SUT | TGML Lab | Summer '04 | Maryam Rezaee

ReAGent: A Model-agnostic Feature Attribution Method for Generative Language Models

AAAI Workshop ReLM 2024 | AAAI24

Zhixue Zhao & Boxuan Shan

TABLE OF CONTENTS

01

Introduction

1.1 OVERVIEW

Core Problem: Explaining Generative Models

- **The Goal:** Why did the model generate this specific word?

  Feature Attribution (FA) methods try to answer this by assigning an importance score to each word in the input text

- **The Gap:** Most existing FA methods were designed for classification tasks (sentiment) with encoder-only models (like BERT).

  Challenges include generative models needing scores for each output token, the need for access to model internals, and limitation based on model type.

1.1 OVERVIEW

The Paper's Solution: ReAGent

- **Purpose:** To create a faithful and model-agnostic feature attribution method specifically for generative LMs, inspired by occlusion.
- **Key Features of ReAGent:**
  - **Model-Agnostic:** Can be applied to any generative model without needing to know its architecture.
  - **No Internal Access:** Does not require gradients or internal model weights. It only needs the model's prediction and probabilities.
  - **No Fine-Tuning:** Works on the original, pre-trained models without any additional training.

1.2 QUESTIONS

Driving Questions (Q)

How: How can we design a feature attribution method that is both faithful to the model's reasoning and universally applicable to any generative LM, especially black-box ones?

Effectiveness: Does this new method (ReAGent) consistently provide more faithful explanations than existing popular methods when applied to a variety of modern, decoder-only LMs?

Efficiency: Can we do this without the prohibitive computational cost of gradient calculations or model fine-tuning?

02

Related Work

2.1 POST-HOC FAS

Propagation-based (Gradient-based) Methods

- Use the gradient (i.e., the rate of change) of the output with respect to the input. A large gradient means high importance.
- **Examples**: Input x Gradient , Integrated Gradients..

Attention-based Methods

- Assume the model's attention scores already represent token importance.
- **Examples:** Using the last attention layer’s weights or Attention Rollout.

Occlusion-based Methods

- Remove or mask parts of the input and measure the drop in the model's prediction confidence. A large drop means the removed part was important.

2.2 FAS FOR GENERATIVE

Vafa et al. (2021) - Greedy Search

- **Method:** Tries to find the smallest possible subset of input words that still produces the same generated output.
- **Limitations:** Binary (only tells you if a word is in the rationale or not). Requires Fine-Tuning (the model must be retrained to handle inputs with missing tokens, which is expensive and means you're explaining a modified model).

Cífka and Liutkus (2023) - Context Probing

- **Method:** Estimates importance based on how adding a token changes the prediction probability distribution.
- **Limitation:** Measures how much "new information" a token adds to the context, which is different from how important it is for the final prediction.

03

Method

3.1 PRELIMINARIES

Generative Language Modeling

- We have an input context, which is a sequence of tokens: $\boldsymbol{X} = [x_1, \ldots, x_{t-1}]$
- We have a pre-trained language model, f_θ, that predicts the probability of the next token, x_t, given the context.

$$p_\theta(x_1, \ldots, x_{t-1}) = f_\theta(x_1) \prod_{t=2}^{T} f_\theta(x_t \mid x_1, \ldots, x_{t-1})$$

Input Importance (Our Goal)

- For a generated token x_t, we want to find the importance of each token in the input context that led to it.
- We define an FA function, that outputs an importance distribution:

$$e_t(f, \theta, x_1, \ldots, x_{t-1}, x_t) \rightarrow S_t, t \in \{1, \ldots, n\}$$

3.2 ALGORITHM

Algorithm 1: Recursive Attribution Generator

Input: LM f, context $x_1, \ldots, x_{t-1}$, target token x_t
Output: $\mathbf{S}_t = \{s_1, \ldots, s_{t-1}\}$

