Advanced Text-As-Data - Winter School - Iesp UERJ

Day 4: Large Language Models

Sebastián Vallejo Vera [1]

July 7, 2025

[1] Dept. of Political Science – Western University

Table of Contents

Transformers in Decoders

Generative Large Language Models

- Large Language Models (LLM) are pre-trained decoders. This means models they have been pre-trained using the Transformers architecture to generate new text from a given prompt.
- In their most basic understanding, LLMs excel at next work prediction. So good, that people (erroneously, in my view) confound LLM responses with human-like behavior.
- The SOTA LLM as of July 2025 are: GPT-4something (OpenAI), Llama4 something (Meta), DeepSeek RSomething, Mistral 3something, Claude 4something, Qwen (?)... it's an industry reminiscent of dot.com bubble (again, in my view). By the time you re-read this, it will be outdated.
- The structure is pretty straightforward: (pre-prompt) $\Rightarrow$ prompt $\Rightarrow$ LLM output $\Rightarrow$ prompt $\Rightarrow$ LLM output etc.

Why Are LLMs So Good, Though? i

- LLMs are trained using gazillions of text data they ~~stole~~ scraped from the internet. For example, GPT-3 (outdated model) was trained using 500 billion tokens and 175 billion parameters.
- Many of these models are proprietary, so we have few clues as to way they are so good. Even the ones that claim to be open-source (like Llama), are not really open-source. The ethical standing of companies developing LLMs is, in my view, is tenuous at best (but we still use these models, so its a cycle of hypocrisy, as is the norm in this late stage of capitalism).
- More importantly, even if there is a truly open-source LLM out there, it would be close to impossible for you or me to actually peruse their code and figure out what is making them work.

Why Are LLMs So Good, Though? ii

- Beyond the fact that they have troves of data which they transform (into tensor networks) and spew back at us (which is bound to make things pretty good at mimicking the people producing that data), one mechanism that seems to be at work is in-context learning.
  - During pre-training, LLMs develop certain 'skills', among them pattern recognition. When prompted, LLMs rapidly recognize the required task–i.e., in-context learning– and adapt their output accordingly.
  - Why this happens is still unclear.
- There is also contextual learning, an updating of weights conditional on the tokens used in the prompt and response.
- Combined, they seem to be part of the driving force that produce the type of results we obtain.

LLM in the Social Sciences

- LLMs have been increasingly used in the social sciences.
- Goal: generate data from text (next token prediction); measure some latent characteristic of a set of texts.
  - Applied: latent dimensions, i.e., ideology (Kato and Cochrane, 2025; Wu et al., 2023); annotators (Timoneda and Vallejo Vera, 2025)
  - Limitations: biases from party cues (Vallejo Vera and Driggers, 2025); bias from language (Walker and Timoneda, 2024).
  - Meta: effect of LLMs on respondents.

Applications and Limitation of LLMs

Applications of LLM

- With LLMs, what researchers can do is really limited by creativity. I will provide the basic code structure to access LLMs through an API, and obtaining and tidying data, but most of the responses will come from prompts. So... prompt engineering? (I do not like this term or this practice but it's a thing).
- There are some approaches that improve responses, and there are some best practices to consider when using LLMs. I will cover them both in the slides and provide examples of how to implement these in the code.

To the code!

Classification Tasks with LLM i

- One of the most common applications of LLMs is text classification (annotation).
  - Ask for a list of topics from sets of texts (unsupervised).
  - Ask for a specific classification from a set of options (supervised).
- Approaches to classifying text:
  - Zero-shot learning: Only instructions in the prompt, no examples.
  - Few-shot learning: Instructions + examples (between 5 and 10).
  - Few-shot with chain-of-thought (CoT) reasoning: Ask model to classify example + ask reasoning behind classification (in steps if necessary) + add (example text + classification + reasoning) to the prompt.
- Few-shot CoT improves performance but has a major problem: examples do not generalize well to unseen new texts, gains are limited, and more examples do not improve performance (overfitting).

Classification Tasks with LLM ii

- (More self-promotion coming up) We (Timoneda and Vallejo Vera, 2025) propose a new approach called memory learning:
  - We allow the model to learn from its own previous classifications, which should improve performance in supervised classification.
  - Similar to how human annotators code data – we learn as we go.
  - More technical: Models already excel with 'in-context learning' and use 'contextual memory', so we can maximize learning by adding more memory

To the code!

Best Practices when Using LLM i

(Note that these are mostly from my own research and application)

- Since LLMs are non-deterministic in their answers, you should run multiple iterations of each annotation task. Internal consistency will vary across models and hyperparameters used.
- Best practices when using LLMs as annotators include multiple iterations of the task, reporting on all the output from LLMs, and clearly explaining the adjudication strategy employed. This increases transparency and allows for replication.
- Still, the constantly-changing LLM environment makes replication complicated and, with time, impossible. Researchers should clearly report the version of the model used and, if possible, the platform on which it was deployed (or if it was locally deployed).

Best Practices when Using LLM ii

- My final recommendation goes back to Grimmer and Stewart (2013) key principle of text analysis more generally: "validate, validate, validate."

Limitations when Using LLM i

(Note that these are also mostly from my own research and application)

- Researchers should take into account that LLMs are not created in a vacuum. They are the partial product of human-generated data, that is embedded with human biases.
- Thus, results will mimic human biases, but in a very LLM way (see Vallejo Vera and Driggers, 2025).
- Fine-tuning a Transformers-based model will still be more accurate for text classification than using a LLM (because we take advantage of the full encoder structure, which we can more easily adapt).

Limitations when Using LLM ii

- Models are trained in high-resource languages... mostly English. They are great at English language tasks. But most of the world is not in English.
- Finally, the broader class/core-periphery/empire considerations of access to knowledge and resources are applied to LLM (and computational text-analysis more broadly).

Questions? Comments? Good-Byes?

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* .

References 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* .