Observabilidad para LLM's

Sara Lamuño García - UO283706

Andrea Acero Suárez - UO287876

Sergio Pérez Arias - UO294130

Índice

Observabilidad en Sistemas

- Frecuentemente malinterpretado por la mayoría de las empresas.
- Comprender el estado de un sistema sin modificarlo directamente.
- Problema: error o comportamiento inesperado no puede reproducirse localmente.
- Solución: analizar el estado del sistema en producción.
  - ¿Dónde está ocurriendo el problema?
  - ¿Por qué está ocurriendo el problema?
- Google y Facebook han trabajado con estos principios desde hace tiempo.

Introducción a los Grandes Modelos de Lenguaje (LLMs)

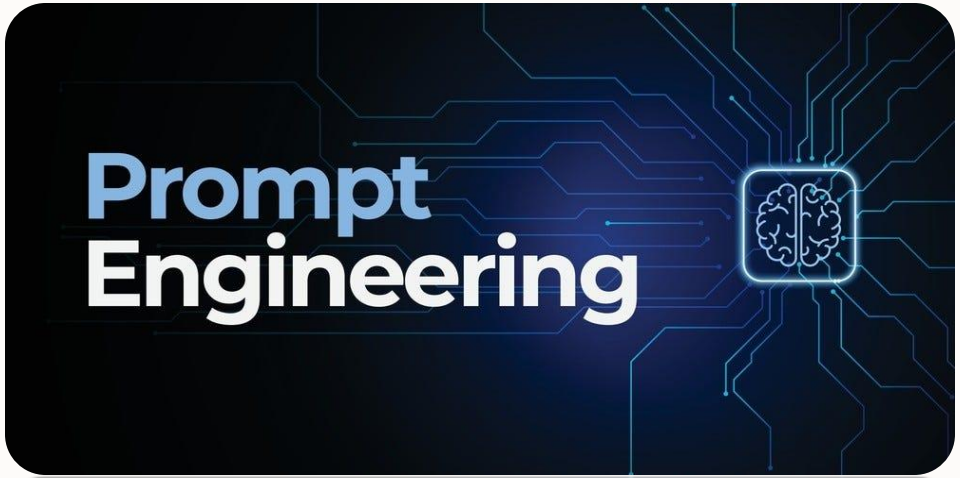
- Definición práctica
  - Reciben texto de entrada.
  - Generan respuestas.
  - Se usan en multitud de sectores.
- Definición técnica
  - Arquitectura Transformer.
  - Concepto de atención.
  - Resuelven un problema clave en el procesamiento del lenguaje natural.

Fine – tuning o ajuste fino

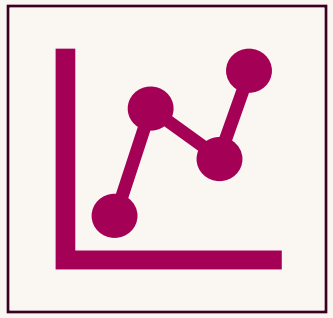
- Los modelos de lenguaje pasan por diferentes fases antes de estar listos para su uso.
- Preentrenamiento, entrenamiento y alineación.
- Fine – tuning: datos más especializados que la alineación.
- Riesgos:
  - Menos flexibilidad si se ajusta demasiado.
  - Malas respuestas.
  - Puede limitar la capacidad del modelo.

Prompt Engineering o Ingeniería de instrucciones

- Arte de diseñar las instrucciones correctas para obtener respuestas adecuadas del modelo.
- Carter lo compara con escribir consultas SQL para bases de datos.
- Implica creatividad y refinamiento.
- Retrieval-Augmented Generation (RAG).
- Carter opina considerarla parte de la ingeniería de IA.



