Token-wise Decomposition of Autoregressive Language Model Hidden States for Analyzing Model Predictions

Byung-Doh Oh
Department of Linguistics
The Ohio State University
`oh.531@osu.edu`

William Schuler
Department of Linguistics
The Ohio State University
`schuler.77@osu.edu`

Abstract

While there is much recent interest in studying why Transformer-based large language models make predictions the way they do, the complex computations performed within each layer have made their behavior somewhat opaque. To mitigate this opacity, this work presents a linear decomposition of final hidden states from autoregressive language models based on each initial input token, which is exact for virtually all contemporary Transformer architectures. This decomposition allows the definition of probability distributions that ablate the contribution of specific input tokens, which can be used to analyze their influence on model probabilities over a sequence of upcoming words with only one forward pass from the model. Using the change in next-word probability as a measure of importance, this work first examines which context words make the biggest contribution to language model predictions. Regression experiments suggest that Transformer-based language models rely primarily on collocational associations, followed by linguistic factors such as syntactic dependencies and coreference relationships in making next-word predictions. Additionally, analyses using these measures to predict syntactic dependencies and coreferent mention spans show that collocational association and repetitions of the same token largely explain the language models' predictions on these tasks.

1 Introduction

Much of contemporary natural language processing (NLP) is driven by Transformer-based large language models, which are trained to make predictions about words in their context by aggregating representations through their self-attention mechanism. The breakthrough in many NLP tasks these models have achieved has led to active research into interpreting their predictions and probing the knowledge embodied by these models (Manning et al., 2020; Rogers et al., 2021; Belinkov, 2022).

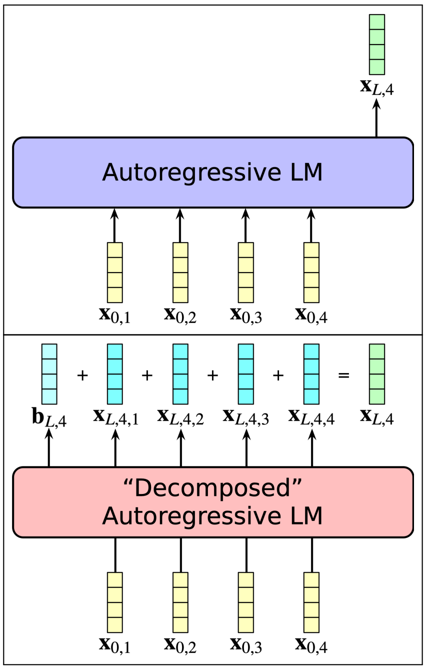

Figure 1: Schematic of input and output representations from Transformer-based autoregressive language models. Standard models (top) calculate one vector of final hidden states at a given timestep ($\mathbf{x}_{L,i}$), which in this work (bottom) is decomposed exactly into the sum of output representations of each input token ($\mathbf{x}_{L,i,k}$) and a cumulative bias term ($\mathbf{b}_{L,i}$).

One line of such research focuses on quantifying the importance of each input token to the models' final output, but due to the complexity of the computations performed within the Transformer layers, analysis has been limited to studying the self-attention mechanism and the feedforward neural network independently (Kobayashi et al., 2020, 2021; Geva et al., 2021, 2022; Mickus et al., 2022) or has relied on e.g. gradient-based attribution methods (Sanyal and Ren, 2021; Zaman and Belinkov, 2022) that yield measures that are not interpretable in terms of output model probabilities.

To address these limitations, this work presents a linear decomposition of final language model

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10105–10117
July 9-14, 2023 ©2023 Association for Computational Linguistics

hidden states into the sum of final output representations of each initial input token and a cumulative bias term, which is schematized in Figure 1. This work focuses on decomposing autoregressive language models, in which the final hidden states are used to calculate a probability distribution over the next token. The decomposition allows the definition of probability distributions that ablate the contribution of specific input tokens, which can be used to study their impact on next-token probabilities with only one forward pass from the model. This decomposition is exact if the activation function of the feedforward neural network is differentiable almost everywhere,[1] and therefore it does not require perturbing the original computations of the language model (e.g. by using approximations) to gauge the influence of input tokens for virtually all contemporary Transformer architectures. Additionally, this work defines an intuitive importance measure for each context token based on the change in next-token log probability, which does not correlate strongly with layer-wise attention weights or gradient norms. Since this measure is defined in terms of log probabilities, they can also be summed to quantify importance in predicting an arbitrary sequence of tokens according to the chain rule of conditional probabilities.

Using the proposed decomposition and associated importance measure, this work characterizes which kinds of context words autoregressive language models leverage most in order to make next-word predictions. Results from stepwise regression analyses suggest that Transformer-based language models rely mainly on collocational associations, followed by linguistic factors such as syntactic dependencies and coreference relationships. Follow-up analyses using these importance measures to predict syntactic dependencies and coreferent mention spans additionally show that collocational association and repetitions of the same token largely explain the language models' predictions on these tasks.

2 Background: Transformer Decoder of Autoregressive Language Models

Transformer-based autoregressive language models (e.g. Radford et al., 2019; Brown et al., 2020; Zhang et al., 2022) use a variant of the multi-layer Transformer decoder (Vaswani et al., 2017). Each decoder layer consists of a masked self-attention block and a feedforward neural network, which together calculate a vector $\mathbf{x}_{l,i} \in \mathbb{R}^d$ for token w_i at layer l:

$$\mathbf{x}_{l,i} = \mathrm{FF}_l(\mathrm{N}_{l,\mathrm{out}}(\mathbf{x}'_{l,i} + \mathbf{x}_{l-1,i})) + (\mathbf{x}'_{l,i} + \mathbf{x}_{l-1,i}), \quad (1)$$

where FF_l is a two-layer feedforward neural network, $\mathrm{N}_{l,\mathrm{out}}$ is a vector-wise layer normalization operation, and $\mathbf{x}'_{l,i} \in \mathbb{R}^d$ is the output representation from the multi-head self-attention mechanism, in which H heads mix representations from the previous context. This output $\mathbf{x}'_{l,i}$ can be decomposed into the sum of representations resulting from each attention head h and a bias vector $\mathbf{v}_l$:

$$\mathbf{x}'_{l,i} = \sum_{h=1}^{H} \mathbf{V}_{l,h}\, [\mathrm{N}_{l,\mathrm{in}}(\mathbf{x}_{l-1,1}) \cdots \mathrm{N}_{l,\mathrm{in}}(\mathbf{x}_{l-1,i})]\, \mathbf{a}_{l,h,i} + \mathbf{v}_l, \quad (2)$$

where $\mathbf{V}_{l,h} \in \mathbb{R}^{d \times d}$ and $\mathbf{v}_l \in \mathbb{R}^d$ represent the weights and biases of the composite value-output transformation[2] respectively, and $\mathbf{a}_{l,h,i} \in \mathbb{R}^i$ is the vector of self-attention weights from each head.

