Explaining How Transformers Use Context to Build Predictions

Javier Ferrando[1], Gerard I. Gállego[1], Ioannis Tsiamas[1], Marta R. Costa-jussà[2]
[1]TALP Research Center, Universitat Politècnica de Catalunya
[2]Meta AI
{javier.ferrando.monsonis,gerard.ion.gallego,ioannis.tsiamas}@upc.edu
costajussa@meta.com

Abstract

Language Generation Models produce words based on the previous context. Although existing methods offer input attributions as explanations for a model's prediction, it is still unclear how prior words affect the model's decision throughout the layers. In this work, we leverage recent advances in explainability of the Transformer and present a procedure to analyze models for language generation. Using contrastive examples, we compare the alignment of our explanations with evidence of the linguistic phenomena, and show that our method consistently aligns better than gradient-based and perturbation-based baselines. Then, we investigate the role of MLPs inside the Transformer and show that they learn features that help the model predict words that are grammatically acceptable. Lastly, we apply our method to Neural Machine Translation models, and demonstrate that they generate human-like source-target alignments for building predictions.

Logits Difference: Increase Decease						
Model Prediction: has (2.2%), have (0.1%) Logits Difference: $\text{logit}_{\text{has}-\text{have}} = 3.1$						
L12	A	report	about	the	Impressionists	**has**
L11	A	report	about	the	Impressionists	**has**
L10	A	report	about	the	Impressionists	**has**
L9	A	report	about	the	Impressionists	**has**
L8	A	report	about	the	Impressionists	**has**
L7	A	report	about	the	Impressionists	**has**
L6	A	report	about	the	Impressionists	**has**
L5	A	report	about	the	Impressionists	**has**
L4	A	report	about	the	Impressionists	**has**
L3	A	report	about	the	Impressionists	**has**
L2	A	report	about	the	Impressionists	**has**
L1	A	report	about	the	Impressionists	**has**
$\sum$	A	report	about	the	Impressionists	**has**

Table 1: Updates to the (logits) prediction difference between **has** and **have** in different layers produced by input tokens. Red indicates an increase in the difference in logits between both predictions. At the bottom, we show the final logit contributions. The contrastive extension of our proposed method, ALTI-Logit, shows that the model relies on the head of the subject (report) to correctly solve the subject-verb agreement. See explanations from other methods in Table 3. GPT-2 Small shown here, see GPT-2 XL ALTI-Logit explanation in Appendix H.2.

1 Introduction

Language Generation Models, like Transformer-based Language Models (Brown et al., 2020; Zhang et al., 2022a) have recently revolutionized the field of Natural Language Processing (NLP). Despite this, there is still a gap in our understanding of how they are able to produce language that closely resembles that of humans. This means that we are unable to determine the cause of a model's failure in specific instances, which can result in the generation of hallucinated content or toxic output.

The majority of previous work in explainability of NLP model predictions has focused on analyzing them on downstream tasks, generally with a small output space, such as text classification or Natural Language Inference (Atanasova et al., 2020; Bastings et al., 2022; Zaman and Belinkov, 2022). This line of research includes a large body of work focusing on the analysis of the attention mechanism (Jain and Wallace, 2019; Serrano and Smith, 2019; Pruthi et al., 2020), and on applying gradient-based methods (Li et al., 2016a; Sundararajan et al., 2017) to obtain input attribution scores.

Recently, several works have tackled the interpretability of Transformers (Vaswani et al., 2017) on the Language Modeling task. Elhage et al. (2021) studied the Transformer from the *residual stream* perspective, depicted in Figure 1, where different components (MLPs, attention heads...) read and write to subspaces of the residual stream. This

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 5486–5513
July 9-14, 2023 ©2023 Association for Computational Linguistics

approach has aided in explaining certain behaviours of language models, like induction heads (Olsson et al., 2022), where attention heads search over the context for previous repetitions of the same token and copy the next token, or even specialized heads solving the Indirect Object Identification (IOI) task (Wang et al., 2023). Similarly, MLPs inside the Transformer have also been studied as elements writing into the residual stream. Geva et al. (2022) observed that MLP blocks can act as key-value memories, where values add to the residual, thus promoting the prediction of words that convey similar semantic meaning.

Furthermore, the *attention mechanism* in the Transformer, composed of attention heads, an output weight matrix, and a layer normalization, can be decomposed into an interpretable operation (Kobayashi et al., 2020, 2021), providing layer-wise explanations which have proven to be highly faithful (Ferrando et al., 2022b,a).

In this work, we propose explaining the predictions of Transformers language generators by combining the residual stream analysis perspective with the attention decomposition. Our approach measures the amount of logit (pre-activation of the softmax) added or subtracted by each token representation at each layer. We then track the logit contributions back to the model's input by aggregating across layers (*Logit* explanation). Additionally, we consider the mixing of information in intermediate layers by using ALTI (Ferrando et al., 2022b) (*ALTI-Logit* explanation).

To evaluate the proposed interpretability methods, we follow the recently introduced contrastive explanations framework (Yin and Neubig, 2022), which aims to explain why the model predicted one token instead of a foil token, *a priori* explained by some linguistic phenomena evidence. Then, we analyze the role of MLPs and show that they aid the model in determining predictions that follow grammar rules. Finally, we demonstrate that NMT models generate human-like source-target alignments for building translations.[1]

2 Approach

2.1 Residual Stream

Given a language generation timestep t, the output of the last layer,[2] $\boldsymbol{x}_t^L \in \mathbb{R}^d$, is projected to the token embedding space by applying the unembedding matrix $\boldsymbol{U} \in \mathbb{R}^{d\times|V|}$ to get the logits of the next token prediction. Then, a softmax function is applied to obtain a probability distribution over the vocabulary:

$$P(\boldsymbol{x}_t^L) = \text{softmax}(\boldsymbol{x}_t^L \boldsymbol{U}) \quad (1)$$

The residual connection in the Transformer can be seen as an information stream (nostalgebraist, 2020; Elhage et al., 2021; Mickus et al., 2022) that gets updated after each block. Let's call $\boldsymbol{o}_t^l$ and $\widetilde{\boldsymbol{x}}_t^l$ the output of the MLP and self-attention blocks at layer l respectively, 'writing' into the residual stream at position t (Figure 1). The last state of the residual stream can be represented as

$$\boldsymbol{x}_t^L = \sum_l^L \boldsymbol{o}_t^l + \sum_l^L \widetilde{\boldsymbol{x}}_t^l + \boldsymbol{x}_t^0 \quad (2)$$

The final logit of a particular next token prediction w can be computed by multiplying the last state of the residual stream with the w-th column[3] of $\boldsymbol{U}$:

$$\begin{aligned} \text{logit}_w &= \boldsymbol{x}_t^L \boldsymbol{U}_w \\ &= \Big(\sum_l^L \boldsymbol{o}_t^l + \sum_l^L \widetilde{\boldsymbol{x}}_t^l + \boldsymbol{x}_t^0\Big)\boldsymbol{U}_w \end{aligned} \quad (3)$$

By linearity:

$$\text{logit}_w = \sum_l^L \boldsymbol{o}_t^l \boldsymbol{U}_w + \sum_l^L \widetilde{\boldsymbol{x}}_t^l \boldsymbol{U}_w + \boldsymbol{x}_t^0 \boldsymbol{U}_w \quad (4)$$

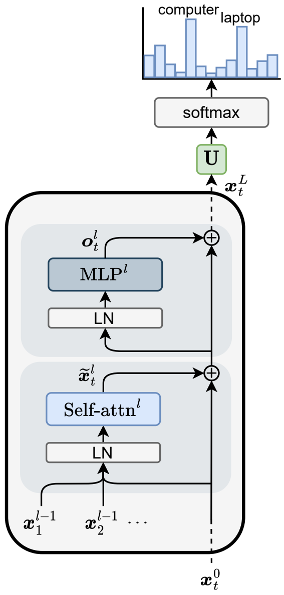

Figure 1: A Transformer Language Model, represented as modules writing into the residual stream.

[1] The code accompanying the paper is available at https://github.com/mt-upc/logit-explanations.

[2] We refer to it as a row vector.

[3] Note that we refer to the j-th column of a matrix $\boldsymbol{B}$ as $\boldsymbol{B}_j$, instead of $\boldsymbol{B}_{:,j}$.

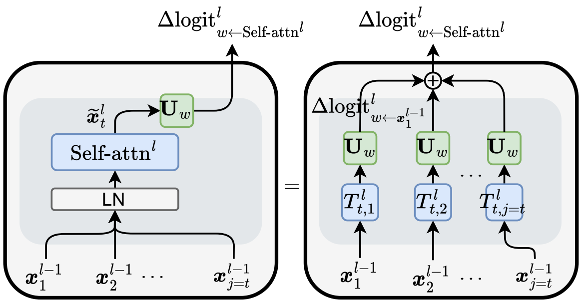

Figure 2: The output of the self-attention block at each layer updates the logit of w (left). The logit's update can be decomposed per input token (right).

2.2 Multi-head Attention as a Sum of Vectors

Inspired by the decomposition of the Post-LN self-attention block done by Kobayashi et al. (2021), we apply a similar approach to the Pre-LN setting, common in current LMs (see full derivation in Appendix A). The output of the self-attention block at each generation step t can be expressed as

$$\tilde{\boldsymbol{x}}_t^l = \sum_j^t T_{t,j}^l(\boldsymbol{x}_j^{l-1}) + \boldsymbol{b}_O^l \tag{5}$$

where $T_{t,j}^l : \mathbb{R}^d \mapsto \mathbb{R}^d$ is an affine transformation applied to each layer's input token representation (or residual stream) $\boldsymbol{x}_j^{l-1} \in \mathbb{R}^d$:

$$T_{t,j}^l(\boldsymbol{x}_j^{l-1}) = \sum_h^H \left(\boldsymbol{x}_j^{l-1} \boldsymbol{L}^l \boldsymbol{W}_V^{l,h} \boldsymbol{A}_{t,j}^{l,h} \boldsymbol{W}_O^{l,h} + \boldsymbol{A}_{t,j}^{l,h} \theta^{l,h} \right) \tag{6}$$

with $\boldsymbol{W}_V^{l,h} \in \mathbb{R}^{d \times d_h}$ the matrix generating the values, $\boldsymbol{W}_O^{l,h} \in \mathbb{R}^{d_h \times d}$ the attention output matrix (per head) and $\boldsymbol{b}_O^l \in \mathbb{R}^d$ its associated bias. $\boldsymbol{A}^{l,h} \in \mathbb{R}^{t \times t}$ is the attention weight matrix of each head, $\theta^{l,h} \in \mathbb{R}^d$ remaining terms originated from biases, and $\boldsymbol{L}^l \in \mathbb{R}^{d \times d}$ combines centering, normalizing, and scaling operations of the layer normalization (see Appendix A).

