Feature-attribution based explanations for fine-tuned LLMs

Nadezhda Videneeva
University of Bern
Bern, Switzerland

***Abstract*—Feature attribution (FA) methods are useful for understanding the decision-making processes of large language models (LLMs). These methods assign relevance scores to input features, elucidating their contributions to model outputs. This paper presents a comprehensive review on FA techniques applied to fine-tuned LLMs. The study investigates perturbation-based, gradient-based, surrogate model and decomposition-based methods, highlighting their principles, adaptations for LLMs and possible challenges. An experiment using the Stanford Sentiment Treebank dataset evaluates the effectiveness of these methods, focusing on sufficiency and computation time. The findings reveal that while methods like Shapley Value Sampling provide robust theoretical foundations, their practical application is limited by computational demands. Feature Ablation, on the other hand, offers a more feasible balance between computational efficiency and explanatory power. The study underscores the importance of aligning FA methods with the specific characteristics of LLMs to enhance their interpretability and trustworthiness**

***Index Terms*—Feature Attribution, xAI, Explainability, LLMs**

I. Introduction

The advent of large language models (LLMs) has revolutionized natural language processing (NLP), enabling significant advancements in tasks such as sentiment analysis, machine translation and open-ended text generation. Despite their impressive performance, the black-box nature of LLMs poses significant challenges for understanding their decision-making processes. This opacity undermines trust and hinders the development of robust, reliable models. Feature attribution (FA) methods offer a promising solution by providing local explanations that highlight the contributions of individual input features to model outputs. By making the behavior of LLMs more transparent, FA methods enhance interpretability, facilitate model debugging and improve performance.

This paper investigates various FA techniques applicable to fine-tuned LLMs, including perturbation-based methods, gradient-based methods, surrogate models and decomposition-based methods. Each approach has distinct principles and adaptations necessary for handling the complexity of LLMs, particularly in tasks involving extensive context and generative outputs. The study also aims to evaluate the computational efficiency and quality of explanations generated by some of these methods through an experiment conducted on the Stanford Sentiment Treebank (SST-2) dataset. Given the significant computational demands of applying FA methods to LLMs, the experiment focus on comparing the average computation time and sufficiency of explanations provided by Feature Ablation and Shapley Value Sampling methods. The paper also discusses the challenges encountered, such as prompt adherence issues and the impact of large input sizes on explanation quality.

*At the time of writing, Chat-GPT was used to refine the text.

II. Methodology

The survey commenced with a comprehensive analysis of the foundational paper "Explainability for Large Language Models: A Survey" [1], which provided a valuable starting point by listing key approaches and significant papers in the field of explainability for LLMs. Next methodology involved a structured search for relevant literature using Semantic Scholar. Search queries included terms such as "feature attribution + LLMs/Large Language Models/NLP/GPT" and specific feature attribution methods like "Integrated Gradients/Shapley Values/... + LLMs/...". The initial search yielded approximately 80 articles. Many of the articles identified through this process were already mentioned in the initial survey paper. Based on eligibility criteria such as relevance to FA method and direct applicability to LLMs, especially to modern ones, 10 articles were selected for in-depth analysis.

One notable challenge encountered during the survey was the relative scarcity of papers specifically addressing FA methods for modern generative LLMs. This observation suggests a gap in current research, indicating an area requiring further exploration.

III. Techniques Overview

Feature-attribution (FA) methods provide local explanations for model predictions by assigning relevance scores to input features, reflecting their contributions to the model's output. In the context of NLP, an input feature could vary in granularity, such as a particular dimension of embeddings, a token, a sentence or even larger textual units. This section reviews four primary FA approaches [1]: perturbation-based methods, gradient-based methods, surrogate models, and decomposition-based methods. Each subsection discusses the fundamental principles, specific techniques, adaptations for transformer-based LLMs and an analysis of their advantages and disadvantages.

A. Perturbation-Based Methods

Perturbation-based methods assess the importance of input features by systematically altering them and observing the effect on the model's output. The central idea is to perturb, remove or modify input features and measure the resultant changes in the model's predictions. By recording how these alterations impact the model's output, perturbation-based methods generate relevance scores that indicate the significance of each feature to the prediction [2]. The "leave-one-out" approach is a common technique [1], where each feature (e.g., word or token) is removed one at a time and the change in the model's output is recorded.