$\mathrm{N}_{l,\alpha}$, where $\alpha \in \{\mathrm{in}, \mathrm{out}\}$,[3] is a vector-wise layer normalization operation (Ba et al., 2016) that first standardizes the vector and subsequently conducts elementwise transformations using trainable parameters $\mathbf{c}_{l,\alpha}, \mathbf{b}_{l,\alpha} \in \mathbb{R}^d$:

$$\mathrm{N}_{l,\alpha}(\mathbf{y}) = \frac{\mathbf{y} - m(\mathbf{y})}{s(\mathbf{y})} \odot \mathbf{c}_{l,\alpha} + \mathbf{b}_{l,\alpha}, \quad (3)$$

where $m(\mathbf{y})$ and $s(\mathbf{y})$ denote the elementwise mean and standard deviation of $\mathbf{y}$ respectively, and $\odot$ denotes a Hadamard product.

The output representation from the last decoder layer L is layer-normalized and multiplied by the projection matrix to yield logit scores for the probability distribution over token w_{i+1}:

$$\mathbf{z}_i = \mathbf{W}\, \mathrm{N}_{L+1,\mathrm{in}}(\mathbf{x}_{L,i}), \quad (4)$$

where $\mathbf{z}_i \in \mathbb{R}^V$ is the vector of logit scores, $\mathbf{W} \in \mathbb{R}^{V \times d}$ is the projection matrix, V is the size of the vocabulary, and $\mathrm{N}_{L+1,\mathrm{in}}$ is the final layer normalization operation with parameters $\mathbf{c}_{L+1,\mathrm{in}}$ and $\mathbf{b}_{L+1,\mathrm{in}}$.

[1] That is, the function is differentiable at all real numbers except a subset of Lebesgue measure zero, such as the rectified linear unit (ReLU; Nair and Hinton, 2010), which has an inflection point at $x = 0$.

[2] For the simplicity of notation, multi-head self-attention is formulated as a sum of 'value-output' transformed representations from each attention head instead of the 'output' transformed concatenation of 'value' transformed representations from each attention head as in Vaswani et al. (2017). To this end, the weights and biases of the 'value' and 'output' transformations are respectively composed into $\mathbf{V}_{l,h}$ and $\mathbf{v}_l$. Refer to Appendix A for the derivation of $\mathbf{V}_{l,h}$ and $\mathbf{v}_l$.

[3] $\mathrm{N}_{l,\mathrm{in}}$ is applied before the masked self-attention block, and $\mathrm{N}_{l,\mathrm{out}}$ is applied before the feedforward neural network.

3 Token-wise Decomposition of Language Model Hidden States

This section provides a mathematical definition of the token-wise decomposition of language model hidden states, which allows the quantification of the contribution of each input token to the conditional probability of the next token.

3.1 Mathematical Definition

In this section, we show that the vector of logits $\mathbf{z}_i$ in Equation 4 can be decomposed into the sum of final output representations of each input token w_k and a 'bias-like' term that accumulates bias vectors throughout the Transformer network, which is exact if the activation function within the feedforward neural network is differentiable almost everywhere:

$$\mathbf{z}_i = \sum_{k=1}^{i} \mathbf{z}'_{i,k} + \mathbf{b}_i, \tag{5}$$

where $\mathbf{z}'_{i,k} \in \mathbb{R}^V$ is the final transformed output at timestep i of the input representation $\mathbf{x}_{0,k}$[4] at timestep k. This $\mathbf{z}'_{i,k}$ is calculated by aggregating the output of all computations performed on $\mathbf{x}_{0,k}$ throughout the Transformer layers:

$$\mathbf{z}'_{i,k} = \mathbf{W}\,\mathbf{n}_{\mathrm{x},L+1,i,k}, \tag{6}$$

where $\mathbf{n}_{\mathrm{x},L+1,i,k}$ is a layer-normalized version of $\mathbf{x}_{L,i,k}$, explained below. Additionally, $\mathbf{b}_i \in \mathbb{R}^V$ is the 'bias-like' term resulting from accumulating computations performed on bias vectors that are difficult to attribute to any specific source position k:

$$\mathbf{b}_i = \mathbf{W}\,\mathbf{n}_{\mathrm{b},L+1,i}, \tag{7}$$

where $\mathbf{n}_{\mathrm{b},L+1,i}$ is a layer-normalized version of $\mathbf{b}_{L,i}$, also explained below.

This decomposition is in turn achieved by maintaining input-specific vectors $\mathbf{x}_{l,i,k} \in \mathbb{R}^d$ and a 'bias-like' vector $\mathbf{b}_{l,i} \in \mathbb{R}^d$ throughout the network. The second index of both $\mathbf{x}_{l,i,k}$ and $\mathbf{b}_{l,i}$ represents each target position i, and the third index of $\mathbf{x}_{l,i,k}$ represents each source position $k \in \{1, ..., i\}$. Therefore, when the third index of $\mathbf{x}_{l,i,k}$ is reduced and the result is added to $\mathbf{b}_{l,i}$, the undecomposed output representation $\mathbf{x}_{l,i} \in \mathbb{R}^d$ is returned:

$$\mathbf{x}_{l,i} = \sum_{k=1}^{i} \mathbf{x}_{l,i,k} + \mathbf{b}_{l,i}. \tag{8}$$