2.3 Layer-wise Contributions to the Logits

Combining Equation (4) and Equation (5) we get[4]:

$$\text{logit}_w = \underbrace{\sum_l^L \boldsymbol{o}_t^l \boldsymbol{U}_w}_{\Delta\text{logit}^l_{w \leftarrow \text{MLP}^l}} + \sum_l^L \underbrace{\sum_j^t T_{t,j}^l(\boldsymbol{x}_j^{l-1}) \boldsymbol{U}_w}_{\Delta\text{logit}^l_{w \leftarrow \text{Self-attn}^l}} + \boldsymbol{x}_t^0 \boldsymbol{U}_w \tag{7}$$

The logit's update of each self-attention, $\Delta\text{logit}^l_{w \leftarrow \text{Self-attn}^l}$, can be expanded into individual updates by each $\boldsymbol{x}_j^{l-1}$ (Figure 2). Therefore, the contribution of each layer's input token representation $\boldsymbol{x}_j^{l-1}$ to an output token w can be defined as its update to the logit of w:

$$\Delta\text{logit}^l_{w \leftarrow \boldsymbol{x}_j^{l-1}} = T_{t,j}^l(\boldsymbol{x}_j^{l-1}) \boldsymbol{U}_w \tag{8}$$

Similarly, logit updates can be computed at the head level ($\Delta\text{logit}^{l,h}_{w \leftarrow \boldsymbol{x}_j^{l-1}}$) by multiplying the unembedding matrix with the head-wise affine transformation in Equation (6).

[4]Biases are removed to save space.

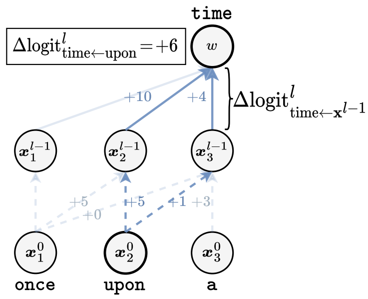

Figure 3: $\boldsymbol{x}_2^{l-1}$ and $\boldsymbol{x}_3^{l-1}$ contribute 10 and 4 logits respectively to the next token prediction $w = \texttt{time}$. Due to the mixing of contextual information across layers, upon contributes $\frac{1}{2}$ to $\boldsymbol{x}_2^{l-1}$ and $\frac{1}{4}$ to $\boldsymbol{x}_3^{l-1}$, which results in upon contributing $10 \cdot \frac{1}{2} + 4 \cdot \frac{1}{4} = 5 + 1 = +6$ logits.

2.4 Tracking Logit Updates to the Input Tokens

If we assume each residual stream preserves its token identity throughout the layers, the total logit update to w produced by input token s can be computed as

$$\Delta\text{logit}_{w \leftarrow s} = \sum_l^L \Delta\text{logit}^l_{w \leftarrow \boldsymbol{x}_{j=s}^{l-1}} \tag{9}$$

that is, the sum of the logit updates performed by the s-th token intermediate representations at every layer. Henceforth, we refer to this as the *Logit* explanation.

However, in intermediate layers, each residual stream represents a mixture of input tokens (Brunner et al., 2020). Therefore, $\Delta\text{logit}^l_{w \leftarrow \boldsymbol{x}_j^{l-1}}$ can't be directly interpreted as the logit update *caused by* the model's input token $s = j$. We propose to track the logit update back to the model inputs by measuring the mixing of contextual information in the residual streams. For that purpose, we use ALTI (Ferrando et al., 2022b). ALTI, as well as other methods relying on the *rollout* method (Abnar and Zuidema, 2020; Mohebbi et al., 2023) assume that token representations are formed by linearly combining the representations from the preceding layer, i.e. $\boldsymbol{x}_i^l = \sum_j c_{i,j}^l \boldsymbol{x}_j^{l-1}$, with $\sum_j c_{i,j}^l = 1$. Each $c_{i,j}^l$ refers to the contribution of $\boldsymbol{x}_j^{l-1}$ to $\boldsymbol{x}_i^l$. By multiplying the layer-wise coefficient matrices, $\boldsymbol{M}^l = \boldsymbol{C}^l \cdot \boldsymbol{C}^2 \cdots \boldsymbol{C}^1$, one can describe each intermediate layer representation as a linear combination of the model input tokens, $\boldsymbol{x}_i^l = \sum_s m_{i,s}^l \boldsymbol{x}_s^0$.

Column s of $\boldsymbol{M}^{l-1}$ contains the proportion of the s-th input token's contribution encoded in each token representation *entering* layer l. We can obtain the update performed by each model input token (Figure 3, right) to the logit of a next prediction token w as

$$\Delta\text{logit}^l_{w\leftarrow s} = \Delta\text{logit}^l_{w\leftarrow \mathbf{x}^{l-1}}\, \boldsymbol{M}_s^{l-1} \tag{10}$$

We refer to Appendix B for a more detailed explanation. The final contribution of the s-th input token to the prediction of token w can be obtained as the sum of its logit updates at each layer:

$$\Delta\text{logit}_{w\leftarrow s} = \sum_l^L \Delta\text{logit}^l_{w\leftarrow s} \tag{11}$$

We denote this method the *ALTI-Logit* explanation. Note that if we don't consider mixing of contextual information, $\boldsymbol{M}^{l-1}$ becomes the identity matrix, and we get the Logit explanation (Equation (9)).

2.5 Contrastive Explanations

Contrastive explanations (Yin and Neubig, 2022) aim to explain why the model predicted one target token w instead of another foil token f. We can explain this decision by determining how much each token contributed to the final logit difference between w and f: $\text{logit}_{(w-f)}$. Following Equation (9) and Equation (11), we can define the Contrastive Logit and Contrastive ALTI-Logit [5] saliency scores of input tokens as their update to the logit difference:

$$\Delta\text{logit}_{(w-f)\leftarrow s} = \Delta\text{logit}_{w\leftarrow s} - \Delta\text{logit}_{f\leftarrow s} \tag{12}$$

3 Experimental Setup

We evaluate the quality of our proposed method through contrastive explanations. Following Yin and Neubig (2022) we use a subset of BLiMP dataset (Warstadt et al., 2020), which contains sentence pairs with small variations in grammatical correctness. The 11 subsets belong to 5 linguistic phenomena: anaphor agreement, argument structure, determiner-noun agreement, NPI licensing, and subject-verb agreement.

Phenomena	ID	Example (Acceptable/Unacceptable)
Anaphor Agreement	aga	Karla could listen to herself/himself.
	ana	Eva approached herself/themselves.
Argument Structure	asp	Gerald is hated by the teachers/pie.
Determiner-Noun Agreement	dna	Eva has scared these children/child.
	dnai	Tammy was observing that man/men.
	dnaa	The driver sees that unlucky person/people.
	dnaai	Phillip liked that smooth horse/horses.
NPI Licensing	npi	Even Danielle also/ever leaves.
Subject-Verb Agreement	darn	The grandfathers of Diana drink/drinks.
	ipsv	Many people have/has hidden away.
	rpsv	Most associations buy/buys those libraries.

Table 2: Examples: in Table 8 of BLiMP phenomenons[6] used by Yin and Neubig (2022), with acceptable and unacceptable continuations in bold. Underlined words represent the linguistic evidence to resolve the phenomena (extracted by the rules).

For each linguistic phenomena, we use spaCy (Honnibal and Montani, 2017) and follow Yin and Neubig (2022) rules to find the evidence (in previous tokens), that is enforcing grammatical acceptability (Table 2). For anaphor agreement, we obtain all context tokens that are coreferent with the target token. For argument structure, we extract the main verb of the sentence. Determiner-noun agreement's evidence is found in the determiner of the target noun. In NPI licensing, "even" word can appear in the acceptable target, but not in the unacceptable. Finally, in the subject-verb agreement phenomenon, the form of the verb has to agree in number with the head of the subject, which we use as evidence. We differ from Yin and Neubig (2022) in that we discard `ipsv` and `rpsv` subsets, due to the large fraction of sentences with a 'quantifier + head of subject + verb' structure, where both the quantifier (many, most...) and the head of the subject could be used by the model to solve the agreement.

We also add to the analysis `SVA` (subject-verb agreement) (Linzen et al., 2016) and the Indirect Object Identification (`IOI`) (Wang et al., 2023; Fahamu, 2022) datasets. The `SVA` dataset includes nouns with an opposite number to that of the main

[5] Throughout the paper we use Logit and ALTI-Logit to refer also to their contrastive variant.

[6] BLiMP IDs. aga: anaphor_gender_agreement; ana: anaphor_number_agreement; asp: animate_subject_passive; dna: determiner_noun_agreement_1; dnai: determiner_noun_agreement_irregular_1; dnaa: determiner_noun_agreement_with_adj_1; dnaai: determiner_noun_agreement_with_adj_irregular_1; npi: npi_present_1; darn: distractor_agreement_relational_noun; ipsv: irregular_plural_subject_verb_agreement_1; rpsv: regular_plural_subject_verb_agreement_1

subject, which makes this dataset well-suited for evaluating saliency methods. Indirect object identification (IOI) is a feature present in sentences that have an initial dependent clause, like "After Lee and Evelyn went to the lake", followed by a main clause, like "Lee gave a grape to Evelyn". The indirect object "Evelyn" and the subject "Lee" are found in the initial clause. In all examples of IOI dataset, the main clause refers to the subject again, which gives an object to the IO. The goal of the IOI task is to predict the final word in the sentence to be the IO. In IOI examples, the rule for predicting the IO is the IO itself being in the first clause.

