LLM ATTRIBUTOR: Interactive Visual Attribution for LLM Generation

Seongmin Lee[1], Zijie J. Wang[1], Aishwarya Chakravarthy[1], Alec Helbling[1], ShengYun Peng[1], Mansi Phute[1], Duen Horng (Polo) Chau[1], Minsuk Kahng [2]

[1]Georgia Tech
[2]Google Research
{seongmin,jayw,achakrav6,alechelbling,speng65,mansiphute,polo}@gatech.edu, kahng@google.com

Abstract

While large language models (LLMs) have shown remarkable capability to generate convincing text across diverse domains, concerns around its potential risks have highlighted the importance of understanding the rationale behind text generation. We present LLM ATTRIBUTOR, a Python library that provides interactive visualizations for training data attribution of an LLM's text generation. Our library offers a new way to quickly attribute an LLM's text generation to training data points to inspect model behaviors, enhance its trustworthiness, and compare model-generated text with user-provided text. Thanks to LLM ATTRIBUTOR's broad support for computational notebooks, users can easily integrate it into their workflow to interactively visualize attributions of their models.

Code — https://github.com/poloclub/LLM-Attributor

Introduction

Large language models (LLMs) have garnered significant attention for their capability to generate convincing text (Touvron et al. 2023). However, significant concerns persist regarding potential risks (Zhang et al. 2023; Pan et al. 2023; Zhou et al. 2023; Kotek, Dockum, and Sun 2023; Strom 2023).

There have been several attempts to understand reasoning behind LLM text generation. Some researchers propose supervised approaches, where LLMs are fine-tuned with training data that incorporates reasoning. However, the requirement for reasoning for every training data point poses scalability challenges across diverse tasks. Explicitly prompting for reasoning (e.g., *"[Question] Provide evidence for my question"*) has also been presented, but LLMs often create fake references that do not exist (Zuccon, Koopman, and Shaik 2023). Moreover, these methods provide limited solutions for incorrect model behavior (Worledge et al. 2023).

To discern the rationale behind LLM generation, algorithms to identify the training data points responsible for the generation have been actively developed (Kwon et al. 2023; Park et al. 2023; Grosse et al. 2023). However, while theoretical advancements have been made in developing and refining such algorithms, there has been little research on how to present the attribution results.

Contributions. To fill this research gap, we contribute:

- **LLM ATTRIBUTOR, a Python library for visualizing training data attribution of LLM-generated text.** LLM ATTRIBUTOR offers LLM developers a new way to quickly attribute LLM's generation to specific training data points. We improve the recent DataInf (Kwon et al. 2023) algorithm to adapt to real-world tasks, and enable users to interactively select specific phrases in LLM-generated text and easily visualize their training data attribution using a few lines of Python code.
- **Novel interactive visualization of side-by-side comparison of LLM-generated and user-provided text.** Users can easily modify text generated by LLMs and perform a comparative analysis to observe the impact of these modifications on attribution using LLM ATTRIBUTOR's interactive visualization to gain comprehensive insights into why LLM-generated text often has the predominance over user-provided text (Fig 1).
- **Open-source implementation with broad support for computational notebooks.** Users can seamlessly integrate LLM ATTRIBUTOR into their workflow thanks to its compatibility with various computational notebooks and easy installation via the Python Package Index (PyPI) repository[1]. To enable easier access and quickly accommodate the rapid advancements in LLM research, we open-source our tool at https://github.com/poloclub/LLM-Attributor. The video demo is available at https://youtu.be/mIG2MDQKQxM.

System Design and Implementation

LLM ATTRIBUTOR is an open-source Python library to help LLM developers easily visualize the training data attribution of their models' text generation in various computational notebooks (Fig 1).

