Manual de Estudio: AWS Certified AI Practitioner

Dominio 3.2: Técnicas Eficaces de Ingeniería de Peticiones

Material de Preparación Detallado

1. ¿Qué es una Petición (Prompt)?

AWS define una petición como el conjunto específico de entradas proporcionadas por el usuario que guían a los Modelos de Lenguaje de Gran Tamaño (LLM) para generar una respuesta apropiada. La calidad de la petición afecta directamente la calidad de la respuesta.

1.1. Componentes de una Petición

Una petición bien estructurada suele combinar:

- **Instrucción/Tarea:** La acción específica que el modelo debe realizar (ej. Resume este texto").
- **Contexto:** Información adicional que ayuda al modelo a situarse (ej. .Eres un experto en leyes").
- **Texto de entrada:** El contenido real que el modelo debe procesar.
- **Indicadores de salida:** Formato deseado (ej. "Dame la respuesta en formato JSON").

2. Técnicas de Ingeniería de Peticiones

Para el examen, es fundamental distinguir entre estas estrategias:

1. **Zero-shot (Cero muestras):** Se le pide al modelo realizar una tarea sin proporcionarle ningún ejemplo previo en la petición.
2. **Few-shot (Pocas muestras):** Se proporcionan algunos ejemplos (1 o más) dentro de la petición para calibrar el resultado. Esto se conoce como **Aprendizaje en contexto**.
3. **Chain of Thought (Cadena de Pensamiento):** Se pide al modelo que desglose su proceso de razonamiento en pasos intermedios. Ideal para tareas lógicas, matemáticas o complejas.

4. **Prompt Tuning (Ajuste de peticiones):** Técnica avanzada donde se reemplaza el texto de la petición por un soporte de incrustación continua optimizado durante el entrenamiento. Es más eficiente que el refinamiento completo (*fine-tuning*) porque mantiene los parámetros del modelo congelados.

3. El Espacio Latente y las Alucinaciones

3.1. ¿Qué es el Espacio Latente?

Es el conocimiento codificado del lenguaje dentro de un LLM. Consiste en patrones estadísticos y relaciones almacenadas durante el entrenamiento previo (usando bases de datos como Wikipedia o Common Crawl).

3.2. Origen de las Alucinaciones

Las alucinaciones ocurren cuando se hace una petición sobre un tema que no está presente o no es suficiente en el **espacio latente** del modelo.

- El modelo no razonaçomo un humano; elige palabras basándose en su **probabilidad condicional**.
- Si el conocimiento falta, el modelo elegirá la çoincidencia estadística más próxima", lo que resulta en una respuesta convincente pero objetivamente falsa.

4. Prácticas Recomendadas para el Éxito

- **Especificidad:** Instrucciones claras sobre formato, estilo, tono y longitud.
- **Iteración:** Probar y modificar las peticiones de forma cíclica.
- **Equilibrio:** Evitar peticiones excesivamente simples (imprecisas) o excesivamente complejas (saturadas).
- **Barreras de protección (Guardrails):** Implementar controles para evitar salidas no deseadas.

5. Riesgos y Seguridad en Peticiones

AWS pone especial énfasis en las vulnerabilidades de los LLMs en Amazon Bedrock:

6. Soluciones de AWS

- **Amazon Bedrock Guardrails:** Permite definir temas bloqueados, umbrales de seguridad para contenido dañino y filtrado de datos confidenciales (PII).
- **Amazon Titan:** Modelos pre-entrenados diseñados para ser controlados mediante estas técnicas.

Riesgo	Descripción
Prompt Injection	Ataque donde un usuario introduce una entrada maliciosa para engañar al modelo y producir respuestas no deseadas.
Jailbreak	Intentos deliberados de eludir las medidas de seguridad y barreras de protección (*Guardrails*) establecidas.
Hijacking (Secuestro)	Intento de cambiar o manipular la instrucción original de la petición con nuevas instrucciones.
Envenenamiento	Inclusión de instrucciones dañinas en fuentes externas (webs, correos) que el modelo podría ingerir.