We use GPT-2 XL (1.5B) model (Radford et al., 2019), as in (Yin and Neubig, 2022), as well as other autoregressive Transformer language models, such as GPT-2 Small (124M), and GPT-2 Large models (774M), OPT 125M (Zhang et al., 2022b), and BLOOM's 560M and 1.1B variants (BigScience Workshop, 2022), through HuggingFace library (Wolf et al., 2020).

Alignment Metrics. Following Yin and Neubig (2022), we define the *evidence* as a binary vector $\boldsymbol{b} \in \mathbb{R}^t$ (with as many components as the number of previous tokens), with all zeros except in the position of the tokens inside the evidence, i.e. the tokens which the prediction depends on, extracted by the rule. Explanations are vectors, also $\in \mathbb{R}^t$. To measure the alignment between an explanation and the evidence we use MRR (Mean Reciprocal Analysis). Sorting the tokens in descending order, MRR evaluates the average of the inverse of the rank of the first token that is part of $\boldsymbol{b}$. Although Yin and Neubig (2022) use also dot-product and Probes Needed metrics for measuring alignments, dot-product favors Grad Norm explanations since it gives positive scores only, and Probes Needed is closely related to MRR, giving redundant results.

4 Contrastive Methods

Yin and Neubig (2022) proposed extending different common input attribution methods to the contrastive setting. In §5 we compare their explanations with the ones obtained with our proposed contrastive methods (Equation (12)).

4.1 Input Erasure

Erasure-based methods remove parts of the input and measure the change in the model's prediction (Li et al., 2016b), where the higher the prediction

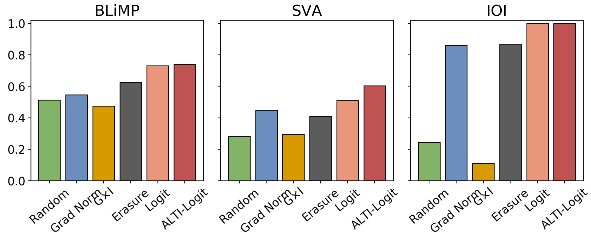

Figure 4: Alignment (MRR ↑) of different explanation methods of GPT-2 Small model predictions with BLiMP, SVA, and IOI datasets.

change, the higher the attribution of that particular token. Specifically, we take the difference between the model's output with the entire input $\mathbf{x}$, and after removing from $\mathbf{x}$ the s-th token, i.e. $m_w(\mathbf{x}) - m_w(\mathbf{x}_{\neg s})$. Yin and Neubig (2022) define the Contrastive Input Erasure as

$$\boldsymbol{c}^e_{(w,\neg f)\leftarrow s} = (m_w(\mathbf{x}) - m_w(\mathbf{x}_{\neg s})) - (m_f(\mathbf{x}) - m_f(\mathbf{x}_{\neg s})) \tag{13}$$

This metric evaluates the extent to which removing x_s from the input increases the likelihood of the foil, and decreases the likelihood of the target in the model's output.

4.2 Gradient Norm

The Transformer model can be approximated by the linear part of the Taylor-expansion at a baseline point (Simonyan et al., 2014), $m(\boldsymbol{X}^0) \approx \nabla m(\boldsymbol{X}^0) \cdot \boldsymbol{X}^0$, where $\boldsymbol{X}^0 \in \mathbb{R}^{t \times d}$ is the sequence of input embeddings. Therefore, $\nabla m_w(\boldsymbol{X}^0)$ represents the sensitivity of the model to each input dimension when predicting w. Following, saliency scores for each token can be computed by taking the norm of the gradient vector corresponding to the token embedding, $\left\|\nabla_{\boldsymbol{x}^0_s} m(\boldsymbol{X}^0)\right\|_1$.

Yin and Neubig (2022) extend this method to the Contrastive Gradient Norm and define it as

$$\boldsymbol{c}^g_{(w,\neg f)\leftarrow s} = \left\|\nabla_{\boldsymbol{x}^0_s} \left(m_w(\boldsymbol{X}^0) - m_f(\boldsymbol{X}^0)\right)\right\|_1 \tag{14}$$

4.3 Gradient × Input

The gradient × input method (Shrikumar et al., 2016; Denil et al., 2014) calculates the dot product between the gradient and the input token embedding. Yin and Neubig (2022) define the Contrastive Gradient × Input as

$$\boldsymbol{c}^{g \times i}_{(w,\neg f)\leftarrow s} = \nabla_{\boldsymbol{x}^0_s} \left(m_w(\boldsymbol{X}^0) - m_f(\boldsymbol{X}^0)\right) \cdot \boldsymbol{x}^0_s \tag{15}$$

Logit
A report about the Impressionists **has**
ALTI-Logit
A report about the Impressionists **has**
Erasure
A report about the Impressionists **has**
Grad Norm
A report about the Impressionists **has**
G×I
A report about the Impressionists **has**

Table 3: Comparison of different contrastive explanation methods described in §4 and ALTI-Logit (**has** vs. **have**). Same example as in Table 1.

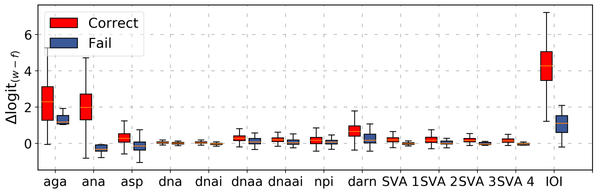

Figure 5: Update to the logit difference between the acceptable and the unacceptable predictions produced by the input tokens inside the linguistic evidence (GPT-2 XL).

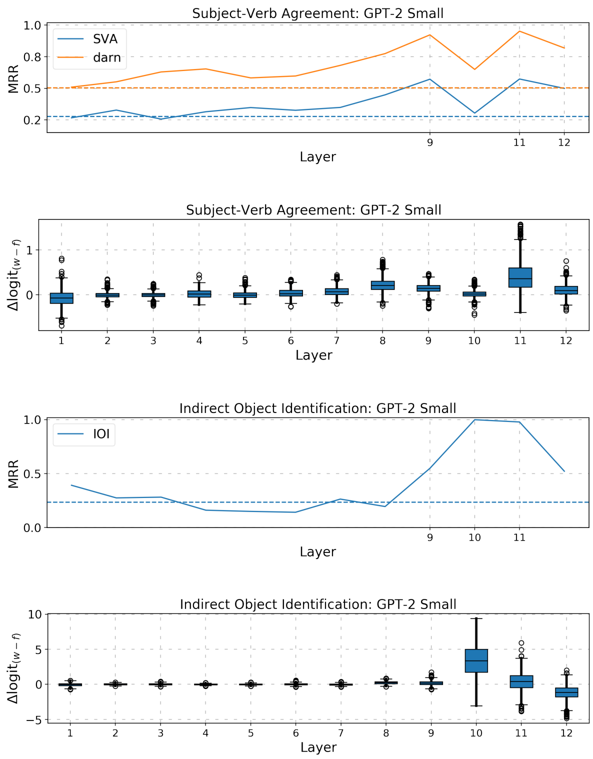

Figure 6: ALTI-Logit MRR alignment scores (line plots) and updates in logit difference by every input token ($\Delta\text{logit}^{l}_{(w-f)\leftarrow\text{Self-attn}^{l}}$) between acceptable and unacceptable predictions (box plots) per layer (GPT-2 Small). Horizontal dashed lines refer to random alignment.

5 Results

In the following sections we provide results on the alignment between the explanations of different methods and linguistic evidence, as well as an analysis of observed model behaviours through the lens of ALTI-Logit.

5.1 Alignment Results

In Figure 4 we present the MRR results of GPT-2 Small averaged across dataset categories, while the extended results for every subset can be found at Appendix C, Table 7. In Appendix C, Figure 11 we expand Figure 4 across different models. We can observe that Logit and ALTI-Logit explanations consistently align better with the evidence of linguistic phenomena than common gradient-based and erasure-based baselines. Note that for BLiMP the average we show in Figure 4 is across 9 different subsets. In Table 3 we show an example comparing different contrastive explanations, where Grad Norm, G×I and Erasure explanations don't align with the evidence to solve the subject-verb agreement (report), and disagree between each other.

We find similar alignment results for Logit and ALTI-Logit methods. However, we observe that ALTI-Logit aligns better at tasks where the tokens of the linguistic evidence are far from the prediction. This is especially noticeable in Subject-verb agreement datasets (including `SVA` and `darn`), where ALTI-Logit shows higher alignments than any other method across all models. This might indicate that incorporating information about contextual mixing is advantageous for dealing with large contexts.

Despite the generally accurate performance of the models examined in this study (Figure 12 and Figure 13, Appendix D), there are cases where the unacceptable token gets predicted with a higher probability. In order to gain a deeper understanding of the variations in model behavior between correct and incorrect predictions, we analyze the logit update generated by the input tokens associated with the linguistic evidence. This analysis, conducted using ALTI-Logit (Figure 5), reveals differences in the distributions. These findings suggest that the tokens representing the linguistic evidence play a crucial role in achieving accurate predictions, and if their contribution is only marginal, the likelihood of failure increases considerably.

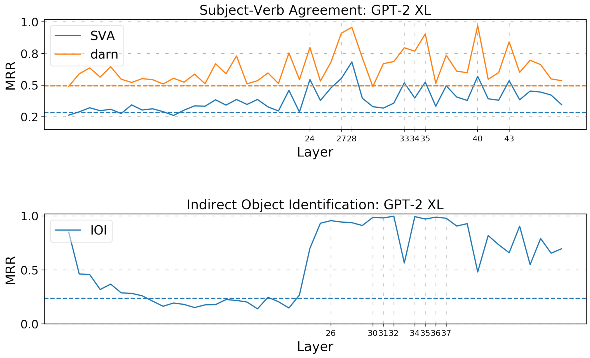

Figure 7: ALTI-Logit MRR alignment scores across layers (GPT-2 XL). Horizontal dashed lines refer to random alignment.

5.2 Layer-wise Analysis with ALTI-Logit

In the line plots in Figures 6 and 7 we provide the MRR alignment results across layers of GPT2-Small and GPT2-XL for two different linguistic phenomena. Models behave similarly across subsets inside the same phenomena, like in Subject-Verb Agreement (`SVA` and `darn`), and Anaphor Agreement (`aga` and `ana`) in Appendix E. The model's alignment trend also stays similar, even though the distance between the prediction and the evidence is different across subsets (`SVA`'s distance is 4 times `darn`'s).

