Do Pre-Trained Language Models Truly Focus on the Content They Are Expected to?

Anonymous ACL submission

Abstract

Pre-trained language models (PLMs) have significantly revolutionized various natural language processing tasks, showcasing extraordinary capabilities in text comprehension and processing. Despite their widespread success, the elucidation of PLMs' interest towards the input texts remains unclear, *i.e.,* which part of the inputs gains models' attention. Existing methods either rely on various stringent assumptions or ignore the intricate dependency relations inherent in natural language, causing inaccurate estimation results. In response to this limitation, this paper introduces a novel perturbation-based method for estimating the PLMs' interest, comprising two crucial designs, *i.e.,* the co-perturbation strategy and an adaptive optimization algorithm. Specifically, the strategy aims to inject noises across all input words, thereby confronting the inherent combinatorial explosion challenge. Furthermore, the proposed adaptive algorithm focuses on the estimation of interest degree for disentangling the output changes caused by the co-perturbation setting. Through extensive experimentation on various PLMs and datasets, we verify the effectiveness of the proposed method.

1 Introduction

As a burgeoning direction, pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; Black et al., 2021; Touvron et al., 2023) have emerged as the cornerstone of natural language processing (NLP) research as they could provide the vast amounts of knowledge encoded in their parameters, showing stunning performance in various downstream tasks (Singh et al., 2020; Han et al., 2021). Despite their success, it remains unclear: *whether these models truly focus on the content they are expected to?* This question underscores the necessity of investigating PLMs' interest towards the input texts, *i.e.,* examining the models' attention degree to each word of inputs. Additionally, this exploration could also provide insights for us to understand the underlying reason for incorrect model predictions (see Figure 1 for an illustration), which could be instrumental in enhancing model performance for downstream applications (see Section 5.4). Therefore, this paper attempts to make a preliminary exploration of quantifying PLMs' interest towards the input texts.

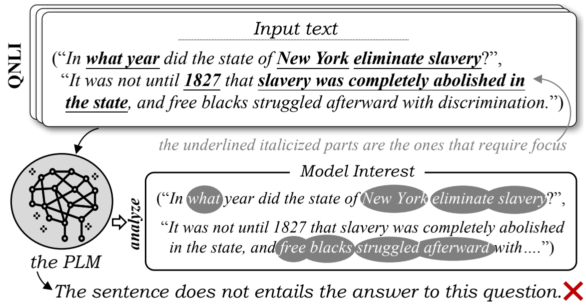


Figure 1: Illustration of the motivation. The underlined words represent the parts expected to garner the model's focus. These words are manually highlighted in this figure to facilitate a clear and direct comparison with the areas of the estimated PLM interest in this sentence (the circled parts).

Although extensive works have delved into several related aspects, including the analysis of the self-attention mechanism through visualization methods (Hoover et al., 2020; Jaunet et al., 2022; Yeh et al., 2023), function-based methods (Barkan et al., 2021; Hao et al., 2021) or probing-based methods (Sorodoc et al., 2020; Mohebbi et al., 2021; Niu et al., 2022) debates over their validity persist (Zhao et al., 2024). Additionally, while several input attribution methods (Lundberg and Lee, 2017; Prabhakaran et al., 2019; Ali et al., 2022; Feng et al., 2024)) in explainable machine learning appear capable of estimating the contribution of input features to model predictions, they will encounter various problems when applied to language models (see Section 2.1). Consequently, current literature lacks a straightforward method to evaluate how PLMs distribute their attention across input texts. To bridge this gap, this paper harnesses the perturbation theory, given its proven efficacy in the realm of machine learning (Ivanovs et al., 2021; Louis et al., 2022). In general, this theory entails introducing noises into input features and monitoring the consequent impact on outputs (Guan et al., 2019; Louis et al., 2022).

Guided by the principle of perturbation theory, it is imperative to perturb the words in input texts one by one and then assess the resultant changes in the model's predictions. To elaborate, under the same perturbation, the magnitude of change in the outputs reflects the model's interest toward that specific word, *i.e.,* a larger change indicates the greater interest and vice versa. However, the intricate dependency relations (Manning et al., 2008) inherent in natural language reveals that quantifying model interest by perturbing each word in isolation may not yield reliable results. For instance, as depicted in Figure 1, the model interest in words such as *New* and *York* are strongly interrelated, indicating their interest estimation should not be conducted separately. This observation illustrates the necessity of manually identifying potential combinations (*e.g.,* selecting the collection "*New York*") and perturbing them together, inevitably leading to the well-known challenge of combinatorial explosion (Khakzar et al., 2019; Ivanovs et al., 2021) and rendering automated model interest estimation. Hence, this situation highlights a conflict: the inaccuracy of assessing interest via word-by-word versus the combinatorial complexity through combination-by-combination. To address this problem, we propose an intuitive co-perturbation strategy that introduces noises to all input words simultaneously, *e.g.,* perturbing all words of the sentence in Figure 1 at once. Unfortunately, this strategy brings a complication in gauging the model's interest towards each individual word, as final changes are the collective results of perturbations applied to each word.