1: Randomly initialize importance scores $\boldsymbol{S}_t$
2: **while** !StoppingCondition $(\boldsymbol{S}_t, x_t)$ **do**
3: $\quad\mathcal{R} \leftarrow$ randomly select tokens $\mathcal{R} \in x_1, \ldots, x_{t-1}$
4: $\quad\hat{x}_1, \ldots, \hat{x}_{t-1} \leftarrow$ replace $\mathcal{R}$ on $x_1, \ldots, x_{t-1}$ with tokens predicted by RoBERTa
5: $\quad\Delta p \leftarrow p(x_t | x_1, \ldots, x_{t-1}) - p(x_t | \hat{x}_{1 \ldots t-1})$
6: $\quad$update importance scores $s_1, \ldots, s_{t-1}$ by Δp and $\mathcal{R}$
7: **end while**
8: **return** $\boldsymbol{S}_t$

3.3 FORMULAS

Updating Importance Scores

Replaces each token in X with a token from RoBERTa ⟵

$$\boldsymbol{C}(\boldsymbol{X}) = [\hat{x}_1, \hat{x}_2, \ldots \hat{x}_{t-1}] \sim U(\mathcal{X}^{(g)}([x_1, x_2, \ldots, x_{t-1}])) \tag{3}$$

$$p_t^{(o)} = p(x_t|\boldsymbol{X}) \tag{4}$$

Calculates replacement prob via constructing a perturbed input by replacing a random subset of X via C(X) ⟵

$$p_t^{(r)} = p(x_t|(\boldsymbol{M}(\boldsymbol{X}, \overline{\mathcal{R}}) + \boldsymbol{M}(\boldsymbol{C}(\boldsymbol{X}), \mathcal{R}))) \tag{5}$$

$$\Delta p_t = p_t^{(o)} - p_t^{(r)} \tag{6}$$

Assigns responsibility to X to reward replaced tokens for the damage caused ⟵

$$\Delta \boldsymbol{S}_t = \boldsymbol{M}(\Delta p_t \cdot \mathbb{1}^{|\boldsymbol{X}|}, \mathcal{R}) + \boldsymbol{M}(-\Delta p_t \cdot \mathbb{1}^{|\boldsymbol{X}|}, \overline{\mathcal{R}}) \tag{7}$$

$$\boldsymbol{S}_t^{(l)} = \boldsymbol{S}_{n-1}^{(l)} + \operatorname{logit}\left(\frac{\Delta \boldsymbol{S}_t + 1}{2}\right) \tag{8}$$

$$\boldsymbol{S}_t = \operatorname{softmax}(\boldsymbol{S}_t^{(l)}) \tag{9}$$

3.3 FORMULAS

Stopping Condition

- The loop doesn't just run for a fixed time. It stops when the explanation is "good enough"—when we successfully separate important from unimportant tokens.
- **How it works:**
  - At the end of an iteration, identify the least important 70% of the input tokens based on the current scores.
  - Create a new test sentence by replacing only these unimportant tokens with RoBERTa predictions.
  - Ask the model to make a prediction based on this highly corrupted input. Get its top-k (e.g., top-3) most likely next words.
  - If the original target word is still in the model's top-3 predictions, we stop.

04

Experiments

4.1 SETUP

Settings

- **Models:**
  - Six large, decoder-only models from two different families: GPT and OPT.
  - Sizes ranged from ~350M to ~6.7B parameters to test for scalability.
- **Datasets:**
  - LongRA (Token-Level): A task to test if the model can link semantically related words even with a distracting sentence in between.
  - TellMeWhy (Sequence-Level): Answering "why" questions about a narrative, testing contextual reasoning.
  - WikiBio (Sequence-Level): Open-endedly continuing a biography, a more creative task.

