ's third type of agent is a set of message clarification agents to increase the clarity of different aspects of the patient's message. **Symptom inquiry** agents extract symptoms from the message and ask clarifying questions about each symptom as needed (e.g., location of abdominal pain). **Self-treatment** agents ask patients to elaborate on how they are treating their symptoms (e.g. patients may self-treat with over-the-counter pain medications or herbal supplements). **Temporal reasoning** agents generate questions to increase clarity in the timeline of presented symptoms, (e.g., duration and frequency of pain). **Message ambiguity** agents target reducing overall ambiguity of the message (e.g. "tell me more about what you mean by you are feeling off."). For each clarification agent $clar_i$,

$$\hat{Q}_{clar_i} = f(T, P_{clar_i}, k) \tag{6}$$

Where P_{clar_i} is a prompt specific to clarification agent $clar_i$. Taking the union over all clarification agents, we end up with question set $\hat{Q}_{clar}$.

Our final question pool $\hat{Q}_p$ consists of questions generated by the differential diagnostic, medical chart, and clarification agents.

$$\hat{Q}_p = \{\hat{Q}_{D_{diff}}, \hat{Q}_{EHR}, \hat{Q}_{clar}\} \tag{7}$$

System Parameters: The value k, which controls the number of questions generated, is specific to each agent and described in detail in Appendix B. In this work, we opted to control the *exact* number of questions generated by each agent to retain granular control of the output size. However, future works may explore more flexible variations, such as generating up-to-k questions or using no constraints.

4.3 Question Filtration

The generation step produces $|\hat{Q}_p|$ questions, which based on hyperparameter choices in the FollowupQ framework (e.g. k) may produce a $\hat{Q}_p$ too large for a patient to answer. So, as an ablation of our core experiments, we investigate ways to systematically reduce the size of $\hat{Q}_p$ to a target size k. Specifically, our framework performs question deduplication and top-k question selection. Our question de-duplication framework uses LLMs to filter non-unique inquiries from $|\hat{Q}_p|$. This is a crucial step as different agents may request the same information in the context of different thought processes. Our top-k selection agent takes the de-duplicated question list and selects the k most important questions to ask the patient. Details of the question filtration method are presented in Appendix C.

5 Experimental Setup

5.1 Dataset Details

We evaluate our methods on a novel benchmark **FollowupBench** (FB) which contains two asynchronous Portal Message datasets: **FB-Real** and **FB-Synth**, which are described in Table 1.

Dataset	# of Messages	Total/Mean # of Questions	Mean # of Sentences
FB-Real	150	514 / 3.4	5.3
FB-Synth	250	2,336 / 9.3	6.5

Table 1: Dataset statistics for FollowupBench (FB). FB-Real has fewer questions on average as they were composed in a live, time-constrained work environment.

FB-Real consists of real messages and EHR records sent from adult patients to their providers between January 2020 and June 2024 at a large university medical center in the United States. From a corpus of over 500k messages, we perform a multi-step filtering process, including extensive human evaluation, to select messages that ensure each patient in our dataset is symptomatic and received a provider response containing follow-up questions. Human reviewers ensure that all clarification questions included in the ground truth are in scope for an AI system to generate (i.e. are grounded to the message or chart and not to prior in-person patient-provider interaction). Additionally, human reviewers ensure all questions are specific to symptom clarifications (i.e. we do not aim to generate logistical questions related to scheduling, insurance coverage, or medication refills). All extracted ground truth questions are broken down into single-topic questions to promote granular evaluation of NLP systems (e.g. *"Do you have any fever or cough?"* is converted to the following two questions: "1. Do you have any fever? 2. Do you have any cough?"). As our human review process is expensive and extremely time-consuming, we limit FB-Real to 150 unique patient messages and the corresponding EHR data of those 150 patients. We provide extensive details of our data curation process in Appendix E. As FB-Real contains protected health information, it can not be shared publicly.

FB-Synth is a semi-synthetic dataset consisting of 250 (medical chart, patient message) pairs with over 2,300 follow-up questions written by a team of 9 physicians, nurses, and medical residents at a large university medical center. This patient data was created by first sampling a random, de-identified medical chart from the corpus of patients used to create FB-Real. Then, we create a synthetic message using a grounded message generation technique inspired by Gatto et al. (2024). Interestingly, FB-Synth has a higher mean number of questions per sample compared to FB-Real (9.3 vs 3.4). This is likely due to various factors, including (a) research subjects often act differently under observation, known commonly as the Hawthorne Effect (McCambridge et al., 2014) (b) providers may request more information when they have more time to reflect on a given patient's needs. The latter point is suggestive of FB-Synth potentially being closer to the target set of questions an AI system should strive to generate. We make this dataset available to the research community. Please see Appendix E for additional FB-Synth details.

5.2 Baseline Methods & Models

We compare FollowupQ to the following baseline prompting approaches:
Unbounded-Generation: We provide an LLM with both patient message and EHR data, and prompt it to write as many questions as necessary to clarify ambiguity and seek missing details. We explore unbounded generation in the context of both 0-shot and few-shot prompting.
k-Question-Generation: The same prompt as unbounded-generation, but with specific guidance to output k questions. This allows us to explore performance correlation with the scale of the generated set. For large values of k, this experiment explores the intrinsic limits of LLMs to reach long-tail questions. We explore $k = 40$ in this study as the mean number of questions output by our FollowupQ framework before filtration is ≈ 35.
Long-Thought Generation: Given the recent success of long-thought models on complex reasoning tasks (DeepSeek-AI, 2025) we explore the performance of models that generate follow-up questions after significant Chain-of-Thought output. This baseline is explicitly instructed to generate as many questions as it sees fit.

Due to the sensitive nature of FB-Real, we run our experiments on a secure computing cluster with no access to the internet and a single NVIDIA A40 GPU. We thus perform all experiments using 4-bit quantized versions of the following models due to computational restraints: (i) Llama3-8b (et. al, 2024a), (ii) Llama3-8b Aloe, a Llama3 variant trained on healthcare data, (Gururajan et al., 2024), (iii) Qwen2.5-32b-instruct (Qwen et al., 2025) — the largest model we have available on our computing platform, and (iv) Qwen2.5-32b distilation of DeepSeek R1, for long-thought baseline. Appendix B contains additional modeling details, and Appendix F contains relevant prompts.