To address this dilemma, we further propose an adaptive estimation algorithm that synergizes the maximum likelihood estimation (MLE) and the maximum entropy principle (MEP) to construct the optimization target. Specifically, the MLE component takes the perturbed and the original predictions as inputs, aiming to constrain the co-perturbation on input texts, thus ensuring the model's outputs remain unchanged. Conversely, the MEP advocates for the maximal introduction of noise across all words by maximizing conditional entropy, pushing the model's tolerance of co-perturbation on input texts to its limits. Through the collaborative effect of these two goals, the proposed algorithm could adaptively estimate the model's interest towards each individual word. Finally, regions capturing heightened model interest will undergo less perturbation, while areas with less attention experience more significant noise induction.

To verify the effectiveness of our method, we conduct extensive experiments on various PLMs and a wide range of datasets. The experimental results show that the proposed algorithm effectively estimates the model's interest towards input texts. Additionally, based on the assessed PLMs' interest, we further explore the potential for improving model classification performance and adjusting the generated texts of PLMs. To summarize, the contributions of this paper are listed as follows:

- We introduce a novel perturbation-based method to investigate the direct quantification of PLMs' interest towards the input texts. To the best of our knowledge, this is a pioneering work in this research topic.
- To achieve this goal, we present a co-perturbation strategy and propose an adaptive estimation algorithm, aiding in the understanding of PLMs' errors.
- Building upon the proposed adaptive estimation algorithm, we conduct extensive experiments across various PLMs and benchmarks. The experimental results verify its effectiveness.

2 Related Work

2.1 Input Attribution Methods

In the domain of explainable machine learning, input attribution methods could provide insights into the importance or contribution of each input unit to the overall output of a complex machine learning model (Ratul et al., 2021; Deng et al., 2023). Layer-wise Relevance Propagation (LRP) (Ali et al., 2022) is such one method, which propagates the relevance or contribution of the final prediction back through the layers, assigning a relevance score to each neuron or unit in the network. However, LRP assumes a direct linear relationship between input features (*e.g.,* words or phrases in NLP) and output decisions, posing challenges in its adaptation to NLP due to the complex and context-dependent nature of natural language (Belinkov and Glass, 2019).

Shapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) is another popular algorithm and builds upon the concept of Shapley Value from cooperative game theory. Unfortunately, it requires consideration of the probabilities associated with various combinations of words in a specific order, leading to the issue of combinatorial explosion (Ivanovs et al., 2021; Khakzar et al., 2019), especially when dealing with high-dimensional data and complex models. Another notable method is LIME (Ribeiro et al., 2016) (Local Interpretable Model-agnostic Explanations), which fits a local simple linear model around the prediction to elucidate the relationship between input features and the output. Although LIME has linear time complexity, it is limited in explaining the network structure of certain classes of models (Chen and Meng, 2020), rendering it ineffective in explaining predictions made by complex networks, particularly LLMs. Feng et al. (2024) propose the token distribution dynamic (TDD) that projects input tokens of hidden layers into the embedding space to estimate their significance. However, the input saliency of TDD is also calculated in a linear manner.

2.2 Exploration on Attention Mechanism

Some work also attempts to explore aspects related to the quantification of model interest. Specifically, visualization-based methods, exemplified by Park et al. (2019), aim to provide visual analytics tools to comprehend the inner mechanisms of the self-attention module. Yeh et al. (2023) provide insights into attention behavior across different layers and positions within transformer models, while Hoover et al. (2020) focus on the analysis of the intricate structures encoded by the models. Jaunet et al. (2022) contribute to a tool tailored for the visual examination of vision and language reasoning. For probing based methods, Li et al. (2021) delve into the layer-wise detection of linguistic anomalies in BERT. Sorodoc et al. (2020) introduce probing techniques examining referential information, while Mohebbi et al. (2021) explore BERT token representations in sentence probing. Function-based methods, including Grad-SAM (Barkan et al., 2021), interpret transformers through the lens of gradient self-attention maps. Hao et al. (2021) focus on interpreting information interactions within transformers, proposing self-attention attribution. Additionally, Geva et al. (2021) shed light on the role of feed-forward layers in transformers, framing them as key-value memories.

