Better Explain Transformers by Illuminating Important Information

Linxin Song[1,4], **Yan Cui**[2], **Ao Luo**[1], **Freddy Lecue**[3] and **Irene Li**[4,5]
[1]Waseda University [2]Kyoto University [3]INRIA [4]University of Tokyo [5]Smartor.me, Inc
{songlx.imse.gt@ruri, luo.ao@toki}.waseda.jp,
yancui@kuicr.kyoto-u.ac.jp, freddy.lecue@inria.fr, ireneli@ds.itc.u-tokyo.ac.jp

Abstract

Transformer-based models excel in various natural language processing (NLP) tasks, attracting countless efforts to explain their inner workings. Prior methods explain Transformers by focusing on the raw gradient and attention as token attribution scores, where non-relevant information is often considered during explanation computation, resulting in confusing results. In this work, we propose highlighting the important information and eliminating irrelevant information by a refined information flow on top of the layer-wise relevance propagation (LRP) method. Specifically, we consider identifying syntactic and positional heads as important attention heads and focus on the relevance obtained from these important heads. Experimental results demonstrate that irrelevant information does distort output attribution scores and then should be masked during explanation computation. Compared to eight baselines on both classification and question-answering datasets, our method consistently outperforms with over 3% to 33% improvement on explanation metrics, providing superior explanation performance. Our anonymous code repository is available at: `https://github.com/LinxinS97/Mask-LRP`

1 Introduction

Transformer (Vaswani et al., 2017) currently serves as the fundamental structure for state-of-the-art models (Kenton and Toutanova, 2019; Radford et al., 2019; Liu et al., 2020; Touvron et al., 2023a,b). The power of these models provides convincing results in multiple Natural Language Processing (NLP) tasks. However, building a robust Transformer-based model to assist trustworthy human decision-making processes requires an understanding of the internal mechanisms of the Transformers (Kovaleva et al., 2019; Jain and Wallace, 2019; Qiang et al., 2022a).

In NLP tasks, tokens are prevalently utilized to signify a word or a fragment of a word (also known as a *subword*), serving as the input for Transformers. To comprehend the influence of input tokens on a Transformer, helping us to understand which part of input the Transformer is most interested in, a typical approach involves determining the *attribution score* of input tokens by leveraging the information captured by the attention matrix obtained from each attention head (Bach et al., 2015; Barkan et al., 2021; Voita et al., 2019; Chefer et al., 2021b,a). A high attribution score signifies that the input token likely plays a pivotal role in the model's decision-making process for a specific class, output word, or answer index.

To derive attribution scores for each input token, recent approaches utilized information within a trained Transformer, such as input-gradients (Shrikumar et al., 2017; Ancona et al., 2019), raw attention matrices (Abnar and Zuidema, 2020) or the combination of input-gradients and attention matrix (Barkan et al., 2021; Qiang et al., 2022b). The underlying premise for those methods is that input token gradients reflect the token's significance during backpropagation, while attention mechanisms capture the between-token interactions. However, both theoretical and empirical results (Chefer et al., 2021b; Qiang et al., 2022b; Ali et al., 2022) indicate that not all types of information embedded within the gradient and attention mechanisms contribute towards the explanations. They either fail to or can only partially aid in understanding which token primarily contributes to the Transformer's decision-making process.

To solve this issue, we follow the line of work known as Layer-wise Relevance Propagation (LRP, Bach et al. (2015)) with refined information flow to derive compelling attribution scores for each token. The information flow within LRP parameterized by each attention head mirrors that of the Transformer, concentrating on distinct portions of the input tokens, and attention heads focusing on irrelevant information can disrupt this flow, causing

Findings of the Association for Computational Linguistics: EACL 2024, pages 2048–2062

explanation confusion. We refine the information flow within LRP by illuminating the attention head that focuses on important information and reducing the attention head that zeroes in on less important information.

To achieve this, we illuminate the important attention head by adopting a head mask generated from dataset statistics. We first label the attention heads concentrating on a specific syntactic relationship as *syntactic* attention heads. Syntactic relations (e.g., nominal subject) are extensively utilized to define the relations between tokens in NLP (Voita et al., 2019), which establish a directional relation between two words. Furthermore, we designate the attention head that predominantly centers on a fixed relative position as a *positional* attention head, which reflects the internal feature (e.g., spatial position) of token embedding. We encapsulate *syntactic* and *positional* within a head mask, which we use to refine the information flow during the LRP process. To further reduce the irrelevant information, we obtain the attribution score by rolling out the relevance of the attention head from each attention blocks with the corresponding gradient (Chefer et al., 2021b).

To evaluate the performance of our method, we compared it with eight strong baselines across five classification datasets and two question-answering datasets. The results reveal that our method outperforms others in explanation performance, demonstrating a distinguished capacity to assign influential tokens from both interaction and internal perspectives. Furthermore, an ablation study uncovers that irrelevant information can obfuscate the LRP process, subsequently leading to a biased explanation of input tokens. The key contributions of our work can be summarized as follows:

1. We refine the information flow within the LRP process by illuminating two types of important information.

2. Through experiments, we demonstrated that irrelevant information hampers the LRP process.

3. Compared to previous state-of-the-art methods, our approach significantly improves explanation performance, achieving over 3.56% improvement in AOPC and LOdds for classification tasks and 33.02% for Precision@20 in question answering tasks.

2 Related Works

To explain a Transformer in NLP tasks, one common approach involves providing a post-hoc interpretable description of the Transformer's behavior. This approach assists users in understanding which input tokens most significantly influence the model's decision-making process. Abnar and Zuidema (2020) achieve this by leveraging the attention heads for defining more elaborate explanation mechanisms, while Wallace et al. (2019) and Atanasova et al. (2020) accomplish this by involving the Integrated Gradients or Input Gradients. Numerous models and domains have employed gradient methods such as Saliency Maps (Zhou et al., 2016; Barkan et al., 2021), Gradient×Input (Shrikumar et al., 2017; Srinivas and Fleuret, 2019; Hesse et al., 2021; Qiang et al., 2022b), or Guided Backpropagation (Zeiler and Fergus, 2014), and these methods have also been effectively transposed and applied to Transformers.

Concurrently, there have been several attempts to implement Layer-Wise Relevance Propagation (LRP, Bach et al. (2015)) in Transformers (Voita et al., 2019; Ali et al., 2022) and other attention-based models (Ding et al., 2017). LRP has been used to explain predictions of diverse models on NLP tasks, including BERT (Kenton and Toutanova, 2019). Other methodologies for LRP / gradient propagation in Transformer blocks can be found in (Chefer et al., 2021b,a), where the relevance scores are determined by combining attention scores with LRP or attention gradients.

Additionally, a few instances exist where perturbation-based methods have employed input reductions (Feng et al., 2018; Prabhakaran et al., 2019), aiming to identify the most relevant parts of the input by observing changes in model confidence or leveraging Shapley values (Lundberg and Lee, 2017; Atanasova et al., 2020). Furthermore, a line of work using tensor decomposition to decompose the attention matrix for a faithful Transformer explanation (Kobayashi et al., 2020, 2021; Modarressi et al., 2022; Ferrando et al., 2022).