5.3 LLM As Judge

In Section 3.2 we formally define the metrics employed in our evaluation. Each metric depends on determining if a provider and LLM-generated question are requesting the same information. To detect matching questions, we employ a fine-tuned PHI-4-14b (et. al, 2024b) based LLM-as-Judge (Zheng et al., 2024) framework to do a pairwise comparison between all true and generated follow-up questions. We use a test set (n=100 question pairs) hand-labeled by a family medicine physician with over twenty years of experience. We find that our judge model can detect matching question pairs that elicit the same information with a macro F1-score of 0.87. Appendix D presents additional details of our Judge model, including fine-tuning procedure, example matches, prompts, and additional LLM performance on our test set.

	Llama3-8b	Llama3-8b-Aloe	Qwen-32b
Zero-Shot-U	0.35 / 11	0.35 / 17	0.35 / 10
Few-Shot-U	0.29 / 9	0.33 / 19	0.36 / 10
Zero-Shot-40	0.40 / 30	0.40 / 30	0.41 / 35
Few-Shot-40	0.37 / 35	0.36 / 35	0.45 / 37
Long-Thought	-	-	0.31 / 10
Follow-up Q	**0.62 / 36**	0.64 / 58	0.54 / 35

Table 2: The (Mean RIM / Mean Number of Questions Asked) for each experiment on FB-Real. We find that Llama3-8b achieves the best performance when balancing number of questions generated vs RIM.

	Llama3-8b	Llama3-8b-Aloe	Qwen-32b
Zero-Shot-U	0.30 / 11	0.30 / 17	0.30 / 11
Few-Shot-U	0.24 / 8	0.27 / 18	0.27 / 10
Zero-Shot-40	0.34 / 30	0.35 / 31	0.43 / 36
Few-Shot-40	0.34 / 35	0.35 / 35	0.41 / 38
Long-Thought	-	-	0.27 / 11
Followup-Q	**0.48 / 35**	0.51 / 59	0.44 / 34

Table 3: The (Mean RIM / Mean Number of Questions Asked) for each experiment on FB-Synth. We find FB-Synth to be slightly more challenging than FB-Real but with Llama3-8b still achieving superior performance.

6 Results

FollowupQ Significantly Outperforms Baseline Methods on Followup Question Generation: In Table 2 we show that FollowupQ (Llama3-8b) achieves a mean RIM score of 0.62 on FB-Real while generating 36 questions. This is a 22-point increase in RIM compared to comparable zero and few-shot baselines using Llama3-8b — showing the effectiveness of FollowupQ. Although Llama3-8b-Aloe achieves an even higher score of 0.64, it also generates additional questions. However, we note that Llama3-8b-Aloe uniquely struggled to follow instructions in ways that support our ability to control output set size. This can result from it's training data (Luo et al., 2025). We thus consider FollowupQ (Llama3-8b) as our top-performing model on the FB-Real dataset.

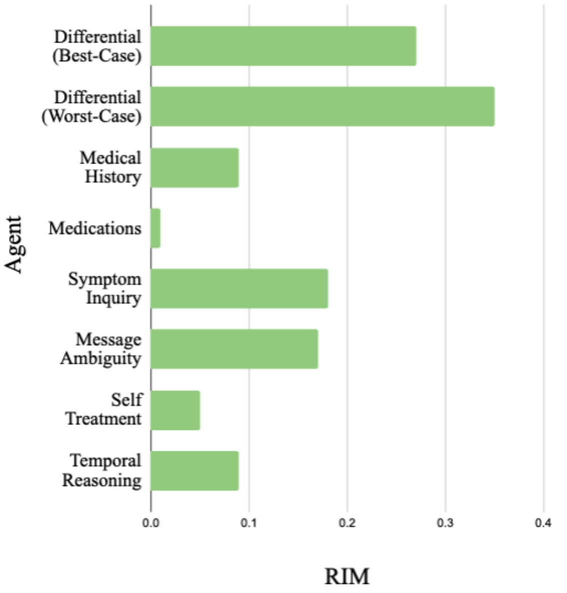


Figure 3: Per-Agent Performance on FB-Real from FollowupQ (Llama3-8b). We find that most of the performance comes from agents trying to rule-out the worst case scenario for a patient.

Interestingly, we find that while baseline solutions do see an increase in performance when encouraged to generate more questions (i.e. Unbounded → 40-question generation), they still struggle to match the performance of FollowupQ. In other words, while FB-Real averages 3.4 follow-up questions per sample, our results demonstrate that even encouraging the LLM to generate over 10x the number of questions written by a real doctor does not solve this problem — motivating a more intricate solution. In summary, FollowupQ's ability to generate diverse questions provides significant improvements over baseline LLMs in both zero and few-shot settings.

On the FB-Synth dataset, we find that FollowupQ with Llama3-8b provides a 5-point improvement over the closest baseline. When employing Qwen-32b, we find that the increase in performance is more subtle, but that FollowupQ still outperforms zero and few-shot baselines while generating fewer questions on average.

FollowupQ Reduces The Number of Information-seeking Messages Providers Need to Send by 34%: In Table 4 we compare each model's ability to achieve a RIM score of 1.0 on FB-Real — indicating that all provider follow-up questions were captured by the model. We find that FollowupQ provides a significant 19% increase over the nearest baseline. This suggests that when patients respond to all FollowupQ questions, providers need to request additional information in 34% fewer symptomatic inquiries, effectively reducing their workload. FollowupQ's provider-centric question-generation strategy produces a broader range of question types, improving performance over baseline LLMs, which often overlook less common concerns in a provider's differential diagnosis.

Method	Message Reduction %
Zero-Shot-40	0.15
Few-Shot-40	0.13
Followup-Q	0.34

Table 4: Number of samples fully matched by each method from our Llama3-8b experiments. This table thus demonstrates potential for workload reduction, with Followup-Q having the potential to reduce need for sending symptom clarification messages by up to 34%.

FollowupQ Surfaces Patterns in Providers' Thought Process: In Figure 3 we show the RIM score of each specialist agent in FollowupQ on FB-Real. We find that most performance comes from the *Differential (worst-case)* agent, displaying how follow-up questions are often concerned with ruling out worst-case scenarios as they would require urgent care. However, a non-trivial amount of inquiries come from other specialist agents focusing on medications, timeline clarification, and ambiguity clarification. Notably $\approx 10\%$ of performance came from inquiries concerning the patient's EHR, highlighting the importance of EHR data in personalized follow-up question generation. Thus, Figure 3 not only demonstrates the interpretability of our framework but also surfaces the underlying thought processes of the providers.