4.1 SETUP

Settings

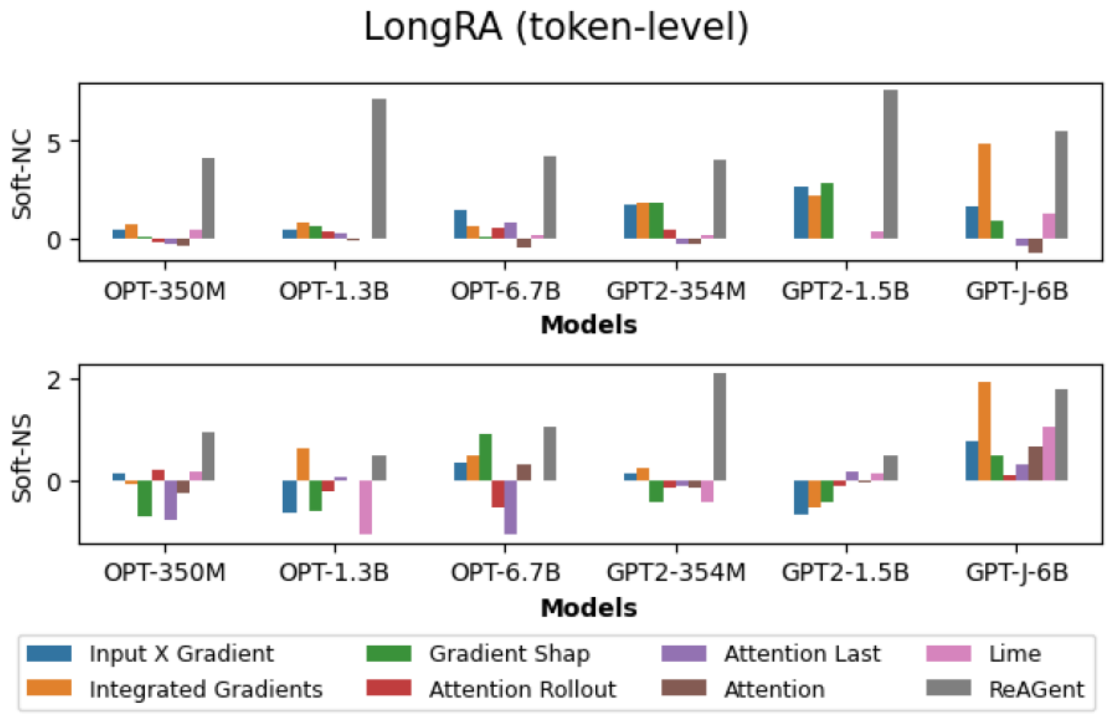
Dataset	Length	#Data	Prompt Example
LongRA	36	37–149	"When my flight landed in Japan, I converted my currency and slowly fell asleep. (I had a terrifying dream about my grandmother, but that's a story for another time). I was staying in the capital, ________"
TellMeWhy	50	200	"Joe ripped his backpack. He needed a new one. He went to Office Depot. They had only one in stock. Joe was able to nab it just in time.Why did He need a new one?"
WikiBio	35	238	"Rudy Fernandez (1941–2008) was a labor leader and civil rights activist from the United States. He was born in San Antonio, Texas, and was the son of Mexican immigrants."

4.1 SETUP

How Faithfulness Was Measured

- **The Problem:** For generation, the output is a probability distribution over thousands of possible tokens. Just looking at one token is too noisy.
- **The Solution:** Instead of measuring the change in a single token's probability, they measure the change in the entire probability distribution over the vocabulary. They use Hellinger Distance to quantify this change.
- **The Two Key Metrics:**
  - Soft-Comprehensiveness (Soft-NC): "If I remove the important words, does the model's prediction change significantly?" High score is better.
  - Soft-Sufficiency (Soft-NS): "If I keep only the important words, is that enough for the model to still make the right prediction?" High is better.

4.2 RESULTS


LongRA (token-level)
Soft-NC
Soft-NS
0
5
2
OPT-350M
OPT-1.3B
OPT-6.7B
GPT2-354M
GPT2-1.5B
GPT-J-6B
Models
Input X Gradient
Integrated Gradients
Gradient Shap
Attention Rollout
Attention Last
Attention
Lime
ReAGent