In the boxplots in Figure 6, we show the distribution of self-attention updates to the logit difference between the acceptable and the unacceptable predictions, $\Delta\text{logit}^{l}_{(w-f)\leftarrow\text{Self-attn}^l}$. As a general pattern, we observe that models tend to update more heavily on the layers where the alignment with linguistic phenomena is higher. This conclusion holds for larger models too, see the `darn` example in Appendix H.2, where large logit updates are found in layers 28, 35, and 40, matching the layers where alignment peaks (Figure 7 Top). In `IOI` and `SVA` tasks both models align with the evidence and increase their logit update towards the last layers. This indicates that models solve these phenomena once they have acquired sufficient contextual information.

Our findings in the `IOI` task support those by Wang et al. (2023). In GPT-2 Small we observe high logit difference updates coming from the Indirect Object (IO) in layers 10 and 11. We further study the heads in those layers (Table 4), where Wang et al. (2023) found 'Name Mover Heads' and 'Negative Mover Heads'. These heads rely on the IO to increase (Name Mover Heads) and decrease

Name Mover Head L10 H7
Then, Yvette and Angie were working at the mountain. Yvette decided to give a banana to **Angie**
Name Mover Head L10 H10
Then, Yvette and Angie were working at the mountain. Yvette decided to give a banana to **Angie**
Name Mover Head L11 H1
Then, Yvette and Angie were working at the mountain. Yvette decided to give a banana to **Angie**
Negative Name Mover Head L11 H8
Then, Yvette and Angie were working at the mountain. Yvette decided to give a banana to **Angie**
Negative Name Mover Head L12 H11
Then, Yvette and Angie were working at the mountain. Yvette decided to give a banana to **Angie**

Table 4: GPT-2 Small updates to the logit prediction difference between **Angie** and **Yvette** in different heads produced by layer input token representations ($\Delta\text{logit}^{l,h}_{(w-f)\leftarrow\boldsymbol{x}^{l-1}_j}$).

(Negative Mover Heads) respectively the logit of the correct prediction. In Appendix H.3 we provide an example of how every model solves the task across layers.

6 Analysis of MLPs

The MLP block in the Transformer contains two learnable weight matrices[7]: $\boldsymbol{W}^l_1 \in \mathbb{R}^{d\times d_{mlp}}$ and $\boldsymbol{W}^l_2 \in \mathbb{R}^{d_{mlp}\times d}$, and an element-wise non-linear activation function α. It takes as input the state of the residual stream at timestep t ($\tilde{\boldsymbol{x}}^l_t$) and computes:

$$\boldsymbol{o}^l_t = \alpha(\text{LN}(\tilde{\boldsymbol{x}}^l_t)\boldsymbol{W}^l_1)\boldsymbol{W}^l_2 \tag{16}$$

Following, $\boldsymbol{o}^l_t$ is added back to the residual stream (Figure 1). Equation (16) can be seen as key-value memories (Geva et al., 2021), where keys are stored in components of $\boldsymbol{k}^l = \alpha(\text{LN}(\boldsymbol{x}^l_t)\boldsymbol{W}^l_1) \in \mathbb{R}^{d_{mlp}}$, and values ($\boldsymbol{v}^l$) are rows of $\boldsymbol{W}_2$. Following the key-value perspective, Equation (16) can be rewritten as

$$\boldsymbol{o}^l_t = \sum_i^{d_{mlp}} k^l_i \boldsymbol{v}^l_i \tag{17}$$

where $\boldsymbol{v}^l_i$ represents the i-th row of $\boldsymbol{W}_2$. Recalling how the final logit of a token w is decomposed by layer-wise updates in Equation (7), the MLP^l updates the logit of w as follows:

$$\begin{aligned}\Delta\text{logit}^l_{w\leftarrow\text{MLP}^l} &= \boldsymbol{o}^l_t \boldsymbol{U}^\intercal_w \\ &= \sum_i^{d_{mlp}} k^l_i \boldsymbol{v}^l_i \boldsymbol{U}^\intercal_w \\ &= \sum_i^{d_{mlp}} \Delta\text{logit}^l_{w\leftarrow k^l_i \boldsymbol{v}^l_i}\end{aligned} \tag{18}$$

Thus, the update of the MLP can be decomposed into sub-updates (Geva et al., 2022) performed by

[7]We omit bias terms.

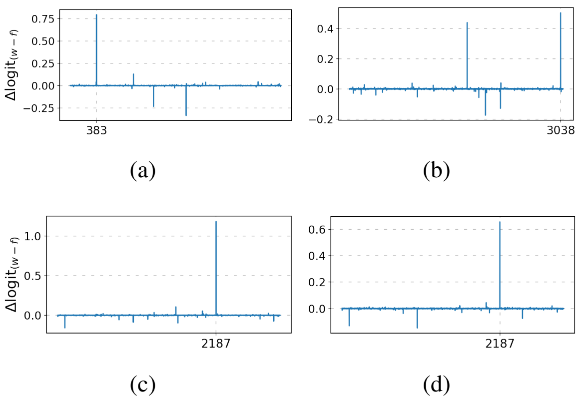

Figure 8: Average (across the dataset) of the updates to the logit difference caused by the weighted values in the MLP (each row i in $\boldsymbol{W}_2^l$), $\Delta\text{logit}^l_{(w-f)\leftarrow k_i^l \boldsymbol{v}_i^l}$. a) dna: dimension $i = 383$ (L11) promotes singular nouns (increases the logit difference between singular and plural nouns) after this/that, b) dna: dimension $i = 3038$ (L11) promotes plural nouns after these/those. Dimension $i = 2187$ (L12) pushes the prediction of singular verbs in different Subject-Verb Agreement datasets c) darn and d) SVA.

each $k_i^l \boldsymbol{v}_i^l$ (weighted row in $\boldsymbol{W}_2^l$). The update in the logit's difference between the target and foil tokens by each value i is therefore:

$$\Delta\text{logit}^l_{(w-f)\leftarrow k_i^l \boldsymbol{v}_i^l} = \Delta\text{logit}^l_{w\leftarrow k_i^l \boldsymbol{v}_i^l} - \Delta\text{logit}^l_{f\leftarrow k_i^l \boldsymbol{v}_i^l} \tag{19}$$

In Figure 8, we show some examples of the contribution of each weighted value $k_i^l \boldsymbol{v}_i^l$ to the logit difference between the acceptable target token and the unacceptable one, at different layers and datasets. We can observe that there is a small subset of values that consistently increase the difference in logits helping to solve the linguistic task. Some of them include the value $i = 383$ in layer 10 (Figure 8 (a)), which increases the logit of singular nouns and reduces the plural ones when the determiner is this or that. For instance, in the sentence "William described this ___", value $i = 383$ increases the logit difference between movie and movies. In dimension 3038 we find a value upweighting the logits of the plural nouns over the singular ones when the determiner is these or those (Figure 8 (b)). These values help solve the linguistic task at hand across different subsets, for instance, the value in dimension $i = 2187$ is in charge of promoting the singular form of the verb when the head of the subject is singular too. This occurs in both darn and SVA subsets.

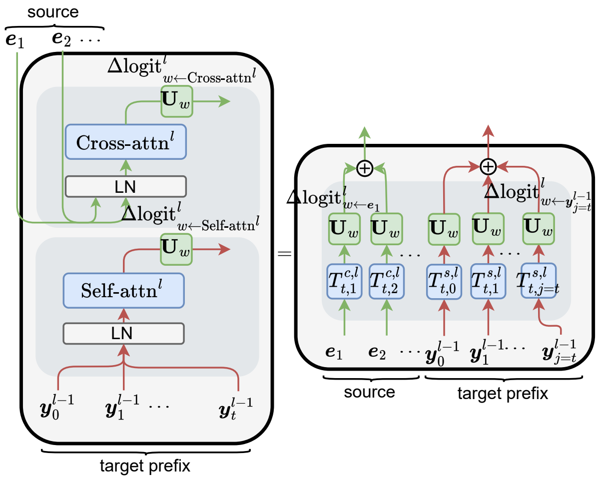

Figure 9: Cross-attention block in the Transformer's decoder (left) and its equivalent using vector transformations (right). Depicted in green and red it's shown the information coming from the encoder and the decoder (target prefix) respectively.

7 Neural Machine Translation

An NMT system estimates the likelihood of a target sequence of tokens, $\mathbf{y} = (y_1, \ldots, y_t)$, given a source sequence of tokens, $\mathbf{x} = (x_1, \ldots, x_I)$:

$$P(\mathbf{y}|\mathbf{x}) = \prod_s^t P(y_s|\mathbf{y}_{<s}, \mathbf{x}) \tag{20}$$

where $\mathbf{y}_{<s} = (y_0, \ldots, y_{s-1})$ is the prefix of y_s, and $x_I = y_0 =$ </s> is a special token used to mark the start and end of the sentence. The encoder processes the source sentence and generates a sequence of contextualized representations, $\mathbf{e} = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_I)$. At each decoding step t, the decoder uses the encoder outputs and the target prefix to compute a probability distribution over the target vocabulary.

Cross-attention. Similar to Equation (6), the output of the cross-attention ($\widetilde{\boldsymbol{y}}_t^{c,l}$) and self-attention ($\widetilde{\boldsymbol{y}}_t^{s,l}$) (Figure 9) of a decoder layer in an encoder-decoder Transformer can be decomposed[8] as

$$\widetilde{\boldsymbol{y}}_t^{c,l} = \sum_i^I T_{t,i}^{c,l}(\boldsymbol{e}_i), \quad \widetilde{\boldsymbol{y}}_t^{s,l} = \sum_j^t T_{t,j}^{s,l}(\boldsymbol{y}_j^{l-1}) \tag{21}$$

As shown in Figure 9, each transformed vector updates the logits of the token predictions by multiplying it with the corresponding column of $\boldsymbol{U}$, as in Equation (8):

$$\Delta\text{logit}^l_{w\leftarrow \boldsymbol{e}_i} = T_{t,i}^{c,l}(\boldsymbol{e}_i)\boldsymbol{U}_w \tag{22}$$

[8]Removing biases.

