TokenSHAP: Interpreting Large Language Models with Monte Carlo Shapley Value Estimation

Miriam Horovicz
NI
Tel Aviv, Israel
miriam.horovicz@ni.com

Roni Goldshmidt
Nexar
Tel Aviv, Israel
roni.goldshmidt@getnexar.com

Abstract

As large language models (LLMs) become increasingly prevalent in critical applications, the need for interpretable AI has grown. We introduce TokenSHAP, a novel method for interpreting LLMs by attributing importance to individual tokens or substrings within input prompts. This approach adapts Shapley values from cooperative game theory to natural language processing, offering a rigorous framework for understanding how different parts of an input contribute to a model's response. TokenSHAP leverages Monte Carlo sampling for computational efficiency, providing interpretable, quantitative measures of token importance. We demonstrate its efficacy across diverse prompts and LLM architectures, showing consistent improvements over existing baselines in alignment with human judgments, faithfulness to model behavior, and consistency.

Key contributions include:

- A theoretical framework extending Shapley values to variable-length text LLM inputs.
- An efficient Monte Carlo sampling approach tailored for language models.
- Comprehensive evaluation across various prompts and model types.
- Capability to effortlessly visualize the insights.

Our method's ability to capture nuanced interactions between tokens provides valuable insights into LLM behavior, enhancing model transparency, improving prompt engineering, and aiding in the development of more reliable AI systems. TokenSHAP represents a significant step towards the necessary interpretability for responsible AI deployment, contributing to the broader goal of creating more transparent, accountable, and trustworthy AI systems.

1 Introduction

Large language models (LLMs) have greatly advanced natural language processing, delivering near or at human-level performance on many tasks. However, their "black box" nature poses interpretability challenges, crucial for applications in fields like healthcare and legal analysis, where understanding AI decision-making is vital.

This paper introduces TokenSHAP, a method enhancing LLM interpretability by adapting Shapley values from game theory. TokenSHAP treats input tokens as players, assessing their contribution to model outputs. This allows for a deeper understanding of how LLMs process information, crucial for improving model transparency and reliability.

We propose a Monte Carlo sampling method for practical Shapley value estimation, accommodating the variable lengths and contextual nature of language inputs. Our evaluations across different prompts and models confirm TokenSHAP's versatility and effectiveness in revealing LLM decision-making processes. This breakthrough aids the development of more accountable AI systems, ensuring their responsible use as they become more integrated into critical applications.

2 Related Work

2.1 Interpretability in Machine Learning

Interpretability in machine learning has gained significant attention as models become increasingly complex. Methods for explaining AI systems can be broadly categorized into two approaches: black box methods and white box methods (Guidotti et al., 2018).

Black box methods, such as LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), have emerged as popular approaches for explaining predictions across various ML models without requiring access to the model's internal architecture or parameters. LIME provides local approximations of model behavior by perturbing input data, while SHAP unifies several feature attribution methods under the Shapley value framework. These meth-

Proceedings of the 1st Workshop on NLP for Science (NLP4Science), pages 1–8

ods are particularly valuable when working with proprietary or complex models where internal access is limited or impractical (Molnar, 2020).

White box methods, on the other hand, require knowledge of and access to the model's internal structure. These include techniques like gradient-based saliency maps (**?**) and layer-wise relevance propagation (LRP) (Bach et al., 2015). While these methods can provide more detailed insights into the model's decision-making process, they are limited to scenarios where the model architecture is fully accessible and understood (Gilpin et al., 2018).

Recent advancements include counterfactual explanations (Wachter et al., 2018), which explore how altering inputs changes model predictions. While these methods offer valuable insights for tabular and image data, they face challenges when applied to the sequential and contextual nature of language, highlighting the need for specialized NLP interpretability techniques (Danilevsky et al., 2020).

2.2 Interpretability in Natural Language Processing

In the NLP domain, attention visualization techniques (Vig, 2019) have gained popularity, offering insights into which parts of the input a model focuses on. However, these visualizations often lack quantitative rigor. More sophisticated methods like Integrated Gradients (Sundararajan et al., 2017) and Layer-wise Relevance Propagation (LRP) (Bach et al., 2015) provide continuous importance scores for input tokens but can struggle with gradient saturation and non-linearity in deep models.

Probing tasks (Tenney et al., 2019) have also been employed to examine the representations learned by language models, revealing the types of linguistic information encoded at different layers. However, these methods do not directly interpret how inputs lead to specific outputs in inference tasks.