In summary, while significant advancements have been made, these studies often rely on various stringent assumptions, causing the lack of effective theoretical frameworks tailored specifically for language models to directly quantify their interest towards input texts.

3 Preliminaries

3.1 Notations

An NLP dataset, denoted as $\mathcal{D}$, comprises a collection of sentences, *i.e.,* $\mathcal{D} = \{s_i | 1 \leq i \leq |\mathcal{D}|\}$, where $|\mathcal{D}|$ represents the number of sentences in this dataset. For each sentence, it is composed of a sequence of words, denoted as $s_i = \{w_{ij} | 1 \leq j \leq |s_i|\}$ with $|s_i|$ indicating the length of the sentence. Additionally, a PLM (indicated as $\mathcal{M}$) can be treated as a complex non-linear function φ (Guan et al., 2019), and we further use φ_k to denote the function fitted by k-*th* layer in $\mathcal{M}$. In this context, the bold symbol $\boldsymbol{s}_i$ and $\boldsymbol{w}_{ij} \in \mathbb{R}^d$ are used to represent the d-dimensional representations of sentence s_i and the word w_{ij}, respectively, where $\mathbb{R}^d$ refers to the feature space.

3.2 The Degree of Model Interest towards Input Texts

For the model interest, it refers to the degree of attention a PLM allocates to each part of the input texts. The greater a PLM's interest towards certain words, the more importance these words hold for the model, thereby intensifying the impact of alterations when perturbing these words. Following the perturbation theory, the model's interest towards the input texts can be characterized by the extent to which the model outputs undergo changes after the introduction of noises to their inputs:

$$\rho(w_{ij}|s_i, \mathcal{M}) \propto ||\varphi(s_i) - \varphi(s_i|\delta(w_{ij}))||^2 \quad (1)$$

where $\rho(w_{ij}|s_i, \mathcal{M})$ (or ρ_{ij}) indicates the model's ($\mathcal{M}$) interest towards the word w_{ij} in sentence s_i. φ refers the non-linear function fitted by $\mathcal{M}$. $\varphi(s_i)$ is the prediction with regard to its input sentence s_i, and $\varphi(s_i|\delta(w_{ij}))$ indicates the prediction after perturbing the word w_{ij} in s_i. The Frobenius norm $||\varphi(s_i) - \varphi(s_i|\delta(w_{ij}))||^2$ measures the distance between original and perturbed predictions.

The definition in Eq. 1 precisely delineates the task objective undertaken in this paper by connecting the magnitude of output changes and the model interest towards input texts. This design behind the equation is straightforward, as depicted in Figure 2.

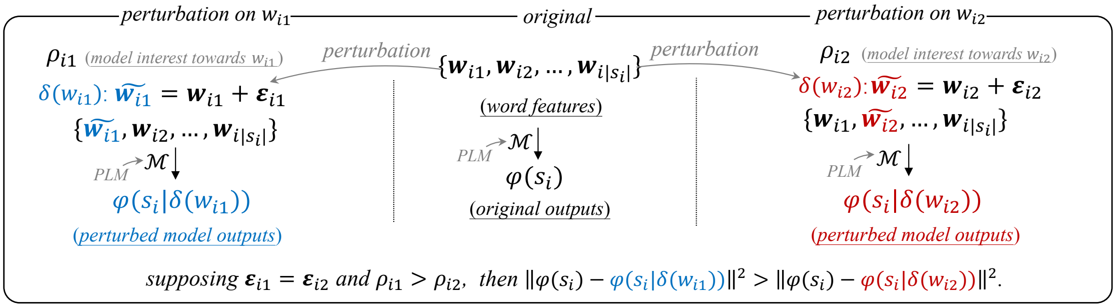


Figure 2: An illustration of the noise injection and model interest estimation. $\rho(w_{ij}|s_i, \mathcal{M})$ (abbreviated as ρ_{ij}) refers to the model interest towards the word w_{ij}. The perturbation of w_{ij}, denoted as $\delta(w_{ij})$, is implemented by introducing noise $\boldsymbol{\epsilon}_{ij}$ into its feature representation $\boldsymbol{w}_{ij}$ (refer to Eq. 3). When the model $\mathcal{M}$ exhibits different levels of interest towards these two words (w_{i1} and w_{i2}), injecting the identical noise ($\boldsymbol{\epsilon}_{i1} = \boldsymbol{\epsilon}_{i2}$) to these two words will result in different impacts on the model outputs ($||\varphi(s_i) - \varphi(s_i|\delta(w_{i1}))||^2 > ||\varphi(s_i) - \varphi(s_i|\delta(w_{i2}))||^2$).

