Investigating Bias in LLM Self-Evaluation

Thesis

Mathematics Expert in Data Analytics and Machine Learning

ELTE | FACULTY OF SCIENCE

Eötvös Loránd University
Faculty of Science

Budapest, 2025

Contents

1 Introduction

1.1 A Brief Introduction to LLMs

A language model is a machine learning model designed to perform a wide range of tasks that involve natural language processing (NLP), including text summarization, translation, sentiment analysis, spam detection, content moderation, text generation, etc.

Significant advancements in deep learning [1, 2, 3] led to the emergence of **large language models** (LLMs) — particularly generative LLMs — which in the early 2020s became productized and widely adopted in both industry and popular discourse.

A generative large language model is a model which has a parameter count on the order of billions or more, and predicts the conditional probability [4]

$$P(w_m|w_0, \cdots, w_{m-1}) \tag{1}$$

where $m \in \mathbb{N}$, w_0 is a special start symbol, and w_k is the k-th token (for $1 \leq k \leq m$) in a sequence of tokens that form a piece of text in some (natural) language. The interpretation of the tokens depends on the exact tokenization strategy used, which may define tokens as words, word pieces, n-grams, or individual characters, and spaces, punctuation marks, etc.

Text generation then is an autoregressive process where given a sequence of tokens as a prefix — known as the **prompt** — the model estimates the probability distribution of the next token, takes a sample from that distribution, appends it to the sequence, and repeats the process with the extended sequence until a given stopping condition is met.

The sampling can usually be controlled via numerical parameters; a commonly used one is the **temperature** [5]: the closer it is to 0, the more the sampling will lean toward the most probable token — making the algorithm more deterministic —, while higher values increase the randomization, making the generated text feel more *creative* until, above a certain threshold, it becomes incoherent and semantically meaningless. If $v \in \mathbb{N}$ denotes the number of all possible tokens available for the model (vocabulary size), and $\mathbf{s} \in \mathbb{R}^v$ is an output vector of the model assigning a score to each token as the continuation of a given input, then the distribution for the sampling, with respect to the temperature $T \in \mathbb{R}$ is calculated via the softmax function:

$$\operatorname{softmax}(\mathbf{s}; T) = \left[\frac{\exp(\frac{s_i}{T})}{\sum_{j=1}^{v} \exp(\frac{s_j}{T})} \right]_{i=1}^{v} \tag{2}$$

In practice, if T is sufficiently close or exactly equal to 0, then the sampling is usually replaced with the deterministic argmax function in order to preserve numerical stability. Non-zero T values control the flatness of the distribution, leading to the aforementioned behavior.

With sufficiently large model complexity and training corpora size and diversity, LLMs start to exhibit capabilities which rival that of top performer humans in a broad class of problems [2, 3]. The versatility of the models is often utilized in a setting where the prompt is composed of two parts, each consisting of instructions given in natural language:

- the **system prompt** can instruct the model to behave in a certain way, for example, to act like a helpful AI assistant, an expert in a domain, or to generate its texts in the style of fictional 18th-century Caribbean pirates,
- and the **user prompt** which describes a task to be carried out by the model, ranging from text translation or summarization to solving complex programming problems or pointing out business risks in legal documents, and more.

Generative models with sufficient generalization capabilities can predict likely continuations of such prompts with so high accuracy that the generated text will often contain an actual solution to the proposed problem. This instruction-following paradigm enables models to perform **few-shot learning** [2] or even **zero-shot**

learning by interpreting tasks directly from the natural language description, based on just a few or zero examples, respectively, without specific training or fine-tuning.

The problem solving performance of LLMs can be improved further by prompt engineering techniques like **chain-of-thought** prompting [6], where the model is provided with a step-by-step example solution to a related problem in the prompt, encouraging it to also articulate intermediate reasoning steps.

1.2 LLM Evaluators, LLM-as-a-Judge

References

[1] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). DOI: https://doi.org/10.48550/arXiv.1706.03762.

[2] OpenAI. "Language Models are Few-Shot Learners". In: *CoRR* abs/2005.14165 (2020). DOI: https://doi.org/10.48550/arXiv.2005.14165.

[3] OpenAI. "GPT-4 Technical Report". In: (2023). DOI: https://doi.org/10.48550/arXiv.2303.08774.

[4] Tong Xiao and Jingbo Zhu. *Foundations of Large Language Models*. 2025. DOI: https://doi.org/10.48550/arXiv.2501.09223.

[5] Enrique Manjavacas et al. "Synthetic Literature: Writing Science Fiction in a Co-Creative Process". In: *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*. Santiago de Compostela, Spain: Association for Computational Linguistics, Sept. 2017, pp. 29–37. DOI: https://doi.org/10.18653/v1/W17-3904.

[6] Jason Wei et al. "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models". In: *CoRR* abs/2201.11903 (2022). DOI: https://doi.org/10.48550/arXiv.2201.11903.