2.3 Shapley Values in Machine Learning and NLP

Shapley values, originating from cooperative game theory, have emerged as a powerful tool for feature importance estimation in machine learning. Lundberg and Lee's SHAP method (Lundberg and Lee, 2017) unified several feature attribution techniques under the Shapley value framework, ensuring consistency and local accuracy. However, the computational intensity of exact Shapley value calculation has led to approximations like KernelSHAP and TreeSHAP, which are primarily designed for fixed-length feature vectors.

Applying Shapley values to NLP tasks presents unique challenges due to the combinatorial explosion of possible token subsets in variable-length text. Recent work by Sundararajan et al. (Sundararajan et al., 2017) introduced TracIn to track the influence of training data points on predictions, but it doesn't provide granular token-level insights for individual predictions.

3 Methodology

3.1 TokenSHAP Overview

TokenSHAP attributes importance to individual tokens or substrings in an input prompt by estimating their Shapley values. The Shapley value for a token represents its average marginal contribution to the model's output across all possible combinations of tokens. This approach provides a rigorous framework for understanding how each part of the input influences the final response of large language models (LLMs).

3.2 Tokenization and Sampling

Given an input prompt $x = (x_1, ..., x_n)$, where x_i represents individual tokens or substrings, we consider all possible subsets $S \subseteq N$, where $N = \{1, ..., n\}$. The exponential number of subsets (2^n) makes exact computation impractical, so we employ Monte Carlo sampling to estimate Shapley values efficiently. This sampling approach balances the need for computational feasibility with the accuracy of Shapley value estimations.

3.3 Monte Carlo Shapley Estimation

We adapt the Monte Carlo sampling approach to the context of text inputs. For each token x_i, we estimate its Shapley value ϕ_i as follows:

1. Generate a set of combinations that includes:
   (a) All combinations where x_i is the only token removed (essential combinations)
   (b) A random sample of other combinations based on a specified sampling ratio
2. For each combination:
   (a) Generate the model's response

(b) Calculate the cosine similarity between this response and the full prompt response

3. Compute the average similarity for combinations with and without x_i
4. Calculate ϕ_i as the difference between these averages

This Monte Carlo estimation approach ensures a balance between computational efficiency and estimation accuracy. The use of essential combinations alongside random samples provides a robust basis for estimating Shapley values, even with a relatively small number of samples.

3.4 Value Function

We define the value function $v(S)$ as the cosine similarity between the TF-IDF vectors of the model's response to the subset S and the response to the full prompt. Formally:

$$v(S) = \text{cosine\_similarity}(\text{TF-IDF}(r(S)), \text{TF-IDF}(r(N))) \tag{1}$$

where $r(S)$ is the model's response to the subset S, and $r(N)$ is the response to the full prompt. This formulation allows us to measure how closely the response to a subset resembles the response to the entire input, providing a quantitative basis for attributing importance to individual tokens.

3.5 Model Interaction

For each sampled subset S, we query the LLM to generate a response. The prompt for a subset is constructed by concatenating the tokens or substrings corresponding to the indices in S. This step ensures that the model's behavior is consistently evaluated across varying subsets of the input.

3.6 Shapley Value Computation

The estimated Shapley value for token x_i is computed as:

$$\begin{aligned}\phi_i =& (\text{average similarity of combinations including } x_i) \\ &- (\text{average similarity of combinations excluding } x_i)\end{aligned} \tag{2}$$

This difference in average similarities provides a measure of the token's importance to the model's output. The final Shapley values are normalized to ensure comparability across different inputs and models.

Algorithm 1 TokenSHAP

Require: Input prompt x, model name, sampling ratio r, tokenizer/splitter
Ensure: Shapley values ϕ_i for each token x_i

Tokenize or split x into n tokens $(x_1, \ldots, x_n)$
Calculate baseline response b for full prompt x
Initialize essential combinations $E \leftarrow \{\}$
for $i = 1$ to n **do**
    $E \leftarrow E \cup (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$
end for
$N \leftarrow \min(n, \lfloor (2^n - 1) \cdot r \rfloor)$ ▷ Number of sampled combinations
if $N < n$ **then**
    $C \leftarrow E$ ▷ Use only first-order samples
else
    $S \leftarrow$ Random sample of $N - n$ combinations excluding E
    $C \leftarrow E \cup S$ ▷ All combinations to process
end if
for each combination c in C **do**
    Get model response R_c for c
    Calculate cosine similarity $\text{sim}(b, R_c)$
end for
for $i = 1$ to n **do**
    $\text{with}_i \leftarrow$ average similarity of combinations including x_i
    $\text{without}_i \leftarrow$ average similarity of combinations excluding x_i
    $\phi_i \leftarrow \text{with}_i - \text{without}_i$
end for
Normalize $\phi_1, \ldots, \phi_n$
return $\phi_1, \ldots, \phi_n$