4 Estimation of Model Interest

4.1 Co-Perturbation Strategy

Owing to the inherent contextual dependencies in natural language (Manning et al., 2008), perturbing words one by one may yield unreliable interest estimations. This is particularly prominent when the model's interest towards certain words is strongly interconnected (*e.g.*, the words "*New*" and "*York*" in the sentence shown in Figure 1). Although manual identification of potential combinations during perturbation is feasible, this will lead to the well-known challenge of combinatorial explosion (Khakzar et al., 2019; Ivanovs et al., 2021) and also make automated model interest estimation impracticable. In response to this issue, we propose a straightforward co-perturbation strategy that injects noise into all words of the texts simultaneously:

$$\begin{aligned}\delta(s_i) &= \{\delta(w_{i1}), \ldots, \delta(w_{ij}), \ldots, \delta(w_{i|s_i|})\} \\ &= \{\widetilde{\boldsymbol{w}}_{i1}, \ldots, \widetilde{\boldsymbol{w}}_{ij}, \ldots, \widetilde{\boldsymbol{w}}_{i|s_i|}\} \\ &= \{\boldsymbol{w}_{i1} + \boldsymbol{\epsilon}_{i1}, \ldots, \boldsymbol{w}_{ij} + \boldsymbol{\epsilon}_{ij}, \ldots, \boldsymbol{w}_{i|s_i|} + \boldsymbol{\epsilon}_{i|s_i|}\}\end{aligned} \tag{2}$$

where $\delta(s_i)$ denotes the injections of noise into all words in sentence s_i with $\delta(w_{ij})$ being the perturbation of the j^{th} word in s_i. $\widetilde{\boldsymbol{w}}_{ij}$ denotes the perturbed word feature with a certain noise vector $\boldsymbol{\epsilon}_{ij}$, defined as:

$$\delta(w_{ij}) : \widetilde{\boldsymbol{w}}_{ij} = \boldsymbol{w}_{ij} + \boldsymbol{\epsilon}_{ij},\ s.t.\ \boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij}) \tag{3}$$

where the noise $\boldsymbol{\epsilon}_{ij}$ follows a Gaussian distribution, *i.e.*, $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij})$. $\boldsymbol{\Sigma}_{ij}$ denotes the covariance matrix. The noise for each word representation in the sentence s_i is initialized under different covariance matrices and the same mean vector. With this strategy, the estimation process is not only simplified but also the accuracy of interest estimation is enhanced by considering the collective effect of perturbations across the entire input[1].

4.2 Adaptive Estimation

While the co-perturbation strategy is conceptually straightforward, its implementation requires sophisticated estimation. This necessity arises from the challenge of disentangling the overall changes in prediction into those from each individual word. Specifically, it involves determining how to separate the model interest in each word $||\varphi(s_i) - \varphi(s_i|\delta(w_{ij}))||^2$ from the final combined changes $||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2$.

To navigate this complexity effectively, we develop an adaptive estimation algorithm designed to complement the co-perturbation strategy. As shown in Eq. 2, each word has a corresponding noise, and these noises serve as the parameters to be estimated. The desired optimization goal can be articulated as follows: through algorithmic estimation, words of higher model interest should exhibit a lower tolerance to the estimated noise, whereas less important words should display a higher noise tolerance. Taking the sentence s_i as an example, the parameters we need to optimize are $\boldsymbol{\epsilon}_i = [\boldsymbol{\epsilon}_{i1}^\top, \ldots, \boldsymbol{\epsilon}_{ij}^\top \ldots, \boldsymbol{\epsilon}_{i|s_i|}^\top]^\top$, and the corresponding optimization objective can be designed as[2]:

$$\begin{aligned}\mathcal{J}(\boldsymbol{\epsilon}_i) &= \underbrace{\mathbb{E}[||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2]}_{MLE} \\ &\quad - \lambda \underbrace{H(\varphi(s_i|\delta(s_i))|\varphi(s_i))}_{MEP} \\ &= \mathbb{E}[||\boldsymbol{s}_i - \widetilde{\boldsymbol{s}}_i||^2] + \frac{d \cdot \lambda}{2} \sum_{j=1}^{|s_i|} \ln \rho_{ij}\end{aligned} \tag{4}$$

[1]Figure 6 in the appendix provides a visual illustration.
[2]Please refer to Section A for the detailed derivation.

where $\mathcal{J}$ is the loss to be optimized. ln represents the napierian logarithm and $\mathbb{E}$ is the mathematical expectation. H indicates the conditional entropy with p being the probability. The first term embodies the maximum likelihood estimation (MLE) of the distribution of $\widetilde{\boldsymbol{w}}_{i\cdot}$. This implies that this term is dedicated to learning a distribution that generates all potentially reasonable input noises corresponding to the predictions. The second term encourages a high conditional entropy, aligning with the maximum entropy principle (MEP). The principle states that in all possible probabilistic distributions, the one with the highest entropy is the best one. The λ balances the MLE loss and MEP loss.

