CampusDual TIC

AI Engineering

Clase 4 - User and Context Assistant

Ingeniería de prompts para Generative AI

Duración: 45 minutos

1. Importancia del Contexto en los Prompts

El contexto proporciona información adicional que guía al modelo a generar respuestas más precisas y coherentes. Incluir un contexto adecuado ayuda al modelo a entender mejor la intención detrás de la solicitud y a enfocar la respuesta en la dirección deseada.

Ejemplo:

- Sin Contexto:

Describe los beneficios de la inteligencia artificial en la medicina.

- Con Contexto:

Contexto: Eres un experto en tecnología médica.
Describe los beneficios de la inteligencia artificial en la medicina.

2. Diferenciación de Roles: Usuario vs. Asistente

Entender y definir claramente los roles de Usuario y Asistente en los prompts puede mejorar la fluidez y efectividad de las interacciones con GenAI.

- Usuario: Es quien realiza la solicitud o proporciona el prompt.
- Asistente: Es el modelo de GenAI que responde a la solicitud.

Implementación en Prompts:

```
class ModeloGenerativo:
 # ... (código anterior)
  def crear_prompt_usuario(self, mensaje):
     return f"Usuario: {mensaje}"
  def crear_prompt_asistente(self, respuesta):
     return f"Asistente: {respuesta}"


Usuario: ¿Cuáles son las aplicaciones de la IA en la educación?
Asistente: La IA en la educación permite la personalización del aprendizaje, adaptándose a las necesidades individuales de cada estudiante.
```

3. Tipos de Prompts: Zero-Shot, One-Shot y Few-Shot

La manera en que estructuramos los prompts puede influir en la capacidad del modelo para generar respuestas adecuadas. Existen tres tipos principales de prompts:

- **Zero-Shot Prompts**

Definición: El modelo responde sin recibir ejemplos previos.
Uso: Útil cuando se espera que el modelo entienda la tarea basándose únicamente en la descripción proporcionada.

Por favor, describe los beneficios de la inteligencia artificial en la agricultura.

- **One-Shot Prompts**

Definición: El modelo recibe un único ejemplo antes de la solicitud.
Uso: Ayuda al modelo a entender mejor el formato y la expectativa de la respuesta.

Tarea: Describe los beneficios de la inteligencia artificial en la educación.
Respuesta: La inteligencia artificial en la educación permite la personalización del aprendizaje, adaptándose a las necesidades individuales de cada estudiante.

- **Few-Shot Prompts**

Definición: El modelo recibe varios ejemplos antes de la solicitud.
Uso: Proporciona múltiples referencias para guiar al modelo hacia respuestas más coherentes y precisas.

Tarea: Describe los beneficios de la inteligencia artificial en la educación.
Respuesta: La inteligencia artificial en la educación permite la personalización del aprendizaje, adaptándose a las necesidades individuales de cada estudiante.

Tarea: Describe los beneficios de la inteligencia artificial en la salud.
Respuesta: La inteligencia artificial en la salud mejora el diagnóstico temprano, optimiza tratamientos y gestiona eficientemente los datos médicos.

Few shot prompting example

Your job is to create content for our client, {{client_name}}. Here is some information about the client {{client_description}}.

Here are a few examples of content we've created in the past from briefs:

"""
Example 1:
Brief: {{brief_1}}
Content: {{content_1}}

Example 2:
Brief: {{brief_2}}
Content: {{content_2}}
"""

Here is the latest brief to create content about:
"""
Brief:{{brief_description}}
Content:

4. Técnicas Avanzadas en Ingeniería de Prompts

Para maximizar la efectividad de los prompts, se pueden emplear técnicas avanzadas como:

- Formateo de Plantillas: Utilizar plantillas predefinidas que estructuren la solicitud de manera consistente.

```
def formatear_prompt(self, plantilla, variables):
    try:
        prompt_formateado = plantilla.format(**variables)
        return prompt_formateado
    except KeyError as e:
        print(f"Error: Falta la variable {e} en el diccionario de variables.")
        return plantilla
```

- Limpieza de Prompts: Asegurar que los prompts estén bien estructurados y libres de errores que puedan confundir al modelo.

```
def limpiar_prompt(self, prompt):
    prompt = prompt.strip()
    prompt = prompt.capitalize()
    return prompt
```

Relación con Generative AI

La ingeniería de prompts es una habilidad crucial para interactuar eficazmente con modelos de GenAI. Al diseñar prompts bien estructurados que incluyan el contexto adecuado y utilizar distintos tipos de prompts según la necesidad, puedes mejorar significativamente la calidad y relevancia de las respuestas generadas. Esta práctica te permitirá desarrollar aplicaciones más precisas y útiles, optimizando la interacción con modelos avanzados de lenguaje y otros sistemas generativos.

Preguntas Frequentes

¿Cómo determino el contexto adecuado para un prompt?

Analiza el propósito de la tarea y qué información adicional puede ayudar al modelo a entender mejor lo que se espera. Por ejemplo, si el asistente debe escribir un artículo científico, especifica su rol y el tema a tratar.

¿Qué tipo de prompt debo usar para obtener una respuesta más detallada?

Los Few-Shot Prompts suelen generar respuestas más detalladas ya que el modelo recibe varios ejemplos que guían su comportamiento.

¿Es posible combinar distintos tipos de prompts en una sola solicitud?

Sí, puedes combinar One-Shot y Few-Shot prompts para adaptar la complejidad y el detalle de las respuestas según tus necesidades específicas.

PREGUNTAS

CampusDual TIC