3 Preliminary

3.1 Problem Formulation

This work focuses on post-hoc explanations of Transformer-based models, like BERT (Kenton and Toutanova, 2019; Liu et al., 2020) and GPT (Radford et al., 2019), across various NLP tasks. Given a dataset D with each input $\boldsymbol{x}_i$ consisting of T tokens, we use a fine-tuned Transformer-based language model, $f(\cdot;\boldsymbol{\theta})$, composed of B self-attention blocks with M attention heads each. We extract each model layer's output for analysis, with layer input denoted as $\boldsymbol{x}^{(n)}$ and n ranging from 1 to N. Here, $\boldsymbol{x}^{(N)}$ and $\boldsymbol{x}^{(1)}$ signify the model input and output, respectively, as information propagation starts from the output to the input.

We aim to understand the attribution of input $\boldsymbol{x}^{(N)} \in D$ to the output $\boldsymbol{x}^{(1)} \in \{c_1...c_K\}$ (K denoting classification task classes or question answering task tokens). We seek an attribution function $\boldsymbol{R}^{(N)} = R(\boldsymbol{x}^{(N)})$ evaluating each token's contribution to output $\boldsymbol{x}^{(N)}$. An ideal $\boldsymbol{R}^{(N)}$ assigns high attribution scores to influential tokens, causing output confidence to flatten or predictions to flip when these tokens are removed or masked.

3.2 Layer-wise Relevance Propagation

The Layer-wise Relevance Propagation (LRP, Bach et al. (2015)) is used to compute the attribution score $\boldsymbol{R}^{(N)}$ of each input token, propagating relevance from the predicted class or index backward to the input tokens.

The LRP applies the chain rule to propagate gradients with respect to the output $\boldsymbol{x}^{(1)}$ at index c, denoted as $\boldsymbol{x}_c^{(1)}$:

$$\nabla \boldsymbol{x}_j^{(n)} = \frac{\partial \boldsymbol{x}_c^{(1)}}{\partial \boldsymbol{x}_j^{(n)}} = \sum_i \frac{\partial \boldsymbol{x}_c^{(1)}}{\partial \boldsymbol{x}_i^{(n-1)}} \frac{\partial \boldsymbol{x}_i^{(n-1)}}{\partial \boldsymbol{x}_j^{(n)}}, \quad (1)$$

where j and i are element indices in $\boldsymbol{x}^{(n)}$ and $\boldsymbol{x}^{(n-1)}$ respectively. The layer operation on two tensors $\boldsymbol{X}$ and $\boldsymbol{Y}$ is denoted as $L^{(n)}$, typically indicating the input feature map and weights for layer n. The relevance propagation follows the Deep Taylor Decomposition (Montavon et al., 2017):

$$\begin{aligned} R_j^{(n)} &= \mathcal{G}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{R}^{(n-1)}) \quad (2) \\ &= \sum_i \boldsymbol{X}_j \frac{\partial L_i^{(n)}(\boldsymbol{X}, \boldsymbol{Y})}{\partial \boldsymbol{X}_j} \frac{R_i^{(n-1)}}{L_i^{(n)}(\boldsymbol{X}, \boldsymbol{Y})}, \end{aligned}$$

with j and i denoting elements in $R^{(n)}$ and $R^{(n-1)}$ respectively. This equation obeys the conservation rule:

$$\sum_j R_j^{(n)} = \sum_i R_i^{(n-1)}. \quad (3)$$

We begin relevance propagation with $R^{(0)}$ as a one-hot vector indicating the target class or index $c \in \boldsymbol{x}^{(1)}$.

LRP presumes non-negative activation functions and is incompatible with functions outputting both positive and negative values, like GELU (Hendrycks and Gimpel, 2016). As Chefer et al. (2021b) done, we overcome this by filtering out negative values and selecting the positive subset of indices $q = \{(i,j)|x_i w_{ij} \geq 0\}$ for relevance propagation:

$$\begin{aligned} R_j^{(n)} &= \mathcal{G}(x, w, q, R^{(n-1)}) \\ &= \sum_{\{i|(i,j)\in q\}} \frac{x_j w_{ji}}{\sum_{\{j'|(j',i)\in q\}} x_{j'} w_{j'i}} R_i^{(n-1)}. \quad (4) \end{aligned}$$

4 Layer-wise Relevance Propagation Through Important Attention Head

In this work, we empirically show that irrelevant information can detrimentally impact the LRP process. Therefore, our focus should be directed toward the important information while concurrently eliminating irrelevant information within the LRP process. In this section, we initially classify two kinds of important information (Sec.4.1), followed by introducing the method to extract this information in each layer (Sec.4.2). Subsequently, we illustrate the technique of concentrating on the important information extracted during Layer-wise Relevance Propagation (LRP, Sec. 4.3).

4.1 Important Information Flows in Transformer

Understanding Transformer-based models in NLP tasks entails grasping the important information each attention head prioritizes. This information in an input sentence comprises internal and interaction information (Voita et al., 2019; Qiang et al., 2022b). Interaction information explores if Transformer's encoder heads focus on tokens tied to core syntactic relationships, while internal information refers to an input where an attention head focuses on a fixed position for token embedding (Voita et al., 2019). In this work, to capture the above types of information, we identify two functions that attention heads might be playing: (1) syntactic: the head points to tokens in a specific syntactic relation, and (2) positional: the head points to a specific relative position. Not all syntactic relations are suitable for defining the core component of a sentence. De Marneffe et al. (2014) classifies the syntactic relations into nominal, clauses, modifier words, and function words. While nominal (subject,

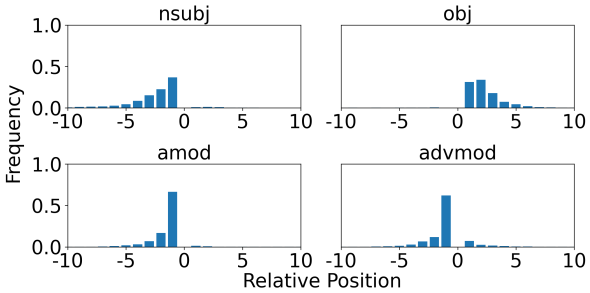

Figure 1: Distributions of the relative positions dependent for different syntactic relations in SST2.

object) and modifier words (adverb, adjectival modifier) are frequent, others like vocatives (common in conversations), expletives (e.g., "it" and "their" in English), and dislocated elements (frequent in Japanese) don't define a sentence's core and explain on them can confuse human understanding. Therefore, we identify four core syntactic relations: nominal subject (*nsubj*), direct object (*dobj*), adjectival modifier (*amod*), and adverbial modifier (*advmod*), which contains the core information of a whole sentence. The selected syntactic relations establish directional links between two words or linguistic units. For example, in "*The car is red*", *car* is the nsubj target for *red*. Hence, in LRP, important information the relevance contains of a layer input $\boldsymbol{x}^{(n_b)}$ in the self-attention block b at layer n_b can be decomposed as:

$$\boldsymbol{R}_{\text{imp}}^{(n_b)} = \boldsymbol{R}_{\text{synt}}^{(n_b)} + \boldsymbol{R}_{\text{pos}}^{(n_b)}, \tag{5}$$

where $\boldsymbol{R}_{\text{imp}}^{(n_b)}$ denotes the important information, $\boldsymbol{R}_{\text{synt}}^{(n_b)}$ and $\boldsymbol{R}_{\text{pos}}^{(n_b)}$ the information from syntactic relations and relative positions, respectively. The next section will detail preserving important information in the LRP process by identifying the important attention heads.