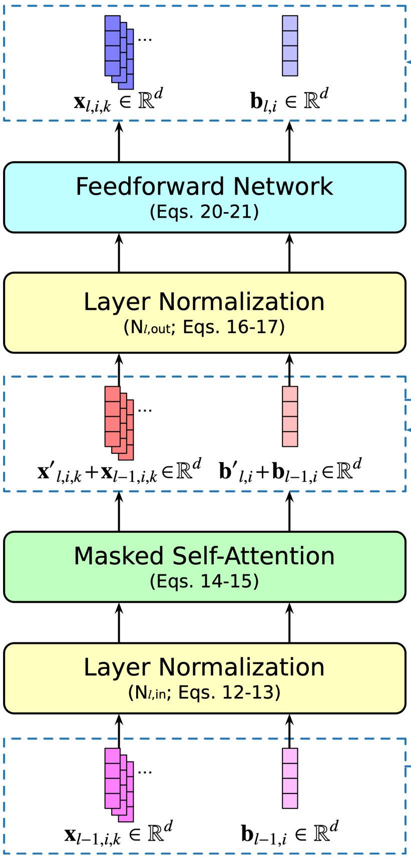
$\mathbf{x}_{l,i,k} \in \mathbb{R}^d$
$\mathbf{b}_{l,i} \in \mathbb{R}^d$
Feedforward Network
(Eqs. 20-21)
Residual Connection
Layer Normalization
($\mathrm{N}_{l,\mathrm{out}}$; Eqs. 16-17)
$\mathbf{x}'_{l,i,k}+\mathbf{x}_{l-1,i,k} \in \mathbb{R}^d$
$\mathbf{b}'_{l,i}+\mathbf{b}_{l-1,i} \in \mathbb{R}^d$
Masked Self-Attention
(Eqs. 14-15)
Residual Connection
Layer Normalization
($\mathrm{N}_{l,\mathrm{in}}$; Eqs. 12-13)
$\mathbf{x}_{l-1,i,k} \in \mathbb{R}^d$
$\mathbf{b}_{l-1,i} \in \mathbb{R}^d$

Figure 2: Alternative formulation of computations performed within one decoder layer of a Transformer-based autoregressive language model, which allows the contribution of each input token w_k to $\mathbf{x}_{l,i}$ to be preserved as $\mathbf{x}_{l,i,k}$.

These decomposed representations are updated by each decoder layer (Eq. 1; Fig. 2) as follows:

$$\mathbf{x}_{l,i,k} = \mathbf{f}_{\mathrm{x},l,i,k} + (\mathbf{x}'_{l,i,k} + \mathbf{x}_{l-1,i,k}), \tag{9}$$
$$\mathbf{b}_{l,i} = \mathbf{f}_{\mathrm{b},l,i} + (\mathbf{b}'_{l,i} + \mathbf{b}_{l-1,i}), \tag{10}$$

where $\mathbf{b}_{0,i} = \mathbf{0}$ and $\mathbf{x}_{0,i,k}$ is a position-sensitive version of $\mathbf{x}_{0,k}$:

$$\mathbf{x}_{0,i,k} = \begin{cases} \mathbf{x}_{0,k} & \text{if } i = k, \\ \mathbf{0} & \text{if } i \neq k, \end{cases} \tag{11}$$

and $\mathbf{f}_{\mathrm{x},l,i,k}$ and $\mathbf{f}_{\mathrm{b},l,i}$ are decomposed versions of the output from the feedforward network for $\mathbf{x}_{l,i,k}$ and $\mathbf{b}_{l,i}$, defined below.

The exact decomposition of hidden states according to each source position is made possible due to the linear nature of computations within the masked self-attention block and a local linear approximation of the activation function within the feedforward neural network. First, layer normalization $\mathrm{N}_{l,\mathrm{in}}$ (Eq. 3) is applied to $\mathbf{x}_{l-1,i,k}$ to yield $\mathbf{n}_{\mathrm{x},l,i,k}$ by centering it, scaling it by the standard deviation of the undecomposed representation $s(\mathbf{x}_{l-1,i})$,

[4] Throughout this paper, the input representation $\mathbf{x}_{0,k}$ denotes the sum of the type-specific embedding for token w_k and the positional embedding for position k.

and obtaining a Hadamard product with trainable vector $\mathbf{c}_{l,\text{in}}$:

$$\mathbf{n}_{\text{x},l,i,k} = \frac{\mathbf{x}_{l-1,i,k} - m(\mathbf{x}_{l-1,i,k})}{s(\mathbf{x}_{l-1,i})} \odot \mathbf{c}_{l,\text{in}}. \tag{12}$$

$\mathrm{N}_{l,\text{in}}$ is also applied to $\mathbf{b}_{l-1,i}$ to yield $\mathbf{n}_{\text{b},l,i}$, except that the bias vector $\mathbf{b}_{l,\text{in}}$ is accumulated by this term:

$$\mathbf{n}_{\text{b},l,i} = \frac{\mathbf{b}_{l-1,i} - m(\mathbf{b}_{l-1,i})}{s(\mathbf{x}_{l-1,i})} \odot \mathbf{c}_{l,\text{in}} + \mathbf{b}_{l,\text{in}}. \tag{13}$$

Subsequently, the masked self-attention mechanism (Eq. 2) is applied to $[\mathbf{n}_{\text{x},l,1,k} \cdots \mathbf{n}_{\text{x},l,i,k}]$ to yield $\mathbf{x}'_{l,i,k}$, which updates the total representation from source position k to target position i using self-attention weights $\mathbf{a}_{l,h,i}$:

$$\mathbf{x}'_{l,i,k} = \sum_{h=1}^{H} \mathbf{V}_{l,h} \, [\mathbf{n}_{\text{x},l,1,k} \cdots \mathbf{n}_{\text{x},l,i,k}] \, \mathbf{a}_{l,h,i}. \tag{14}$$

The self-attention mechanism is also applied to $[\mathbf{n}_{\text{b},l,1} \cdots \mathbf{n}_{\text{b},l,i}]$ to yield $\mathbf{b}'_{l,i}$. Similarly to layer normalization, the bias vector $\mathbf{v}_l$ is accumulated by this term:

$$\mathbf{b}'_{l,i} = \sum_{h=1}^{H} \mathbf{V}_{l,h} \, [\mathbf{n}_{\text{b},l,1} \cdots \mathbf{n}_{\text{b},l,i}] \, \mathbf{a}_{l,h,i} + \mathbf{v}_l. \tag{15}$$