Performance After Filtration Step In addition to our primary evaluations, we explore question filtration on FB-Real using outputs from the best-performing LLM, Llama-3-8b. In this experiment, we filter $|\hat{Q}_p|$ to a size of 10. The choice of k=10 is motivated by both conversations with clinical experts and the mean number of questions asked in our synthetic dataset (see Table 1). We find that our top-performing model, on average, generates 36 questions. After de-duplication, we go from 36 → 22 questions per sample with a mean RIM score of 0.62 → 0.57.

Once we perform Top-10 filtering on the de-duplicated set, the RIM score drops from 0.57 →

0.42. This is likely not due to the quality of the questions, but rather our inability to model the subjective preferences of the provider who wrote the ground truth responses, as dialogue data is highly prone to response variability. Our result still demonstrates the highest 10-question performance across all experiments. However, as Tables 2 and 3 demonstrate, simply solving the *recall problem* (i.e., covering all ground-truth doctor questions without limiting generated question set size) is extremely difficult. Thus, we consider deeper investigation of question filtration to future works including developing provider-specific agents to personalize the question selection process.

7 Discussion

In this study, we introduce a novel task and dataset for advancing information-seeking LLM agents in the context of asynchronous medical conversations in primary care. We find that off-the-shelf LLMs struggle to generate question sets written by real providers. This result motivates our FollowupQ framework, which takes a multi-agent approach to question generation targeting a diverse and domain-specific line of medical inquiry. Our results demonstrate that FollowupQ significantly outperforms baseline LLMs and has the potential to reduce the asynchronous messaging workload of healthcare providers.

8 Limitations

This work is limited in that we are only able to explore a limited number of LLMs due to the computational restrictions of our secure computing environment. Additionally, future works will explore the search for optimal FollowupQ hyperparameters leveraging insights from our preliminary results. This was considered out of scope for the current submission.

Both a strength and weakness of our dataset is that it is sourced from a single hospital in a rural community. This is a strength as rural populations may be underrepresented in medical NLP datasets. However, this is a weakness as our patient population may be biased towards certain sub-populations and our providers may be biased towards asking questions common to patients they frequently care for at this single hospital. Another limitation of our dataset is that it may be the case that different doctors will respond to the same patient message with different sets of questions. This phenomenon is the product of a variety of factors including (i) where and when was the doctor trained? (ii) what past experiences have the doctor had? (iii) how busy is the doctor at the time of reading the message? However, we note this is an unavoidable characteristic common to any medical dialogue dataset.

Additionally, while our agentic framework shows strong performance, future iterations may explore additional model training to improve performance of certain agents. For example, recent datasets studying Differential Diagnosis (DDX) (Fansi Tchango et al., 2022) may prove useful to our DDX agents. However, this was considered out of scope for this submission.

Finally, we highlight that we do not explore our methods in the context of synchronous medical dialogue. However, future works may be inspired by the FollowupQ framework to generate questions for synchronous dialogue.

9 Ethical Considerations

The study protocol was approved by the Institutional Review Boards at the submitting authors' institution (STUDY02001534). The FB-Real dataset contains sensitive patient information that cannot be publicly shared. This dataset was only utilized in a secure computing environment and handled by researchers who have completed HIPAA training.

While there are no risks to any human subjects as a result of this study, we highlight that real-world utilization of our framework might have potential risks to patients if certain precautions are not taken such as employing rail-guarded LLMs to prevent the generation of harmful, offensive, and misleading content. Additionally, any medical NLP system utilizing patient data must consider patient data privacy policies and protect user data from being stored or utilized inappropriately.

Acknowledgments

This work was partially supported by a pilot grant from the Hitchcock Foundation and the NSF Research Traineeship, Transformative Research, and Graduate Education in Sensor Science, Technology and Innovation grant (DGE- 2125733).

References

Jeffrey Budd. 2023. Burnout related to electronic health record use in primary care. *Journal of primary care & community health*, 14:21501319231166921.

Lang Cao. 2024. Diaggpt: An llm-based and multi-agent dialogue system with automatic topic management for flexible task-oriented dialogue.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.

Xuanzhong Chen, Ye Jin, Xiaohao Mao, Lun Wang, Shuyang Zhang, and Ting Chen. 2024. Rareagents: Autonomous multi-disciplinary team for rare disease diagnosis and treatment.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Aaron Grattafiori et. al. 2024a. The llama 3 herd of models.

Marah Abdin et. al. 2024b. Phi-4 technical report.

Arsene Fansi Tchango, Rishab Goel, Zhi Wen, Julien Martel, and Joumana Ghosn. 2022. Ddxplus: A new dataset for automatic medical diagnosis. *Advances in neural information processing systems*, 35:31306–31318.

Fred F Ferri. 2010. *Ferri's Differential Diagnosis E-Book: A Practical Guide to the Differential Diagnosis of Symptoms, Signs, and Clinical Disorders*. Elsevier Health Sciences.

Joseph Gatto, Parker Seegmiller, Timothy E. Burdick, and Sarah Masud Preum. 2024. In-context learning for preserving patient privacy: A framework for synthesizing realistic patient portal messages.

Yubin Ge, Ziang Xiao, Jana Diesner, Heng Ji, Karrie Karahalios, and Hari Sundaram. 2023. What should I ask: A knowledge-driven approach for follow-up questions generation in conversational surveys. In *Proceedings of the 37th Pacific Asia Conference on Language, Information and Computation*, pages 113–124, Hong Kong, China. Association for Computational Linguistics.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges.

Ashwin Kumar Gururajan, Enrique Lopez-Cuena, Jordi Bayarri-Planas, Adrian Tormos, Daniel Hinjos, Pablo Bernabeu-Perez, Anna Arias-Duart, Pablo Agustin Martin-Torres, Lucia Urcelay-Ganzabal, Marta Gonzalez-Mallo, Sergio Alvarez-Napagao, Eduard Ayguadé-Parra, and Ulises Cortés Dario Garcia-Gasulla. 2024. Aloe: A family of fine-tuned open healthcare llms.