It is noteworthy that the right part focuses on maximizing the conditional entropy, thereby striving to introduce as much noise as possible to each word. While the left part seeks to minimize the difference between perturbed results and original predictions. As a result, when the j^{th} word can endure substantial changes without affecting the predictions, the ρ_{ij} will be small. In contrast, for an important word, the interest degree will be large.

Algorithm 1 sketches the process of the proposed adaptive estimation method. It begins with the initialization of an noise matrix $\boldsymbol{\epsilon}_i$ ($\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{ij})$) for each sentence s_i in the dataset $\mathcal{D}$. Then, the adaptive optimization iterates until the algorithm converges. In each iteration, the random noise matrix is utilized to compute the perturbed word features based on Eq.3. Then, the loss is calculated according to Eq. 4, and compute the gradient to optimize $\boldsymbol{\epsilon}_i$. Finally, the estimated noise matrix is used to compute the interest vector $\boldsymbol{\rho}_i$.

Algorithm 1: Adaptive Estimation

Input: The dataset $\mathcal{D}$, the PLM $\mathcal{M}$ and λ.
Output: The set of model interest $\{\boldsymbol{\rho}_i\}$.

Fine-tune $\mathcal{M}$ on the training set of $\mathcal{D}$
for $s_i \in \mathcal{D}$ **do**
  Generate the random noise matrix $\boldsymbol{\epsilon}_i$
  while *Not Converge* **do**
    Perturb s_i according to Eq.2
    Compute loss according to Eq.4
    Estimate gradients and optimize $\boldsymbol{\epsilon}_i$
  Compute interest vector $\boldsymbol{\rho}_i$ based on $\boldsymbol{\epsilon}_i$

5 Experiments

5.1 Experimental Setup

PLMs and Datasets: To evaluate the proposed adaptive perturbation algorithm, we employ a diverse set of well-established PLMs, including BERT (Devlin et al., 2019) (in versions of 110M and 340M), GPT-2 (Radford et al., 2019) (124M, 355M, 774M and 1.5B) and OPT (Zhang et al., 2022) (125M, 350M and 1.3B). Different models with varying parameter sizes are considered, resulting in a total of nine models in this paper.

A diverse array of datasets is also leveraged, encompassing various NLP tasks, including Sentiment Analysis (SST2 (Socher et al., 2013)), Natural Language Inference (QNLI (Rajpurkar et al., 2016)) and Paraphrasing/Sentence Similarity (QQP (Iyer et al., 2017)). They collectively offer a thorough assessment of the proposed method[3].

Research Questions: To outline the experiments, we raise three primary research questions:

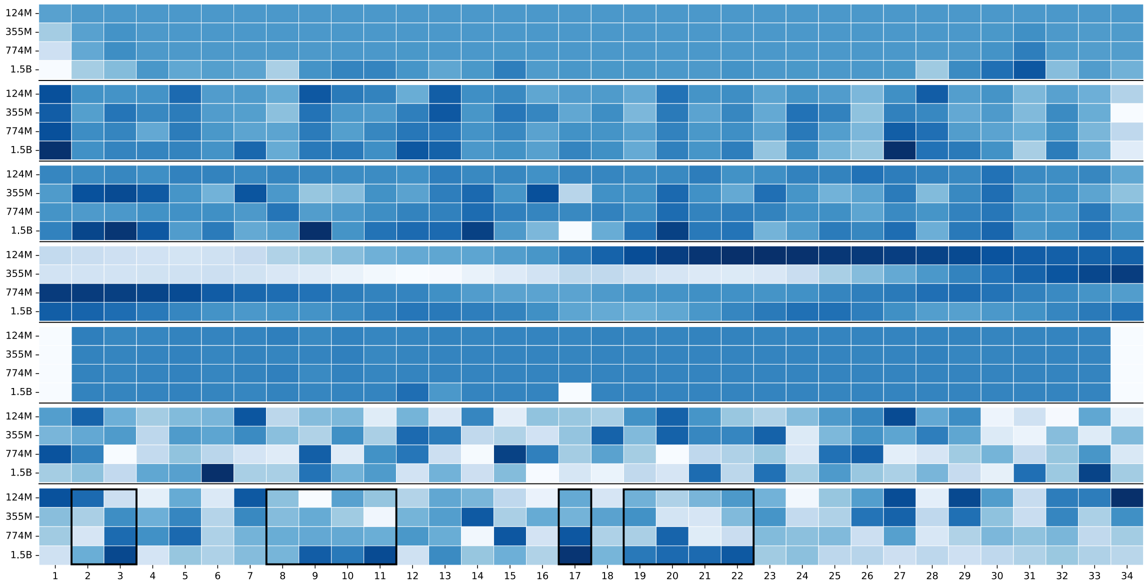
- RQ1: Does the proposed adaptive algorithm effectively assess model interest and how does the parameter λ affect the estimation results?
- RQ2: In addition to assessing the model-level interest, how do the intermediate layers put their focus on the input texts? What variations in interest are observed across different layers?
- RQ3: What are the potential practical applications of analyzing model interest, particularly for large language models?