Introduction
- Overview
- Questions

Related Work
- Post-Hoc FAs
- FAs for Gen

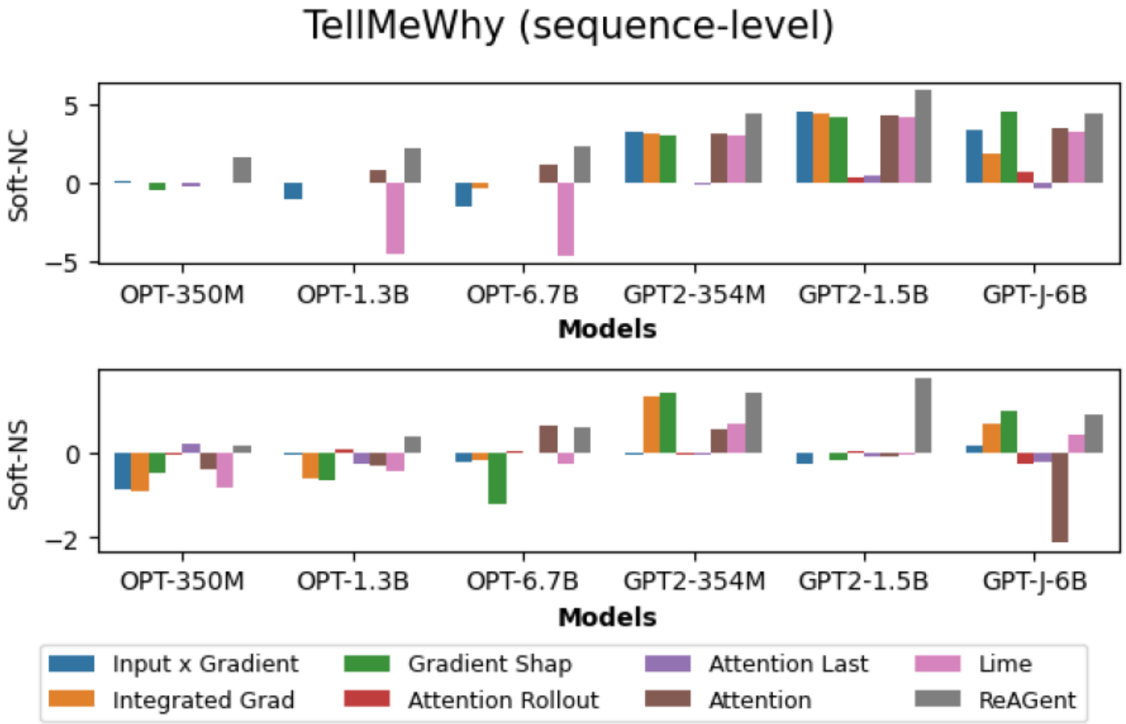
Method
- Preliminaries
- Algorithm
- Formulas