Michael A Hansen, Rebecca Chen, Jacqueline Hirth, James Langabeer, and Roger Zoorob. 2023. Impact of covid-19 lockdown on patient-provider electronic communications. *Journal of Telemedicine and Telecare*, page 1357633X221146810.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Ky Huynh, Juan P Brito, Carma L Bylund, Larry J Prokop, and Naykky Singh Ospina. 2023. Understanding diagnostic conversations in clinical practice: A systematic review. *Patient Education and Counseling*, page 107949.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Yubin Kim, Chanwoo Park, Hyewon Jeong, Yik Siu Chan, Xuhai Xu, Daniel McDuff, Hyeonhoon Lee, Marzyeh Ghassemi, Cynthia Breazeal, and Hae Won Park. 2024. Mdagents: An adaptive collaboration of llms for medical decision-making.

Wei-Jen Ko, Te-yuan Chen, Yiyan Huang, Greg Durrett, and Junyi Jessy Li. 2020. Inquisitive question generation for high level text comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6544–6555, Online. Association for Computational Linguistics.

Younghun Lee, Dan Goldwasser, and Laura Schwab Reese. 2024. Towards understanding counseling conversations: Domain knowledge and large language models. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2032–2047, St. Julian's, Malta. Association for Computational Linguistics.

Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. 2024. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning.

Jianyu Liu, Yi Huang, Sheng Bi, Junlan Feng, and Guilin Qi. 2025. From superficial to deep: Integrating external knowledge for follow-up question generation using knowledge graph and LLM. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 828–840, Abu Dhabi, UAE. Association for Computational Linguistics.

Siru Liu, Aileen P Wright, Allison B Mccoy, Sean S Huang, Julian Z Genkins, Josh F Peterson, Yaa A Kumah-Crystal, William Martinez, Babatunde Carew, Dara Mize, Bryan Steitz, and Adam Wright. 2024. Using large language model to guide patients to create efficient and comprehensive clinical care message. *Journal of the American Medical Informatics Association*, 31(8):1665–1670.

Meng Lu, Brandon Ho, Dennis Ren, and Xuan Wang. 2024. TriageAgent: Towards better multi-agents collaborations for large language model-based clinical triage. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5747–5764, Miami, Florida, USA. Association for Computational Linguistics.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2025. An empirical study of catastrophic forgetting in large language models during continual fine-tuning.

Jim McCambridge, John Witton, and Diana R Elbourne. 2014. Systematic review of the hawthorne effect: new concepts are needed to study research participation effects. *Journal of clinical epidemiology*, 67(3):267–277.

Daniel McDuff, Mike Schaekermann, Tao Tu, Anil Palepu, Amy Wang, Jake Garrison, Karan Singhal, Yash Sharma, Shekoofeh Azizi, Kavita Kulkarni, Le Hou, Yong Cheng, Yun Liu, S Sara Mahdavi, Sushant Prakash, Anupam Pathak, Christopher Semturs, Shwetak Patel, Dale R Webster, Ewa Dominowska, Juraj Gottweis, Joelle Barral, Katherine Chou, Greg S Corrado, Yossi Matias, Jake Sunshine, Alan Karthikesalingam, and Vivek Natarajan. 2023. Towards accurate differential diagnosis with large language models.

Daniel McDuff, Mike Schaekermann, Tao Tu, Anil Palepu, Amy Wang, Jake Garrison, Karan Singhal, Yash Sharma, Shekoofeh Azizi, Kavita Kulkarni, et al. 2025. Towards accurate differential diagnosis with large language models. *Nature*, pages 1–7.

Yan Meng, Liangming Pan, Yixin Cao, and Min-Yen Kan. 2023. Followupqg: Towards information-seeking follow-up question generation. *arXiv preprint arXiv:2309.05007*.

Chiara Valentina Misischia, Flora Poecze, and Christine Strauss. 2022. Chatbots in customer service: Their relevance and impact on service quality. *Procedia Computer Science*, 201:421–428. The 13th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 5th International Conference on Emerging Data and Industry 4.0 (EDI40).

Shu-Ting Pi, Michael Yang, and Qun Liu. 2024. Contact complexity in customer service.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report.

Shaina Raza, Deepak John Reji, Femi Shajan, and Syed Raza Bashir. 2022. Large-scale application of named entity recognition to biomedicine and epidemiology. *PLOS Digital Health*, 1(12):e0000152.

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.

Julia Shaver. 2022. The state of telehealth before and after the covid-19 pandemic. *Primary Care: Clinics in Office Practice*, 49(4):517–530.

Michael Stillman. 2023. Death by Patient Portal. *JAMA*, 330(3):223–224.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2024. MedAgents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 599–621, Bangkok, Thailand. Association for Computational Linguistics.

Tao Tu, Anil Palepu, Mike Schaekermann, Khaled Saab, Jan Freyberg, Ryutaro Tanno, Amy Wang, Brenna Li, Mohamed Amin, Nenad Tomasev, Shekoofeh Azizi, Karan Singhal, Yong Cheng, Le Hou, Albert Webson, Kavita Kulkarni, S Sara Mahdavi, Christopher Semturs, Juraj Gottweis, Joelle Barral, Katherine Chou, Greg S Corrado, Yossi Matias, Alan Karthikesalingam, and Vivek Natarajan. 2024. Towards conversational diagnostic ai.

Amos Tversky. 1977. Features of similarity. *Psychological review*, 84(4):327.

Deeksha Varshney, Aizan Zafar, Niranshu Behera, and Asif Ekbal. 2022. CDialog: A multi-turn covid-19 conversation dataset for entity-aware dialog generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11373–11385, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ziyu Wang, Hao Li, Di Huang, and Amir M. Rahmani. 2024. Healthq: Unveiling questioning capabilities of llm chains in healthcare conversations.

Caleb Winston, Cleah Winston, Claris Winston, and Chloe Winston. 2024. Medical question-generation for pre-consultation with llm in-context learning. In *GenAI for Health: Potential, Trust and Policy Compliance*.

Zhentao Xu, Mark Jerome Cruz, Matthew Guevara, Tie Wang, Manasi Deshpande, Xiaofeng Wang, and Zheng Li. 2024. Retrieval-augmented generation with knowledge graphs for customer service question answering. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR 2024, page 2905–2909. ACM.

Zihao Yi, Jiarui Ouyang, Yuwen Liu, Tianhao Liao, Zhe Xu, and Ying Shen. 2024. A survey on recent advances in llm-based multi-turn dialogue systems.

Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, Hongchao Fang, Penghui Zhu, Shu Chen, and Pengtao Xie. 2020. MedDialog: Large-scale medical dialogue datasets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250, Online. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

A Additional Results

Below, we show the global question matching scores, which describe the overall percentage of questions matched across all samples. Note that this is distinct from RIM, which measures sample-level coverage.