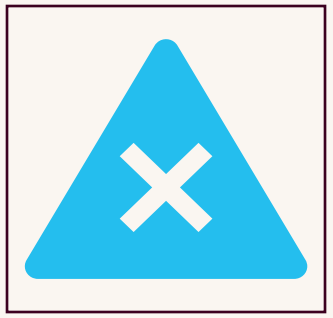
Similitudes entre la observabilidad en los LLMs y sistemas más "convencionales"



Referencia base de datos:

Parametrización de los datos y decisión de como realizarla

Resultado defectuoso para usuario, aunque la consulta este bien



Latencia:

No gusta algo que es lento

En LLMs no solo es culpa del modelo

Desafíos que la observabilidad en LLMs ayuda a resolver

Latencia:

Entender de donde proviene

El prompt y la cantidad salida generada influyen

Chain of Thought Prompting

Desarrollo impulsado por la observabilidad:

Eliminar barrera entre desarrollo y producción

Sistema vivo

Problemas que afectan usuarios

El trabajo comienza cuando estas en producción

Enfoque del desarrollo en los LLMs antes de pasar a producción

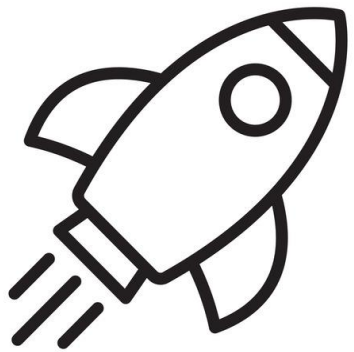
- Interés de la empresa
- La mayoría de las herramientas no son aplicables (algunas como control de versiones son muy útiles)
- No sirven procesos convencionales de QA
- Usuarios para recopilar información

Incrementalidad y Lanzamientos rápidos

Esencial tener la capacidad de realizar despliegues diarios

Flujo de trabajo proactivo:

- Analizar interacciones recientes
- Identificar errores
- Implementar soluciones



Herramientas observabilidad como Service Level Objectives(SLOs)

Equilibrar confiabilidad y la creatividad en las salidas del sistema

Importancia de la observabilidad

- Lanzamiento inicial no satisface todas las necesidades de los usuarios
- Limitaciones al enfrentar preguntas complejas. (¿Por qué va tan lento?)
- Honeycomb implementa funcionalidad bubble up

Importancia de entender los objetivos y necesidades de los usuarios

- Adivinar el objetivo del usuario para dar la respuesta que busca
- Entregar valor de forma gradual
- Casos no abordados (Preguntas sin respuesta)
- Nuevas funcionalidades de IA

Observabilidad en Modelos de Lenguaje y su Impacto en la Ingeniería de Indicaciones

Capacidad de recopilar, analizar y comprender señales que afectan el comportamiento del sistema.

Indicaciones de usuarios

Datos de entrada

Respuestas generadas

Factores contextuales

Impacto en la Ingeniería de Indicaciones:

Generación Programática: Indicaciones dinámicas, personalización y adaptación en tiempo real.

Análisis: Evaluación de entradas y salidas para precisión y coherencia.

Mejora Continua: Ajuste y optimización mediante métricas de observabilidad.

Implementación de Estrategias de Observabilidad

Herramientas Clave:

Registros Estructurados: Capturan información detallada de solicitudes y respuestas, facilitando auditorías y depuración.

Telemetría Abierta: Proporciona métricas y datos de rastreo estandarizados, permitiendo correlacionar eventos y analizar patrones.

Mapeo de Procesos: Visualiza la ejecución de funciones, identificando cuellos de botella y puntos de falla en el sistema.

Gestión de Errores y Desafíos en la Observabilidad

Errores Comunes:

Tiempos de Espera Prolongados: Retrasos que afectan la experiencia del usuario.

Interrupciones del Servicio: Caídas que impactan la disponibilidad del sistema.

Errores Sutiles:

- Estructuras JSON Incompletas.
- Incoherencias Semánticas.
- Interpretaciones Erróneas del Contexto.

Principales Desafíos:

Automatización Limitada: Instrumentación en bases de datos y frameworks de IA en desarrollo.

Manejo de Datos Complejos: Dificultad en la agregación y análisis de entradas y salidas diversas.

Falta de Estándares: Ausencia de métricas uniformes para evaluar la efectividad de las indicaciones

Perspectivas Futuras de la Observabilidad en Modelos de Lenguaje

- Instrumentación Automática: Mejora en la captura y análisis de datos en tiempo real.
- Manejo de Alta Cardinalidad: Técnicas avanzadas para patrones más precisos.
- Estándares en Ingeniería de Indicaciones: Optimización de procesos y mejores prácticas.
- Integración de Aprendizaje Automático: Detección automática de problemas y optimización continua.