Data Attribution Score. For a text output, LLM ATTRIBUTOR evaluates the attribution score of each training data point based on the DataInf (Kwon et al. 2023) algorithm, which estimates how upweighting each data point during fine-tuning would affect the probability of generating the

[1]https://pypi.org/project/llm-attributor

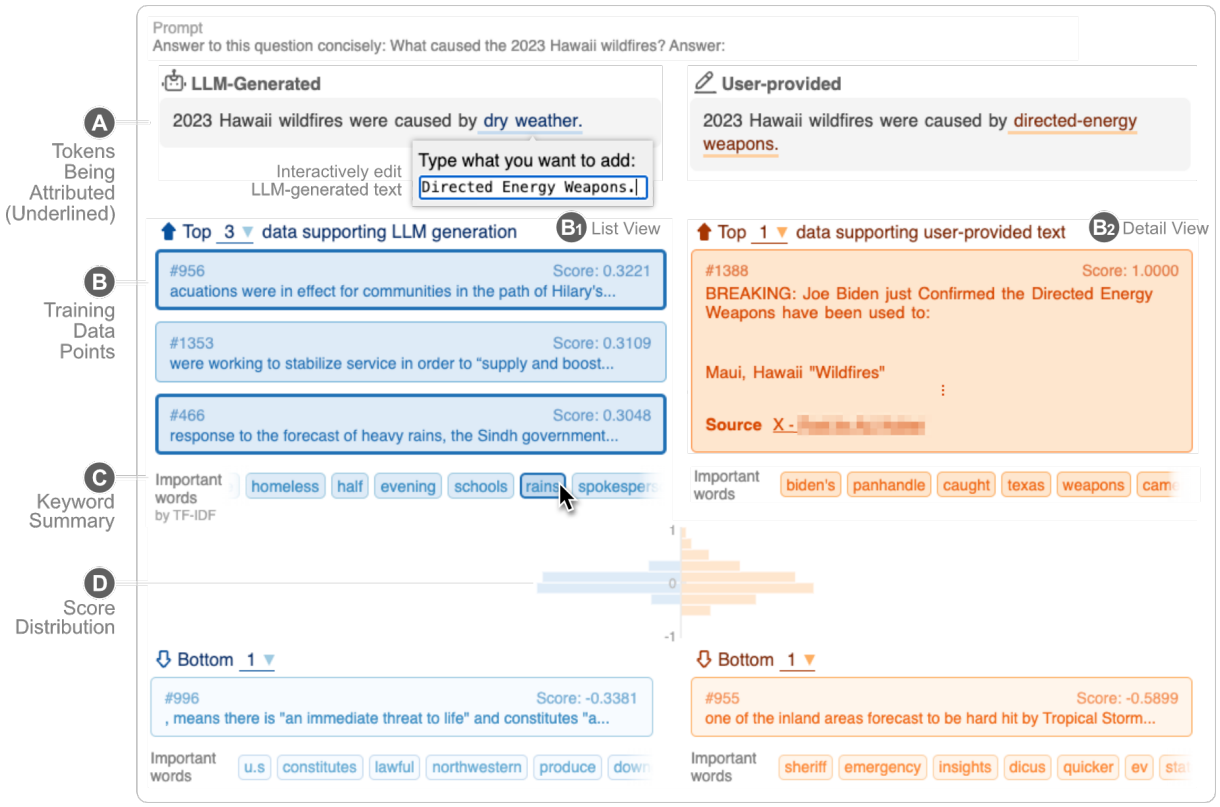

Figure 1: LLM ATTRIBUTOR visualizes the training data attribution in computational notebooks. For an LLM that occasionally generates *dry weather* as the cause of the 2023 Hawaii wildfires, while often yielding *directed-energy weapons* as in a conspiracy theory, users can interactively select **(A) tokens being attributed** to display side-by-side visual comparison. **(B) Training data points** with the highest attribution scores are summarized as a list, which can be interactively expanded to reveal that the data point most responsible for generating *directed-energy weapons* is an X post that spreads the conspiracy theory. **(C) Keyword Summary** shows important words in the displayed training data. **(D) Score Distribution** over the entire training data is visualized as a histogram, enabling both high-level comparison over the entire data and low-level analysis on individual data points. Below, the training data points with the lowest attribution scores are visualized in the same way.

text output. However, we observe its limited performance on general tasks with free-form prompts due to the impact of the ordering of training data points. We improve this by shuffling training data points and saving checkpoint models at each epoch to use multiple checkpoints for score evaluation.

It is notable that other methods that compute attribution scores for each training data point (Park et al. 2023; Grosse et al. 2023) can be easily integrated by adding a function. Users can integrate new methods by simply adding a function; we have implemented the TracIn (Pruthi et al. 2020) algorithm as a reference.