	Llama3-8b	Llama3-8b-Aloe	Qwen-32b
Zero-Shot-U	0.34 / 11	0.34 / 17	0.38 / 10
Few-Shot-U	0.28 / 9	0.33 / 19	0.35 / 10
Zero-Shot-40	0.42 / 30	0.40 / 30	0.44 / 35
Few-Shot-40	0.37 / 35	0.38 / 35	0.45 / 37
Long-Thought	-	-	0.32 / 10
Follow-up Q	0.58 / 36	0.61 / 58	0.52 / 35

Table 5: FB-Real: (Percentage of total questions matched by each model / mean number of questions asked).

	Llama3-8b	Llama3-8b-Aloe	Qwen-32b
Zero-Shot-U	0.28 / 11	0.29 / 17	0.27 / 11
Few-Shot-U	0.22 / 8	0.26 / 18	0.25 / 10
Zero-Shot-40	0.32 / 30	0.34 / 31	0.41 / 36
Few-Shot-40	0.32 / 35	0.33 / 35	0.40 / 38
Long-Thought	-	-	0.25 / 11
Followup-Q	0.46 / 35	0.49 / 59	0.41 / 34

Table 6: FB-Synth: (Percentage of total questions matched by each model / mean number of questions asked)

B Modeling Details

B.1 Hyperparameters

For all of the experiments in Tables 2 and 3, we use temperature = 0.6 with the remaining sampling parameters left as default.

FollowupQ has various values of k used for each agent. We choose a reasonable k value for our experiments based on our conversations with providers, but plan to explore a broader hyperparameter search in future works. EHR Agents: $k = 1$ (i.e., 1 question for medical history, 1 question for medications). Differential Diagnostics: $k = 3$ per diagnosis. Message Clarification Agents $\rightarrow$ Symptom inquiry agents write $k = 2$ questions per symptom. Message ambiguity and temporal reasoning use $k = 3$. Self-treatment agents use $k = 2$.

B.2 Computing the Differential Diagnosis

Computing the Differential Diagnosis (DDX) for a patient is itself a challenging sub-task which is the subject of prior work (McDuff et al., 2025;

Fansi Tchango et al., 2022). Given that the motivation of this paper is to generate follow-up questions, not generate DDXs, we pre-compute the best and worst-case DDX for each dataset before running FollowupQ. This design choice ensures that question generation quality for each model is evaluated fairly and not bottle-necked by a pre-question computation. Thus, each model uses the DDX from a Qwen-2.5-32b-Instruct model but future iterations may benefit from LLMs specifically trained to compute the DDX. Note the questions that pertain to each pre-computed diagnoses are the outputs from each individual model. This design choice thus standardizes question generation experiments, but future works may diversify the DDX model as needed.

B.3 Experimental Cost

We use an NVIDIA A40 GPU to run our experiments. We estimate our results require between 100-150 hours of computing time.

C Filtration Step Details

De-Duplication: For a given set of follow-up questions $\hat{Q}$, we first embed each $\hat{q}_i \in \hat{Q}$ using SBERT (Reimers and Gurevych, 2019) and compute k question clusters $X = \{x_1, \ldots, x_k\}$ using K-Means clustering. We then prompt a de-duplication agent to reduce each cluster into a set of unique information requests $\hat{Q}_{x_i} = f(x_i, P_{redundant})$.

The filtered question list $\hat{Q}_{filter}$ is the union of filtered question clusters $\hat{Q}_{x_i} \forall x_i \in X$. Note we de-duplicate on a per-cluster basis to improve accuracy when employing smaller, open-source LLMs, which we find are unable to properly filter large question lists in a single shot. By clustering, we reduce the complexity of the task by trying to maximize the overlap of questions in a given filtering step, making it clear what elements are redundant.

Top-K Selection: Finally, we design a top-k filtration agent to return the top-k most important questions to reduce ambiguity in the patient's message.

$$\hat{Q} = f(T, \hat{Q}_{filter}, P_{top_k}) \tag{8}$$

This agent takes the set of redundancy-filtered questions $\hat{Q}_{filter}$ and uses the top-k filtering prompt P_{top_k} to generate the final set of predicted questions $\hat{Q}$.

Note that for both De-Duplication and Top-K Sampling, we use Qwen2.5-32b-instruct to perform these tasks. This model was chosen as it was the largest parameter model available on our HIPAA-compliant compute server. The prompt we use for the top-k selection step is as follows:

Top-K Selection Prompt:
Task Description
You are a clinician who has been given a list of questions to ask a patient. You feel the list is too long, and only want to ask 10 questions max.
Rules
- Output the 10 most important questions from the provided list.
- Important questions are defined as those that fill in the most critical pieces of missing information from the patient's original message.
Make sure you only output the question list and nothing else.
Patient Message
{patient message}
All Questions
{all questions}
Top-10 Questions as Numbered List

D Auto-Evaluation Details

Results in this study use our metric, Requested Information Match (RIM), to compare model performance. RIM requires a pairwise comparison between all true and predicted question sets. Recall that a pair of matching questions means that the LLM's response should elicit the same information as the provider's question. In this section, we describe (i) a Physician-crafted test set (ii) prompts (iii) data generation & fine-tuning procedure for auto-evaluation.

D.1 Physician-Crafted Test Set

Our auto-eval test set was created by sampling 50 random questions from FB-Synth. We had to limit the size of our test set due to limited access to expert physicians for annotation. To create the test set, the authors then sourced a set of primarily challenging candidates for positive and negative question matches for each question. By challenging, we mean (i) non-matching candidates with high token overlap and (ii) matching candidates with high semantic similarity but potential structural differences. This design choice helps thoroughly vet the model, as questions that are near exact duplicates or on completely different topics are easy for a model to detect. Finally, a family physician with 20+ years of experience annotated our test set, and the resulting match data included 56 matches and 44 non-matches. We provide this dataset as a supplement to this work.

To illustrate, consider the following examples:
Question A: Do you have any abdominal pain?
Question B: Does your lower stomach hurt?
Label: No-Match
Analysis: While these questions seem similar, inquiry about lower-stomach pain is often directed at identifying the presence of a specific illness. Question A here is more of a general inquiry into stomach pain, with the physician casting a wide net for information collection. Thus, these questions have unequal specificity, making them a non-match.
Question A: Does your bladder feel tight?
Question B: Does your bladder feel different than usual?
Label: Match
Analysis: Here is an example where the physician feels "tight" and "different than usual" will both give them enough information to rule out their suspected issue.

