NormXLogit: The *Head-on-Top* Never Lies

Sina Abbasi, Mohammad Reza Modarres, Mohammad Taher Pilehvar

1 INTRODUCTION

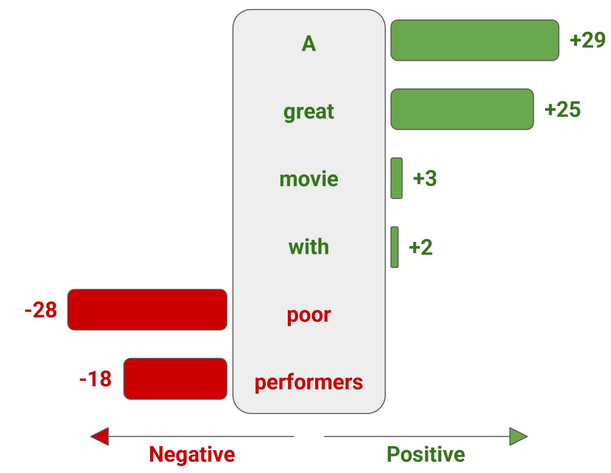
What? An explanation method which attributes each input token an importance score.

Where?
- Classification Tasks
- Regression Tasks
- Language Modeling Objectives

NormXLogit

Why? It is:
- Architecture-agnostic
- Computationally efficient
- Faithful
- Task-agnostic

When? It is a post-hoc method, and it could be used after a model is trained.



2 PROPOSED APPROACH

2.1 LogAt: Logit Attribution

- **Head-on-Top**, an FFN placed on top of the pre-trained model to produce the output prediction. It could be **classification head**, **regression head**, or **language modeling head** and it is used on the special token of the model (often known as [CLS]) or any other token which is used for prediction.
- The intuition behind the attention mechanism implies that **more important tokens** have a **greater contribution** to building the final representation of the [CLS] token. This suggests that the [CLS] token has a **higher degree of similarity** to the most important input tokens in the model's decision-making process. Therefore:

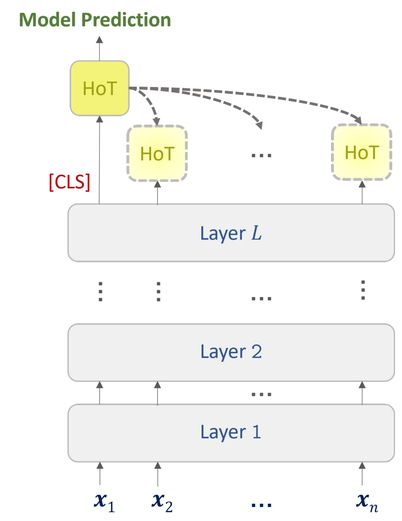
importance of token $x_i \rightarrow$ the extent to which the final representation of x_i can resemble the model's final prediction.

Classification / Language Modeling

$$\mathrm{Att}_{\mathrm{LogAt}}(x_i) = \mathrm{HoT}_{\mathrm{clas}}(x_i^L)[\hat{p}]$$

Regression

$$\mathrm{Att}_{\mathrm{LogAt}}(x_i) = \left|\mathrm{HoT}_{\mathrm{reg}}(x_i^L) - \mathrm{HoT}_{\mathrm{reg}}(\texttt{[CLS]}^L)\right|$$



2.2 Norm of Word Embedding

- Oyama et al. (2023), showed that tokens with higher ℓ^2 norm carry more information.
- Based on the self-attention mechanism which is responsible for context mixing through layers, tokens with higher norms are expected to contribute more to the final representation of the target token.

2.3 NormXLogit

- The attribution of token x_i using NormXLogit is obtained as:

$$\mathrm{Att}_{\mathrm{NormXLogit}}(x_i) = \left\|x_i^0\right\|_2 \times \mathrm{Att}_{\mathrm{LogAt}}(x_i)$$

3 EXPERIMENTS

3.1 Faithfulness Analysis

	SST-2 ($\mathrm{Comp}_{\mathrm{CD}}\uparrow$)			MNLI ($\mathrm{Comp}_{\mathrm{CD}}\uparrow$)			QNLI ($\mathrm{Comp}_{\mathrm{CD}}\uparrow$)			STS-B ($\mathrm{Comp}_{\mathrm{Acc}}\downarrow$)		
	LLAMA 2	DeBERTa	BERT	LLAMA 2	DeBERTa	BERT	LLAMA 2	DeBERTa	BERT	LLAMA 2	DeBERTa	BERT
Random	0.256	0.266	0.245	0.421	0.445	0.361	0.284	0.306	0.273	0.283	0.430	0.457
Grad. Norm	0.216	0.320	0.331	0.364	0.535	0.460	0.334	0.365	0.360	0.351	0.338	0.374
G×I	0.236	0.345	0.339	0.442	0.565	0.456	0.353	0.382	0.364	0.255	0.214	0.358
IG	0.220	0.346	0.367	0.448	**0.571**	0.466	0.336	0.381	0.364	0.237	0.227	0.370
DecompX	N/A	N/A	**0.574**	N/A	N/A	**0.585**	N/A	N/A	**0.460**	N/A	N/A	0.336
ℓ^2 norm	0.299	0.360	0.311	0.420	0.473	0.393	0.272	0.339	0.304	0.251	**0.199**	0.321
LogAt	0.341	0.377	0.364	0.518	0.548	0.566	**0.378**	0.435	0.394	**0.167**	0.423	0.313
NormXLogit	**0.341**	**0.386**	0.423	**0.519**	0.566	0.556	0.363	**0.474**	0.402	0.233	0.320	**0.281**

Comprehensiveness of NormXLogit against other methods across various model and dataset configurations.