After adding the residual representations, layer normalization $\mathrm{N}_{l,\text{out}}$ is applied to $\mathbf{x}'_{l,i,k} + \mathbf{x}_{l-1,i,k}$ and $\mathbf{b}'_{l,i} + \mathbf{b}_{l-1,i}$ in a similar manner to Equations 12 and 13 to yield $\mathbf{n}'_{\text{x},l,i,k}$ and $\mathbf{n}'_{\text{b},l,i}$ respectively, by centering each vector, scaling them by the standard deviation of their corresponding undecomposed representation $s(\mathbf{x}'_{l,i} + \mathbf{x}_{l-1,i})$, and applying the learned parameters $\mathbf{c}_{l,\text{out}}$ and $\mathbf{b}_{l,\text{out}}$:

$$\mathbf{n}'_{\text{x},l,i,k} = \frac{\mathbf{x}'_{l,i,k} + \mathbf{x}_{l-1,i,k} - m(\mathbf{x}'_{l,i,k} + \mathbf{x}_{l-1,i,k})}{s(\mathbf{x}'_{l,i} + \mathbf{x}_{l-1,i})} \odot \mathbf{c}_{l,\text{out}}, \tag{16}$$

$$\mathbf{n}'_{\text{b},l,i} = \frac{\mathbf{b}'_{l,i} + \mathbf{b}_{l-1,i} - m(\mathbf{b}'_{l,i} + \mathbf{b}_{l-1,i})}{s(\mathbf{x}'_{l,i} + \mathbf{x}_{l-1,i})} \odot \mathbf{c}_{l,\text{out}} + \mathbf{b}_{l,\text{out}}. \tag{17}$$

Finally, if the activation function within the feedforward neural network from Equation 1 is differentiable almost everywhere,[5] local linear approximation can be used to calculate its output values:

$$\begin{aligned} \mathrm{FF}_l(\mathbf{y}) &= \mathbf{F}_{l,2} \, \sigma(\mathbf{F}_{l,1} \, \mathbf{y} + \mathbf{f}_{l,1}) + \mathbf{f}_{l,2} & (18) \\ &= \mathbf{F}_{l,2}(\mathbf{s} \odot (\mathbf{F}_{l,1} \, \mathbf{y} + \mathbf{f}_{l,1}) + \mathbf{i}) + \mathbf{f}_{l,2}, & (19) \end{aligned}$$

where $\mathbf{F}_{l,1}, \mathbf{F}_{l,2}$ and $\mathbf{f}_{l,1}, \mathbf{f}_{l,2}$ are the weights and biases of the feedforward neural network, σ is the activation function, and $\mathbf{s}$ and $\mathbf{i}$ are respectively the vector of slopes and intercepts of tangent lines specified by each element of the input vector $\mathbf{F}_{l,1} \, \mathbf{y} + \mathbf{f}_{l,1}$.[6] This reformulation of the activation function allows the feedforward neural network to apply to each decomposed vector $\mathbf{n}'_{\text{x},l,i,k}$ and $\mathbf{n}'_{\text{b},l,i}$ to yield $\mathbf{f}_{\text{x},l,i,k}$ and $\mathbf{f}_{\text{b},l,i}$ respectively:

$$\mathbf{f}_{\text{x},l,i,k} = \mathbf{F}_{l,2} \, \mathbf{s}_{l,i} \odot \mathbf{F}_{l,1} \, \mathbf{n}'_{\text{x},l,i,k}, \tag{20}$$

$$\mathbf{f}_{\text{b},l,i} = \mathbf{F}_{l,2}(\mathbf{s}_{l,i} \odot (\mathbf{F}_{l,1} \, \mathbf{n}'_{\text{b},l,i} + \mathbf{f}_{l,1}) + \mathbf{i}_{l,i}) + \mathbf{f}_{l,2}, \tag{21}$$

where $\mathbf{s}_{l,i}$ and $\mathbf{i}_{l,i}$ are the vector of slopes and intercepts of tangent lines specified by each element of the undecomposed $\mathbf{F}_{l,1} \, \mathrm{N}_{l,\text{out}}(\mathbf{x}'_{l,i} + \mathbf{x}_{l-1,i}) + \mathbf{f}_{l,1}$. As with other operations, the bias vectors $\mathbf{f}_{l,1}$, $\mathbf{f}_{l,2}$, and $\mathbf{i}_{l,1}$ are accumulated by $\mathbf{f}_{\text{b},l,i}$.

3.2 Proposed Importance Measure ΔLP: Change in Next-Word Probabilities

Based on the decomposition outlined in Section 3.1, the importance of each input token $w_{1..i}$ to the probability of the next token $\mathsf{P}(w_{i+1} \mid w_{1..i})$ can be quantified. To this end, the probability distribution over the next token that ablates the contribution of w_k is defined as follows:

$$\mathsf{P}(w_{i+1} \mid w_{1..i \setminus \{k\}}) = \underset{w_{i+1}}{\textsc{SoftMax}}(\mathbf{z}_i - \mathbf{z}'_{i,k}). \tag{22}$$

Subsequently, the importance measure of w_k to the prediction of w_{i+1} is calculated as the difference between log probabilities of w_{i+1} given the full context ($w_{1..i}$) and the context without it ($w_{1..i \setminus \{k\}}$):

$$\begin{aligned} &\Delta\mathsf{LP}(w_{i+1} \mid w_{1..i}, w_{k \in \{1,\ldots,i\}}) = & (23) \\ &\log_2 \mathsf{P}(w_{i+1} \mid w_{1..i}) - \log_2 \mathsf{P}(w_{i+1} \mid w_{1..i \setminus \{k\}}). \end{aligned}$$

This measure captures the intuition that an input token that is more crucial to predicting the next token w_{i+1} will result in larger decreases in $\mathsf{P}(w_{i+1} \mid w_{1..i})$ when its contribution to the logit scores is ablated out. It is also possible for ΔLP to be negative, or in other words, $\mathsf{P}(w_{i+1} \mid w_{1..i})$ can increase as a result of ablating an input token w_k. However, a preliminary analysis showed that negative ΔLP values were much less commonly observed than positive ΔLP values and input tokens with negative ΔLP values were not in an easily interpretable relationship with the predicted token. Therefore, the experiments in this work focus on

[5] Virtually all widely used activation functions such as the rectified linear unit (ReLU; Nair and Hinton, 2010) and the Gaussian error linear unit (GELU; Hendrycks and Gimpel, 2016) satisfy this property.