The full dataset can be found at https://huggingface.co/datasets/PortalPal-AI/Question-Matching

D.2 Prompt

Below is the prompt used for our LLM-as-Judge model.

> You are presented with two medical questions written by two different doctors. Determine if the answer to question B will elicit the information required to answer question A. Keep in mind that question B should be equally as specific or vague as question A in order for two questions to elicit the same information.
> Output yes or no and nothing else.
> Question A: {A}
> Question B: {B}
> Answer:

Note that the instruction "Keep in mind that question B should be equally as specific or vague as question A" was found to help the model understand the physician's behavior of asking intentionally generic/vague and specific questions. Using our previous example:
Question A: Do you have any abdominal pain?
Question B: Does your lower stomach hurt?

This prompt structure will help the model identify that the physician in Question A is looking for answers *beyond* lower stomach pain and is thus a non-match.

D.3 Data Generation & Fine-Tuning Procedure

Our evaluation procedure can be computationally expensive due to the need for pair-wise comparison between predicted and ground truth question sets. Thus, we employ PHI-4 (et. al, 2024b) as our evaluator as it is (i) a smaller, more efficient LLM (only 14B parameters), making it suited for scalable inference and (ii) PHI-4 is separate from the models used in our main results table thus avoiding any biases. In other words, out of precaution, we use a separate LLM for evaluation to ensure no favoritism is shown by our judge model towards questions generated by the same LLM.

To help PHI-4 better understand our task, we perform synthetic data generation to fine-tune the model using the prompt in Appendix D.2. Specifically, we write five canonical examples of question matching in the form of contrastive learning examples and teach an LLM to generate similar data points based on randomly sampled medical symptoms. Consider one canonical example.

```
Root: What did you stub your toe on?
Positive: To be clear, did you stub your
toe on something? What exactly
did you hit?
Negative: Does your toe hurt?

## Task ##
-> Now, write a new contrastive sample,
this time about {topic}.
```

We use such examples to generate new contrastive data points about {topic}. We consider the root samples to be provider questions and the positive / negative samples to be LLM-generated questions for fine-tuning. We generate 1,000 contrastive samples, which leads to 2,000 fine-tuning examples for our judge model. Note that we perform n-gram de-duplication with our test set (n=5) to ensure we are not accidentally generating any test samples. The result of our fine-tuning can be found below.

Model	Precision	Recall	F1
Phi-4 (Base)	0.86	0.86	0.85
Phi-4 (Fine-Tuned)	0.88	0.87	0.87

We find that this gives us a 2-point boost on our physician-crafted test set. We additionally note that empirical inspection of model outputs shows strong performance in identifying question matches. We fine-tune the model using Q-LoRA (Hu et al., 2021; Dettmers et al., 2023) the SFT trainer from TRL [2] for 1 epoch with a learning rate of 2e-5 and batch size of 32. In the future, we will explore larger models as the basis of our LLM-as-Judge framework.

E FollowupBench Dataset Details

In this section, we outline (i) how we created FB-Real from asynchronous patient conversations sourced from an online EHR portal and (ii) how we created FB-Synth using semi-synthetic data and real provider follow-up questions.

E.1 Creation of FollowupBench-Real

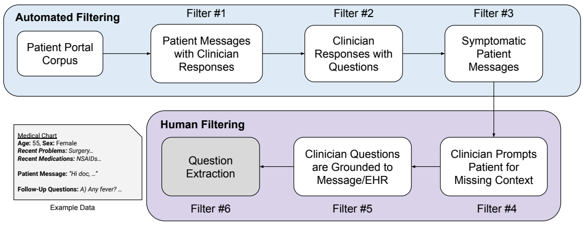

Automated Filtering
Filter #1
Filter #2
Filter #3
Patient Portal Corpus
Patient Messages with Clinician Responses
Clinician Responses with Questions
Symptomatic Patient Messages
Human Filtering
Question Extraction
Clinician Questions are Grounded to Message/EHR
Clinician Prompts Patient for Missing Context
Filter #6
Filter #5
Filter #4
Medical Chart
Age: 55, Sex: Female
Recent Problems: Surgery...
Recent Medications: NSAIDs...
Patient Message: "Hi doc, ..."
Follow-Up Questions: A) Any fever? ...
Example Data

Figure 4: Visualization of the data collection process for FollowupBench-Real.

The goal of this study is to explore the capacity of LLMs to generate follow-up questions that pertain to any *missing context* required to make an informed decision. Thus, we aim to focus on symptomatic patients who do not provide the full breadth of information required to be properly diagnosed. We visualize our sample selection pipeline in Figure 4. First, we filter for patient-initiated conversations that received at least one provider response (Filter #1). Next, we filter for conversations where the provider response contains at least one question mark '?' (Filter #2). Finally, in an effort to emphasize symptomatic patients and de-emphasize logistical inquiries such as medication or appointment-centric messages, we employ a BERT-based (Devlin et al., 2019) symptom extraction model introduced in (Raza et al., 2022). Based on this symptom tagger, we only consider messages that contain at least one symptom (Filter #3). Based on these criteria, we have a pool of eligible symptomatic conversations. Our final filtering step (Filter #4-6) leverages human annotation and is described in the following section.

Human Data Annotation

From the pool of eligible conversations, two human annotators were tasked with selecting samples that met the following two criteria.

Criteria #1: The provider response must prompt the patient for missing information about their symptoms. Any responses that solely pertain to logistical matters, such as medication refills or appointment scheduling, were not to be included.

Criteria #2: The follow-up questions asked by the provider must be grounded to either the patient's message or their Chart/EHR. In other words, questions that appear based on information from a prior in-person encounter are deemed out-of-scope for the LLM and are thus not included in the final dataset for the sake of fairness. Additional out-of-scope message types are (i) those based on images submitted by patients (which we do not have access to) (ii) temporally dependent questions such as "how are you feeling now?" which presumes significant time has elapsed since the message was processed (iii) questions which are healthcare suggestions (e.g. would you like to try this treatment?).

If the human annotators deem a provider response meets both Criteria #1 and Criteria #2, we include the sample in our evaluation. Once a sample is considered eligible, the human annotator is tasked with extracting the questions from the provider response to be used later as our ground truth test set for LLM evaluation. Note that provider responses that contain both eligible and ineligible questions are still utilized, with all out-of-scope questions being excluded during evaluation. Our human annotators sampled messages until the target evaluation set size of (n=150) was reached. Note that for the sake of facilitating the auto-evaluation task, the human annotators were allowed to slightly rephrase the provider questions to be complete sentences.