4.2 Identifying Important Heads

To illuminate the influence of the attention heads that are oriented towards important information, we create a head mask denoted as $\mathcal{M} \in \mathbb{R}^{B\times M}$ by combining two separate masks: $\mathcal{M}_{\text{synt}}$ and $\mathcal{M}_{\text{pos}}$. The mask $\mathcal{M}$ is constructed as follows:

$$\mathcal{M} = \mathcal{M}_{\text{synt}} + \mathcal{M}_{\text{pos}}. \tag{6}$$

$\mathcal{M}_{\text{synt}}$ represents the syntactic mask generated based on the statistical analysis of syntactic relations within each text, while the positional mask $\mathcal{M}_{\text{pos}}$ is derived from the positional analysis of the specific Transformer-based model chosen for the study.

Syntactic mask. We first obtain the distribution of the k-th syntactic relation at each token position, denoted as $\boldsymbol{\lambda}_k$. Here, λ_k^i represents the probability of the k-th syntactic relation appearing at position i (as depicted in Fig. 1). The attention head mask for syntactic relations, denoted as $\mathcal{M}_{\text{synt}}^{(b,m)}$, can be derived as follows:

$$\mathcal{M}_{\text{synt}}^{(b,m)} = \sum_{k\in K} \mathbb{1}_{\left\{\alpha_k^{(b,m)} > \max(\boldsymbol{\lambda}_k)+\xi_{\text{synt}}\right\}}, \tag{7}$$

where $K = \{\text{nsubj, dobj, amod, advmod}\}$ represents the set of core syntactic relations, $\alpha_k^{(b,m)} \in [0,1]$ denotes the frequency of the m-th attention head at block b assigning its highest attention weight to the k-th syntactic relation. The threshold ξ_{synt} determines the level of probability at which an attention head is considered syntactic relation-specific. In this work, we set $\xi_{\text{synt}} = 0.1$ to ensure that the selected attention head is not solely focused on a specific token position but exhibits a substantial probability of capturing syntactic relations.

Positional mask. We also examine attention heads that exhibit a high degree of focus on specific relative positions (e.g., ..., $-1, +1, +2$, ...). We refer to these attention heads as "positional" if, most of the time, their maximum attention weight is assigned to a specific relative position. To identify these attention heads, we utilize a positional mask denoted as $\mathcal{M}_{\text{pos}}^{(b,m)}$, which collects the indices of attention heads that satisfy the positional criteria. The positional mask is defined as follows:

$$\mathcal{M}_{\text{pos}}^{(b,m)} = \sum_{i\in I} \mathbb{1}_{\left\{\alpha_i^{(b,m)} > \xi_{\text{pos}}\right\}}, \tag{8}$$

where $\alpha_i^{(b,m)} \in [0,1]$ denotes the frequency of the m-th attention head at block b assigning its highest attention weight to the i-th relative position, $I = \{..., -1, +1, ...\}$ denotes the set of relative positions and ξ_{pos} is set to 0.8, as previously mentioned, to ensure that we capture attention heads primarily focusing on the positional information.

4.3 Layer-wise Relevance Propagation Through Important Heads

To gain deeper insights into the important information within the Transformer model, we specifically focus on the Layer-wise Relevance Propagation

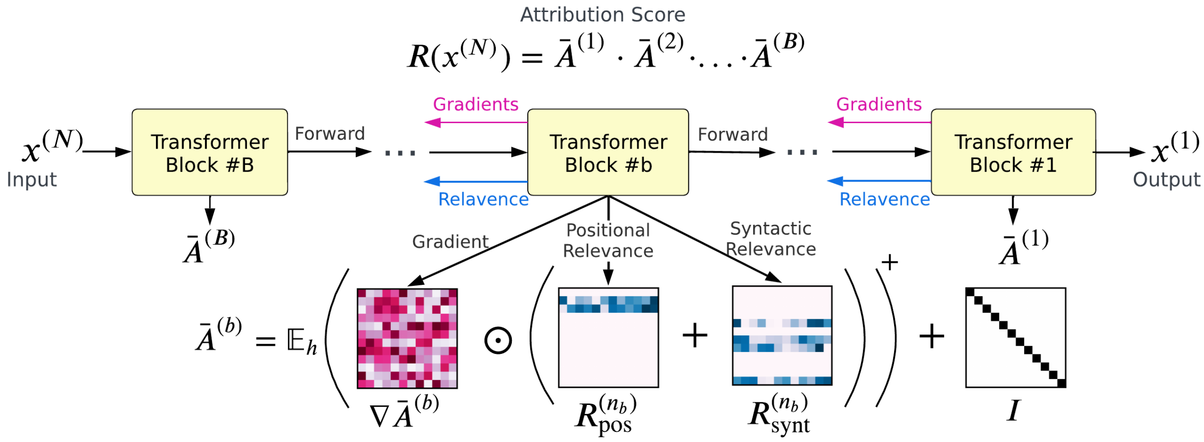

Figure 2: Illustration of our method. Gradients and relevance are propagated through the Transformer block from the final layer to the first layer. We extract two types of important information during the LRP process in all blocks by identifying the important heads.

(LRP) process between important attention heads across different layers and obtain the final attribution score. The process of our proposed method is illustrated in Fig. 2.