[6] That is, $\mathbf{s} = \sigma'(\mathbf{F}_{l,1} \, \mathbf{y} + \mathbf{f}_{l,1})$, and $\mathbf{i} = \sigma(\mathbf{F}_{l,1} \, \mathbf{y} + \mathbf{f}_{l,1}) - \sigma'(\mathbf{F}_{l,1} \, \mathbf{y} + \mathbf{f}_{l,1}) \odot (\mathbf{F}_{l,1} \, \mathbf{y} + \mathbf{f}_{l,1})$.

characterizing input tokens with high ΔLP values, which are the tokens that drive a large increase in $\mathsf{P}(w_{i+1} \mid w_{1..i})$.

4 Experiment 1: Correlation with Other Importance Measures

This work first compares the decomposition-based ΔLP defined in Section 3.2 with other measures of importance that have been used in the literature to examine the degree to which ΔLP may be redundant with them. To this end, Pearson correlation coefficients were calculated between the proposed ΔLP and attention weights and gradient norms at a token level.

4.1 Procedures

The first experiment used the English section of the Conference on Natural Language Learning shared task corpus (CoNLL-2012; Pradhan et al., 2012) as well as the Wall Street Journal corpus of the Penn Treebank (WSJ; Marcus et al., 1993). Both corpora include text from the newswire domain, and the CoNLL-2012 corpus additionally includes text from broadcasts, magazines, telephone conversations, weblogs, and the Bible. The development sets of the two corpora were used in this experiment, which consist of 9,603 and 1,700 sentences respectively.

To calculate importance measures on the two corpora, the Open Pre-trained Transformer language model (OPT; Zhang et al., 2022) with ~125M parameters was used for efficiency. In addition to ΔLP defined in Section 3.2,[7] the following importance measures were calculated for each context token $w_{k\in\{1,...,i\}}$ at timestep i:

- Layer-wise attention weights (Vaswani et al., 2017): Average attention weights over w_k from all heads within each layer, i.e. $\frac{1}{H}\sum_{h=1}^{H}\delta_k^\top \mathbf{a}_{l,h,i}$, where $\delta_k \in \mathbb{R}^i$ is a Kronecker delta vector consisting of a one at element k and zeros elsewhere, and $l \in \{1, ..., L\}$.
- Gradient norms (Simonyan et al., 2014): Norm of gradient of next-token log probability w.r.t. the input $\mathbf{x}_{0,k}$, i.e. $||\nabla_{\mathbf{x}_{0,k}} \log \mathsf{P}(w_{i+1} \mid w_{1..i})||_n$, where $n \in \{1, 2\}$.
- Input × gradient norms (Shrikumar et al., 2017): $||\mathbf{x}_{0,k} \odot \nabla_{\mathbf{x}_{0,k}} \log \mathsf{P}(w_{i+1} \mid w_{1..i})||_n$, where $n \in \{1, 2\}$.

Each article of the CoNLL-2012 and WSJ corpora was tokenized according OPT's byte-pair encoding (BPE; Sennrich et al., 2016) tokenizer and was provided as input to the OPT model. In cases where each article did not fit into a single context window, the second half of the previous context window served as the first half of a new context window to calculate importance measures for the remaining tokens.[8] Finally, Pearson correlation coefficients were calculated between token-level ΔLP and attention-/gradient-based importance measures on each corpus (163,309,857 points in CoNLL-2012; 25,900,924 points in WSJ).

4.2 Results

The results in Figure 3 show that across both corpora, the proposed ΔLP shows weak correlation with both attention weights and gradient norms, which suggests that ΔLP does not capture a redundant quantity from importance measures that have been used in previous work to examine language model predictions. The gradient norms are more correlated with ΔLP, which is likely due to the fact that the gradients calculated with respect to the original input representation $\mathbf{x}_{0,k}$ accumulate all computations performed within the network like the token-wise decomposition. However, one crucial difference between ΔLP and gradient norms is that gradient norms can 'saturate' and approach zero when the model makes accurate predictions, as $\nabla_{\mathbf{z}_i} \log \mathsf{P}(w_{i+1} \mid w_{1..i}) \approx \mathbf{0}$ when $\mathsf{P}(w_{i+1} \mid w_{1..i}) \approx 1$. This means that the importance measures of all context tokens will be systematically underestimated for high-probability target tokens, which may be especially problematic for analyzing large language models that have been trained on billions of training tokens. For average attention weights, they seem to correlate with ΔLP most at layer 1, where they are calculated over layer-normalized input representations $[\mathrm{N}_{1,\mathrm{in}}(\mathbf{x}_{0,1}) \cdots \mathrm{N}_{1,\mathrm{in}}(\mathbf{x}_{0,i})]$. In contrast, the attention weights at higher layers seem to correlate less with ΔLP, as they are calculated over representations that have been 'mixed' by the self-attention mechanism.

5 Experiment 2: Characterizing High-Importance Context Words

Having established that ΔLP provides a novel method to quantify the importance of each context

[7] Code for calculating decomposed OPT representations and their associated ΔLP is publicly available at https://github.com/byungdoh/llm_decomposition.

[8] In practice, most articles fit within one context window of 2,048 tokens.

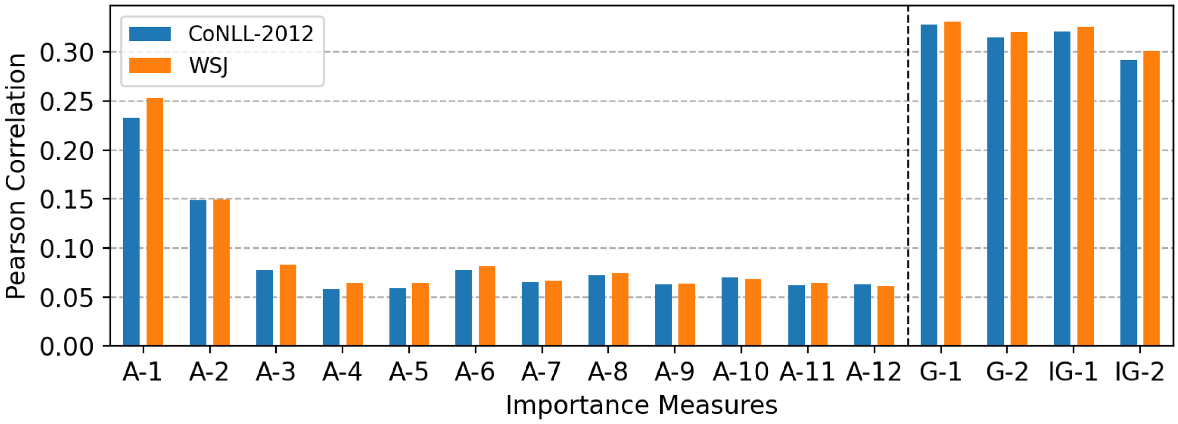

Figure 3: Pearson correlation coefficients between ΔLP and other importance measures for each context token. A-l is average attention weight at layer l; G-n is n-norm of gradient; IG-n is n-norm of input × gradient.

token to language model predictions, the second experiment conducts a series of regression analyses to characterize high-importance context words (i.e. words with high ΔLP values) and shed light on which kinds of context words language models leverage most in order to make predictions about the next word.