Method	AER (↓) Bilingual	M2M
Attention weights	48.6	96.4
SD-SmoothGrad (Ding et al., 2019)	36.4	-
Vector Norms (Kobayashi et al., 2020)	41.4	-
Distance Vectors-Output (Ferrando et al., 2022a)	38.8	36.4
Proposed alignment extraction	**26.0**	**27.3**

Table 5: Mean AER of the cross-attention contributions in the best layer of the bilingual and M2M models. For the bilingual model, we show the average on five different seeds.

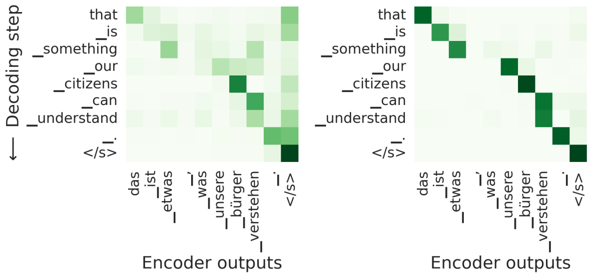

Figure 10: Left: attention weights in the cross-attention in the penultimate layer. Right: contributions obtained as logit updates to token predictions in the penultimate layer.

Alignment. Source-target alignments derived from attention weights in NMT systems can be unreliable (Zenkel et al., 2019; Li et al., 2019; Garg et al., 2019), with upper layers producing better alignments. A limitation of using this method to interpret model predictions is that the ground truth target word may not match the model's actual prediction. However, by measuring how the encoder token representations update the logits of the reference words, $\Delta\text{logit}^{l}_{w\leftarrow \boldsymbol{e}_i}$, we can more precisely explain which source word causes the final logit of the reference word, even if it is not one of the top predictions.

Following Kobayashi et al. (2020) and Ding et al. (2019) setting, we train a 6-layer Transformer model for the German-English (De-En) translation task using Europarl v7 corpus[9] Koehn (2005). We also evaluate on M2M, a 12 layer multilingual model (Fan et al., 2021). We use Vilar et al. (2006) dataset, consisting of 508 De-En human annotated sentence pairs with alignments, and compare them with our extracted alignments using Alignment Error Rate (AER). We also show results of other attention-based alignments extraction methods. Vector Norms take the norm of the transformed vectors in Equation (21), Distance Vectors-Output measures the distance between the transformed vectors and the attention block output $\widetilde{\boldsymbol{y}}^{c,l}_{t}$. SD-SmoothGrad relies on gradients to extract alignments. In Table 5 we show that our proposed method achieves lower AER values, which indicates that NMT models generate human-like alignments for building model predictions.

8 Related Work

The projection of LMs representations and model parameters to the vocabulary space has been a subject of previous research (Belrose et al., 2023; Din et al., 2023). Geva et al. (2021, 2022) view feed-forward layers as performing updates to the probability distribution of the token predictions. Mickus et al. (2022) study how the different Transformer modules contribute to the hidden representations, and Dar et al. (2022) directly interpret Transformer static parameters in the embedding space. In this work, our focus lies in interpreting the influence of input tokens and its representations in the model predictions.

Furthermore, work on mechanistic interpretability (Olah, 2022) has discovered 'circuits' within LMs in charge of solving tasks (Wang et al., 2023; Geva et al., 2023). In contrast to their methods, our approach does not rely on causal interventions in the computations of Transformers. More broadly, our work can be related to those explaining the prediction process of LMs (Tenney et al., 2019; Voita et al., 2019; Sarti et al., 2023).

9 Conclusions

In this paper, we introduce a new procedure for analyzing language generation models by combining the residual stream perspective with interpretable attention decomposition, and tested our approach using contrastive examples in Transformer LMs. We found that the explanations provided by our proposed methods, Logit and ALTI-Logit, align better with available linguistic evidence in the context of the sentence, compared to common gradient-based and erasure-based baselines. We also analyzed the role of MLPs and showed that they assist the model in determining predictions that conform to the grammar rules. Additionally, we applied our method to a Machine Translation model and demonstrated that it generates human-like alignments for building predictions. Overall, our results suggest that decomposing the logit scores is an effective way to analyze language generation models.

[9] http://www.statmt.org/europarl/v7

10 Limitations

The experimental methodology employed in this study for both contrastive explanations and NMT is not directly extensible to languages other than English, due to the scarcity of resources such as models and annotations.

The datasets employed in this study to evaluate contrastive explanations across various linguistic paradigms are restricted to sentences that possess a well-defined structure. As a result, it is possible that the conclusions drawn may not be generalizable to the broader distribution of sentences.

Lastly, it should be noted that the method proposed in this study should not be used as a definitive explanation of model predictions in any other context. It is recommended to use the method as a debugging tool and should be employed in conjunction with other methods to gain a comprehensive understanding of model predictions.

11 Ethics statement

It is acknowledged that the experiments reported in this study are limited to high-resource languages. However, the methodology employed is language-independent and may be applied to other languages in the future, provided that adequate annotated data becomes available.

12 Acknowledgements

We would like to thank the anonymous reviewers for their useful comments. Javier Ferrando, Gerard I. Gállego and Ioannis Tsiamas are supported by the Spanish Ministerio de Ciencia e Innovación through the project PID2019-107579RB-I00 / AEI / 10.13039/501100011033.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, Online. Association for Computational Linguistics.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Jasmijn Bastings, Sebastian Ebert, Polina Zablotskaia, Anders Sandholm, and Katja Filippova. 2022. "will you find these shortcuts?" a protocol for evaluating the faithfulness of input salience methods for text classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 976–991, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens.

BigScience Workshop. 2022. BLOOM (revision 4ab0472).

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *International Conference on Learning Representations*.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2022. Analyzing transformers in embedding space.

Misha Denil, Alban Demiraj, and Nando de Freitas. 2014. Extraction of salient sentences from labelled documents. *CoRR*, abs/1412.6815.

Alexander Yom Din, Taelin Karidi, Leshem Choshen, and Mor Geva. 2023. Jump to conclusions: Short-cutting transformers with linear transformations.

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*.

Fahamu. 2022. ioi (revision 223da8b).

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Michael Auli, and Armand Joulin. 2021. Beyond english-centric multilingual machine translation. *Journal of Machine Learning Research*, 22(107):1–48.

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022a. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8756–8769*.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022b. Measuring the mixing of contextual information in the transformer. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 8698–8714*.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly Learning to Align and Translate with Transformer Models. *arXiv:1909.02074 [cs]*. ArXiv: 1909.02074.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. Incorporating Residual and Normalization Layers into Analysis of Masked Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4547–4568, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. Understanding neural networks through representation erasure.

Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. On the word alignment from neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Trans. Assoc. Comput. Linguistics*, 4:521–535.

Timothee Mickus, Denis Paperno, and Mathieu Constant. 2022. How to Dissect a Muppet: The Structure of Transformer Embedding Spaces. *Transactions of the Association for Computational Linguistics*, 10:981–996.

Hosein Mohebbi, Willem Zuidema, Grzegorz Chrupała, and Afra Alishahi. 2023. Quantifying context mixing in transformers. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3378–3400, Dubrovnik, Croatia. Association for Computational Linguistics.

nostalgebraist. 2020. interpreting gpt: the logit lens. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`.

Chris Olah. 2022. Mechanistic interpretability, variables, and the importance of interpretable bases. `https://transformer-circuits.pub/2022/mech-interp-essay/index.html`.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.

A. Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. In *OpenAI Blog*.

Gabriele Sarti, Nils Feldhus, Ludwig Sickert, Oskar van der Wal, Malvina Nissim, and Arianna Bisazza. 2023. Inseq: An interpretability toolkit for sequence generation models. *ArXiv*, abs/2302.13942.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Learning important features through propagating activation differences. *CoRR*, abs/1605.01713.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328, International Convention Centre, Sydney, Australia. 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, Florence, Italy. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

David Vilar, Maja Popovic, and H. Ney. 2006. Aer: do we need to "improve" our alignments? In *IWSLT*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019. The bottom-up evolution of representations in the transformer: A study with machine translation and language modeling objectives. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4396–4406, Hong Kong, China. Association for Computational Linguistics.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 184–198*.

Kerem Zaman and Yonatan Belinkov. 2022. A multilingual perspective towards the evaluation of attribution methods in natural language inference.

Thomas Zenkel, Joern Wuebker, and John DeNero. 2019. Adding interpretable attention to neural translation models improves word alignment. *CoRR*, abs/1901.11359.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022a. Opt: Open pre-trained transformer language models.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel

Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022b. Opt: Open pre-trained transformer language models.

A Pre-LN Self-attention Decomposition

$\boldsymbol{x}_j^{l-1} \in \mathbb{R}^{d \times d_h}$	Layer Input (Residual Stream position j)
$\boldsymbol{A}^{l,h} \in \mathbb{R}^{t \times t}$	Attention Matrix
$\boldsymbol{W}_V^{l,h} \in \mathbb{R}^{d \times d_h}$	Values Weight Matrix
$\boldsymbol{W}_O^{l,h} \in \mathbb{R}^{d_h \times d}$	Output Weight Matrix (per head)
$\boldsymbol{b}_V^{l,h} \in \mathbb{R}^{d_h}$	Value bias
$\boldsymbol{b}_O^{l} \in \mathbb{R}^{d}$	Output bias
$H \in \mathbb{R}$	Number of heads
$\mathrm{LN}^l : \mathbb{R}^d \mapsto \mathbb{R}^d$	Layer Normalization

Table 6: Components of the self-attention module.