According to the type of information a relevance contains, the relevance of each attention head in the self-attention block at layer n_b can be defined as a combination of two types of relevance w.r.t. attention heads: important relevance and irrelevant relevance. Recalling the Eq. (2) and (5), we have:

$$\begin{aligned} \boldsymbol{R}^{(n_b)} &= \mathcal{G}\left(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{R}^{(n_b-1)}_{\text{imp}} + \boldsymbol{R}^{(n_b-1)}_{\text{others}}\right) \\ &= \mathcal{G}\left(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{R}^{(n_b-1)}_{\text{synt}} + \boldsymbol{R}^{(n_b-1)}_{\text{pos}} + \boldsymbol{R}^{(n_b-1)}_{\text{others}}\right), \end{aligned} \tag{9}$$

in each Transformer block. Here, $\boldsymbol{R}^{(n_b-1)}_{\text{others}}$ corresponds to the relevance output from attention heads that are not specific to important information. To highlight the important relevance $\boldsymbol{R}^{(n_b-1)}_{\text{imp}}$ in the LRP process, we employ the b-th block's mask $\mathcal{M}^{(b)}$ obtaining from Eq. (6):

$$\boldsymbol{R}^{(n_b)} := \boldsymbol{R}^{(n_b)}_{\text{synt}} + \boldsymbol{R}^{(n_b)}_{\text{pos}} = \mathcal{G}(\boldsymbol{X}, \boldsymbol{Y}, \mathcal{M}^{(b)} \boldsymbol{R}^{(n_b-1)}).$$

To keep the conservation after adopting the mask, we apply normalization to $\boldsymbol{R}^{(n_b)}_{\text{synt}}$ and $\boldsymbol{R}^{(n_b)}_{\text{pos}}$ as follows:

$$\boldsymbol{R}^{(n_b)}_{\text{synt}} := \boldsymbol{R}^{(n_b)}_{\text{synt}} \frac{\left|\sum R^{(n_b)}_{\text{synt}}\right|}{\left|\sum R^{(n_b)}\right|} \cdot \frac{\sum R^{(n_b-1)}}{\sum R^{(n_b)}_{\text{synt}}},$$

$$\boldsymbol{R}^{(n_b)}_{\text{pos}} := \boldsymbol{R}^{(n_b)}_{\text{pos}} \frac{\left|\sum R^{(n_b)}_{\text{pos}}\right|}{\left|\sum R^{(n_b)}\right|} \cdot \frac{\sum R^{(n_b-1)}}{\sum R^{(n_b)}_{\text{pos}}}.$$

The normalization step ensures the conservation rule is maintained, i.e., $\sum R^{(n_b)}_{\text{synt}} + \sum R^{(n_b)}_{\text{pos}} = \sum R^{(n_b-1)}$. Note that we have omitted the subscript of the index (e.g., i, j) to enhance readability.

We output the final attribution $\boldsymbol{R}^{(N)}$ by leveraging the rollout of weighted attention relevance (Chefer et al., 2021b) of each block b:

$$\bar{\boldsymbol{A}}^{(b)} = \mathbb{E}_h\left(\nabla \boldsymbol{A}^{(b)} \odot \left(\boldsymbol{R}^{(n_b)}_{\text{synt}} + \boldsymbol{R}^{(n_b)}_{\text{pos}}\right)\right)^{+} + I \tag{10}$$

$$\boldsymbol{R}(x^{(N)}) = \bar{\boldsymbol{A}}^{(1)} \cdot \bar{\boldsymbol{A}}^{(2)} \cdot \ldots \cdot \bar{\boldsymbol{A}}^{(B)}, \tag{11}$$

where $\odot$ denotes the Hadamard product, $\boldsymbol{A}^{(b)} = \text{softmax}(\boldsymbol{Q}^{(b)} \cdot \boldsymbol{K}^{(b)\top}/\sqrt{d_h})$ is the attention matrix obtain from query $\boldsymbol{Q}$ and key $\boldsymbol{K}$ in block b, and $\nabla \boldsymbol{A}^{(b)}$ denotes the corresponding gradient. We use the superscript a^{+} to denote the operation $max(0, a)$.

5 Experiment

5.1 Experiment Setup

Implementation details. For the classification task, we use pretrained $\text{BERT}_{\text{base}}$ (Kenton and Toutanova, 2019) with a 512 token input limit and attribute the [CLS] token as the classifier input. For question answering, we compare our method with three baselines using pretrained $\text{BERT}_{\text{base}}$, GPT-2 (Radford et al., 2019), and RoBERTa (Liu et al., 2020), assessing the effect of model scale and tokenizer on information flow. We evaluate the attribution of the start and end answer indices.

Our model-agnostic method can apply to various Transformer-based models with minimal modifications. We obtain all results from the validation set across all methods, focusing on the post-hoc explanation with fixed model parameters. Variance is limited to the baseline using a randomly generated mask.

Datasets. We choose the validation set on seven datasets across the sentiment classification: SST-2 (Socher et al., 2013), IMDB (Maas et al., 2011), Yelp Polarity (Zhang et al., 2015), duplicated question classification: QQP (Chen et al., 2018), natural language inference: MNLI (Williams et al., 2018) and question answering: SQuADv1 (Rajpurkar et al., 2016) and SQuADv2 (Rajpurkar et al., 2018) to evaluate all methods. SST-2, IMDB, and Yelp Polarity take a single sentence as input, while QQP and MNLI use a pair of sentences for their target. Specifically, we extract the data marked as *duplicate* (with ground truth label 1) in QQP for evaluation. Details of the model and datasets are in Appendix C.

Evaluation metrics. We use AOPC and LOdds for classification evaluation, and precision@20 for question-answering evaluation. To evaluate post-hoc explanation interpretability in a classification task, we measure model confidence for a specific class before and after masking influential tokens, using both linear (AOPC) and non-linear (LOdds) metrics (Qiang et al., 2022b). AOPC and LOdds aim to detect the change of confidence before and after the influential tokens are removed, which are formularized as:

$$\text{AOPC}(k) = \frac{1}{T}\sum_{t=1}^{T} f_{\widehat{y}}\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right) - f_{\widehat{y}}\left(\widetilde{\boldsymbol{x}}i^k;\boldsymbol{\theta}\right), \tag{12}$$

$$\text{LOdds}(k) = \frac{1}{T}\sum_{t=1}^{T} \log\frac{f\left(\widetilde{\boldsymbol{x}}_i^k;\boldsymbol{\theta}\right)}{f\left(\boldsymbol{x}_i;\boldsymbol{\theta}\right)}, \tag{13}$$

where $\widetilde{\boldsymbol{x}}_i^k$ denotes the top-$k\%$ masked input tokens ranked by the attribution score $R(\boldsymbol{x}_i^{(N)})$. $f_{\widehat{y}}(\cdot;\boldsymbol{\theta})$ denotes the model's max confidence w.r.t label $\widehat{y}$. Furthermore, we use precision@20 to evaluate the question answering task (SQuADv1 and SQuADv2). In QA tasks, precision@20 will not introduce bias because it will not remove the ground truth answer from the input, and the model that has a low precision@20 means that the model cannot capture a correct mapping between the answer part and the ground truth index.

Hyperparameters In this work, we use two hyperparameters: ξ_{synt} and ξ_{pos} for the corresponding masks. As we mentioned in the main context, we choose 0.1 for ξ_{synt} and 0.8 for ξ_{pos}. One reason why we choose these values is that we empirically found that the highest frequency for the syntactic relations is almost lower than 0.7 for a specific relative position. Therefore, $\xi_{\text{synt}} = 0.1$ ensure the syntactic mask effectively filters out the attention head, which is focusing on irrelevant information, or just focusing on a specific position, and $\xi_{\text{pos}} = 0.8$ help us to capture the rest attention heads that are focusing mainly on a specific relative position, which is filtered by the syntactic mask. Although the two masks are complementary, many attention heads still focus on various relative positions so that we cannot identify their function and mark them as irrelevant attention heads.

