Transformers and Large Language Models - Winter School - PUCP

Parte III: Large Language Models

Sebastián Vallejo Vera [1]

August 18, 2025

[1] Dept. of Political Science – Western University

Índice

Transformers en Decoders

Generative Large Language Models

- Los Large Language Models (LLM) son decodificadores (decoders) preentrenados. Esto significa que han sido preentrenados usando la arquitectura Transformers para generar texto nuevo a partir de un prompt dado.
- En su explicación más básica, los LLMs destacan en la predicción del siguiente token. Tan bien, que la gente (erróneamente, en mi opinión) confunde las respuestas de LLM con comportamientos similares a los humanos.
- Los LLMs SOTA a agosto de 2025 son: GPT-5algo (OpenAI), Llama4 algo (Meta), DeepSeek RAlgo, Mistral 3algo, Claude 4algo, Qwen (?)... es una industria que recuerda a la burbuja puntocom (otra vez, en mi opinión). Para cuando releas esto, estará desactualizado.
- La estructura es bastante sencilla: (pre-prompt) $\Rightarrow$ prompt $\Rightarrow$ output de LLM $\Rightarrow$ prompt $\Rightarrow$ output de LLM, etc.

¿Por Qué Son Tan Buenos los LLMs? ¡

- Los LLMs se entrenan usando una cantidad inmensa de datos textuales que ~~robaron~~ recopilaron de internet. Por ejemplo, GPT-3 (modelo desactualizado) fue entrenado con 500 mil millones de tokens y 175 mil millones de parámetros.
- Muchos de estos modelos son propietarios, así que tenemos pocas pistas sobre por qué son tan buenos. Incluso los que dicen ser open-source (como Llama), no lo son realmente. La ética de las empresas que desarrollan LLMs es, en mi opinión, cuando menos cuestionable (pero seguimos usando estos modelos, así que es un ciclo de hipocresía, como es la norma en esta etapa avanzada del capitalismo).
- Más importante aún, aunque existiera un LLM verdaderamente open-source, sería casi imposible para ti o para mí revisar su código y entender qué los hace funcionar.

¿Por Qué Son Tan Buenos los LLMs? ii

- Más allá del hecho de que tienen un montón de datos que transforman (en redes tensoriales) y nos devuelven (lo que seguramente hace que imiten muy bien a las personas que generaron esos datos), un mecanismo que parece estar en juego es el in-context learning.
  - Durante el preentrenamiento, los LLMs desarrollan ciertas 'habilidades', entre ellas el reconocimiento de patrones. Cuando se les da un prompt, reconocen rápidamente la tarea requerida–es decir, in-context learning–y adaptan su salida en respuesta a esto.
  - Por qué sucede esto aún no está claro.
- También existe el contextual learning, una actualización de pesos condicional a los tokens usados en el prompt y la respuesta.
- Combinados, parecen ser parte de la fuerza motriz que produce el tipo de resultados que obtenemos.

LLM en las Ciencias Sociales

- Los LLMs se utilizan cada vez más en las ciencias sociales.
- Objetivo: generar datos a partir de texto (predicción del siguiente token); medir alguna característica latente de un conjunto de textos.
  - Aplicado: dimensiones latentes, es decir, ideología (Kato and Cochrane, 2025; Wu et al., 2023); anotadores (Timoneda and Vallejo Vera, 2025)
  - Limitaciones: sesgos por señales partidistas (Vallejo Vera and Driggers, 2025); sesgo por idioma (Walker and Timoneda, 2024).
  - Meta: efecto de los LLM en usuarios.

Aplicaciones y Limitaciones de los LLM

Aplicaciones de los LLM

- Con los LLM, lo que los investigadores pueden hacer está realmente limitado por la creatividad. Proporcionaré la estructura básica de código para acceder a los LLM mediante una API, y para obtener y organizar datos, pero la mayoría de las respuestas vendrán de los prompts. Así que... ¿ingeniería de prompts? (No me gusta este término ni esta práctica, pero existe).
- Hay algunos enfoques que mejoran las respuestas, y hay algunas buenas prácticas a considerar al usar LLM. Los cubriré tanto en las diapositivas como con ejemplos de implementación en el código.

¡Al código!