At position t, each head of a Pre-LN self-attention mechanism computes:

$$\boldsymbol{z}_t^{l,h} = \sum_j^t \underbrace{\left(\mathrm{LN}^l(\boldsymbol{x}_j^{l-1})\boldsymbol{W}_V^{l,h} + \boldsymbol{b}_V^{l,h}\right)}_{j\text{-th value}} \boldsymbol{A}_{t,j}^{l,h} \tag{23}$$

By representing attention heads as parallel independent components, we can express the output of the self-attention as

$$\widetilde{\boldsymbol{x}}_t^l = \sum_h^H \boldsymbol{z}_t^{l,h}\boldsymbol{W}_O^{l,h} + \boldsymbol{b}_O^l \tag{24}$$

leading to:

$$\widetilde{\boldsymbol{x}}_t^l = \sum_j^t \sum_h^H \left(\mathrm{LN}^l(\boldsymbol{x}_j^{l-1})\boldsymbol{W}_V^{l,h} + \boldsymbol{b}_V^{l,h}\right) \boldsymbol{A}_{t,j}^{l,h}\boldsymbol{W}_O^{l,h} + \boldsymbol{b}_O^l \tag{25}$$

The layer normalization computes:

$$\mathrm{LN}^l(\boldsymbol{x}_j^{l-1}) = \frac{\boldsymbol{x}_j^{l-1} - \mu(\boldsymbol{x}_j^{l-1})}{\sigma(\boldsymbol{x}_j^{l-1})} \odot \gamma^l + \beta^l \tag{26}$$

with μ and σ computing the mean and standard deviation, and $\gamma^l \in \mathbb{R}^d$ and $\beta^l \in \mathbb{R}^d$ refer to learned element-wise transformation and bias respectively. Considering $\sigma(\boldsymbol{x}_j^{l-1})$ as a constant, LN can be treated as a constant affine transformation:

$$\mathrm{LN}(\boldsymbol{x}_j^{l-1}) = \boldsymbol{x}_j^{l-1}\boldsymbol{L}^l + \beta^l \tag{27}$$

where $\boldsymbol{L}^l \in \mathbb{R}^{d \times d}$ represents a matrix that combines centering, normalizing, and scaling operations together.

Using Equation (27) in Equation (25):

$$\begin{aligned}
\widetilde{\boldsymbol{x}}_t^l &= \sum_j^t \sum_h^H \left(\left((\boldsymbol{x}_j^{l-1}\boldsymbol{L}^l + \beta^l)\boldsymbol{W}_V^{l,h} + \boldsymbol{b}_V^{l,h}\right) \boldsymbol{A}_{t,j}^{l,h}\boldsymbol{W}_O^{l,h}\right) + \boldsymbol{b}_O^l \\
&= \sum_j^t \sum_h^H \left(\left(\boldsymbol{x}_j^{l-1}\boldsymbol{L}^l\boldsymbol{W}_V^{l,h} + \beta^l\boldsymbol{W}_V^{l,h} + \boldsymbol{b}_V^{l,h}\right) \boldsymbol{A}_{t,j}^{l,h}\boldsymbol{W}_O^{l,h}\right) + \boldsymbol{b}_O^l \\
&= \sum_j^t \sum_h^H \left(\boldsymbol{x}_j^{l-1}\boldsymbol{L}^l\boldsymbol{W}_V^{l,h}\boldsymbol{A}_{t,j}^{l,h}\boldsymbol{W}_O^{l,h} + \beta^l\boldsymbol{W}_V^{l,h}\boldsymbol{A}_{t,j}^{l,h}\boldsymbol{W}_O^{l,h} + \boldsymbol{b}_V^{l,h}\boldsymbol{A}_{t,j}^{l,h}\boldsymbol{W}_O^{l,h}\right) + \boldsymbol{b}_O^l
\end{aligned}$$

$$= \sum_j^t \sum_h^H \left(\boldsymbol{x}_j^{l-1} \boldsymbol{L}^l \boldsymbol{W}_V^{l,h} \boldsymbol{A}_{t,j}^{l,h} \boldsymbol{W}_O^{l,h} + \boldsymbol{A}_{t,j}^{l,h} \left(\beta^l \boldsymbol{W}_V^{l,h} \boldsymbol{W}_O^{l,h} + \boldsymbol{b}_V^{l,h} \boldsymbol{W}_O^{l,h} \right) \right) + \boldsymbol{b}_O^l \tag{28}$$

Considering $\theta^{l,h} = \left(\beta^l \boldsymbol{W}_V^{l,h} + \boldsymbol{b}_V^{l,h} \right) \boldsymbol{W}_O^{l,h}$

$$\widetilde{\boldsymbol{x}}_t^l = \sum_j^t \sum_h^H \left(\boldsymbol{x}_j^{l-1} \boldsymbol{L}^l \boldsymbol{W}_V^{l,h} \boldsymbol{A}_{t,j}^{l,h} \boldsymbol{W}_O^{l,h} + \boldsymbol{A}_{t,j}^{l,h} \theta^{l,h} \right) + \boldsymbol{b}_O^l \tag{29}$$

For each j-th input term, H affine transformations are applied to $\boldsymbol{x}_j$. Furthermore, all heads' operations can be further grouped into a single affine transformation:

$$\widetilde{\boldsymbol{x}}_t^l = \sum_j^t \left(\boldsymbol{x}_j^{l-1} \boldsymbol{L}^l \sum_h^H \boldsymbol{W}_V^{l,h} \boldsymbol{A}_{t,j}^{l,h} \boldsymbol{W}_O^{l,h} + \sum_h^H \boldsymbol{A}_{t,j}^{l,h} \theta^{l,h} \right) + \boldsymbol{b}_O^l \tag{30}$$

So, we can write $\widetilde{\boldsymbol{x}}_t^l$ as a sum of t affine transformations, and the output bias:

$$\widetilde{\boldsymbol{x}}_t^l = \sum_j^t T_{t,j}^l(\boldsymbol{x}_j^{l-1}) + \boldsymbol{b}_O^l \tag{31}$$

B Tracking Logits to the Input with Rollout

The rollout method (Abnar and Zuidema, 2020) assumes any intermediate representation is a linear combination of the model inputs, $\boldsymbol{x}_j^{l-1} = \sum_s m_{j,s}^{l-1} \boldsymbol{x}_s^0$, where $m_{j,s}^{l-1}$ is a score indicating the contribution of input token s to the $l-1$ representation (or residual path) of token j. By dividing the logit update performed by $\boldsymbol{x}_j^{l-1}$ among the model inputs ($\Delta\text{logit}_{w,j\leftarrow \boldsymbol{x}_s^0}^l$) based on their contributions to $\boldsymbol{x}_j^{l-1}$, we obtain:

$$\begin{aligned} \Delta\text{logit}_{w\leftarrow \boldsymbol{x}_j^{l-1}}^l &= \sum_s \Delta\text{logit}_{w,j\leftarrow \boldsymbol{x}_s^0}^l \\ &= \sum_s m_{j,s}^{l-1} \Delta\text{logit}_{w\leftarrow \boldsymbol{x}_j^{l-1}}^l \end{aligned} \tag{32}$$

Based on the total logit update produced in layer l, we have that:

$$\begin{aligned} \Delta\text{logit}_{w\leftarrow \text{Self-attn}^l}^l &= \sum_j \Delta\text{logit}_{w\leftarrow \boldsymbol{x}_j^{l-1}}^l \\ &= \sum_j \sum_s \Delta\text{logit}_{w,j\leftarrow \boldsymbol{x}_s^0}^l \\ &= \sum_j \sum_s m_{j,s}^{l-1} \Delta\text{logit}_{w\leftarrow \boldsymbol{x}_j^{l-1}}^l \\ &= \sum_s \sum_j m_{j,s}^{l-1} \Delta\text{logit}_{w\leftarrow \boldsymbol{x}_j^{l-1}}^l \\ &= \sum_s \Delta\text{logit}_{w\leftarrow s}^l \end{aligned} \tag{33}$$

So, we have obtained Equation (10):

$$\Delta\text{logit}_{w\leftarrow s}^l = \Delta\text{logit}_{w\leftarrow \mathbf{x}^{l-1}}^l \, \boldsymbol{M}_s^{l-1} \tag{34}$$