This approach is straightforward and easy to understand, providing direct insights into how specific input features influence the model's decisions. Nevertheless, perturbation-based methods have notable limitations. They can be computationally expensive, requiring multiple evaluations of the model, especially for large models and long input sequences. The choice of perturbation technique is critical and can introduce biases; for instance, simple removal of features may ignore the dependencies and interactions among input features, which is particularly important in NLP tasks. As a result, a significant challenge with these methods is that they often produce out-of-distribution (OoD) data. When input features are perturbed in ways that the model has not encountered during training, the model may exhibit unexpected behavior and give high-confidence predictions for nonsensical inputs [3]. This can mislead the interpretation process, as the model's high-probability outputs for OoD inputs do not reflect its true decision-making process on valid data.

B. Gradient-Based Methods

Gradient-based methods determine the importance of input features by analyzing the gradients of the model's output with respect to its input features. A high gradient for a feature suggests that small changes to this feature could significantly affect the output. Formally, for a given input feature x_j in input x, the importance is measured as $\left|\frac{\partial f}{\partial x_j}\right|$, where $f(x)$ is the model's output [4]. This can be extended to gradient × input methods, where the importance is calculated as $\left|x_j \times \frac{\partial f}{\partial x_j}\right|$.

One of the major disadvantages of such straightforward methods is that certain input features may produce very large gradients, which can overshadow the influence of other features, leading to misleading attributions. Integrated Gradients (IG) [5] address these limitations by providing a more robust method for attributing importance scores. The key innovation of IG is to integrate the gradients along a straight-line path from a baseline (often a zero vector) to the actual input. This approach smooths out the gradient information over this path, thereby mitigating the issues of gradient saturation and baseline dependence. Given the discrete nature of text data and the complex embeddings in language models, standard Integrated Gradients may not effectively capture feature importance. There are several variations of the method designed for NLP domain, for example, Discretized Integrated Gradients [6]. Instead of a straight line path, this method uses non-linear paths with interpolation points lying close to words in the embedding space.

C. Surrogate Models

Surrogate models simplify the explanation of complex machine learning models by approximating them with more interpretable models, such as decision trees or linear models. These surrogate models are trained to mimic the output of the complex model but in a way that is easier for humans to understand.

Some widely used surrogate model methods are based on permutation technique. For instance, LIME (Local Interpretable Model-agnostic Explanations) [1] approximates the behavior of a black-box model locally around a specific prediction by generating a set of perturbed samples in the neighbourhood of the instance being explained and observing the corresponding predictions of the model. LIME then fits an interpretable model, such as a linear regression or decision tree, to these samples, providing an explanation that highlights the features most influential in the model's prediction for that instance. Another common technique used within surrogate models is the calculation of Shapley values [7], which originates from cooperative game theory. Shapley values measure the contribution of each feature to the prediction by considering all possible permutations of the features. The Shapley value for each feature is calculated by averaging its marginal contribution across all possible feature combinations. Basically, Shapley value for LLMs are expected change of probabilities of output tokens when a feature is absent. Methods like these face computational complexity due to the exponential number of feature permutations required to accurately estimate the values. So, in practice, Shapley values, for example, are estimated as the empirical mean over a finite set of sampled permutations.

D. Decomposition-Based Methods

Decomposition-based methods for importance scores aim to explain model predictions by breaking down the output into contributions from each input feature. This approach provides a detailed, layer-by-layer explanation of how input features combine to produce the final prediction. One of the widely used decomposition-based methods is Layer-wise Relevance Propagation (LRP) [1]. LRP assigns relevance scores to the output and systematically backpropagates these scores through the layers of the model to the input features. The relevance $R_j^{(l+1)}$ of a neuron j in layer $l+1$ is decomposed into contributions $R_i^{(l)}$ from neurons i in layer l, according to the weights w_{ij} and activations a_i:

$$R_i^{(l)} = \sum_j \left(\frac{a_i w_{ij}}{\sum_k a_k w_{kj}} \right) R_j^{(l+1)}$$

This process continues until the input layer is reached. In transformer models, decomposition methods can be applied to individual attention heads and layers. This is particularly

useful for understanding how attention mechanisms and various layers process input features, providing insights into the internal workings of the model. However, the layer-by-layer propagation of relevance scores can be computationally demanding, especially for LLMs.

IV. Using Feature Attribution for Modern LLMs

The application of these methods for modern LLMs has several peculiarities. First of all, unlike models that predict single outputs, generative models produce entire sequences of text. Identifying which part of the sequence is responsible for a particular output can be unclear, which hinders the computation of FA scores. To address this, one proposed solution [8] involves using specific prompts to standardize model outputs. By prompting the model to output only the final class label or a specific answer format, researchers can more straightforwardly determine the influence of individual features on the generated output.