5.1 Procedures

In order to characterize high-importance context words that drive next-word predictions, linear regression models were fit in a stepwise manner to ΔLP values on the development set of the CoNLL-2012 corpus, which contains manual annotations of both syntactic structures and coreference relationships. To this end, the ΔLP values were calculated for each context word at a word level (following the Penn Treebank tokenization conventions such that they align with the annotations) using the OPT model with ~125M parameters. Whenever the predicted word consisted of multiple tokens, the ΔLP values were added together to calculate:

$$\Delta\mathsf{LP}(w_{i+1}, w_{i+2} \mid w_{1..i}, w_k) = \qquad (24)$$
$$\Delta\mathsf{LP}(w_{i+2} \mid w_{1..i+1}, w_k) + \Delta\mathsf{LP}(w_{i+1} \mid w_{1..i}, w_k),$$

which is well-defined by the chain rule of conditional probabilities. Likewise, when the context word consisted of multiple tokens, the contributions of all component tokens were ablated simultaneously (Eq. 22) to calculate the ΔLP of that context word.[9] In order to keep the regression models tractable, the ΔLP value of the most important context word for each predicted word (i.e. highest ΔLP value) provided the response data for this experiment. This resulted in a total of 162,882 observations, which are visualized in Figure 4.

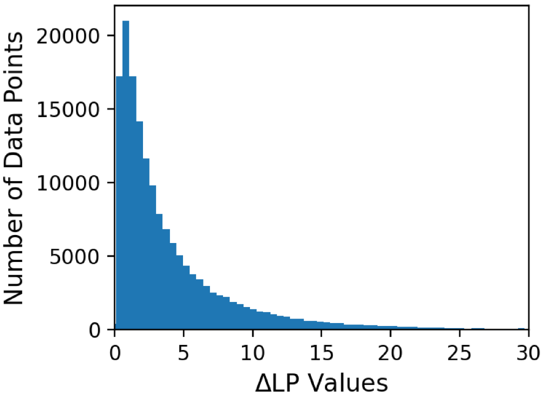

Figure 4: Histogram of top ΔLP values for each predicted word on the development set of the CoNLL-2012 corpus calculated from the OPT model.

Subsequently, a 'baseline' regression model that contains baseline predictors was fit to the set of ΔLP values. These baseline predictors include the index of the predicted word (i.e. how many words are in the context), the linear distance between the context word and the predicted word, and $\log \mathsf{P}(w_{i+1} \mid w_{1..i})$, which may be correlated with ΔLP values. Additionally, in order to guide the identification of factors underlying the ΔLP values of high-importance context words, each data point was associated with the following predictors of interest that capture associations between the predicted word and the context word:

[9] This ability to quantify the contribution of each context token in predicting multiple target tokens or the simultaneous contribution of multiple context tokens in model prediction is another advantage of ΔLP over attention weights or gradient norms, which are inherently defined at a single-token level.

- Pointwise mutual information (PMI): $\log_2 \frac{\mathsf{P}(w_k, w_{i+1})}{\mathsf{P}(w_k)\mathsf{P}(w_{i+1})}$, which is calculated using unigram and bigram probabilities estimated from the Gigaword 4 corpus (Parker et al., 2009). Two variants of PMI are explored in this work, which capture associations of word pairs in contiguous bigrams ($\text{PMI}_{\text{bigram}}$) and document co-occurrences (PMI_{doc}).[10]
- Syntactic dependency: A binary variable indicating whether the context word and the predicted word form a syntactic dependency. The CoreNLP toolkit (Manning et al., 2014) was used to convert annotated constituency structures to dependency representations.
- Coreference relationship: A binary variable indicating whether the context word and the predicted word are in coreferent spans.

These predictors of interest were included in a stepwise manner, by including the one predictor that contributes most to regression model fit at each iteration and testing its statistical significance through a likelihood ratio test (LRT). All predictors were centered and scaled prior to regression modeling, so the regression coefficients β are defined in units of standard deviation and are comparable across predictors.

5.2 Results

The results in Table 1 show that among the predictors of interest, both variants of PMI made the biggest contribution to regression model fit, followed by syntactic dependency and coreference relationship.[11] This suggests that Transformer-based autoregressive language models rely primarily on collocational associations in making next-word predictions (e.g. *wedding* predicting *groom*, *medical* predicting *hospital*). Linguistic factors like syntactic dependencies and coreference relationships explained additional variance in ΔLP values, although their contribution was not as large.

The baseline predictors also shed light on the characteristics of context words that have a large influence on next-word probabilities. Most notably, the linear distance between the predicted word and the context word was a positive predictor of ΔLP, which indicates that language models can leverage words far back in the context and that the contribution of such context words is large when they do. Moreover, ΔLP values were negatively correlated with log probability, which indicates that the contribution of context words generally decreases when the model is making confident predictions about the next word. Finally, although there was a positive correlation between word index and ΔLP values, its strength was too weak to draw conclusive interpretations.

Predictor	β	t-value	ΔLL
Word index	0.034	1.919	-
Distance	1.126	62.755	-
Log prob.	-0.083	-5.350	-
$\text{PMI}_{\text{bigram}}$	1.220	70.857	6151.262*
PMI_{doc}	1.286	73.952	3194.815*
Dependency	1.055	63.720	1981.778*
Coreference	0.123	7.195	25.883*

Table 1: Regression coefficients from the final stepwise regression model and increase in regression model likelihood (ΔLL) from including each predictor of interest. The predictors of interest are presented in the order they were included during stepwise regression (i.e. strongest predictor at each iteration). *: $p < 0.001$.