5.2 Baselines

We categorize eight baselines into three groups based on their characteristics with one additional random baseline:

Attention maps : **RawAtt** (Abnar and Zuidema, 2020) uses the mean attention weights from the final Transformer block as attribution scores, while **Rollout** (Abnar and Zuidema, 2020) rolls out average attention weights from all Transformer blocks.

Relevance-based : **LRP** (Bach et al., 2015) uses output-to-input layer relevance as attribution scores. **PartialLRP** (Voita et al., 2019) calculates relevance at the model's final layer. **GAE** (Chefer et al., 2021a) propagates attention gradients to the final layer to obtain attribution scores.

Gradient-based : **CAM** (Zhou et al., 2016) and **GradCAM** (Barkan et al., 2021) use the final layer gradient and its weighted version by final layer attention respectively as attribution scores. **AttCAT** (Qiang et al., 2022b) combines the summation of attention weight from each Transformer block with input gradient.

In addition, we include **Random**, a baseline using a randomly generated mask (maintaining the same mask rate, i.e., $\|\mathcal{M}_{\text{random}}\| = \|\mathcal{M}_{\text{ours}}\|$, as our method) to show that our method effectively identifies the crucial head in the Transformer model.

5.3 Results

We assessed the explanation performance of each method within classification tasks by computing mean AOPC and LOdds across five benchmark

Methods	SST-2		IMDB		Yelp		MNLI		QQP	
	AOPC ↑	LOdds ↓	AOPC ↑	LOdds ↓	AOPC ↑	LOdds ↓	AOPC ↑	LOdds ↓	AOPC ↑	LOdds ↓
RawAtt	0.374	-0.992	0.354	-1.593	0.376	-1.513	0.135	-0.399	0.447	-5.828
Rollout	0.337	-0.911	0.334	-1.456	0.244	-0.770	0.137	-0.396	0.437	-5.489
LRP	0.336	-0.888	0.288	-1.271	0.163	-0.464	0.131	-0.395	0.438	-5.745
PartialLRP	0.396	-1.052	0.370	-1.726	0.401	-1.688	0.136	-0.401	0.445	-5.718
GAE	0.423	-1.171	0.384	-1.853	0.404	-1.682	0.144	-0.421	0.447	-5.923
CAM	0.399	-1.086	0.365	-1.883	0.298	-1.473	0.132	-0.386	0.450	-5.988
GradCAM	0.341	-0.855	0.236	-0.974	0.104	-0.229	0.126	-0.369	0.449	-5.953
AttCAT	0.405	-1.110	0.340	-1.697	0.397	**-2.034**	0.138	-0.419	0.447	-5.897
Random	$0.432_{\pm.005}$	$-1.205_{\pm.004}$	$0.387_{\pm.004}$	$-1.898_{\pm.003}$	$0.426_{\pm.005}$	$-1.886_{\pm.007}$	$0.142_{\pm.002}$	$-0.415_{\pm.021}$	$0.448_{\pm.001}$	$-5.998_{\pm.012}$
Ours	**0.438**	**-1.208**	**0.392**	**-1.906**	**0.434**	-1.898	**0.148**	**-0.445**	**0.451**	**-6.001**

Table 1: AOPC and LOdds results of all methods in explaining $\text{BERT}_{\text{base}}$ model on each dataset. The best results are marked in bold. Note that a method with high AOPC and low LOdds is desirable, indicating a strong ability to mark influential tokens. The results of the Random mask are average and standard deviation between five runs. We also provide the comparison with SOTA tensor decomposition method in Appendix B.

Method	SQuADv1			SQuADv2		
	$\text{BERT}_{\text{base}}$	GPT-2	RoBERTa	$\text{BERT}_{\text{base}}$	GPT-2	RoBERTa
Rollout	4.62	5.86	8.04	6.15	5.54	5.87
RawAtt	36.33	28.97	45.61	4.69	27.85	18.09
AttCAT	31.44	17.53	47.32	18.81	16.99	23.39
Ours	**52.97**	**51.62**	**67.31**	**27.03**	**49.63**	**56.41**

Table 2: Precision@20 results of the selected explanation methods on SQuAD datasets. Higher Precision@20 is better, indicating the marked influential tokens highly overlap with the answer text.

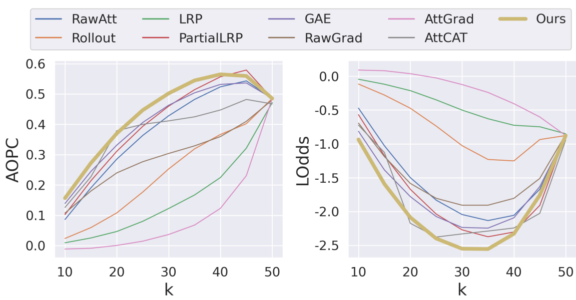

Figure 3: AOPC and LOdds scores of different methods in explaining $\text{BERT}_{\text{base}}$ against the corruption rate k on SST-2. Note that higher AOPC and lower LOdds scores are better.

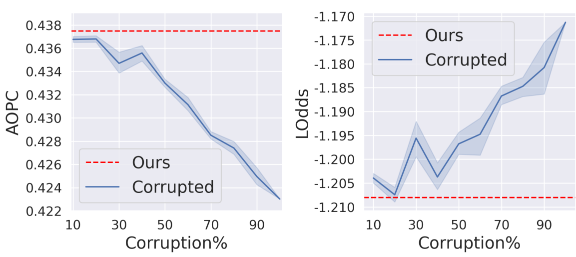

Figure 4: Comparison before and after corrupting the generated mask on SST-2. The blue line combines the solid line (average values) and shadow areas (standard deviation). The method's ability to explain becomes dropped after adding corruption.

datasets, detailed in Tab.1. Remarkably, the performance across all post-hoc explanation methods remained stable, independent of random initialization, except for a randomly initialized mask method. Our approach generally surpassed others, achieving the highest AOPC and lowest LOdds, indicating superior accuracy in identifying influential tokens. Fig.3 displays performance curves against pruning rate k, endorsing our method's performance at every rate. It consistently outperformed gradient-based methods, particularly in handling lengthy token lists. Attention information from larger matrices often includes irrelevant details that assign high attribution to non-influential tokens, reducing the quality of explanations (see Sec.5.4 for more). For the question-answering task, we evaluated Precision@20 on two SQuAD datasets. As per Tab.2, our method consistently outperformed the baselines, demonstrating accurate attribution to influential answer tokens.

5.4 Assessing the Impact of Important and Irrelevant Information

In this section, we seek to address two key questions: (1) does our method effectively identify the attention head that focuses on important information? and (2) does the residual, irrelevant information that other heads concentrate on adversely affect the explanation?