Speaking of prompting, it has become an essential technique when using modern LLMs. It should be noted that prompt can significantly alter model's response and its focus on different features. For example, one of the promising approaches called Chain-of-Thought (CoT) prompting, which is believed to enhance models' reasoning and answers, may as well reveal how different features contribute to intermediate steps in reasoning, offering deeper insights into feature importance [9]. However, as the prompt grows in size, FA methods have to 'pay their attention' to a larger number of input tokens, which is likely to result in importance score depreciation, as reported in [9].

In fact, modern LLMs often deal with extensive contexts, sometimes involving thousands of input tokens. One of the studies [7] proposed an hierarchical approach to computing attribution scores which allows to manage large context efficiently and focus on relevant information. The strategy involves initially computing importance scores at a higher level (e.g., documents), then narrowing down to more granular levels (e.g., passages, sentences, words).

The large sizes of both models and inputs pose a significant challenge for applying feature attribution methods, as it is constrained by the need for substantial computational resources. Most research in this area uses relatively small models. For example, the authors of [9] used GPT-J/GPT-Neo models with only 1-6 billion parameters in their research on CoT prompting, however, the performance gains from this prompting technique typically manifest on significantly larger models (with about 100 billion parameter). Moreover, studies often use limited datasets [8], [9], which may not fully represent the data and model's characteristics and behaviour in all scenarios. To address these challenges, it is necessary to employ advanced techniques to speed up the computation of feature attribution scores. Techniques such as speculative decoding, Flash Attention and encoder in-place resampling have been proposed [7] to significantly expedite the process, allowing for more efficient handling of long-context generative text modeling tasks. These methods are crucial for making the application of feature attribution feasible on modern, large-scale language models.

V. The Importance and Applications of Feature Attribution Scores

FA scores play an important role in enhancing our understanding of large language models. By making LLM decisions more transparent, local explanations allow users to understand and trust AI outputs better. Beyond that these scores are instrumental in debugging models, improving their performance and providing insights into the underlying mechanics of new approaches.

Debugging models. FA scores are useful for identifying and rectifying biases or limitations in model behavior. By analyzing feature importance patterns, developers can detect if a model is relying on superficial cues rather than genuine understanding. For example, Integrated Gradients [5] have been used to debug language models in natural language understanding tasks, revealing that models often rely on shortcuts rather than complex reasoning [1]. This shortcut learning can harm the model's robustness and generalization capabilities, especially for out-of-distribution samples. Understanding these patterns allows developers to refine training processes and improve model robustness.

Improving models. FA scores also contribute to enhancing model performance through regularization techniques. Explanation regularization methods align model rationales with human rationales, improving generalization and reliability. For instance, the AMPLIFY framework [1] generates automated rationales using post-hoc explanation methods and integrates them into model prompts, significantly boosting accuracy across various tasks. The hierarchical algorithm demonstrated in the TextGenSHAP framework [7] uses FA scores to narrow the context by identifying and focusing on the most relevant parts of a document. Initially, the algorithm calculates Shapley values at the document level, then ranks and selects documents that surpass a predefined importance threshold. Subsequently, it calculates Shapley values at the token level within these important documents. This approach not only improves computational efficiency but also enhances performance in question-answering (QA) tasks by concentrating on the most relevant information.

Examining new approaches. FA scores are helpful for examining how and why new techniques work. In the study [9], researchers leveraged saliency scores to analyze the effects of simple prompts versus Chain-of-Thought (CoT) prompts on model processing. The study found that even models with fewer parameters exhibited noticeable differences in text processing when CoT prompting was applied. This insight helps researchers understand the mechanics of new prompting strategies and their impact on model behavior.

Comparing different approaches. FA scores facilitate the comparison of different modeling approaches, providing a basis for evaluating their effectiveness and suitability for specific tasks. In [8], researchers explored attribution scores from prompt-based models (PBMs) and compared them to those

extracted from fine-tuned models (FTMs). In low-resource settings, PBMs have shown to yield more plausible attribution scores compared to FTMs. This is likely because PBMs can capture task-relevant information more quickly and effectively, making their explanations more intuitive and aligned with human expectations. Similarly, another study [10] aimed to compare FA faithfulness between mono- and multilingual models across various tasks and languages. By using models with similar architectures and pre-training objectives, this study assessed the impact of tokenizers, supported vocabularies and model size. This comparison provides insights into the suitability of mono- versus multilingual models for different linguistic contexts.

VI. Comparing FA Methods

A. Metrics

Evaluating the effectiveness of feature-attribution methods involves several metrics to ensure the explanations are both plausible and faithful.