6 Experiment 3: Syntactic Dependency and Coreference Prediction Using ΔLP

The previous experiment revealed that compared to measures of collocational association, syntactic dependency and coreference relationships were not as strong predictors of ΔLP. Experiment 3 further examines the connection between high-importance context words and syntactic dependency and coreference relationships by using ΔLP to predict them independently and analyzing the extent to which each relationship type aligns with ΔLP.

6.1 Procedures

This experiment used ΔLP to make predictions about context words in syntactic dependency and coreference relationships on the development sets of the WSJ and CoNLL-2012 corpora respectively.

First, on the WSJ corpus, the precision scores for syntactic dependency relations were calculated by counting how many times context words with high ΔLP match words in syntactic dependency relations. While each word has exactly one incoming typed edge from its head in a typical depen-

[10] The corpus was tokenized following the Penn Treebank conventions for consistency. PMI was defined to be 0 for word pairs without unigram or bigram probability estimates.

[11] Refer to Appendix B for regression results from the first iteration of the stepwise analysis, which evaluates each predictor independently on top of the baseline regression model.

Relation	ΔLP	Base.	PMI_b	PMI_d
Nom. subj.	61.15	39.79	1.38	1.44
Direct obj.	70.43	22.01	0.91	1.57
Oblique	52.54	24.31	-0.68	1.54
Compound	80.44	39.56	4.97	2.93
Nom. mod.	53.84	26.09	-0.41	1.84
Adj. mod.	82.55	36.02	4.36	2.17
Determiner	52.03	36.52	1.51	1.08
Case marker	52.38	27.96	-0.29	1.08
Microavg.	56.20	29.22	1.11	1.58

Table 2: Precision scores calculated using ΔLP, random word baseline, and average PMI of frequent syntactic dependency relations in the WSJ corpus. The less frequent relations are not presented separately but are included in the microaverage. PMI_b is average PMI based on contiguous bigrams; PMI_d is average PMI based on document co-occurrences.

Mention head POS	ΔLP	Base.	Rep.%
Personal pronoun	26.55	36.80	30.92
Possessive pronoun	23.29	36.45	30.59
Proper noun (sg.)	61.21	23.19	68.80
Proper noun (pl.)	70.67	57.33	68.00
Common noun (sg.)	43.39	12.55	48.75
Common noun (pl.)	47.01	24.73	55.03
Possessive ending	46.28	30.58	40.91
Microavg.	38.21	28.65	43.26

Table 3: Precision scores calculated using ΔLP, most recent head POS baseline, and Rep. % of frequent types of coreferent spans in the CoNLL-2012 corpus. The less frequent types are not presented separately but are included in the microaverage. Rep. % is the proportion of mention spans whose head words are repeated from previous coreferent spans.

dency syntax representation, since autoregressive language models have no access to the forward context, all edges between word pairs were treated as undirected edges and were evaluated at the later word in the pair. For each predicted word w_{i+1} that is in n syntactic dependency relationships, the top-n context words were selected based on ΔLP within the same sentence and compared to the n words that are in syntactic dependency relationships with w_{i+1}. The syntactic dependency representations converted using the CoreNLP toolkit (Manning et al., 2014) were used to evaluate the performance on the WSJ corpus. As a baseline, the expected precision scores from randomly selecting n previous words within the same sentence are also reported.

Similarly, antecedent selection precision scores for coreference relations were calculated by counting how many times the context word with the highest ΔLP value matched words in spans denoting the same entity. For each mention span, ΔLP quantifying the impact of every context word on the prediction of the entire span (Eq. 24) was calculated. Subsequently, the context word with the highest ΔLP was evaluated in terms of whether it belonged to any antecedent spans denoting the same entity. As a baseline, precision scores from selecting the most recent word with the same part-of-speech as the head word of the span are reported.

6.2 Results

The syntactic dependency results in Table 2 reveal a discrepancy in performance according to the type of relation that is being predicted. Generally, context words with high ΔLP values corresponded most closely to words in adjectival modifier and compound relations, followed by those in subject and direct object relations, which are core arguments in English. Performance on adjunct nouns such as nominal modifiers and oblique nouns, as well as function words like determiners and case markers was lower. This trend in turn seems to be generally driven by the strength of collocational associations, as can be seen by the corresponding average PMI values in Table 2. This corroborates the regression results of Experiment 2 and further suggests that the seeming connection between language model predictions and syntactic dependencies may underlyingly be the effects of collocational association. One counterexample to this trend seems to be the syntactic dependency between the main verb and its direct object, which shows close correspondence to ΔLP despite not having high average PMI values.

The coreference results in Table 3 show an even larger gap in performance according to the type of entity mention. Generally, context words with high ΔLP values corresponded most closely to previous mentions of proper nouns and common nouns. In contrast, they did not correspond well to antecedents of personal and possessive pronouns, showing lower precision scores than a simple baseline that chooses the most recent pronoun. A follow-up analysis of the ΔLP values showed that when the language model has to predict a head word that has already been observed in its context, the earlier occurrence of that head word contributes substantially to its prediction. The proportion of mention spans whose head words are repeated from head words of previous coreferent spans in Table 3 shows that the close correspondence between ΔLP and previous mentions of proper nouns is driven by the fact that these proper nouns are often repeated verbatim in the corpus. In contrast, the prediction of pronouns does not seem to be mainly driven by context words that denote their antecedents.

7 Discussion and Conclusion

This work advances recent efforts to interpret the predictions of Transformer-based large language models. To this end, a linear decomposition of final language model hidden states into the sum of final output representations of each initial input token and a cumulative bias term was presented. This decomposition is exact as long as the activation function of the feedforward neural network is differentiable almost everywhere, and therefore it is applicable to virtually all Transformer-based architectures. Additionally, this decomposition does not require perturbing any intermediate computations nor re-running the language model to examine the impact of each input token. The decomposition in turn allows the definition of probability distributions that ablate the influence of input tokens, which was used to define the importance measure ΔLP that quantifies the change in next-token log probability. The first experiment in this work demonstrated that ΔLP does not capture a redundant quantity from importance measures that have been used in previous work to examine language model predictions such as layer-wise attention weights or gradient norms.

