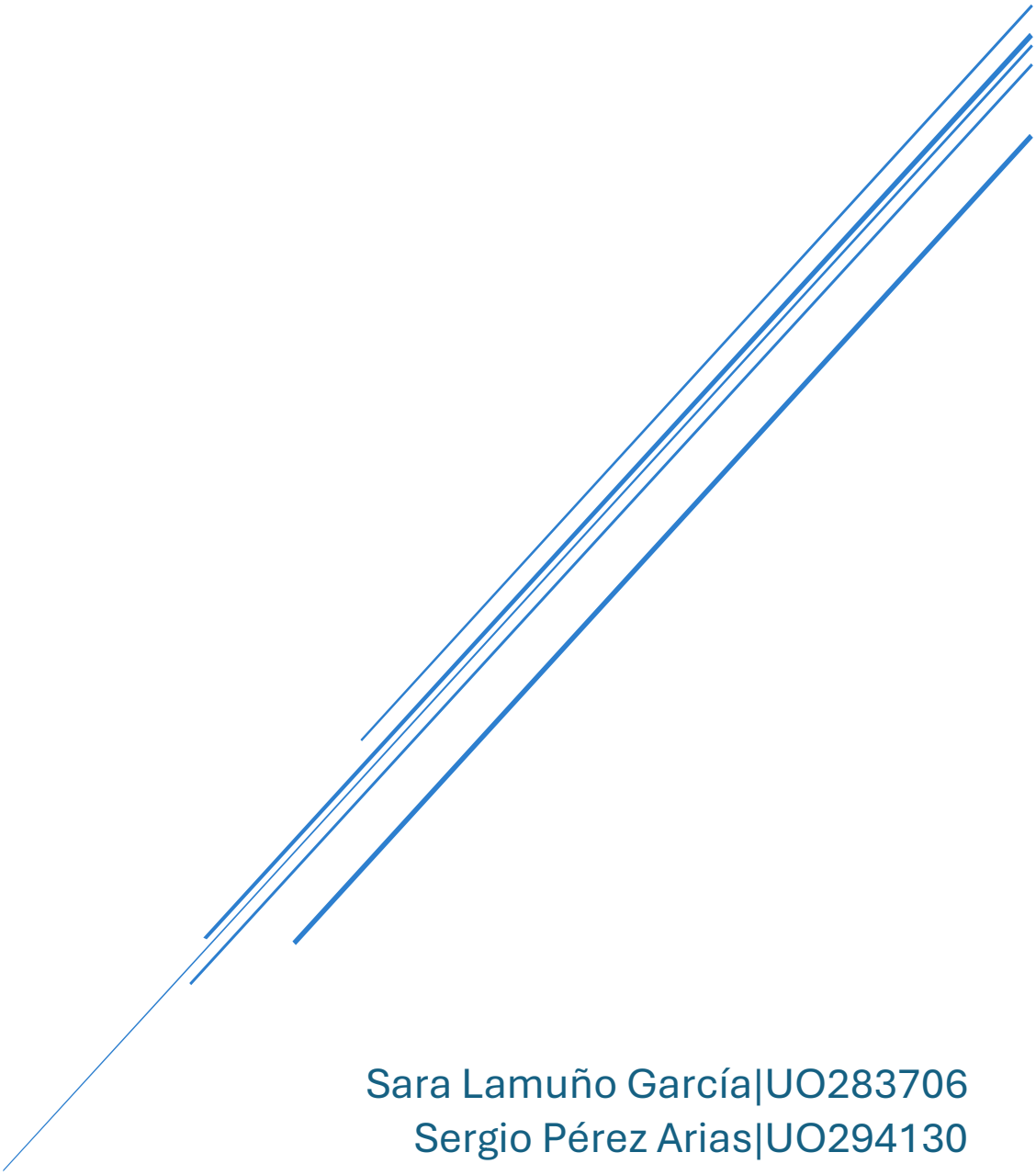
PHILLIP CARTER ON OBSERVABILITY FOR LARGE LANGUAGE MODELS

SE RADIO 610



Sara Lamuño García|UO283706
Sergio Pérez Arias|UO294130
Andrea Acero Suárez | UO287876
Arquitectura de Software

Observabilidad en Sistemas

Concepto de observabilidad es frecuentemente malinterpretado en la industria, ya que cada empresa que vende herramientas de observabilidad lo define de manera distinta. Se trata de comprender el estado de un sistema sin modificarlo directamente.

