ChatGPT

Prompt Engineering for Healthcare & Call Center Applications

Introduction

Large language models (LLMs) like GPT-4 are increasingly applied in healthcare and call center settings to automate tasks such as patient triage, claims processing, and customer support. By guiding these models with precise prompts, organizations can improve efficiency and care quality [1] [2]. However, unstructured LLM outputs can be misleading or dangerous in clinical contexts, so *prompt engineering* is essential to ensure accuracy and safety [3] [4].

Prompt Design Best Practices

- **Be Specific:** Clearly define the task, role, and output format to avoid ambiguity [5] [4]. For example, instruct "summarize" and set limits (e.g. "limit each summary to 300 words" [4]).
- **Provide Context:** Include relevant background (patient history, symptoms, call transcript excerpts, etc.) so the model has enough information to answer accurately [6].
- **Give Examples:** Use few-shot prompts with input-output examples or instruct the model to show its reasoning (chain-of-thought) to teach the desired style and content [7] [8]. Negative examples (what **not** to output) can also clarify expectations.
- **Specify Structure:** Tell the model exactly how to format its answer (bullet list, JSON fields, numbered steps, etc.) [4]. Explicit format instructions greatly improve consistency.
- **Iterate & Validate:** Test and refine prompts using sample data. Have clinicians or domain experts review LLM outputs, and update prompts based on feedback [9]. Continuous evaluation (especially in pilot deployments) helps catch errors or biases early.
- **Stay Compliant:** Omit any protected health information (PHI) in prompts and outputs. Instruct the model to use authoritative sources or peer-reviewed guidance for medical/insurance content [10] [11]. Remember that GPT-style LLMs are not inherently HIPAA-compliant, so do not feed them real patient records.

Zero-Shot Prompting

Zero-shot prompts provide a direct instruction to the LLM without any examples [12]. They rely on the model's pre-trained knowledge and must be very clear. For high-stakes tasks like triage or scheduling, explicitly state the role, input details, and required output format. Below are sample zero-shot templates with placeholders:

- **Prompt (Patient Triage, Zero-Shot):**

```
You are a medical triage assistant. A patient reports [insert patient
symptoms]. Based on standard triage guidelines, classify the case as *Non-
Urgent*, *Urgent*, or *Emergency*, and recommend the next steps.
```

Example Output: "Emergency – possible appendicitis given severe abdominal pain and fever. Advise immediate ER visit."

- **Prompt (Appointment Reminder, Zero-Shot):**

```
You are a clinic assistant. Write a friendly reminder message for patient
**[Patient Name]** about their upcoming appointment on **[Appointment
Date]** at **[Clinic Name]**. Include rescheduling instructions if needed.
```

Example Output: "Hello [Patient Name], this is a reminder of your appointment at [Clinic Name] on [Appointment Date]. If you need to reschedule or have questions, please call us at [Phone Number]. Thank you!"

Few-Shot Prompting

Few-shot prompts include a handful of example interactions (typically 1–5) to demonstrate the desired input-output pattern [13] [7]. This helps the model learn the format, tone, and detail level you expect. Provide representative examples covering your use case, then ask the model to perform a similar task on new inputs. For example:

- **Prompt (Insurance Claim Summary, Few-Shot):**

```
You are an insurance assistant. Summarize the customer calls below,
focusing on the claim issue, actions taken, and next steps.

Example 1:
Customer: "My health insurance claim #12345 was denied after surgery."
Summary: "The customer’s post-surgery claim #12345 was initially denied. We
verified coverage and discovered incomplete paperwork. The agent told the
customer we will resubmit the claim with missing documents and follow up in
3 days."

Example 2:
Customer: "I updated my policy but my deductible reset already?"
Summary: "The customer is concerned that their policy update caused a
deductible reset. The agent checked the account and confirmed a system
error. The agent assured the customer the deductible will be corrected and
notified billing. The issue will resolve in 2 days."

Example 3:
```

```
Customer: "[insert customer call transcript or issue placeholder]"
Summary:
```

Example Output: "The customer's claim was [issue summary]. The agent [actions taken]. We informed the customer [next steps]."

- **Prompt (Patient Education, Few-Shot):**

```
You are a health educator. For each medical condition below, provide a
patient-friendly explanation using simple language.

Medical condition: Hypertension
Explanation: "Hypertension, or high blood pressure, means your heart is
working too hard and your blood is pushing too strongly against your blood
vessels. Lifestyle changes like diet, exercise, and sometimes medication
can help control it."

Medical condition: Diabetes
Explanation: "Diabetes is a condition where the body can't use sugar
properly. This happens because of a problem with insulin, a hormone that
helps sugar enter your cells for energy. It can be managed with healthy
eating, exercise, and medication."

Medical condition: [insert_condition]
Explanation:
```

Example Output: (Depends on [insert_condition], e.g. "Asthma is when your airways become inflamed...")