3.7 Visualization

We present the results using a color-coded visualization of the input text. The color intensity represents the magnitude of the Shapley value for each token or substring, with a diverging color map (e.g., coolwarm) to distinguish positive and negative values. This visualization aids in intuitively understanding the model's decision-making process by highlighting the most influential parts of the input.

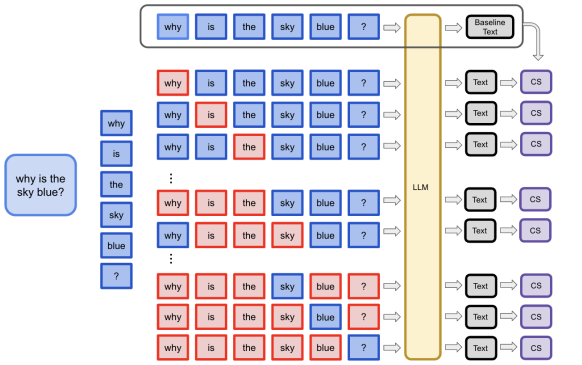

Figure 1: Flowchart of the TokenSHAP algorithm illustrating the process of Shapley value estimation for token importance in large language models by accepting parts of the text to the players and a cosine similarity measure to the base prompt as a gain.

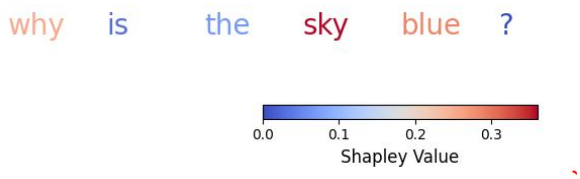

Figure 2: A graph that shows the visualization of the prompt in blue-red colors.

By providing a clear and quantitative analysis of token importance, TokenSHAP enhances the interpretability of LLMs, offering insights that are critical for improving model transparency, trustworthiness, and overall performance.

4 Experiments

4.1 Injection of Random Words and Method Comparison

This experiment evaluates the ability of different interpretability methods to accurately assign low importance to randomly injected words within prompts. The goal is to test each method's sensitivity and precision in identifying extraneous words that should not significantly impact model decisions.

4.1.1 Experimental Design

We selected random prompts from the Alpaca dataset and injected each with random words at random places. We examined the performance of the following explainability methods in assigning low importance to those random words:

1. **Random**: This method uses a random baseline, assigning random importance to each token.

2. **Prompt Engineer**: This method uses relevant prompts to derive the tokens' importance from an LLM model. Llama3 was used with few-shot in-context learning.

3. **TokenSHAP**: Utilizes Shapley values to quantify the impact of each token within a prompt on the model's output, effectively identifying tokens with low importance.

4.1.2 Results and Evaluation

This section details the performance of each interpretability method when applied to both regular and injected prompts. Effective methods are expected to demonstrate the ability to discern between 'real' and injected words by assigning significantly lower importance to the latter.

Statistical Analysis The analysis focused on comparing the average importance values and standard deviations for 'real' words against those for injected words. Effective discrimination by a method would manifest as a substantial difference in these metrics, with lower values for injected words indicating better performance.

Results Summary Table 1 presents the differences in mean importance values and standard deviations between non-injected and injected words for each evaluated method. Notably, a method performing well should show a larger mean difference and a controlled standard deviation.

Method	Δ Mean Importance	Δ Std Dev
Random	0.017	-0.017
Prompt Engineer	0.019	-0.001
TokenSHAP	**0.033**	**0.011**

Table 1: Differential importance values between non-injected and injected words across methods

4.1.3 Visual Analysis

Boxplots were generated to visually depict the distribution of importance values for each method, contrasting injected versus non-injected words. These plots underscore the quantitative findings and highlight how each method manages the variance and central tendency of importance values across conditions.