Tareas de Clasificación con LLM i

- Una de las aplicaciones más comunes de los LLM es la clasificación de texto (anotación).
  - Pedir una lista de temas a partir de conjuntos de textos (no supervisado).
  - Pedir una clasificación específica de un conjunto de opciones (supervisado).
- Enfoques para clasificar texto:
  - Aprendizaje zero-shot: solo instrucciones en el prompt, sin ejemplos.
  - Aprendizaje few-shot: instrucciones + ejemplos (entre 5 y 10).
  - Few-shot con *chain-of-thought reasoning* (CoT): Pedir al modelo que clasifique un ejemplo + pedir el razonamiento detrás de la clasificación (en pasos si es necesario) + añadir (texto del ejemplo + clasificación + razonamiento) al prompt.

Tareas de Clasificación con LLM ii

- Few-shot CoT mejora el rendimiento pero tiene un problema importante: los ejemplos no se generalizan bien a textos nuevos no vistos, las ganancias son limitadas, y más ejemplos no mejoran el rendimiento (*overfitting*).
- (Se viene más autopromoción) Nosotros (Timoneda and Vallejo Vera, 2025) proponemos un nuevo enfoque llamado aprendizaje con memoria:
  - Permitimos que el modelo aprenda de sus propias clasificaciones previas, lo que mejora el rendimiento en clasificación supervisada.
  - Similar a cómo los anotadores humanos codifican datos–aprendemos sobre la marcha.
  - Más técnico: Los modelos ya destacan con 'in-context learning' y usan 'contextual memory', así que podemos maximizar el aprendizaje agregando más memoria.

¡Al código!

Buenas Prácticas al Usar LLM i

(Nota que estas son mayormente de mi propia investigación y aplicación)

- Dado que los LLM no son determinísticos en sus respuestas, deberías ejecutar múltiples iteraciones de cada tarea de anotación. La consistencia interna variará según el modelo y los hiperparámetros usados.
- Las buenas prácticas al usar LLM como anotadores incluyen múltiples iteraciones de la tarea, reportar todas las respuestas de los LLM, y explicar claramente la estrategia de adjudicación empleada. Esto aumenta la transparencia y permite la replicación.

Buenas Prácticas al Usar LLM ii

- Aún así, el entorno en constante cambio de los LLM hace que la replicación sea complicada y, con el tiempo, imposible. Los investigadores deben reportar claramente la versión del modelo usada y, si es posible, la plataforma en la que fue desplegado (o si fue desplegado localmente).
- Mi recomendación final vuelve al principio clave del análisis de texto de Grimmer and Stewart (2013): "validar, validar, validar."

Limitaciones al Usar LLM i

(Nota que estas también son mayormente de mi propia investigación y aplicación)

- Los investigadores deben tener en cuenta que los LLM no se crean en el vacío. Son el producto parcial de datos generados por humanos, que están impregnados de sesgos humanos.
- Por lo tanto, los resultados imitarán sesgos humanos, pero de una forma muy LLM (ver Vallejo Vera and Driggers, 2025).
- En general, el fine-tuning de un modelo Transformers es todavía más preciso para la clasificación de texto que un LLM (porque aprovechamos toda la estructura del encoder, que podemos adaptar más fácilmente).

Limitaciones al Usar LLM ii

- Los modelos se entrenan en idiomas con muchos recursos... mayormente inglés. Son excelentes en tareas en inglés. Pero la mayoría del mundo no habla inglés.
- Finalmente, las consideraciones más amplias sobre clase/centro-periferia/imperio de acceso al conocimiento y recursos se aplican a los LLM (y al análisis computacional de texto en general).

¿Preguntas? ¿Comentarios? ¿Despedidas?

References

Grimmer, Justin and Brandon M Stewart. 2013. "Text as data: The promise and pitfalls of automatic content analysis methods for political texts." *Political analysis* 21(3):267–297.

Kato, Ken and Christopher Cochrane. 2025. "KOKKAI DOC: An LLM-driven framework for scaling parliamentary representatives." *arXiv preprint arXiv:2505.07118* .

Timoneda, Joan C and Sebastián Vallejo Vera. 2025. "Memory Is All You Need: Testing How Model Memory Affects LLM Performance in Annotation Tasks." *arXiv preprint arXiv:2503.04874* .

Referencias ii

Vallejo Vera, Sebastián and Hunter Driggers. 2025. "Bias in LLMs as Annotators: The Effect of Party Cues on Labelling Decision by Large Language Models." *arXiv preprint arXiv:2408.15895* .

Walker, Christina and Joan C Timoneda. 2024. "Identifying the sources of ideological bias in GPT models through linguistic variation in output." *arXiv preprint arXiv:2409.06043* .

Wu, Patrick Y, Jonathan Nagler, Joshua A Tucker and Solomon Messing. 2023. "Large language models can be used to estimate the latent positions of politicians." *arXiv preprint arXiv:2303.12057* .