5.2 RQ1: Analysis of Adaptive Estimation

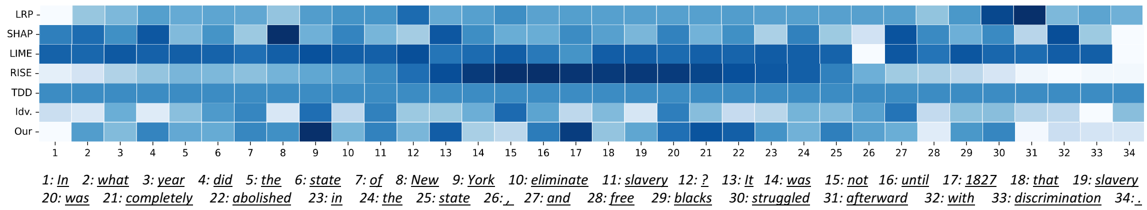
To evaluate the effectiveness of the proposed method, we compare it with six strong baselines, *i.e.,* five quite popular input attribution models[4] (including LIME (Ribeiro et al., 2016), SHAP-value (Lundberg and Lee, 2017), RISE (Petsiuk et al., 2018), LRP (Ali et al., 2022) and TDD (Feng et al., 2024)) and the word-by-word perturbation method. Notably, the interest estimation is performed at the token-level due to the underlying mechanism of model processing. For words composed of multiple tokens, we select the interest of the tail/head tokens as their interest.

Figure 3a presents the experimental results of all compared models evaluated on various sizes of GPT-2, where the parts marked by bounding boxes highlight the content expected to gain model interest. Generally, the proposed method consistently

[3] See Table 2 in appendix for data and fine-tuning details.
[4] Section B provides detailed information on these models.



(a) Comparison among different sizes of GPT-2 models, where the heatmaps from top to bottom are produced by LRP, SHAP, LIME, RISE, TDD, individual-perturbation and the adaptive model, respectively. For clarity and direct illustration, the parts expected to gain model interest are marked with bounding boxes.



(b) Model interest of all compared models on the OPT (1.3B), where "Idv." indicates the individual perturbation.

Figure 3: Comparison of the model interest estimated by all compared models.

provides more accurate assessments of model interest across different sizes of GPT-2 compared to all baseline models. Specifically, the model interest estimated by LRP and TDD tends to be similar values, indicating its inability to distinguish between important and non-important words, thus demonstrating its ineffectiveness for language models. While SHAP estimates varying levels of interest for different words, it focuses more on function words (*e.g.,* "*In*", "*It*", "*and*" *etc.*). LIME and individual perturbation could identify a few meaningful words, but they still miss many significant words (*e.g.,* "*1827*"). Consequently, these baseline models either erroneously prioritize more frequent words due to their failure to capture contextual dependencies or miss the decisive words. In contrast, our method accurately estimates model interest by refining its assessment based on the impact of all words on the model outputs. Additionally, the RISE produces completely different results, focusing more on the parts in the middle or on both sides of the input texts, and these focal points will also change with the model size.

In summary, all comparison models fail to achieve satisfactory results when applied to analyzing the PLMs' interest towards input texts, which verifies the effectiveness of the proposed method and further illustrates the necessity of an input analysis method tailored for PLMs. It is worth noting that as the model parameter size increases, the proposed adaptive evaluation method can effectively capture the relatively more important parts of the input text, allowing a better assessment, whereas the other methods do not exhibit this effect.

Figure 3b compares our method with baseline models on another different PLM, *i.e.,* OPT (1.3B). It can be seen that even on a different model, our method consistently excels in accurately assessing models' interest in input texts. However, these baselines still exhibits unsatisfactory results, ignoring the significant words in inputs (*e.g.,* "*year*" or "*1827*"). These findings further underscore the effectiveness and adaptability of our method in assessing model interest. It suggests that the proposed model can autonomously adjust its criteria based on the word features and structures of different models, thereby offering a more accurate reflection of the model's interest towards input texts.

