DIVING DEEP: HOW DO LLMS LEARN ON THE FLY?

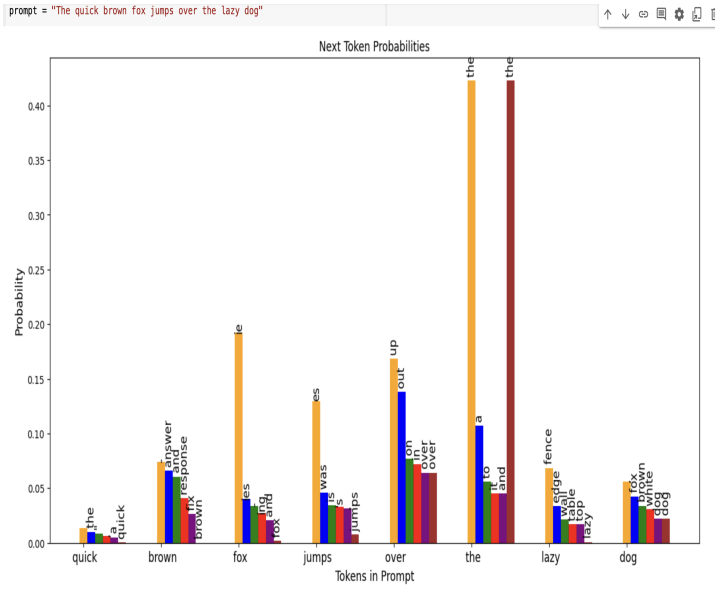
Charlie Kerfoot[1,†], Rashfiqur Rahman[2,†], Amy Wu[3,†], Andrew Tang[4] and Vishal Misra[4]

[1]*Horace Mann School, New York – <charlie_kerfoot@horacemann.org>*
[2]*Boston University, Boston – <rash@bu.edu>*
[3]*University of Florida, Florida – <amy.wu@ufl.edu>*
[4]*Columbia University, New York – <at3456@columbia.edu,vishal.misra@columbia.edu>*
[†]*These authors contributed equally to this work*

1. INTRODUCTION

Large Language Models (LLMs) have revolutionized natural language processing, demonstrating remarkable capabilities in text generation and understanding. One of the many phenomenons in LLMs is in-context learning, where task-specific responses are generated by providing the model with task-specific prompts [1]. This study investigates the behavior of LLMs through the lens of Bayesian learning, where the model constantly updates its token generating distribution based on new evidence. To further explore this topic, we propose a practical open-source tool for visualizing token probabilities during text generation, providing empirical insight into the models' next token prediction process. This lets us analyze how the model is constantly learning in real-time.

Figure 1. Example of token probability distribution given a prompt



2. METHODS

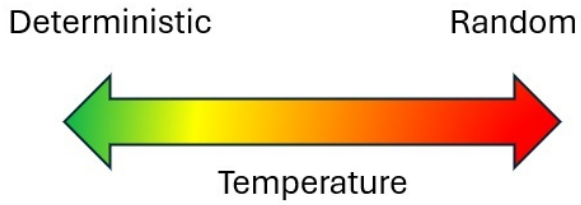
To retrieve token probabilities within prompts, we tokenized the prompt and performed a single forward pass to obtain logits for each token. We then applied the softmax function to these logits to derive the probabilities of the next token. To investigate in-context learning, we designed prompts that tested the model's ability to adapt to new patterns or information within the input sequence. By analyzing the probabilities in both the prompt, and its completion, this approach allowed us to observe how the probabilities of each token affect the model generation through output manipulation [2]. We utilized both OpenAI GPT-2 and Meta LLaMA-3 models to investigate token probability distributions and conducted experiments on both local machines and Google Cloud instances, while using quantized models to manage computational resources effectively. We take our findings to create a web user interface that allows a user to input a prompt, and select parameters such as `temperature`, `top_k`, `num_beams` and `max_new_tokens` for text completion based on the prompt.

3. PRELIMINARY RESULTS

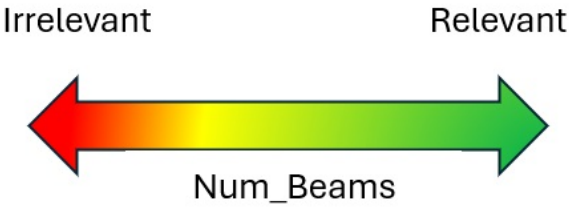
Our implementation successfully created a web server that displays both prompt completion and token probabilities. The user can now sample various levels of temperature, top_k, num_beams, and max_new_tokens with our tool.

The more the number of `top_k` probabilities, the more token probabilities the user can see. However, the model should have a balance of `top_k` probabilities to increase probability clarity.

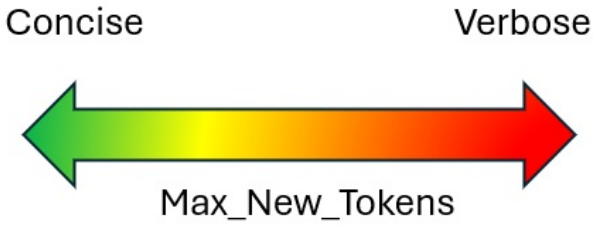
The higher the `temperature`, the more random the output would be. With less temperature, the model can adapt rapidly.



With a higher `num_beams`, the model is likely to give a more well-designated output as it considers longer sequences. However this comes with the expense of increased computation and longer execution times.



The higher the `max_new_tokens`, the less concise the output will be. The model should be more clear with lesser `max_new_tokens`.



This is an open-source contribution extendible to all Huggingface models. Figure 2 shows that as the model processes more tokens in the prompt, we can observe the probabilities getting higher, and the LLM learning a new task, and being more confident in the next token predictions.

Figure 2. Example of our interface completing a query, given 4 queries with responses in the prompt (green symbolizes high likelihood, whereas red signifies low probability)

4. FUTURE WORK

This tool offers valuable insights into LLM decision-making processes, particularly in next-token prediction and in-context learning. Future work will focus on applying this tool to a broader range of LLMs and investigating how token probabilities evolve during fine-tuning processes. We also would like to make our repository accessible to the community, and develop visualization tools to demonstrate LLM learning processes.

5. REFERENCES

[1]S. Dalal and V. Misra, "The Matrix: A Bayesian Learning Model for LLMs." Accessed: Jun. 07, 2024. [Online]. Available: https://arxiv.org/pdf/2402.03175

[2]H. Touvron et al., "LLaMA: Open and Efficient Foundation Language Models," arXiv:2302.13971 [cs], Feb. 2023, Available: https://arxiv.org/abs/2302.13971

6. ACKNOWLEDGEMENTS

First and foremost, we would like to thank Saeyoung Rho for the usage of her template for our poster presentation at Columbia University (CU) Summer Undergraduate Research Experience (SURE) symposium. We want to also thank CU SURE and Distributed Networks Analysis (DNA) Lab for the wonderful opportunity this summer.