To answer the first question, we carry out an ablation study where we replace our mask with a randomly generated mask, maintaining the same mask rate as discussed in Sec.5.2, to examine if this

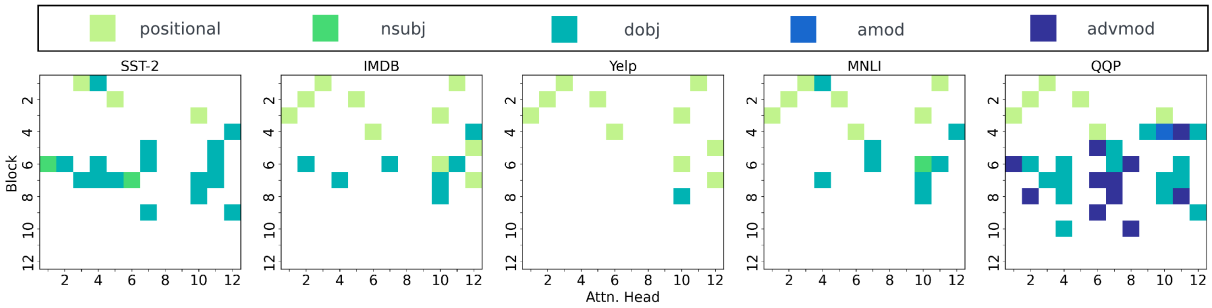

Figure 5: Different types of important heads in $\text{BERT}_{\text{base}}$ model cross different dataset. The x-axis denotes the position of the attention head, while the y-axis is the position of the Transformer block. It is obvious that attention heads in previous blocks tend to focus on simple internal information (e.g., position), while attention heads in later blocks tend to focus on the complex interactions between tokens (e.g., syntactic relations).

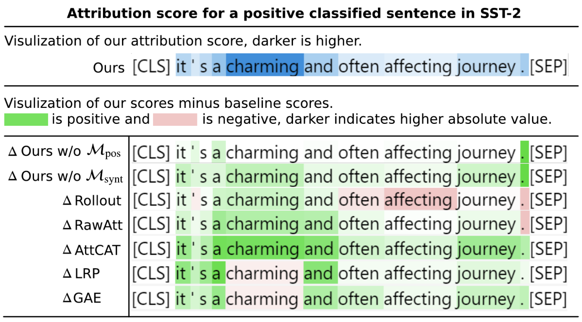

Figure 6: The comparison of attribution scores between our method (shown in the first line) and baselines on a positive classified sentence. Tokens highlighted in green represent those receiving more attention from our method than the baseline, while those in red signify the opposite. Our method emphasizes more on both internal and interaction information. We put results of other datasets in Appendix.D

alteration impacts the explanatory capacity. The results, as reported on line 12 in Tab.1, clearly demonstrate that our method consistently outperforms the variant with a randomly generated mask. This underscores that our method is capable of identifying a set of attention heads that can robustly explain the information flow within a Transformer.

For the second question, we derive our answer by collating findings from Tab.1 and Fig.4. We discover from Tab. 1 that even with a random mask, our method exhibits superior explanation performance than other relevance-based methods such as GAE because of the less focus on irrelevant information. This suggests that irrelevant information flow in the Transformer greatly affects the LRP, thereby confusing the explanation of input tokens. In addition, we conducted another ablation study where we randomly switched a portion of the remaining zeros in $\mathcal{M}_{\text{ours}}$. These zeros in the mask correspond to the irrelevant information the Transformer focuses on, and their alteration can be interpreted as a corruption of the generated mask. If our method employs a 100% corrupted mask (a mask filled with ones), it degenerates to GAE. We observed the variance in explanation performance at different corruption rates (ranging from 10% to 100%) on SST-2, the results of which are displayed in Fig. 4. Notably, it is clear that the rate of performance decline is closely related to the corruption rate and ultimately converges to the performance of GAE. This evidence substantiates the notion that irrelevant information can interfere with the LRP process at each layer, thereby resulting in a perplexing explanation.

5.5 Visualizing and Analyzing Extracted Attention Heads

We visualize both $\mathcal{M}_{\text{synt}}$ and $\mathcal{M}_{\text{pos}}$ that our method extracted from $\text{BERT}_{\text{base}}$ according to each dataset. The resulting visualizations are presented in Fig. 5. We discovered that positional attention heads are predominantly concentrated in the earlier blocks, whereas syntactic attention heads tend to gather in the later blocks. This observed phenomenon suggests that Transformers initially learn the simplistic internal information and subsequently propagate this internal information to the subsequent layers. This aids the attention heads in these later layers in capturing the interaction information between tokens. Additionally, we found that during model training on more datasets with long input tokens, such as IMDB and Yelp, there are only a few heads with unipolar function, that is, a head focusing solely on a single pattern, and those heads are filtered by our mask. Yet, as the experiment results in Sec. 5.3 illustrate, the attribution scores assigned

Method	SST-2		QQP	
	AOPC	LOdds	AOPC	LOdds
Ours	0.438	-1.208	0.451	-6.001
Ours w/o $\mathcal{M}_{\text{pos}}$	0.438	-1.208	0.450	-6.001
Ours w/o $\mathcal{M}_{\text{synt}}$	0.437	-1.205	0.449	-5.998

Table 3: Explanation performance comparison of different masks. Only use $\mathcal{M}_{\text{pos}}$ or $\mathcal{M}_{\text{synt}}$ still have strong explanation performance.

solely by these heads are representative enough to provide a persuasive explanation. This implies that for binary classification tasks, the important information flow can be remarkably simple, even in the context of complex inputs. We also examine the explanation performance differences when using $\mathcal{M}$ compared to solely utilizing $\mathcal{M}_{\text{pos}}$ or $\mathcal{M}_{\text{synt}}$ in Tab. 3. Interestingly, we discover that eliminating one type of mask doesn't substantially impact the explanation performance. This can be attributed to the fact that a single mask type does not alter the ranking of output attribution but rather enriches its detail. Additional insights are provided in the subsequent paragraph.

To delve deeper into the attributions assigned by these important heads, we visualized the difference in attribution scores allocated by our method and other baseline methods in Fig. 6. The sentence, randomly selected from the SST-2 dataset and depicted in Fig.6, is annotated with a positive sentiment. Compared to attention-based methods (Rollout, RawAtt, AttCAT), our approach de-emphasizes less crucial tokens like *affecting*, emphasizing important ones like *charming*. Also, unlike relevance-based methods (LRP, GAE) that overlook *journey*, our method pays attention to it due to its link with *charming* via *and*. Thus, our method successfully extracts interaction information, attributing scores based on both single tokens' internal information and their interplay.

6 Conclusion

In this study, we propose that irrelevant information in the gradient and attention hampers the explanation process. To address this, we improve the information flow in the LRP process by masking irrelevant attention heads. By illuminating the important information, we show that explanations become more convincing. Our method outperforms nine baseline methods in classification and question answering tasks, consistently delivering better explanation performance.