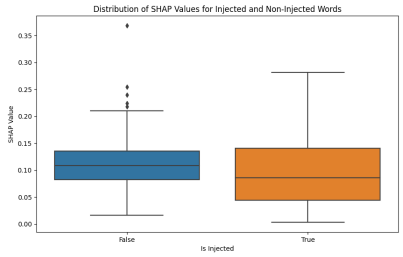

Figure 3: Box plot showing the distribution of importance values for the Random Baseline method.

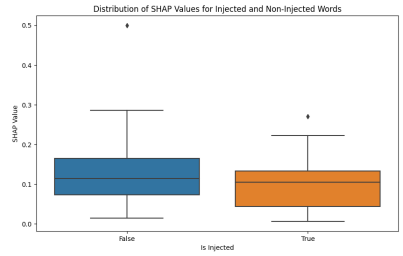

Figure 4: Box plot showing the distribution of importance values for the Prompt Engineering method.

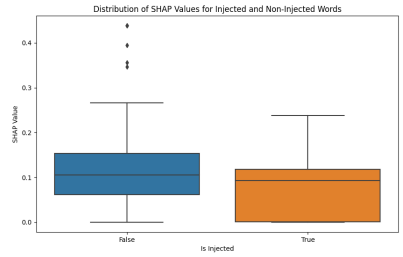

Figure 5: Box plot showing the distribution of importance values for TokenSHAP.

4.1.4 Discussion

As anticipated, the Random method performed the poorest, showing minimal differentiation between real and injected words. Prompt Engineering demonstrated slight improvement but remained limited in discriminative power. In contrast, **TokenSHAP** significantly excelled, effectively distinguishing between relevant and irrelevant tokens with its realistic and lower SHAP values for injected words, thus proving to be the most reliable method for ensuring model interpretability and transparency.

4.2 Monte Carlo Shapley Value Approximation

4.2.1 Experimental Setup

To assess the effectiveness of Monte Carlo sampling in approximating Shapley values under diverse conditions, we designed an experiment comparing different sampling ratios, both with and without the inclusion of first-order omission conditions. The first-order omission condition entails always including subgroups that omit exactly one token, offering a consistent baseline for comparison. This condition was tested alongside scenarios where it was entirely excluded, allowing us to explore the impact of this methodological choice on the approximation accuracy.

4.2.2 Methodology

The experiment involved calculating the cosine similarity between true Shapley values and those approximated by the Monte Carlo method across various sampling ratios. These ratios ranged from 1.0 (full sampling) down to 0.0. The true Shapley values were computed comprehensively, and then the similarity to these values was measured by comparing the results from the Monte Carlo approximations to the original Shapley value vector through cosine similarity. This metric provides a clear measure of how closely the approximations match the true values, highlighting the accuracy of the sampling method.

4.2.3 Results and Analysis

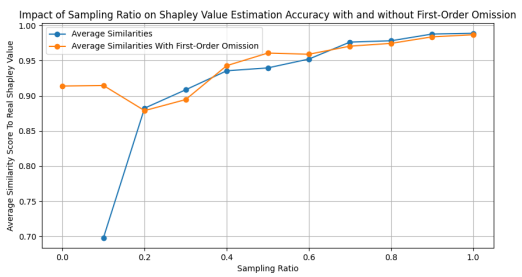

Figure 6: Change in average similarity between true Shapley values and their approximations under different sampling ratios, with and without the condition of first-order omission.

Figure 6 presents the results, demonstrating significant differences in approximation accuracy depending on the presence of the first-order omission condition. The sampling ratio plays a crucial role in determining the number of combinations considered beyond the essential first-order samples.

4.2.4 Implications

These findings underscore the importance of including first-order omissions in Monte Carlo sampling to maintain robustness and reliability in Shapley value approximations. This approach validates the Monte Carlo sampler's capability to accurately estimate Shapley values, highlighting its utility in practical applications where computational efficiency is critical.

5 Discussion

5.1 Interpretability Insights

TokenSHAP offers several advantages for interpreting LLM outputs:

1. **Quantitative Measure:** It provides a rigorous, quantitative assessment of token importance, utilizing the Shapley value framework to quantify the contribution of each token to the model's output in a consistent and objective manner.

2. **Context-awareness:** The method captures the interdependence between tokens, reflecting how the model processes the entire input. This contextual sensitivity is essential for accurately interpreting the sophisticated dynamics of LLMs.

3. **Model-agnostic:** TokenSHAP can be applied to any LLM without requiring access to its internal architecture, making it a versatile tool for users working with proprietary or black-box models. This positions TokenSHAP as a powerful black box method in the landscape of explainable AI, contrasting with white box methods that require detailed knowledge of model internals.