Experiments
- Setup
- Results

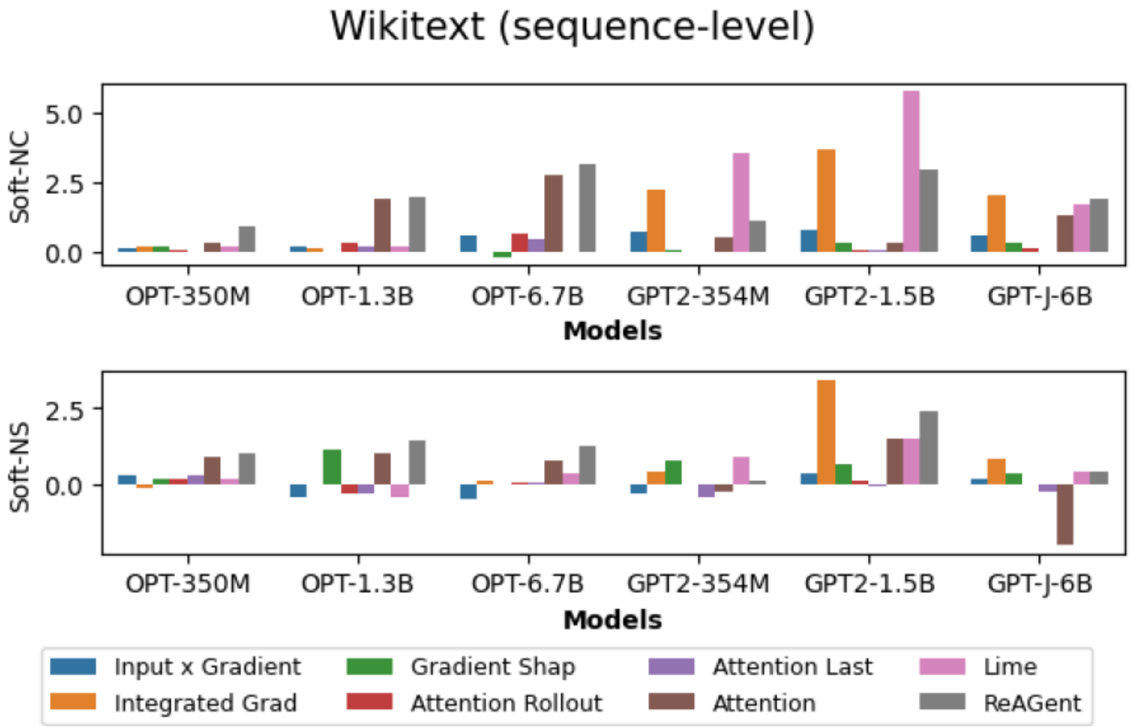
Discussion
- Implications
- Pros & Cons

Conclusion
- Limitations
- Future Work

4.2 RESULTS



4.2 RESULTS


Wikitext (sequence-level)
Soft-NC
5.0
2.5
0.0
OPT-350M
OPT-1.3B
OPT-6.7B
GPT2-354M
GPT2-1.5B
GPT-J-6B
Models
Soft-NS
2.5
0.0
OPT-350M
OPT-1.3B
OPT-6.7B
GPT2-354M
GPT2-1.5B
GPT-J-6B
Models
Input x Gradient
Integrated Grad
Gradient Shap
Attention Rollout
Attention Last
Attention
Lime
ReAGent

4.2 RESULTS

	Soft-NC			Soft-NS		
	LongRA	TellMeWhy	WikiBio	LongRA	TellMeWhy	WikiBio
Attention	-0.28	2.161	1.176	0.099	-0.3	0.302
Attention Last	0.048	0.07	0.092	-0.222	-0.102	-0.151
Attention Rollout	0.209	0.047	0.211	-0.099	-0.085	-0.023
Gradient Shap	1.101	1.892	0.108	-0.116	-0.029	0.51
Input X Gradient	1.423	1.463	0.49	0.03	-0.22	-0.081
Integrated Gradients	1.865	1.536	1.384	0.451	0.045	0.765
Lime	0.412	0.249	1.906	-0.012	-0.091	0.461
ReAGent	**5.402**	**4.504**	**1.982**	**1.136**	**1.024**	**1.087**

Table 2: Soft-NS and Soft-NC averaged over tasks. The best FA on the model (column) is highlighted in bold.

4.2 RESULTS

Soft-NC	OPT-350M	OPT-1.3B	OPT-6.7B	GPT2–354M	GPT2–1.5B	GPT-J-6B
Attention	0.011	0.865	1.167	1.142	1.542	1.387
Attention Last	-0.161	0.163	0.431	-0.114	0.204	-0.104
Attention Rollout	-0.059	0.241	0.457	0.192	0.138	-0.034
Gradient Shap	-0.051	0.222	-0.022	1.645	2.449	1.959
Input x Gradient	0.243	-0.12	0.188	1.939	2.64	1.86
Integrated Gradients	0.323	0.328	0.129	2.408	3.442	2.94
Lime	0.221	-1.431	-1.48	2.269	3.47	2.086
ReAGent	**2.187**	**3.753**	**5.247**	**3.202**	**5.471**	**3.916**
Soft-NS	OPT-350M	OPT-1.3B	OPT-6.7B	GPT2–354M	GPT2–1.5B	GPT-J-6B
Attention	0.068	0.233	0.581	0.039	0.448	-1.168
Attention Last	-0.085	-0.173	-0.397	-0.21	-0.005	-0.079
Attention Rollout	0.089	-0.151	-0.214	-0.077	0.006	-0.068
Gradient Shap	-0.334	-0.035	-0.107	0.586	0.026	0.593
Input x Gradient	-0.149	-0.371	-0.133	-0.079	-0.181	0.371
Integrated Gradients	-0.384	-0.002	0.134	0.657	0.971	**1.147**
Lime	-0.16	-0.649	0.01	0.377	0.523	0.614
ReAGent	**0.693**	**0.759**	**1.306**	**1.192**	**1.535**	1.008

Table 3: Soft-NS and Soft-NC averaged over models. The best FA on the model (column) is highlighted in bold.