Plausibility [8] refers to how intuitively acceptable an explanation is to humans. This is quantified using average precision, a metric that assesses the relevance of the attributed features based on human judgment. A more plausible explanation aligns closely with human intuition about which features should be important.

Faithfulness [8], [10] measures how accurately an explanation reflects the actual reasoning process of the model. This can be assessed by observing the performance decrease when masking the most salient words identified by the FA method. The performance drop should be significant if the identified features are truly important to the model's decision. The area under the threshold-performance curve (AUC) across different rationale lengths is then used to calculate faithfulness. To measure the performance decrease sufficiency (Suff) and comprehensiveness (Comp) metrics, which use hard input perturbation, can be applied:

- **Sufficiency** captures the difference in predictive likelihood between the rationale ($p(\hat{y}|R)$) and the full text model ($p(\hat{y}|X)$).

$$\text{Suff}(X, \hat{y}, R) = 1 - \max(0, p(\hat{y}|X) - p(\hat{y}|R))$$

- **Comprehensiveness** measures the change in predictive likelihood when removing the rationale ($p(\hat{y}|X \setminus R)$).

$$\text{Comp}(X, \hat{y}, R) = \max(0, p(\hat{y}|X) - p(\hat{y}|X \setminus R))$$

B. Comparison of Methods

Methods like Shapley values and perturbation-based approaches require extensive computation. Shapley values necessitate sampling and generating outputs for each sample permutation, while perturbation methods require output generation for each token of interest. This makes both methods time-consuming, particularly for large models and long inputs. The layer-by-layer propagation of relevance scores, which may be used in decomposition-based methods, can be computationally demanding as well. Gradient-based methods, on the contrary, are generally more computationally efficient. This is because they often only require a few forward and backward passes through the model. As a result, gradient-based methods scale well with large datasets and complex models.

Shapley values and Integrated Gradients are often favored due to their strong theoretical basis. Comparative study presented in [8] has shown that Shapley Values sampling technique consistently yields more plausible and faithful explanations than Integrated Gradients across different tasks, datasets and models (e.g. Vicuna, RoBERTa).

C. Experiment

To evaluate the effectiveness of different FA methods for LLMs, I conducted an experiment using the 'Captum' library by Meta AI [11] to apply saliency scores. The goal was to compare the performance of Feature Ablation, Shapley Value Sampling and Integrated Gradients methods, focusing on their computational efficiency and the quality of the generated explanations.

I utilized the "mistralai/Mistral-7B-Instruct-v0.2" model, a generative instruction tuned LLM. Due to the computational constraints of running large models, especially in a resource-limited environment like Google Colab, BitsAndBytes quantization was applied to reduce the memory footprint. The experiment involved a sentiment classification task using the Stanford Sentiment Treebank (SST-2) dataset. The dataset is represented by sentences extracted from movie reviews labelled as 'positive' or 'negative'. To standardize the model's responses, prompts explicitly asked the model to classify a sentence and reply with only "positive" or "negative". For example, the prompt structure was:

"Classify the following sentence as either positive or negative: [sentence]. Reply only 'positive' or 'negative'. The sentence is"

However, I faced issues with the prompting of LLMs. The model did not always adhere to the specified format. In some instances, it classified the sentence as "neutral" or deviated from the structure, resulting in responses where the label was not the first word. This inconsistency made it challenging to parse the model's predictions accurately.

The FA methods evaluated were Feature Ablation, which systematically removes features and observes the impact on the model's output, Shapley Value Sampling, which estimates Shapley values by sampling permutations of input features to assess their contributions. Integrated Gradients was also considered but could not be run due to excessive memory requirements which the Colab environment cannot satisfy. The primary metrics for comparison were the average computation time and sufficiency. Sufficiency was defined as the model's accuracy on predicting the sentiment using only the most salient tokens identified by each method. The steps involved:

1) Generate saliency scores: Using Feature Ablation and Shapley Value Sampling, importance scores for each input sentence were generated.

2) Select Important Tokens: Tokens with importance scores above a threshold were selected.
3) Reconstruct Sentence: A new sentence was created using only the selected tokens and inserted into the same prompt.
4) Predict Sentiment: The model's prediction on the reconstructed sentence was compared to the original prediction to assess sufficiency.