- **Problema:** Cuando un error o comportamiento inesperado no puede reproducirse localmente debido a múltiples factores (como interconectividad de sistemas).
- **Solución:** Recopilar señales para analizar el estado del sistema en producción, permitiendo responder dos preguntas clave:
    1. ¿Dónde está ocurriendo el problema?
    2. ¿Por qué está ocurriendo?

No solo es una herramienta reactiva para resolver incidentes, sino que también debe integrarse en el desarrollo de software para prevenir problemas antes de que lleguen a producción. Grandes empresas como Google y Facebook han trabajado con estos principios desde hace tiempo, pero ahora se están adoptando en toda la industria.

Introducción a los Modelos de Lenguaje (LLMs)

1. Definición práctica:

   Son sistemas que reciben texto de entrada y generan respuestas basadas en un gran volumen de información almacenada.

2. Definición técnica:

   Basados en la arquitectura Transformer y el concepto de atención. Resuelven un problema clave en el procesamiento del lenguaje natural: comprender cómo diferentes partes de un texto se relacionan entre sí para generar respuestas más precisas.

Fine-tuning

Los modelos de lenguaje pasan por varias fases antes de estar listos para su uso. Primero, se entrenan con grandes volúmenes de datos, proceso que puede dividirse en preentrenamiento y entrenamiento. Luego, en la fase de alineación, se ajusta el modelo para que sus respuestas sean correctas, evitando, por ejemplo, generar contenido dañino.

El fine-tuning entra en juego cuando se quiere especializar un modelo en un dominio específico. A diferencia de la alineación, que busca ajustes generales, el fine-tuning permite adaptar el modelo a un conjunto de datos propio. Este proceso tiene riesgos: si se ajusta demasiado a un contexto, el modelo puede volverse menos flexible y no responder bien fuera de ese ámbito.

Prompt Engineering

El prompt engineering se describe como el arte de diseñar las instrucciones correctas para obtener respuestas adecuadas del modelo. Carter lo compara con escribir consultas SQL para bases de datos: así como una consulta bien formulada mejora la eficiencia de una base de datos, un buen prompt optimiza la respuesta del modelo.

Aunque las indicaciones parezcan claras, el modelo puede entenderlas de forma inesperada. Por eso, el prompt engineering implica creatividad y refinamiento. Además, puede incluir técnicas como el Retrieval-Augmented Generation (RAG). Carter destaca que, debido a su

complejidad, esta disciplina debería considerarse parte de la ingeniería de IA, ya que abarca más que solo la formulación de instrucciones y también influye en cómo se diseñan los sistemas basados en modelos de lenguaje.

Similitudes observabilidad para LLM y sistemas “convencionales”

Al igual que en bases de datos, la forma en que se parametrizan las consultas puede afectar los resultados, lo que resalta la importancia de la latencia.
En LLMs, la lentitud puede deberse a decisiones de implementación, no solo al modelo en sí.

Desafíos que la Observabilidad en LLMs ayuda a resolver

Latencia: Es crucial identificar las causas de la latencia, que pueden incluir la formulación del prompt y la cantidad de salida generada. Técnicas como "chain of thought prompting" pueden mejorar la precisión, pero aumentan la latencia.

Desarrollo impulsado por la observabilidad: Se busca eliminar la barrera entre desarrollo y producción, enfocándose en problemas que afectan a los usuarios. La observabilidad permite recopilar datos sobre el comportamiento del sistema, facilitando la identificación de patrones y errores.