C Results

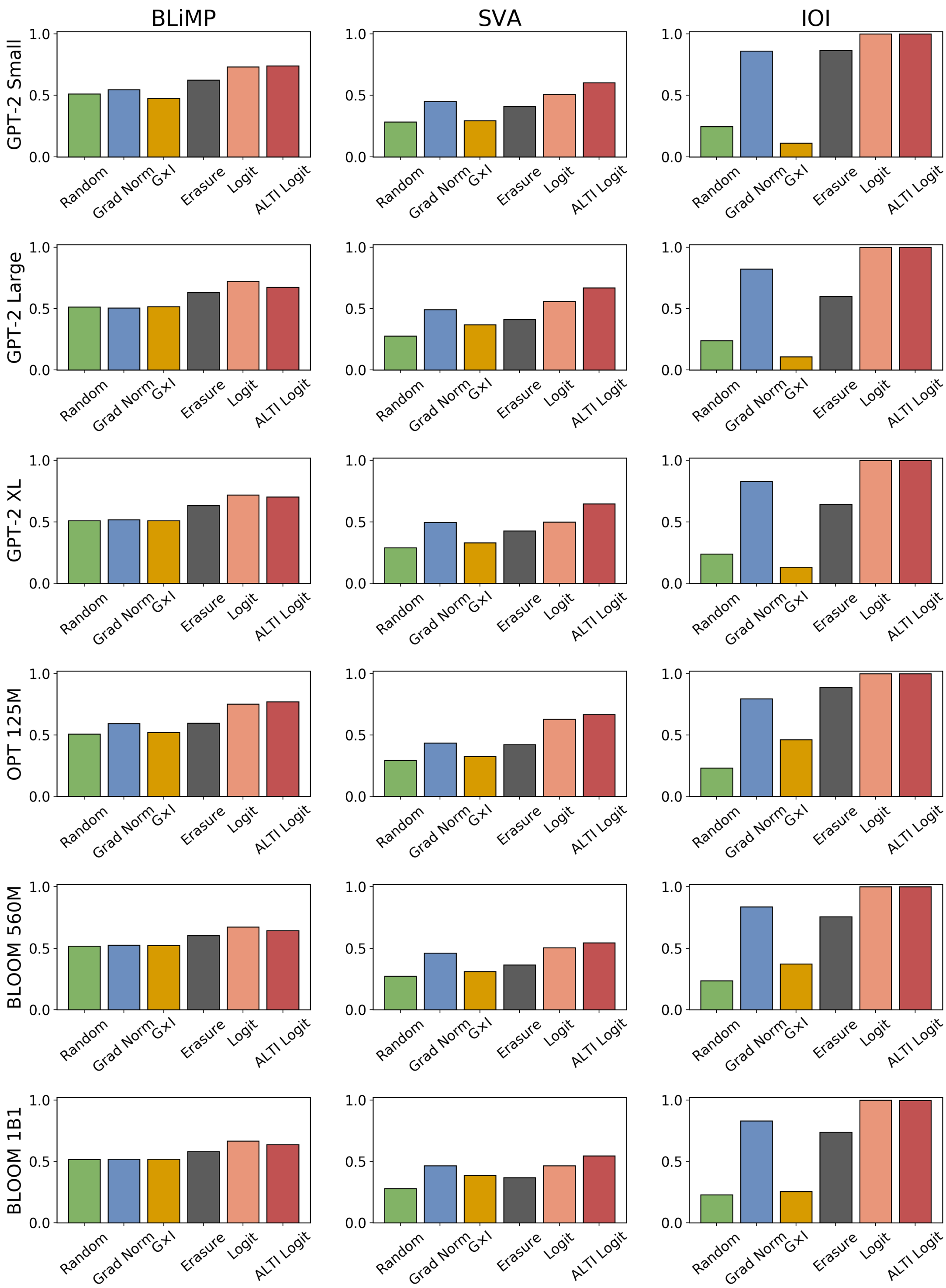

Figure 11: Alignment (MRR ↑) of different explanation methods of GPT-2 Small, Large, and XL, OPT 125M, BLOOM 560M, and BLOOM 1B1 model predictions with BLiMP, SVA, and IOI datasets.

C.1 GPT-2 Small Results

Dataset	Erasure	Logit	ALTI-Logit	Grad Norm	G×I	Random	Distance
aga	0.959	0.827	0.964	0.793	0.791	0.699	3.2
ana	0.963	0.817	0.976	0.675	0.739	0.716	3.2
asp	0.492	0.386	0.499	0.751	0.409	0.381	3.3
dna	0.35	0.737	0.646	0.363	0.387	0.459	1
dnai	0.374	0.711	0.637	0.408	0.432	0.466	1
dnaa	0.61	0.951	0.807	0.263	0.321	0.397	2.1
dnaai	0.659	0.9	0.757	0.263	0.339	0.406	2.1
npi	0.663	0.445	0.417	0.785	0.495	0.599	3.2
darn	0.557	0.802	0.949	0.617	0.363	0.488	3.9
SVA 1	0.389	0.558	0.641	0.432	0.298	0.333	8
SVA 2	0.425	0.57	0.606	0.421	0.303	0.292	11.6
SVA 3	0.454	0.459	0.603	0.51	0.356	0.259	12.9
SVA 4	0.371	0.454	0.566	0.433	0.222	0.249	16.4
IOI	0.865	1.0	1.0	0.86	0.111	0.245	14.9

Table 7: MRR Alignment of different explanation methods on GPT-2 Small predictions on every dataset. The average distance to the linguistic evidence tokens is shown in the last column.

C.2 GPT-2 XL Results

Dataset	Erasure	Logit	ALTI-Logit	Grad Norm	G×I	Random	Distance
aga	0.974	0.79	0.974	0.778	0.713	0.681	3.2
ana	0.945	0.777	0.964	0.721	0.655	0.71	3.2
asp	0.506	0.368	0.514	0.721	0.44	0.369	3.3
dna	0.326	0.655	0.539	0.255	0.486	0.465	1
dnai	0.366	0.598	0.524	0.264	0.515	0.453	1
dnaa	0.631	0.932	0.615	0.205	0.352	0.413	2.1
dnaai	0.644	0.874	0.529	0.205	0.359	0.393	2.1
npi	0.735	0.602	0.711	0.82	0.586	0.594	3.2
darn	0.576	0.873	0.945	0.686	0.477	0.51	3.9
SVA 1	0.416	0.564	0.638	0.467	0.365	0.352	8
SVA 2	0.455	0.558	0.646	0.489	0.353	0.269	11.6
SVA 3	0.424	0.455	0.678	0.535	0.343	0.31	12.9
SVA 4	0.411	0.418	0.625	0.489	0.256	0.226	16.4
IOI	0.643	1.0	1.0	0.829	0.131	0.239	14.9

Table 8: MRR Alignment of different explanation methods on GPT-2 XL predictions on every dataset. The average distance to the linguistic evidence tokens is shown in the last column.

D Model Predictions

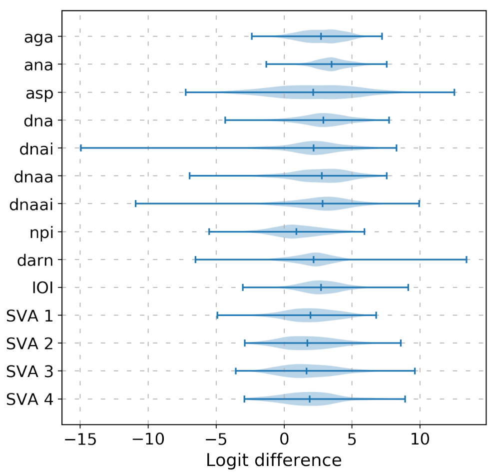

Figure 12: Logit difference between the acceptable and the unacceptable predictions of a GPT-2 Small on every dataset.

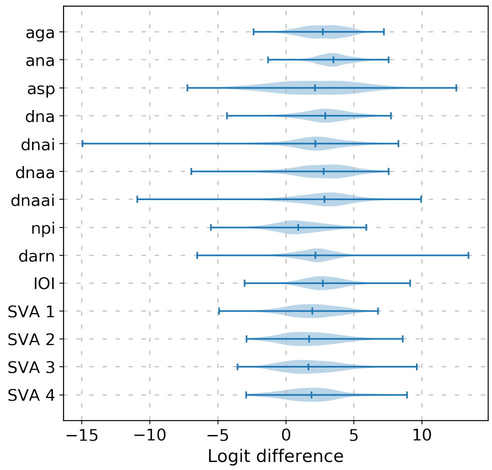

Figure 13: Logit difference between the acceptable and the unacceptable predictions of a GPT-2 XL on every dataset.

E MRR Alignment across layers

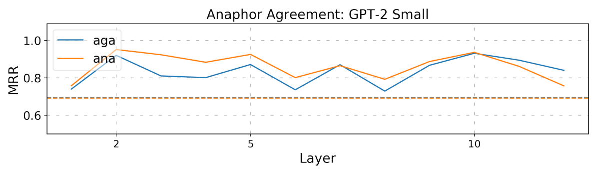

Figure 14: ALTI-Logit MRR alignment scores across layers on Anaphor Agreement datasets (GPT-2 Small).

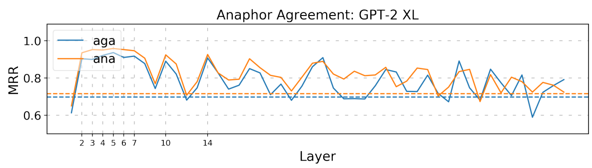

Figure 15: ALTI-Logit MRR alignment scores across layers on Anaphor Agreement datasets (GPT-2 XL).

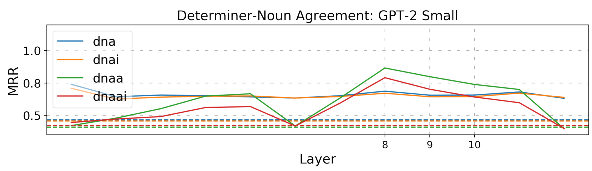

Figure 16: ALTI-Logit MRR alignment scores across layers on Determiner-Noun Agreement datasets (GPT-2 Small).

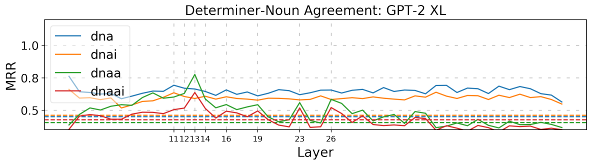

Figure 17: ALTI-Logit MRR alignment scores across layers on Determiner-Noun Agreement datasets (GPT-2 XL).

F MLPs Logit Difference Update

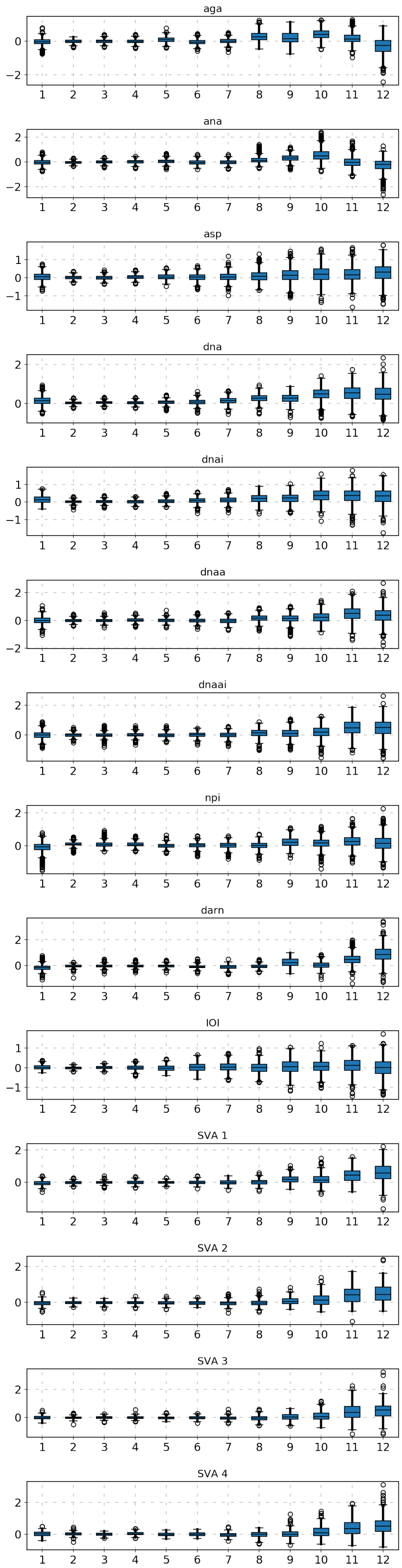

Figure 18: MLPs update to the logit difference $\Delta\text{logit}^{l}_{(w-f)\leftarrow \text{MLP}^{l}}$ across layers (GPT-2 Small).