Limitations

Though our method is model-agnostic, limitations in computational resources prevent us from fully exploring its implications for Large Language Models (LLMs) like LLAMA and LLAMA-2 (Touvron et al., 2023a,b), but we provided the implementation in our repository. We conjecture that LLMs may learn advanced interaction information surpassing the syntactic relationships we defined. This high-level interaction information could potentially allow LLMs to grasp the interplay between sentences or even broader structures like topics, complementing existing research on Transformers' topic learning capability via self-attention mechanisms (Li et al., 2023). Additionally, while we've empirically shown that irrelevant information hinders the LRP process, the origins and contents of this irrelevant information remain obscure. We will delve deeper into the nature of such information in future work.

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.

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. 2022. Xai for transformers: Better explanations through conservative propagation. In *International Conference on Machine Learning*, pages 435–451. PMLR.

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2019. *Gradient-Based Attribution Methods*, pages 169–191. Springer International Publishing, Cham.

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.

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

Oren Barkan, Edan Hauon, Avi Caciularu, Ori Katz, Itzik Malkiel, Omri Armstrong, and Noam Koenigstein. 2021. Grad-sam: Explaining transformers via gradient self-attention maps. In *Proceedings of the*

30th ACM International Conference on Information & Knowledge Management*, pages 2882–2887.

Hila Chefer, Shir Gur, and Lior Wolf. 2021a. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 397–406.

Hila Chefer, Shir Gur, and Lior Wolf. 2021b. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791.

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning. 2014. Universal stanford dependencies: A cross-linguistic typology. In *LREC*, volume 14, pages 4585–4592.

Yanzhuo Ding, Yang Liu, Huanbo Luan, and Maosong Sun. 2017. Visualizing and understanding neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1150–1159, Vancouver, Canada. Association for Computational Linguistics.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. 2021. Fast axiomatic attribution for neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 19513–19524. Curran Associates, Inc.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. In *Proceedings of NAACL-HLT*, pages 3543–3556.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. 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 4365–4374.

Yuchen Li, Yuanzhi Li, and Andrej Risteski. 2023. How do transformers learn topic structure: Towards a mechanistic understanding. *arXiv preprint arXiv:2303.04245*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Ro{bert}a: A robustly optimized {bert} pretraining approach.

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

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. 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 5740–5745, Hong Kong, China. Association for Computational Linguistics.

Yao Qiang, Chengyin Li, Marco Brocanelli, and Dongxiao Zhu. 2022a. Counterfactual interpolation augmentation (cia): A unified approach to enhance fairness and explainability of dnn. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI*, pages 732–739.

Yao Qiang, Deng Pan, Chengyin Li, Xin Li, Rhongho Jang, and Dongxiao Zhu. 2022b. Attcat: Explaining transformers via attentive class activation tokens. In *Advances in Neural Information Processing Systems*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

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*, volume 70 of *Proceedings of Machine Learning Research*, pages 3145–3153. PMLR.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Suraj Srinivas and François Fleuret. 2019. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Eric Wallace, Jens Tuyls, Junlin Wang, Sanjay Subramanian, Matt Gardner, and Sameer Singh. 2019. AllenNLP interpret: A framework for explaining predictions of NLP models. 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): System Demonstrations*, pages 7–12, Hong Kong, China. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2018*, pages 1112–1122. Association for Computational Linguistics (ACL).

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929.

A Why do we choose nsubj, dobj, amod, and advmod?

Many syntactic relations exist, but not all are suitable for defining the core component of a sentence. De Marneffe et al. (2014) classifies the syntactic relations into nominals, clauses, modifier words, and function words. While nominals (subject, object) and modifier words (adverb, adjectival modifier) are frequent, others like vocatives (common in conversations), expletives (e.g., "it" and "their" in English), and dislocated elements (frequent in Japanese) don't define a sentence's core and explain on them can confuse human understanding.

B Extra experiment comparing with tensor decomposition method

We provide the comparison results between ours and the SOTA tensor decomposition method ALTC (Ferrando et al., 2022) in Table 4.

Methods	SST-2		IMDB		Yelp	
	AOPC ↑	LOdds ↓	AOPC ↑	LOdds ↓	AOPC ↑	LOdds ↓
ALTC	0.369	-0.866	0.342	-0.748	0.363	-1.428
Ours	**0.438**	**-1.208**	**0.392**	**-1.906**	**0.434**	**-1.898**

Table 4: AOPC and LOdds results of ALTC and ours in explaining $\text{BERT}_{\text{base}}$ model on SST-2, IMDB, and Yelp. The best results are marked in bold. Note that a method with high AOPC and low LOdds is desirable, indicating a strong ability to mark influential tokens.

C Extra Implementation Details

Environment We run all experiments on the device with the following specs:

- System: Ubuntu 20.04.4 LTS
- CPU: Intel(R) Xeon(R) Platinum 8368 @ 2.40GHz (36 Cores / 72 Threads)
- GPU: NVIDIA A100 SXM4 40GB
- Memory: 230GB

With the above specs, we can complete the evaluation of one dataset within one hour by adopting the multi-process.

Datasets The task, amount of training, validation, and testing set numbers are shown in Tab. 5. Note that the dataset of IMDB and Yelp Polarity does not contain a validation set, so we use the test set for our experiment. Moreover, in QQP, data points are annotated with a binary label as *duplicated* or *not duplicated*. If we remove the influential tokens in those data marked as *not duplicated*, the model's prediction does not change because the two questions remain different. Therefore, we select the data marked as *duplicated* for our experiments to see the changing of the model's prediction from *duplicated* to *not duplicated*.

Dataset	Task	Train	Valid	Test
SST-2	Classification	6,920	872	1,821
IMDB	Classification	25,000	-	25,000
Yelp Polarity	Classification	560,000	-	38,000
QQP	Question Paring	363,846	40,430	390,965
MNLI	Natural Language Inference	392,702	20,000	20,000
SQuADv1	Question Answering	87,599	10,570	9,533
SQuADv2	Question Answering	130,319	11,873	8,862

Table 5: Statistics for the benchmark dataset we used in this work. Note that IMDB and Yelp Polarity only contains training and test set.

Models In this work, we use different pretrained models archived in Hugging Face [1] for each task and modify them to adjust for LRP in our implementation. The models we use for different tasks are shown in Tab. 6. Note that there does not exist GPT-2 model pretrained on SQuADv2, so we adopt the model trained on SQuADv1 for SQuADv2 experiments, which also provides convincing performance.

Dataset	Model	Huggingface Repo
SST-2	$\text{BERT}_{\text{base}}$	textattack/bert-base-uncased-SST-2
IMDB	$\text{BERT}_{\text{base}}$	textattack/bert-base-uncased-imdb
Yelp	$\text{BERT}_{\text{base}}$	abriceyhc/bert-base-uncased-yelp_polarity
QQP	$\text{BERT}_{\text{base}}$	modeltc/bert-base-uncased-qqp
MNLI	$\text{BERT}_{\text{base}}$	textattack/bert-base-uncased-MNLI
SQuADv1	$\text{BERT}_{\text{base}}$	csarron/bert-base-uncased-squad-v1
	GPT-2	anas-awadalla/gpt2-span-head-finetuned-squad
	RoBERTa	thatdramebaazguy/roberta-base-squad
SQuADv2	$\text{BERT}_{\text{base}}$	ericRosello/bert-base-uncased-finetuned-squad-frozen-v2
	GPT-2	anas-awadalla/gpt2-span-head-finetuned-squad
	RoBERTa	21iridescent/roberta-base-finetuned-squad2-lwt

Table 6: Baseline models of different datasets and their Hugging Face repositories.