Datasets	124M	355M	774M	1.5B
QNLI	0.0327	0.0522	0.0784	0.1242
SST-2	0.1250	0.2501	0.3125	0.3750
QQP	0.0220	0.0311	0.0410	0.0468

Table 1: Improvement of accuracy (#Correct/#Total) for various sizes of GPT-2 models on three datasets.

5.2.1 Effects of λ

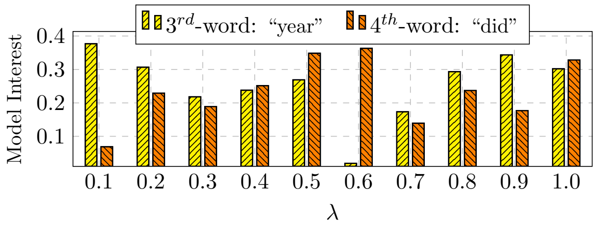
As shown in Eq. 4, the parameter λ is used to balance the *MEP* loss, with a larger value indicating a preference for smaller optimization in this part of the loss. To examine its impact on model interest evaluation, we selected two representative words, "*year*" and "*did*", from the example (see Figure 1). The expectation is that "*year*" should attract more model interest, whereas "*did*" should receive less.

Figure 4a demonstrates that variations in λ significantly influence the estimated interest in the selected words. Specifically, with a lower λ, the interest in "*did*" erroneously surpasses that in "*year*", which contrasts starkly with the anticipated results Conversely, as λ increases, particularly at $\lambda = 0.6$, the interest in "*year*" appropriately exceeds that in "*did*". However, further increases in λ lead to incorrect interest assessments again, indicating the critical influence of this parameter.

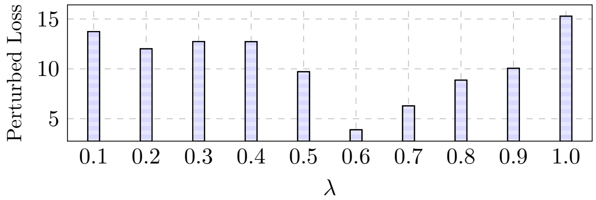
In addition, we also calculate the impact of the estimated word interest on the model output, measured by the MLE loss (see Eq. 4). Figure 4b illustrates the model prediction changes with varying λ. It can be seen that both excessively high and low λ values significantly impact model output. When this parameter is an optimal value ($\lambda = 0.6$), there will be minimal losses. This outcome suggests that appropriate λ tuning is also essential for accurate model interest evaluation and simultaneously maintaining minimal impact on model outputs. The possible reason is that larger λ values may result in suboptimal MEP loss optimization, introducing inappropriate noise and thus skewing interest evaluation and significantly affecting outputs. Conversely, smaller values might lead to an overemphasis on MEP optimization at the expense of other critical factors.

5.3 RQ2: Layer-wise Interest Toward Input Texts

By further adapting Eq. 4 to $\mathbb{E}[||\varphi_k(s_i) - \varphi_k(s_i|\delta(s_i))||^2] - \lambda H(\varphi_k(s_i|\delta(s_i)) \mid \varphi_k(s_i))$, it enables the estimation of layer-wise interest towards



(a) Model interest in "*year*" and "*did*" with varying λ.



(b) Loss of perturbed predictions under different λ.

Figure 4: Analysis of GPT-2's (1.5B) interest (estimated by the proposed algorithm) towards two representative words in the sentence (see Figure 3) and the loss in perturbed predictions of various λ.

the input texts, thereby allowing the investigation of the model's internal dynamics of ρ. Figure 5 showcases the layer-wise interest in GPT-2 (1.5B), focusing on the initial and final 4 layers for brevity, as the complete model comprises 48 layers.

Specifically, Figure 5 reveals that the model's first layer predominantly selects crucial combinations in the input text, such as "*in what year*" or "*eliminate slavery*". Concurrently, it also discards less relevant information, like "*free blacks*". However, the lower layers still maintain focus on additional potential words, such as "*in the state*", to ensure the maximal degree of information retention. As the information is processed through the model's layers, increasingly relevant data are emphasized by higher layers, and the key words receive heightened interest, exemplified by "*year*" and "*1872*". This pattern suggests that the model effectively processes and understands the input information. In summary, our proposed method offers a novel avenue for examining the interest patterns of internal layers in PLMs, thereby enriching our comprehension of their decision-making processes.