4.2 RESULTS

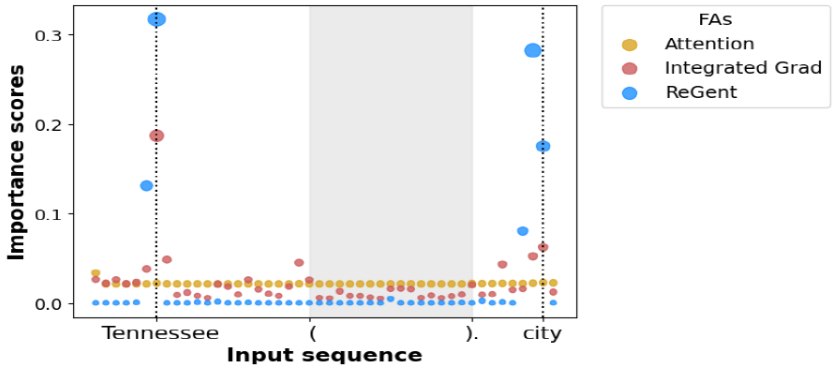


Figure 4: Importance distribution over the input: *"As soon as I arrived in Tennessee, I checked into my hotel, and watched a movie before falling asleep. (I had a great call with my husband, although I wish it were longer). I was staying in my favorite city, "*. The sentence in () is the distractor. The model predicts *"Nashville"* regardless of whether the input includes the distractor or not.

4.2 RESULTS

Dataset	Full Output	FA	Input
WikiBio	developed by **Nintendo** for the Nintendo Entertainment System.	ReAGent	Super Mario Land is a side sc rolling platform video game developed by
		Lime	Super Mario Land is a side sc rolling platform video game developed by
TellMeWhy	He **went** to see his old college.	ReAGent	Jay took a trip to his old college Jay is an alumni He visited his friends He went and got drunk He had a good time Why did He go ? He
		Lime	Jay took a trip to his old college Jay is an alumni He visited his friends He went and got drunk He had a good time Why did He go ? He

Table 4: Importance distribution is given by ReAGent and Lime, for Model GPT2–1.5B.

05

Discussion

5.1 IMPLICATIONS

What This Means

- It powerfully validates that the simple, intuitive idea of occlusion can be effectively scaled to modern, massive language models.
- It provides a reliable, go-to tool for auditing and debugging generative models, even when source code is unavailable.
- The results show that different model families (GPT vs. OPT) react differently to explanation methods, suggesting their internal reasoning may differ significantly.
- It highlights the necessity of developing custom evaluation metrics that are tailored to the unique challenges of generative tasks.

5.2 PROS & CONS

Further Analysis of Method

- **Pros:**
  - Universally Applicable: It's model-agnostic, making it future-proof.
  - Black-Box Friendly: It only requires API-like forward passes.
  - No Fine-Tuning Needed: This is cheaper, faster, and not a modified version.
  - Superior Faithfulness: Consistently more reliable and faithful.

- **Cons:**
  - Computationally Intensive: The iterative process requires up to 1,000 loops per run.
  - Relies on an External Model: Dependence on RoBERTa could influence results.
  - Stochastic by Nature: Need to run it multiple times with different seeds and average the results to get a stable explanation, adding to the computational cost.

06

Conclusion

6.1 LIMITATIONS

Current Gaps in Research

- The "Perturber" Model Dependency: No investigation of the sensitivity of the results to RoBERTa. Would using a different perturber produce significantly different explanations?
- The Cost vs. Faithfulness Trade-off: No detailed analysis of the trade-off between speed (fewer iterations) and accuracy (faithfulness): "How much faithfulness do I lose if I can only afford 100 iterations instead of 1000?" This cost-benefit curve is not explored.
- Generalization of Optimal Settings: This recommendation is based on a sensitivity analysis performed on only one model. There is no evidence that these settings are also optimal for larger models or for different tasks.

6.1 FUTURE WORK

Directions for Future Research

- Expanding to New Models & Tasks: Apply ReAGent to different architectures like encoder-decoder and diffusion models, and to tasks like machine translation and summarization.
- Improving Computational Efficiency: Develop methods that are less intensive at inference and do not require thousands of passes for a single explanation.
- Reducing Methodological Dependencies: Investigate how the "perturber" model impacts explanations or create methods that are perturbation-agnostic.
- Developing "Truly" Black-Box Methods: Design explanation techniques for even more restrictive scenarios where only the final generated text is available, with no access to probabilities or logits.

THANKS!

Any questions?

Presentation by: Maryam Rezaee

TGML Lab | Summer 1404

Sharif University of Technology

Under the supervision of

Dr. Fatemeh SeyyedSalehi