D Additional Visualization Results

In this section, we provide visualization results of the attribution score difference in MNLI (Fig. 7, 8 and 9), IMDB (Fig. 10 and 11), and Yelp (Fig. 12 and 13), which include the task of classification of sentence pair and long text and each dataset, we randomly obtain a data from each class. For all of the above figures, as we mentioned in Fig. 6, tokens highlighted in green represent those receiving more attention from our method than the baseline,

[1] https://huggingface.co/

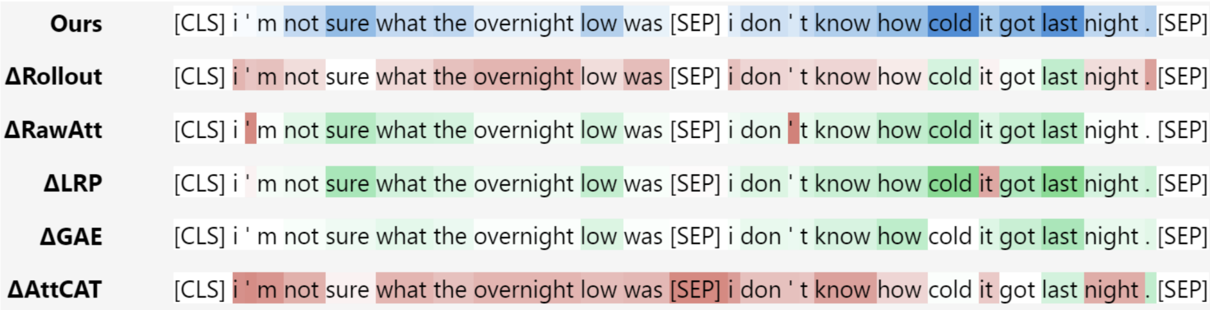

Figure 7: The comparison of attribution scores between our method (shown in the first line) and baselines on an **entailment** classified sentence pair in MNLI.

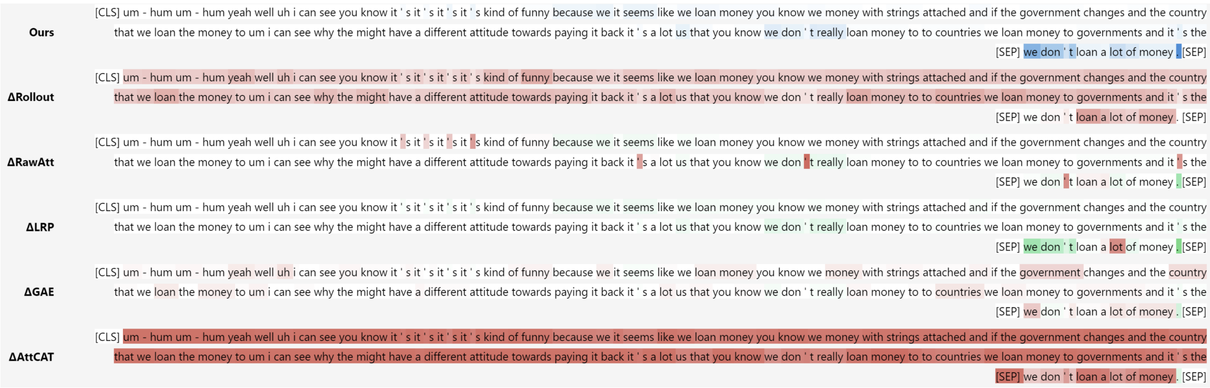

Figure 8: The comparison of attribution scores between our method (shown in the first line) and baselines on a **neutral** classified sentence pair in MNLI.

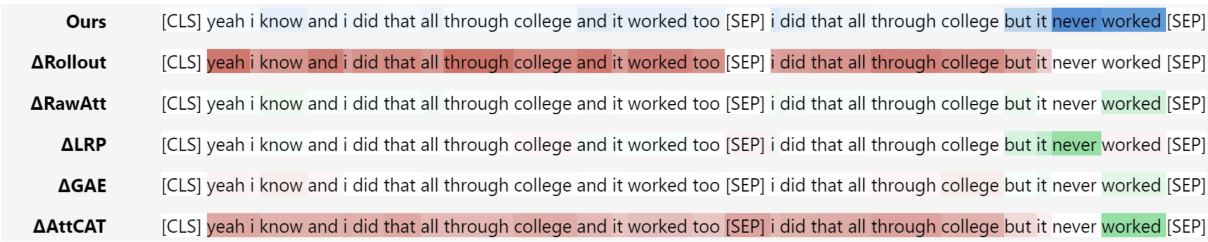

Figure 9: The comparison of attribution scores between our method (shown in the first line) and baselines on a **contradiction** classified sentence pair in MNLI.

while those in red signify the opposite. Our method emphasizes more on both internal and interaction information.

Figure 10: The comparison of attribution scores between our method (shown in the first line) and baselines on a **negative** classified comment in IMDB.

Figure 11: The comparison of attribution scores between our method (shown in the first line) and baselines on a **positive** classified comment in IMDB.

Figure 12: The comparison of attribution scores between our method (shown in the first line) and baselines on a **negative** classified comment in Yelp Polarity.

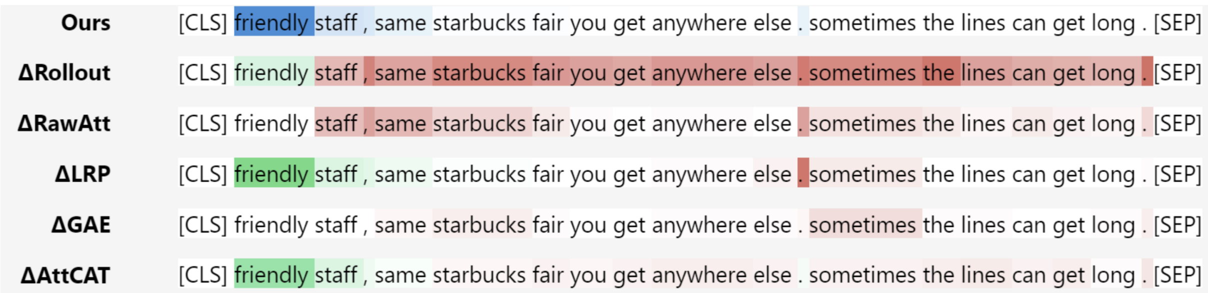

Figure 13: The comparison of attribution scores between our method (shown in the first line) and baselines on a **positive** classified comment in Yelp Polarity.