G Self-attention Logit Difference Update

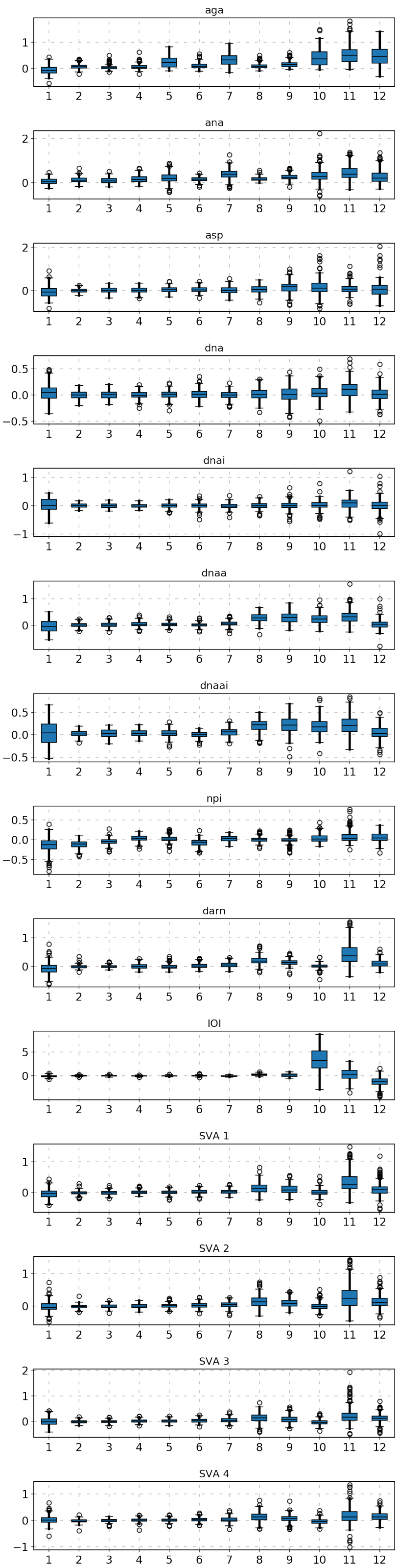

Figure 19: Self-attention update to the logit difference $\Delta\text{logit}^l_{(w-f)\leftarrow\text{Self-attn}^l}$ across layers (GPT-2 Small).

H Qualitative Contrastive Exaplantions

H.1 Explanations of Different Contrastive Methods

Logit
a 2006 guide to the churches of anglesey **says**
ALTI-Logit
a 2006 guide to the churches of anglesey **says**
Erasure
a 2006 guide to the churches of anglesey **says**
Grad Norm
a 2006 guide to the churches of anglesey **says**
G×I
a 2006 guide to the churches of anglesey **says**

Table 9: Comparison of different contrastive explanations on a GPT-2 Small SVA example (why **says** instead of **say**).

Logit
Diane should complain about these unconvinced **drivers**
ALTI-Logit
Diane should complain about these unconvinced **drivers**
Erasure
Diane should complain about these unconvinced **drivers**
Grad Norm
Diane should complain about these unconvinced **drivers**
G×I
Diane should complain about these unconvinced **drivers**

Table 10: Comparison of different contrastive explanations on a GPT-2 Small dnaa example (why **drivers** instead of **driver**).

Logit
Amanda isn't respected by the **children**
ALTI-Logit
Amanda isn't respected by the **children**
Erasure
Amanda isn't respected by the **children**
Grad Norm
Amanda isn't respected by the **children**
G×I
Amanda isn't respected by the **children**

Table 11: Comparison of different contrastive explanations on a GPT-2 Small asp example (why **children** instead of **cups**).

H.2 GPT-2 XL ALTI-Logit across layers

L48	A report about the Impressionists **has**
L47	A report about the Impressionists **has**
L46	A report about the Impressionists **has**
L45	A report about the Impressionists **has**
L44	A report about the Impressionists **has**
L43	A report about the Impressionists **has**
L42	A report about the Impressionists **has**
L41	A report about the Impressionists **has**
L40	A report about the Impressionists **has**
L39	A report about the Impressionists **has**
L38	A report about the Impressionists **has**
L37	A report about the Impressionists **has**
L36	A report about the Impressionists **has**
L35	A report about the Impressionists **has**
L34	A report about the Impressionists **has**
L33	A report about the Impressionists **has**
L32	A report about the Impressionists **has**
L31	A report about the Impressionists **has**
L30	A report about the Impressionists **has**
L29	A report about the Impressionists **has**
L28	A report about the Impressionists **has**
L27	A report about the Impressionists **has**
L26	A report about the Impressionists **has**
L25	A report about the Impressionists **has**
L24	A report about the Impressionists **has**
L23	A report about the Impressionists **has**
L22	A report about the Impressionists **has**
L21	A report about the Impressionists **has**
L20	A report about the Impressionists **has**
L19	A report about the Impressionists **has**
L18	A report about the Impressionists **has**
L17	A report about the Impressionists **has**
L16	A report about the Impressionists **has**
L15	A report about the Impressionists **has**
L14	A report about the Impressionists **has**
L13	A report about the Impressionists **has**
L12	A report about the Impressionists **has**
L11	A report about the Impressionists **has**
L10	A report about the Impressionists **has**
L9	A report about the Impressionists **has**
L8	A report about the Impressionists **has**
L7	A report about the Impressionists **has**
L6	A report about the Impressionists **has**
L5	A report about the Impressionists **has**
L4	A report about the Impressionists **has**
L3	A report about the Impressionists **has**
L2	A report about the Impressionists **has**
L1	A report about the Impressionists **has**

Table 12: GPT-2 XL `darn` (why **has** instead of **have**).

L48	Katherine can't help **herself**
L47	Katherine can't help **herself**
L46	Katherine can't help **herself**
L45	Katherine can't help **herself**
L44	Katherine can't help **herself**
L43	Katherine can't help **herself**
L42	Katherine can't help **herself**
L41	Katherine can't help **herself**
L40	Katherine can't help **herself**
L39	Katherine can't help **herself**
L38	Katherine can't help **herself**
L37	Katherine can't help **herself**
L36	Katherine can't help **herself**
L35	Katherine can't help **herself**
L34	Katherine can't help **herself**
L33	Katherine can't help **herself**
L32	Katherine can't help **herself**
L31	Katherine can't help **herself**
L30	Katherine can't help **herself**
L29	Katherine can't help **herself**
L28	Katherine can't help **herself**
L27	Katherine can't help **herself**
L26	Katherine can't help **herself**
L25	Katherine can't help **herself**
L24	Katherine can't help **herself**
L23	Katherine can't help **herself**
L22	Katherine can't help **herself**
L21	Katherine can't help **herself**
L20	Katherine can't help **herself**
L19	Katherine can't help **herself**
L18	Katherine can't help **herself**
L17	Katherine can't help **herself**
L16	Katherine can't help **herself**
L15	Katherine can't help **herself**
L14	Katherine can't help **herself**
L13	Katherine can't help **herself**
L12	Katherine can't help **herself**
L11	Katherine can't help **herself**
L10	Katherine can't help **herself**
L9	Katherine can't help **herself**
L8	Katherine can't help **herself**
L7	Katherine can't help **herself**
L6	Katherine can't help **herself**
L5	Katherine can't help **herself**
L4	Katherine can't help **herself**
L3	Katherine can't help **herself**
L2	Katherine can't help **herself**
L1	Katherine can't help **herself**

Table 13: GPT-2 XL `aga` (why **herself** instead of **himself**).

H.3 ALTI-Logit (IOI) across Models

L12	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L11	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L10	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L9	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L8	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L7	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L6	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L5	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L4	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L3	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L2	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L1	</s> When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**

Figure 20: OPT 125M `IOI` (why **Paula** instead of **Martha**).

L24	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L23	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L22	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L21	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L20	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L19	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L18	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L17	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L16	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L15	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L14	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L13	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L12	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L11	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L10	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L9	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L8	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L7	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L6	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L5	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L4	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L3	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L2	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L1	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**

Table 14: BLOOM 560M `IOI` (why **Paula** instead of **Martha**).

L24	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L23	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L22	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L21	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L20	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L19	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L18	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L17	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L16	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L15	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L14	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L13	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L12	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L11	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L10	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L9	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L8	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L7	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L6	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L5	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L4	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L3	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L2	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L1	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**

Table 15: BLOOM 1B1 `IOI` (why **Paula** instead of **Martha**).

L12	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L11	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L10	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L9	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L8	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L7	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L6	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L5	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L4	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L3	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L2	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L1	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**

Table 16: GPT-2 Small `IOI` (why **Paula** instead of **Martha**).

L36	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L35	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L34	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L33	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L32	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L31	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L30	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L29	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L28	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L27	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L26	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L25	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L24	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L23	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L22	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L21	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L20	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L19	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L18	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L17	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L16	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L15	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L14	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L13	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L12	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L11	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L10	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L9	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L8	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L7	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L6	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L5	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L4	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L3	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L2	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L1	When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**

Table 17: GPT-2 Large `IOI` (why **Paula** instead of **Martha**).

L48 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L47 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L46 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L45 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L44 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L43 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L42 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L41 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L40 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L39 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L38 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L37 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L36 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L35 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L34 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L33 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L32 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L31 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L30 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L29 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L28 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L27 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L26 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L25 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L24 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L23 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L22 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L21 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L20 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L19 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L18 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L17 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L16 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L15 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L14 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L13 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L12 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L11 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L10 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L9 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L8 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L7 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L6 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L5 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L4 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L3 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L2 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**
L1 | When Paula and Martha got a coconut at the zoo, Martha decided to give the coconut to **Paula**

Table 18: GPT-2 XL `IOI` (why **Paula** instead of **Martha**).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 9

☑ A2. Did you discuss any potential risks of your work?
Section 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 2

☑ B1. Did you cite the creators of artifacts you used?
Section 2

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 9

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3

C ☑ Did you run computational experiments?

Section 3 and 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☐ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Not applicable. Left blank.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.