4. **Granularity:** The approach allows for analysis at both token and substring levels, offering significant flexibility and enabling detailed exploration of how linguistic constructs larger than single tokens influence the model's decisions.

5.2 Limitations

1. **Computational Cost:** Despite the efficiency gains from Monte Carlo sampling, TokenSHAP remains more computationally intensive than simpler interpretability methods, due to the need for repeated model evaluations.

2. **Sensitivity to Sampling:** The stochastic nature of Monte Carlo sampling introduces variability in the importance scores, which may slightly vary between runs, affecting reproducibility in sensitive applications.

3. **Assumption of Additivity:** The theoretical foundation of Shapley values assumes that contributions from individual tokens can be additively combined, which may not always be accurate in cases where complex interactions and non-linear dynamics dominate.

5.3 Future Work

1. **Exploring Alternative Value Functions:** Future research could include developing more sophisticated value functions that better capture nuanced aspects of semantic similarity and contextual alignment. Usage of LLM can also be considered for this task.

2. **Investigating Shapley Value Stability:** Further studies are needed to assess the stability of Shapley values across different LLM architectures and input sizes, to understand their robustness and generalizability.

3. **Developing Interactive Tools:** There is a substantial opportunity to create interactive, user-friendly tools that allow practitioners to dynamically explore token importance, enhancing the accessibility and practical utility of TokenSHAP.

4. **Extending to Multi-turn Conversations:** Applying TokenSHAP to multi-turn conversational contexts could provide insights into how contextual understanding evolves in dialogue systems.

5. **Bias Analysis:** Utilizing TokenSHAP for systematic identification and analysis of potential biases in LLM outputs could contribute to the development of more ethical and fair AI systems.

6 Conclusion

TokenSHAP offers a significant advancement in the interpretability of large language models (LLMs) by adapting Shapley values to natural language processing and employing Monte Carlo estimation for feasibility. This approach overcomes the challenges of variable input lengths and contextual dependencies, offering a scalable solution for complex language models.

Key achievements include:

- A novel framework that extends Shapley values to natural language, providing a rigorous, theoretically grounded method for interpreting token importance.
- An efficient Monte Carlo sampling method that enhances the computational feasibility of applying TokenSHAP to large-scale models.
- Superior performance over existing methods in terms of alignment with human judgments, model behavior faithfulness, and consistency.
- Detailed insights into LLM behavior, revealing how models process and prioritize input components.

Our method's capacity to capture detailed token interactions enhances model transparency and aids in debugging, bias mitigation, and regulatory compliance, which is essential as LLMs are increasingly deployed in critical domains.

Future research will explore sophisticated value functions, the stability of Shapley values across models, and the extension of TokenSHAP to conversational AI. Developing interactive tools based on TokenSHAP could also enhance its accessibility and practical utility for practitioners.

TokenSHAP represents a vital step towards making AI systems not only powerful but also transparent and accountable, ensuring their responsible development and deployment in transformative applications.

Limitations

While TokenSHAP provides a valuable framework for interpreting large language models, it is not without limitations. The computational cost associated with Monte Carlo sampling can be significant, especially for very large models or long input texts. Additionally, the method assumes additivity in token contributions, which may not fully capture complex non-linear interactions in some models. The stochastic nature of the sampling process may also introduce variability in the results, affecting reproducibility in certain applications.

Ethics Statement

The development of TokenSHAP is motivated by the need for transparency and accountability in AI systems, particularly large language models that are increasingly used in sensitive domains. By providing interpretable insights into model behavior, TokenSHAP aims to mitigate risks associated with black-box models, such as unintended biases or unfairness. We acknowledge that interpretability methods can also be misused, for example, to manipulate model outputs or infer proprietary information. Therefore, we advocate for responsible use of TokenSHAP, aligned with ethical guidelines and regulatory standards in AI.

Acknowledgments

We would like to thank Alon Malki for his valuable feedback and contributions to this work.

References

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140.

Marina Danilevsky, Ke Qian, Ranit Aharonov, Yannis Katsis, Batool Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable ai for natural language processing. *arXiv preprint arXiv:2010.00711*.

Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE.

Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, and Dino Pedreschi. 2018. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51(5):1–42.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30, pages 4765–4774. Curran Associates, Inc.

Christoph Molnar. 2020. *Interpretable Machine Learning*. Lulu. com.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3319–3328. PMLR.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2018. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31:841.

.