Subsequently, based on the proposed ΔLP, a stepwise regression analysis was conducted to shed light on the characteristics of context words that autoregressive language models rely on most in order to make next-word predictions. The regression results show that Transformer-based language models mainly leverage context words that form strong collocational associations with the predicted word, followed by context words that are in syntactic dependencies and coreference relationships with the predicted word. The high reliance on collocational associations is consistent with the mathematical analysis of Transformers that a layer of self-attention effectively functions as a lookup table that tracks bigram statistics of the input data (Elhage et al., 2021), as well as empirical observations that Transformer-based autoregressive language models have a propensity to 'memorize' sequences from the training data (Carlini et al., 2022).

Finally, as a follow-up analysis, ΔLP was used to predict syntactic dependencies and coreferent mentions to further examine their relationship to high-importance context words. The precision scores on both tasks revealed a large discrepancy in performance according to the type of syntactic dependency relations and entity mentions. On syntactic dependency prediction, ΔLP corresponded closer to words in relations with high collocational association such as compounds and adjectival modifiers, providing further support for its importance in a language model's next-word prediction. Moreover, on coreferent antecedent selection, ΔLP more accurately identified previous mentions of proper nouns and common nouns that were already observed verbatim in context. This is consistent with the tendency of Transformer-based language models to predict identical tokens from its context (Sun et al., 2021), which seems to be enabled by dedicated 'induction heads' (Elhage et al., 2021; Olsson et al., 2022) that learn such in-context copying behavior.

Taken together, these results suggest that collocational association and verbatim repetitions strongly drive the predictions of Transformer-based autoregressive language models. As such, the connection drawn between a large language model's computations and linguistic phenomena such as syntactic dependencies and coreference observed in previous work (e.g. Manning et al., 2020) may underlyingly be the effects of these factors.

Acknowledgments

We thank the reviewers for their helpful comments. This work was supported by the National Science Foundation grant #1816891. All views expressed are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Limitations

The connection between factors underlying the predictions of Transformer-based autoregressive language models and linguistic factors drawn in this work is based on a model trained on English text and annotated corpora of English text. Therefore, this connection may not generalize to other languages with e.g. more flexible word order. Additionally, although the alternative formulations of Transformer hidden states yielded insights about language model predictions, they are more computationally expensive to calculate as they rely on an explicit decomposition of the matrix multiplication operation, which in undecomposed form is highly optimized for in most packages.

Ethics Statement

Experiments presented in this work used datasets from previously published research (Pradhan et al., 2012; Marcus et al., 1993), in which the procedures for data collection, validation, and cleaning are outlined. These datasets were used to study a large language model's predictions about coreference resolution and dependency parsing respectively, which is consistent with their intended use. As this work focuses on studying the factors underlying the predictions of large language models, its potential risks and negative impacts on society seem to be minimal.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *arXiv preprint*, arXiv:1607.06450v1.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

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 *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1877–1901.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. *arXiv preprint*, arXiv:2202.07646v2.

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*.

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.

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.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (GELUs). *arXiv preprint*, arXiv:1606.08415v4.

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*, pages 7057–7075.

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.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 55–60.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

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.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning*, page 807–814.

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*.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2009. English Gigaword Fourth Edition LDC2009T13. *Linguistic Data Consortium*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Olga Uryupina, and Yuchen Zhang. 2012. CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in OntoNotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Technical Report*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3145–3153.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop Track Proceedings of the 2nd International Conference on Learning Representations*.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. 2021. Do long-range language models actually use long-range context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 807–822.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Kerem Zaman and Yonatan Belinkov. 2022. A multilingual perspective towards the evaluation of attribution methods in natural language inference. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1556–1576.

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. 2022. OPT: Open pre-trained Transformer language models. *arXiv preprint*, arXiv:2205.01068v4.

A Composition of 'Value' and 'Output' Transformations

In Vaswani et al.'s (2017) formulation of multi-head attention, the 'value' transformation is defined at the head level with weights $\mathbf{W}^{\mathrm{V}}_{l,h} \in \mathbb{R}^{(d/H)\times d}$ and biases $\mathbf{b}^{\mathrm{V}}_{l,h} \in \mathbb{R}^{(d/H)}$, and the 'output' transformation is defined at the layer level with weights $\mathbf{W}^{\mathrm{O}}_{l} \in \mathbb{R}^{d\times d}$ and biases $\mathbf{b}^{\mathrm{O}}_{l} \in \mathbb{R}^{d}$. $\mathbf{V}_{l,h}$ and $\mathbf{v}_l$ defined in Equation 2 are equal to:

$$\mathbf{V}_{l,h} = \mathbf{W}^{\mathrm{O}}_{l}(\delta_h \otimes \mathbf{W}^{\mathrm{V}}_{l,h}), \qquad (25)$$

$$\mathbf{v}_l = \sum_{h=1}^{H} \mathbf{W}^{\mathrm{O}}_{l}(\delta_h \otimes \mathbf{b}^{\mathrm{V}}_{l,h}) + \mathbf{b}^{\mathrm{O}}_{l}, \qquad (26)$$

where $\delta_h \in \mathbb{R}^H$ is a Kronecker delta vector consisting of a one at element h and zeros elsewhere, and $\otimes$ denotes a Kronecker product.

B Additional Regression Results

Regression results from the first iteration of the stepwise analysis in Experiment 2, which evaluates each predictor of interest independently on top of the baseline regression model, are outlined in Table 4.

Predictor	β	t-value	ΔLL
$\text{PMI}_{\text{bigram}}$	1.832	113.043	6151.262*
PMI_{doc}	1.643	102.341	5075.541*
Dependency	1.462	88.912	3859.854*
Coreference	0.362	21.877	238.948*

Table 4: Regression coefficients and increase in regression model likelihood (ΔLL) from regression models that include one predictor of interest on top of the baseline regression model. *: $p < 0.001$.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Unnumbered "Limitations" section

☑ A2. Did you discuss any potential risks of your work?
Unnumbered "Ethics Statement" section

☑ 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 4.1

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The datasets used in this work are widely used in NLP research.

☑ 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)?
Unnumbered "Ethics Statement" section

☑ 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?
Unnumbered "Ethics Statement" section

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1

☑ 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 4.1

C ☑ Did you run computational experiments?

Sections 4, 5, 6

☑ 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 4.1

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?
Sections 4.1, 5.1, 6.1

☑ 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?
Sections 4.2, 5.2, 6.2

☑ 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.)?
Sections 4.1, 5.1, 6.1

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.?
No response.

□ 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)?
No response.

□ 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?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.