Chain-of-Thought Prompting

Chain-of-thought (CoT) prompts explicitly ask the model to break down its reasoning into steps [8]. This is useful for complex tasks like diagnosing or analyzing detailed conversations. You can prompt the model to "think step by step" or list numbered rationale points before answering. CoT often yields more accurate and transparent results for multi-step problems.

- **Prompt (Clinical Decision, Chain-of-Thought):**

```
You are a medical analyst. A patient presents with [insert symptoms].
Determine whether this is an emergency. Show your reasoning step by step
before giving the conclusion.
```

Example Output:

1. The patient's symptoms include [...], which are consistent with [interpretation].
2. Based on symptom analysis, this suggests [...].
3. Other concerning factors: [...].
   Conclusion: "Urgent – symptoms suggest [condition]; advise emergency care."

- **Prompt (Call Analysis, Chain-of-Thought):**

```
You are a customer service coach. Review the following call excerpt and
identify the customer's main concern and the outcome. Show your reasoning
steps.
Call: "[insert call transcript or conversation snippet]"
```

Example Output:

1. Customer's statement indicates concern about [...].
2. Agent's response indicates [...].
   Summary: "The customer was concerned about [issue], and the agent agreed to [resolution]."

Function-Calling / Structured Output

Modern LLM platforms can **invoke functions or APIs** by returning structured JSON [14]. This is ideal for automating tasks like scheduling or database queries. Define the function name and parameters ahead of time, and instruct the model to output a JSON object with those fields. Emphasize JSON formatting or function-call syntax in your prompt. For example:

- **Prompt (Schedule Appointment, Function-Calling):**

```
You are integrated with the clinic's scheduling system. A patient [Patient
Name] wants to reschedule appointment [Appointment ID] to [New DateTime]
due to [reason]. Provide a JSON function call to `rescheduleAppointment`
with arguments: patientId, appointmentId, newTime, and reason.
```

Example Output:

```
{
  "function": "rescheduleAppointment",
  "arguments": {
    "patientId": "12345",
    "appointmentId": "A9876",
    "newTime": "2025-08-01T14:00:00",
    "reason": "Vacation conflict"
```

```
  }
}
```

- **Prompt (Check Claim Status, Function-Calling):**

```
You are a virtual assistant for insurance. A user asks: "I want to check my claim status." Extract the claim ID and provider from their request and output a JSON function call to `checkClaimStatus(claimId, provider)`. Use placeholders [insert claim ID] and [insert provider].
```

Example Output:

```
{
  "function": "checkClaimStatus",
  "arguments": {
    "claimId": "12345",
    "provider": "MyHealthInsure"
  }
}
```

Summary of Key Guidelines

- **Iterate and Validate:** Test prompts with representative data and refine them. Involve domain experts (doctors, nurses, senior agents) to review outputs and provide feedback [9].
- **Specify Output Format:** Always instruct the model on the response structure (e.g. bullet list, numbered steps, JSON fields) to ensure consistency [4].
- **Maintain Privacy:** Do not include real patient identifiers or PHI in prompts or LLM outputs. Remind the model to rely on general medical knowledge or non-sensitive data [10] [11].
- **Role-Play:** Framing the assistant as a specific role (doctor, triage nurse, claims agent, etc.) helps it use the appropriate tone and information.
- **Control Randomness:** For critical tasks, use a lower temperature setting to make responses more predictable.

By following these best practices and using the above templates, healthcare and customer support teams can craft precise, reusable prompts. This enables LLMs to handle patient triage, claim summaries, customer service scripts, patient education, appointment reminders, and other operational tasks reliably. Always monitor the system, apply human oversight, and update prompts as requirements evolve.

Sources: Prompt engineering and LLM best practices are informed by recent industry guidance [4] [5], including healthcare-specific examples of summarization and patient communication [2] [3].

---

1 4 7 9 10 Prompt Engineering in Healthcare: Best Practices, Strategies & Trends | HealthTech
https://healthtechmagazine.net/article/2025/04/prompt-engineering-in-healthcare-best-practices-strategies-trends-perfcon

2 3 11 12 13 How Does Prompt Engineering Help?
https://capestart.com/resources/blog/healthcare-and-chatgpt-how-does-prompt-engineering-help/

5 6 HIPAA Compliant LLM - AI Prompts for patient care and insurance
https://hathr.ai/hipaa-compliant-llm-prompt-library/

8 Chain-of-Thought Prompting
https://learnprompting.org/docs/intermediate/chain_of_thought?srsltid=AfmBOoqhV7x5dErFsBK2R4R5LRG1ZBYf03WlcqpkK9VVEshSSHb9NAqu

14 How to master prompt engineering, function calling, and the use of GPT | Sendbird
https://sendbird.com/developer/tutorials/ai-chatbot-prompt-engineering-function-calling