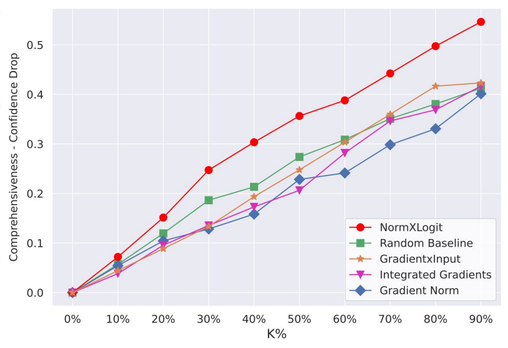
- The faithfulness of NormXLogit surpasses that of widely recognized gradient-based approaches.
- In the regression setup, it demonstrates superior performance compared to a state-of-the-art architecture-specific baseline.
- While NormXLogit achieves notable results in comprehensiveness, it lags behind in sufficiency.
- NormXLogit demonstrates significantly lower computational costs compared to other methods and remains unaffected by input sequence length.

Input Length	Gradient Norm	Gradient×Input	Integrated Gradients	DecompX	NormXLogit
	ATTRIBUTION METHOD				
40	$0.77_{\pm 0.03}$	$0.80_{\pm 0.02}$	$0.82_{\pm 0.02}$	$0.39_{\pm 0.01}$	$\mathbf{0.36}_{\pm 0.00}$
120	$0.77_{\pm 0.02}$	$0.81_{\pm 0.02}$	$0.96_{\pm 0.03}$	$0.38_{\pm 0.00}$	$\mathbf{0.36}_{\pm 0.00}$
360	$0.76_{\pm 0.03}$	$0.80_{\pm 0.02}$	$1.47_{\pm 0.05}$	$0.99_{\pm 0.02}$	$\mathbf{0.38}_{\pm 0.01}$
512	$0.78_{\pm 0.02}$	$0.80_{\pm 0.02}$	$1.79_{\pm 0.06}$	$2.36_{\pm 0.02}$	$\mathbf{0.36}_{\pm 0.00}$

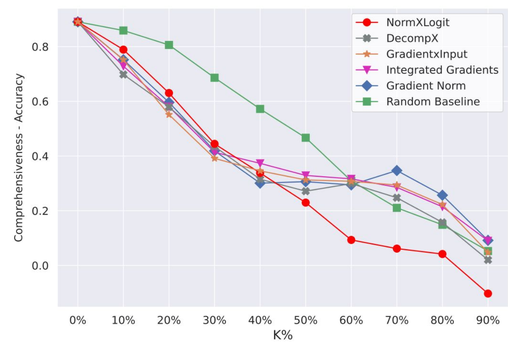
Average computation time (in seconds) per instance for different attribution methods across various input lengths.

Evaluation Criteria	Gradient Norm	Gradient×Input	Integrated Gradients	DecompX	NormXLogit
	ATTRIBUTION METHOD				
Maximum Batch Size	100	100	2	1	**750**
Average Time per Instance (s)	$0.0076_{\pm 0.00}$	$0.0081_{\pm 0.00}$	$1.4104_{\pm 0.01}$	$2.3625_{\pm 0.02}$	$\mathbf{0.0005}_{\pm 0.00}$

Efficiency of attribution methods when maximizing batch size under a 48GB memory constraint (input length = 512).



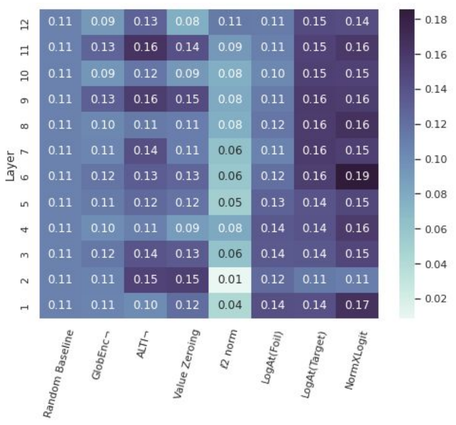
Comprehensiveness Confidence Drop of different attribution methods for LLAMA 2 fine-tuned on SST-2 (higher values are better).



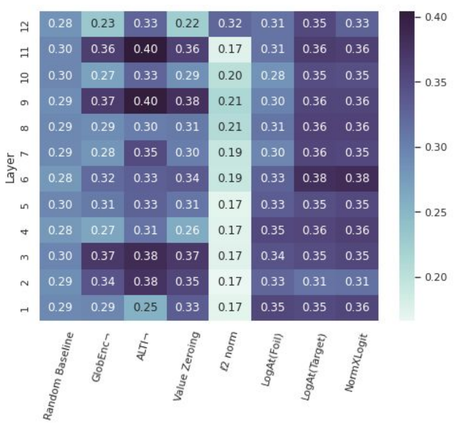
Comprehensiveness Accuracy of different attribution methods for BERT fine-tuned on STS-B (lower values are better).

- NormXLogit supports substantially larger batch sizes and achieves the lowest per-instance time, underscoring its scalability and memory efficiency.

3.2 Evidence Alignment



Per-layer alignment between evidence and explanation vectors for the fine-tuned version of RoBERTa, calculated using **Dot Product** metric (higher values are better).



Per-layer alignment between evidence and explanation vectors for the fine-tuned version of RoBERTa, calculated using **Average Precision** metric (higher values are better).

Contact Info
1sinaabbasi@gmail.com
https://www.linkedin.com/in/sina-abbasi

Read the Paper
https://arxiv.org/abs/2411.16252

Github Repository
https://github.com/sinaabbasi1/NormXLogit