Observability for LLMs

Autores

- Natalia Blanco Agudín (UO295340)
- Claudia Rodríguez Fuertes (UO288406)
- Javier Sanabria Miranda (UO293758)

Definición de Observabilidad

La observabilidad se define mejor por los problemas que resuelve. Cuando un error es difícil de reproducir o implica múltiples factores, el debugging tradicional no es suficiente.

La observabilidad permite entender qué está pasando mediante la recopilación y análisis de datos (telemetría) de distintos escenarios de la aplicación.

Aunque existe una definición académica que la describe como "*entender el estado de un sistema sin modificarlo*", el autor considera más útil enfocarse en su aplicación práctica.

Definición de LLM

Es como una caja negra a la que le envías texto y que tiene un montón de información comprimida dentro que le permite analizar dicho texto y llevar a cabo acciones con instrucciones que previamente se le han dado, de modo que puede generar texto en un formato específico que contenga la información que estás buscando

Con esto como base, hay muchas cosas que se pueden hacer, hay modelos de lenguaje que son mejores para poesía, código...

Definición de “Fine Tuning”

El fine-tuning es una fase de especialización en los modelos de lenguaje (LLM). Antes de él, el modelo pasa por dos etapas:

1. **Entrenamiento:** Se entrena con un gran corpus de texto para aprender a operar con el lenguaje.
2. **Alineación:** Se ajusta para que sus respuestas sean seguras y alineadas con los principios del sistema, evitando contenido dañino.

Luego, el **fine-tuning** permite especializar el modelo en un dominio específico mediante datos propios, mejorando su precisión en ciertos casos, aunque con la posible pérdida de flexibilidad en otros temas

Definición de “Prompt Engineering”

Los prompts en los modelos de lenguaje son como consultas SQL en bases de datos: ambos son instrucciones diseñadas para obtener respuestas específicas. Sin embargo, cada modelo puede interpretar el mismo prompt de manera diferente, por lo que a veces es necesario ajustarlo.

Un LLM es comparado con un niño pequeño que no siempre entiende las instrucciones como se espera, por lo que se requiere claridad y creatividad al formular los prompts.

Observabilidad en el Contexto de los LLM

La observabilidad es fundamental en sistemas que utilizan LLMs, ya que estos son no reproducibles, no depurables y no deterministas en sus respuestas. A diferencia del software tradicional, donde los errores pueden diagnosticarse con pruebas unitarias, en los LLMs los usuarios interactúan de forma impredecibles, lo que hace imposible prever todas las entradas.

Dado que modificar o reentrenar un modelo es costoso y complejo, la observabilidad permite entender su comportamiento sin necesidad de cambiarlo. Para ello, es clave recopilar señales antes y después de la llamada al modelo, identificando los factores que afectan la calidad del output.

Similitudes en la Observabilidad de LLMs y la Observabilidad de Software Convencional

La observabilidad en el rendimiento de los LLMs es clave, ya que los problemas no siempre provienen del modelo, sino de factores externos, al igual que ocurre en bases de datos, una consulta puede dar un resultado incorrecto debido a decisiones previas que afectaron los parámetros usados.

Aunque los LLMs tienen una latencia inherente, otros problemas del sistema pueden agravar la percepción de lentitud. La observabilidad permite medir la latencia real diferenciar entre fallos del modelo y errores en la infraestructura.

Desafios en la Adopción de los LLM en Organizaciones Tradicionales

Los equipos que intentan trabajar con LLMs en organizaciones tradicionales pronto descubren que sus herramientas convencionales de QA, como las pruebas unitarias e integración, no son efectivas. En lugar de confiar en estas pruebas

tradicionales, los equipos se dan cuenta de que deben capturar datos de usuarios reales para comprender cómo el sistema realmente se comporta en producción.

Importancia de las Fast Releases para los LLMs

La capacidad de realizar lanzamientos diarios es crucial para el éxito con modelos de lenguaje, ya que los patrones de comportamiento de los usuarios cambian constantemente y es necesario reaccionar rápidamente para adaptarse a estos cambios. Este enfoque permite crear un ciclo en el que se observan los patrones, se identifican los problemas, se resuelve uno y luego se despliega para verificar si el problema se solucionó sin causar regresiones, es decir, sin haber perdido funcionalidad o eficiencia en otras tareas del sistema.

Como usar la información obtenida para mejorar el LLM

A medida que los usuarios interactúan con el sistema, surgirán limitaciones fundamentales. En este caso, los usuarios pueden preguntar cosas que no pueden ser respondidas directamente con el sistema actual, lo que requiere ajustes y mejoras continuas.

Importancia del Seguimiento de Errores en los LLMs

Los errores afectan la confiabilidad del sistema y pueden clasificarse en diferentes tipos.

Fallos críticos antes de la llamada al modelo: El programa se bloquea antes de invocar al modelo. Solución: Implementar **reintentos** o usar un modelo de respaldo

Errores Corregibles: Puede ser que el JSON de output del LLM esté incompleto por haber limitado sus tokens. Se puede solucionar aumentando ese límite de tokens o ajustando el prompt. También puede ser que la estructura sea incorrecta, que el JSON tenga un campo mal nombrado o falte un dato esencial. Se puede solucionar con validación y corrigiendo el dato automáticamente en caso de ser posible.

Límites de la Observabilidad en LLMs

Falta de reconocimiento de patrones automático: La identificación de tendencias en entradas y salidas es manual, ya que no existen herramientas eficientes para analizar grandes volúmenes de datos.

La observabilidad es un proceso iterativo y complejo: No se puede activar la observabilidad y esperar resultados inmediatos; requiere ajustes constantes, prueba y error, y selección adecuada de métricas.

No existen mejores prácticas ni estándares: No hay estándares universales por lo que cada equipo debe desarrollar su propio enfoque a través de la experiencia y la experimentación.

Mejoras en la Observabilidad de LLMs en los Próximos Años

Instrumentación automática: Actualmente, herramientas como OpenTelemetry no ofrecen auto-instrumentación para modelos de lenguaje, pero se espera que en el futuro lo integren.

Mejor análisis de datos de texto: Los modelos de lenguaje generan entradas y salidas con alta cardinalidad (gran variedad de valores únicos), lo que dificulta el análisis y aumenta costos.

Mejorar la capacidad de manejar datos altamente variables permitirá obtener información útil sin afectar el rendimiento ni los costos.