Our human annotators are senior Ph.D. students experienced in clinical NLP. Our data-inclusion

[2] https://huggingface.co/docs/trl/en/sft_trainer

criteria were developed in collaboration with a primary care provider with 20+ years of experience working in a large healthcare system. Our expert collaborator has significant experience in writing patient portal message responses. Each of our human annotators was trained over multiple sessions with our expert on how to identify quality samples for evaluation as well as parse provider messages into a list of questions. We highlight that, in general, medical expertise is not needed to determine if a question is predominantly logistical. For example, our human annotators were tasked with filtering out samples with questions such as We have an appointment available today, would you like to come in?" or “which pharmacy did you want the medication sent to?".

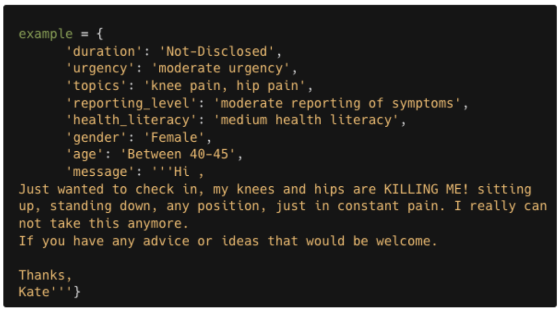
In summary, we use this process to curate the 150 samples used in FB-Real.

E.2 Creation of FollowupBench-Synth

Recall that FollowupBench-Synth is a semi-synthetic dataset containing (patient messages, linked EHR data, and follow-up question set) triplets. All patient messages in FB-Synth are 100% synthetic. All EHR data is real and is randomly sampled from a de-identified pool of patient EHR records at a large university medical center. The follow-up questions were written by a team of providers who were presented with the synthetic messages and real EHR data. In the following sections, we describe each step in the synthetic data generation process.

E.2.1 Synthesizing Patient Messages

Inspired by Gatto et al. (2024), our synthetic data generation pipeline uses grounded generation to promote sample realism. In this work, our goal is to map a randomly sampled set of categorical features into a new patient message. We use the following categorical features: (1) topics/symptoms, (2) duration of symptoms, (3) message urgency, (4) reporting level, or how forthcoming the patient is with their symptoms, (5) level of health literacy, (6) age, and (7) gender. These categories were identified as relevant after discussion with clinical experts and helped promote sample diversity during generation. See Figure 5 for an example set of categories and corresponding messages.

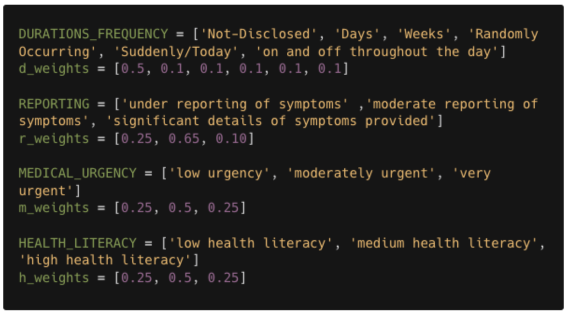


```
example = {
      'duration': 'Not-Disclosed',
      'urgency': 'moderate urgency',
      'topics': 'knee pain, hip pain',
      'reporting_level': 'moderate reporting of symptoms',
      'health_literacy': 'medium health literacy',
      'gender': 'Female',
      'age': 'Between 40-45',
      'message': '''Hi ,
Just wanted to check in, my knees and hips are KILLING ME! sitting
up, standing down, any position, just in constant pain. I really can
not take this anymore.
If you have any advice or ideas that would be welcome.

Thanks,
Kate'''}
```

Figure 5: Example sample used in grounded generation. The goal is to map all non-message features to a novel synthetic message based on randomly sampled features. We provide few-shot examples to guide the generation.

To generate a new sample, we (1) randomly sample 1-2 topics/symptoms from a list of common patient message topics curated by the authors and reviewed by a provider. The list of topics used for generation is provided as a supplement to this submission. For steps (2-5) we randomly sample from the values shown in Figure 6 using the displayed categories and category weights. (6-7) Age and Gender are taken from the EHR data, which will be matched with the synthetic message.



```
DURATIONS_FREQUENCY = ['Not-Disclosed', 'Days', 'Weeks', 'Randomly
Occurring', 'Suddenly/Today', 'on and off throughout the day']
d_weights = [0.5, 0.1, 0.1, 0.1, 0.1, 0.1]

REPORTING = ['under reporting of symptoms' ,'moderate reporting of
symptoms', 'significant details of symptoms provided']
r_weights = [0.25, 0.65, 0.10]

MEDICAL_URGENCY = ['low urgency', 'moderately urgent', 'very
urgent']
m_weights = [0.25, 0.5, 0.25]

HEALTH_LITERACY = ['low health literacy', 'medium health literacy',
'high health literacy']
h_weights = [0.25, 0.5, 0.25]
```

Figure 6: Synthetic data generation sampling categories for Duration, Reporting-Level, Urgency, and Health Literacy.

We thus feed an LLM (Llama-3.3-70B-Instruct) 3 in-context samples of how to map a set of categorical variables to a realistic patient message. The three in-context samples were generated by the authors to mimic realistic patient message style. We use the outputs of this synthetic message generation framework in FB-Synth.

E.2.2 EHR Data Selection

EHR data is randomly sampled from a pool of patients at a large university medical center. We take snapshots of patient EHRs corresponding to details outlined in Section 3.1. See Appendix H for an example of an EHR record used in this study.

E.2.3 Data annotation

Pairs of synthetic patient messages and real EHR records were presented to practicing medical providers at a large university hospital for annotation. We discuss the details of the data annotation strategy below.

Annotator Details Nine providers were employed as annotators to create the FB-Synth dataset. Our annotators all work as nurses, medical residents, or primary care providers who have experience responding to patient portal messages. These annotators were recruited for our study via referral from a collaborating provider. All 9 annotators are female and provide healthcare services in the United States. All annotators were informed on how their data would be used as part of this study.

Payment Each provider was paid a fair wage of $50 (USD) per hour to participate in this study.