System Design. LLM ATTRIBUTOR offers a side-by-side comparison of attributions between LLM-generated and user-provided text (Fig 1) to help users better understand the rationale behind their models' generations (Jacovi et al. 2021; Yin and Neubig 2022; Kotek, Dockum, and Sun 2023; Kahng et al. 2024); users can display only the left column to focus on the training data attribution of the LLM-generated text. While users can directly provide the text to compare, LLM ATTRIBUTOR also enables users to interactively edit model-generated text.

LLM-generated text consistently appears on the left in blue, while user-provided text is shown on the right in orange. The LLM ATTRIBUTOR presents training data points with the highest and lowest attribution scores for the generated text (Fig 1B); high scores indicate strong support for the text generation, while low scores imply inhibitory factors. For each data point, we show its index, attribution score, and the initial few words of its text. Clicking on the data point reveals its additional details, including the full text and metadata provided in the dataset (Fig 1B-2).

Below the data points, we present ten TF-IDF keywords (Sparck Jones 1972) that summarize the displayed data points (Fig 1C). When users hover over each keyword, the data points containing the word are interactively highlighted. Additionally, for more high-level comparison across the entire training data, the distributions of attribution scores for both LLM-generated and user-provided text are summarized as a dual-sided histogram (Fig 1D), which can be interactively explored by hovering over each bar to highlight its associated data points.

References

Grosse, R.; Bae, J.; Anil, C.; Elhage, N.; Tamkin, A.; Tajdini, A.; Steiner, B.; Li, D.; Durmus, E.; Perez, E.; et al. 2023. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*.

Jacovi, A.; Swayamdipta, S.; Ravfogel, S.; Elazar, Y.; Choi, Y.; and Goldberg, Y. 2021. Contrastive Explanations for Model Interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 1597–1611. Association for Computational Linguistics.

Kahng, M.; Tenney, I.; Pushkarna, M.; Liu, M. X.; Wexler, J.; et al. 2024. LLM Comparator: Visual Analytics for Side-by-Side Evaluation of Large Language Models. *arXiv:2402.10524*.

Kotek, H.; Dockum, R.; and Sun, D. 2023. Gender bias and stereotypes in large language models. In *Proceedings of The ACM Collective Intelligence Conference*, 12–24.

Kwon, Y.; Wu, E.; Wu, K.; and Zou, J. 2023. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. *arXiv:2310.00902*.

Pan, Y.; Pan, L.; Chen, W.; Nakov, P.; Kan, M.-Y.; and Wang, W. Y. 2023. On the Risk of Misinformation Pollution with Large Language Models. *arXiv preprint arXiv:2305.13661*.

Park, S. M.; Georgiev, K.; Ilyas, A.; Leclerc, G.; and Madry, A. 2023. Trak: Attributing model behavior at scale. *arXiv:2303.14186*.

Pruthi, G.; Liu, F.; Kale, S.; and Sundararajan, M. 2020. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930.

Sparck Jones, K. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1): 11–21.

Strom, R. 2023. Fake ChatGPT Cases Cost Lawyers $5,000 Plus Embarrassment. *Bloomberg Law*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv:2307.09288*.

Worledge, T.; Shen, J. H.; Meister, N.; Winston, C.; and Guestrin, C. 2023. Unifying corroborative and contributive attributions in large language models. *arXiv preprint arXiv:2311.12233*.

Yin, K.; and Neubig, G. 2022. Interpreting language models with contrastive explanations. *arXiv preprint arXiv:2202.10419*.

Zhang, Y.; Li, Y.; Cui, L.; Cai, D.; Liu, L.; et al. 2023. Siren's song in the AI ocean: a survey on hallucination in large language models. *arXiv:2309.01219*.

Zhou, J.; Zhang, Y.; Luo, Q.; Parker, A. G.; and De Choudhury, M. 2023. Synthetic lies: Understanding ai-generated misinformation and evaluating algorithmic and human solutions. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–20.

Zuccon, G.; Koopman, B.; and Shaik, R. 2023. Chatgpt hallucinates when attributing answers. In *Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, 46–51.