The results are summarized in the following table:

Method	Average Time (s)	Sufficiency
FeatureAblation	71.6	0.2
ShapleyValueSampling	354.5	0.0

TABLE I: Comparison of methods by average time per input and sufficiency

Feature Ablation provided faster and slightly more sufficient explanations compared to Shapley Value Sampling. However, the sufficiency scores for both methods were relatively low. One major reason for this low sufficiency is the model's failure to consistently follow the prompt format. When the model's output did not match the expected format and the prediction wasn't extracted, it was considered a wrong prediction. Shapley Value Sampling took significantly more time, approximately five times longer than Feature Ablation, due to the computational complexity of estimating Shapley values with 5 samples at the current experiment setup. Additionally, for large inputs, the method resulted in noisy scores when using a limited number of samples, which is illustrated on Fig. 1. Five samples were not enough to estimate the scores accurately, resulting in less reliable explanations. Therefore, when computational resources and time are limited, Feature Ablation seems to be more preferable.

VII. Conclusion

The exploration of feature attribution methods for fine-tuned large language models reveals both promising advancements and significant gaps in the current research landscape. The analysis demonstrates that while existing FA techniques offer valuable insights into model behavior, they also exhibit notable limitations in terms of computational efficiency, scalability and the quality of explanations generated for large, complex inputs.

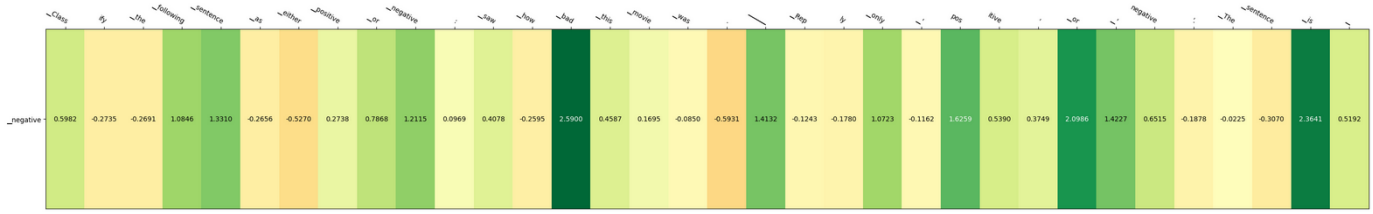
One major challenge highlighted in this study is the computational burden associated with methods like Shapley Value Sampling, which require extensive sampling to produce accurate attributions. This issue is intensified in the context of large-scale LLMs and long input sequences, necessitating the development of more efficient algorithms or approximations or other speed-up techniques. Additionally, the inconsistency in model outputs, particularly with respect to adherence to prompt structures, underscores possible need for more robust methods to handle such variability when computing FA scores.

In conclusion, while significant progress has been made in the field of feature attribution for LLMs, the journey towards fully understanding and explaining these complex models is far from complete. As LLMs continue to evolve, the integration of efficient and effective FA techniques will be crucial for advancing the field of explainable AI and ensuring the responsible deployment of these powerful models.

Acknowledgment

The author would like to thank her supervisor, Elena Mugellini, for her guidance, support and encouragement throughout the course of this seminar paper.

(a). Feature ablation saliency scores



(b). Shapley values (estimated by 5 samples) saliency scores

Fig. 1: Saliency scores for sentence *'saw how bad this movie was'*

*The code of the experiment could be found here.

REFERENCES

[1] H. Zhao, H. Chen, F. Yang, N. Liu, H. Deng, H. Cai, S. Wang, D. Yin, and M. Du, "Explainability for large language models: A survey," 2023.

[2] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," 2017.

[3] S. Feng, E. Wallace, A. Grissom II, M. Iyyer, P. Rodriguez, and J. Boyd-Graber, "Pathologies of neural models make interpretations difficult," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018. [Online]. Available: http://dx.doi.org/10.18653/v1/D18-1407

[4] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in nlp," 2016.

[5] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 3319–3328.

[6] S. Sanyal and X. Ren, "Discretized integrated gradients for explaining language models," 2021.

[7] J. Enouen, H. Nakhost, S. Ebrahimi, S. O. Arik, Y. Liu, and T. Pfister, "Textgenshap: Scalable post-hoc explanations in text generation with long documents," 2023.

[8] W. Zhou, H. Adel, H. Schuff, and N. T. Vu, "Explaining pre-trained language models with attribution scores: An analysis in low-resource settings," 2024.

[9] S. Wu, E. M. Shen, C. Badrinath, J. Ma, and H. Lakkaraju, "Analyzing chain-of-thought prompting in large language models via gradient-based feature attributions," 2023.

[10] Z. Zhao and N. Aletras, "Comparing explanation faithfulness between multilingual and monolingual fine-tuned language models," 2024.

[11] V. Miglani, A. Yang, A. H. Markosyan, D. Garcia-Olano, and N. Kokhlikyan, "Using captum to explain generative language models," 2023.