Enfoque de desarrollo de los LLMs antes de pasar a producción

Las empresas enfrentan desafíos al adoptar nuevas tecnologías, ya que las herramientas y procesos convencionales no siempre son aplicables. Es necesario involucrar a los usuarios y adaptarse a nuevas prácticas, como la trazabilidad.

Incrementalidad y los lanzamientos rápidos

La capacidad de realizar despliegues diarios es esencial para reaccionar a los cambios en el comportamiento de los usuarios. Un ciclo de lanzamientos y monitoreo permite equilibrar confiabilidad y creatividad en las salidas del sistema.

Importancia de la observabilidad

El lanzamiento inicial de un producto puede no satisfacer todas las necesidades de los usuarios. A través de iteraciones, se puede alcanzar un estado estable con ciertas características, que, aunque probablemente sea bueno en muchas áreas, tendrá limitaciones al enfrentar preguntas complejas, que el sistema no está preparado para responder como “¿por qué va tan lento?”. Para abordar estas limitaciones, Honeycomb ha implementado una funcionalidad llamada “bubble up”, que permite identificar patrones y ofrecer soluciones, mejorando el producto continuamente.

Importancia de entender los objetivos y necesidades de los usuarios

Entender los objetivos y necesidades de los usuarios es esencial para desarrollar sistemas de inteligencia artificial (IA) efectivos. Muchos usuarios fracasan al crear

soluciones personalizadas debido a la complejidad y falta de experiencia. Es fundamental adoptar un enfoque incremental, entregando valor gradualmente y

adaptándose a nuevas necesidades. Al introducir funcionalidades de IA, las expectativas pueden variar, lo que puede generar sorpresas y nuevas demandas, complicando su integración en el producto.

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

La observabilidad en modelos de lenguaje permite comprender el comportamiento del sistema mediante el análisis de señales como las indicaciones del usuario, los datos de entrada, el contexto y las respuestas generadas. Su propósito es optimizar la precisión y relevancia de las respuestas a través de un proceso iterativo de ajuste basado en datos reales de uso.

Elementos Clave:

La *generación programática de indicaciones* se realiza dinámicamente, utilizando estructuras predefinidas y datos parametrizados que permiten personalizar las respuestas en tiempo real según el contexto del usuario. El *análisis de entradas y salidas* implica evaluar tanto la cadena estructurada de entrada como la coherencia y alineación de la salida con la intención del usuario. Finalmente, la *iteración y mejora continua* se basa en métricas de observabilidad que permiten refinar las indicaciones y elevar la calidad de las respuestas del modelo.

Implementación de Estrategias de Observabilidad

Los desarrolladores pueden mejorar la observabilidad en modelos de lenguaje mediante herramientas como registros estructurados, que permiten almacenar y auditar solicitudes y respuestas, facilitando la depuración de problemas. La telemetría abierta estandariza la recopilación de métricas y datos de rastreo, ayudando a identificar patrones de comportamiento del sistema. Además, el mapeo de procesos e intervalos en rastreos permite visualizar la ejecución de funciones clave, facilitando la detección de cuellos de botella y fallos en el procesamiento.

Gestión de Errores y Desafíos en la Observabilidad

El monitoreo de errores es clave para mantener la confiabilidad del sistema, abarcando tanto fallos críticos como errores sutiles que afectan la calidad de las respuestas. Los principales desafíos incluyen la automatización limitada en herramientas específicas, el manejo de datos complejos y la falta de estándares claros para evaluar indicaciones, lo que complica el análisis y comparación de resultados.

Perspectivas Futuras de la Observabilidad en Modelos de Lenguaje

Las perspectivas futuras de la observabilidad en modelos de lenguaje incluyen mejoras en herramientas de instrumentación automática, técnicas avanzadas para manejar datos complejos, la definición de estándares para ingeniería de indicaciones y la integración de modelos de aprendizaje automático en el monitoreo.