Annotation Details All providers employed in this study are familiar with writing follow-up questions to patient messages. Thus, annotators required little training. However, we highlight a few guidelines told to annotators during a brief live presentation:

1. Your job is to write a list of questions you wish the AI would have asked the patient before they click send!

2. Your goal is to ask questions that clarify a patient's symptoms, self-treatment, and health status.

3. We are not looking for logistical questions! E.g., appointments, refills

4. The list does not have to be perfect! There is no one right answer! Just use your best judgment.

5. You should write 3-10 questions per message. However, it is okay to go over or under this amount.

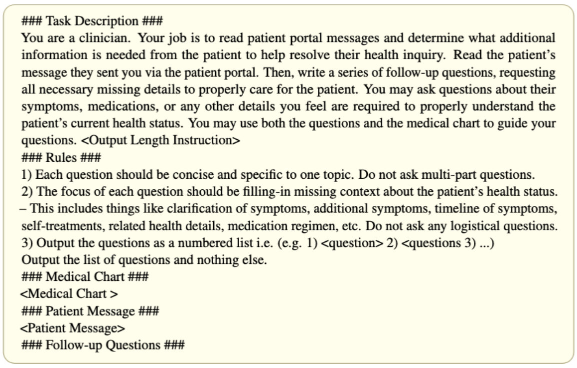
Annotators wrote follow-up questions at their own pace and reported their time spent accordingly. Most annotators in this study provided $\approx$ 3 hours of annotation.

F Prompts

In this section, we discuss the prompts used in our study.

F.1 Baseline Prompts

Figure 7 shows the base prompt for all of our zero, few-shot, and long-thought experiments. Note that <Output Length Instruction> is a placeholder for a string instructing *the unbounded samples* vs *generation of K samples*. The unbounded prompt states “Write as many questions as you need" whereas where k-generation states “write exactly k questions". The few-shot variants add examples in the prompt using the same set of instructions/rules.



```
### Task Description ###
You are a clinician. Your job is to read patient portal messages and determine what additional
information is needed from the patient to help resolve their health inquiry. Read the patient's
message they sent you via the patient portal. Then, write a series of follow-up questions, requesting
all necessary missing details to properly care for the patient. You may ask questions about their
symptoms, medications, or any other details you feel are required to properly understand the
patient's current health status. You may use both the questions and the medical chart to guide your
questions. <Output Length Instruction>
### Rules ###
1) Each question should be concise and specific to one topic. Do not ask multi-part questions.
2) The focus of each question should be filling-in missing context about the patient's health status.
– This includes things like clarification of symptoms, additional symptoms, timeline of symptoms,
self-treatments, related health details, medication regimen, etc. Do not ask any logistical questions.
3) Output the questions as a numbered list i.e. (e.g. 1) <question> 2) <questions 3) ...)
Output the list of questions and nothing else.
### Medical Chart ###
<Medical Chart >
### Patient Message ###
<Patient Message>
### Follow-up Questions ###
```

Figure 7: Core prompt used for zero-shot and few-shot baseline approaches.

F.2 Multi-Agent Prompts

FollowupQ uses 12 prompts at various stages of inference. Due to the considerable space this requires to display, we show the Differential Diagnostics-related questions as an example and refer the reader to the code base for the full suite of prompts.

```
### Task Description ###
You are a doctor. You have received a message from your patient.
Given the message, you suspect they have ### Suspected Issue ### However, you need more information before being sure. Write {k} questions you would want to ask this patient to confirm if they do, or do not, have ### Suspected Issue ###.
### Rules ###
- Output the questions as a numbered list. E.g. 1) <question> 2) <question> ...
- Make questions as concise as possible. Do not ask multi-part questions.
- Only ask the k most important questions needed to clarify this specific diagnosis.
- Make the questions specific to ruling out your suspected issue.
### Patient Message ###
{msg}
### Suspected Issue ###
{suspected_issue}
### {k} Follow-up Questions ###:
```

G Limitations of Self-Reported Patient Information

In this section, we provide an example to illustrate why EHR data modeling is crucial in asynchronous medical dialogue, highlighting the risks of solely relying on patients to provide their medical context. The following example was created and analyzed by one of our collaborating physicians.

Consider the following patient message and accompanying medication list.

Medication List: Furosemide

Message: Hi, been having a low-grade fever this week and a mild cough. Not sure I can make it into work! Heck, I could hardly get my shoes on this morning, I'm feeling so swollen. Need a doctor's note, please.

Analysis: The medical urgency of this message is dependent on an EHR element, Furosemide, that could reasonably be omitted in a self-report. Furosemide reduces swelling, and if the patient can't get their shoes on, their medication may not be functioning, making them prone to heart failure. Thus, it is crucial that future patient messaging systems take patient EHR data into close consideration.

H Example Data

Below is an example Chart, Message, and set of Follow-up questions. Note that this is a chart derived from MT-Samples [3], which we used during early development of our annotation guidelines for FB-Synth (please note MT-Samples is not included in FB-Synth, and that this chart is for illustration only). The message is synthetic, and the follow-up questions are generated by a practicing triage nurse at our partner hospital, a large academic medical center.

```
<Example Medical Chart>
###Demographics###
Age: 50
Gender: Male
###Full Active Problem List###
Diabetes Mellitus type II - hypertension - coronary artery disease - atrial fibrillation - smokes 2 packs of cigarettes per day - Family history of coronary artery disease (father & brother).
###Recent Encounters (Max 10)###
status post PTCA in 1995 by Dr. ABC
###Medications (Outpatient)###
- Aspirin 81 milligrams QDay
- Humulin N. insulin 50 units in a.m.
- HCTZ 50 mg QDay.
- Nitroglycerin 1/150 sublingually
- PRN chest pain.
```

<Patient Message>
Hey Dr. [Doctor's Last Name],
Getting a little nervous... I noticed I am breathing really rapidly, and it's tough to catch my breath. On top of that, I'm coughing up blood, which has me pretty freaked out. This all started quite suddenly, and I'm not sure what's going on. Could we possibly arrange to see each other soon or talk about what I should do next? Can you call me?

[3] https://mtsamples.com/

<Follow-up Questions generated by Real Provider>
1) You say it started suddenly, but when exactly did it start?
2) How much blood are you coughing up? A teaspoon? Flecs?
3) Describe the color of the blood. (Dark vs light)
4) Are you short of breath during any specific activities? E.g. sitting down vs moving.
5) Are you able to talk in a full sentence without pausing?
6) Any dizziness?
7) Has this ever happened before?
8) Have you noticed any blood in your stool?