5.4 RQ3: Practical Applications

5.4.1 Boost Model Classification Performance

The adaptive estimation described in Section 4.2 allows us to calculate the model interest towards each word in the input texts, thereby revealing the model's comprehension of these texts. However, a crucial consideration arises: *if the model exhibits incorrect interest towards the input texts and pro-*

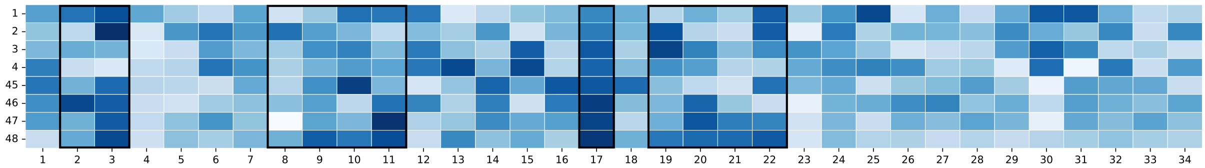


Figure 5: Layer-wise interest of GPT-2 (1.5B) towards the sentence (see Figure 3).

duces unexpected results, is it possible to rectify the predictions based on the estimated interest without modifying the model parameters? In other words, we hope to improve the model classification performance to some extent by leveraging the model interest. One potential solution involves suppressing the effects of the content that currently captures the model's attention, as these parts may not align with the desired focus, potentially interfering with its predictions. This operation could expose the content that the model should prioritize, increasing the probability that the model focuses on the crucial parts.

Building on this insight, we conduct experiments on the misclassified portions of PLMs within the datasets. Table 1 shows the results of various sizes of GPT-2 models across three datasets, where the intersection of "error" cases among these models for a specific dataset is taken as the benchmark to ensure a fair comparison[5]. Generally, the input modification strategy consistently brings performance improvement across all tested models and datasets, validating the effectiveness of this strategy and the precision of our model interest assessments. Notably, models with larger parameter counts exhibit more substantial performance gains. This may be attributed to such models being more susceptible to data biases, and their performance can be significantly boosted by excising these distracting elements.

5.4.2 Adjust Text Generation: A Case Study

To further verify the applicability of the proposed method, this section will briefly discuss its application in text generation[6] using the Llama3-8B-Instruct (Touvron et al., 2023). Taking the prompt "*The impact of climate change has become more evident in recent years.*" as an example, the original response generated by the model is "*The average global temperature has risen by about* $1\,^{\circ}\mathrm{C}$ *since the late 1800 ...*". After analyzing the model interest, we found that the model places relatively high attention on the temporal adverbial phrase (*i.e.*, "*in recent years*"), as evidenced by the generated response. However, if we want to elicit more content about the "*impact*" from the model and reduce the influence of the temporal adverbial phrase to some extent for producing texts that are more aligned with this focus without modifying the original prompt, the proposed interest estimation provides a feasible method. The estimated model interest shows the distribution of the model's understanding of the input text and offers insights into how to influence the final outputs. By suppressing the words "*in recent years*", the model produces "*Climate change is also exacerbating existing social and economic inequalities, disproportionately affecting vulnerable populations such as low-income communities, ...*". It can be observed that output text adjusted by model interest is more coherent in the desired context. This case study highlights the practical benefits of incorporating model interest into the text generation of LLMs, which is particularly valuable in applications requiring high-quality text generation, such as automated content creation, chatbots, and narrative generation (van Stegeren and Theune, 2019; Prabhumoye et al., 2020).

6 Conclusion

This paper probed a fundamental question regarding the interest of Pre-trained Language Models (PLMs) in the contents of input texts and introduced a novel perturbation-based method. This method was grounded in the design of translating model interest into quantifiable shifts in predictions after injecting controlled noise into input words. It encompassed a co-perturbation strategy and an adaptive estimation algorithm, aiming to address the challenges of combinatorial explosion and the intricacies involved in accurately assessing the model's interest towards each individual word. Extensive experiments across diverse PLMs and datasets confirmed the effectiveness of our method. Moreover, we explored the potential applications of enhancing model classification performance and adjusting the text generation of LLMs based on the identified model interest.

[5] See Table 3 for additional results of the nine models across three datasets, including the performance of the original models and those enhanced by ρ.

[6] See Appendix C for detailed information.

7 Limitations

Although this paper has devised an effective method to assess the PLMs' interest towards the input texts, it still lacks a method to correct the model's interest. This correction could enable PLMs to more accurately capture the relationship between inputs and outputs, thereby enhancing robustness to unexpected inputs. As a potential direction for future research, a feasible method could involve incorporating causal intervention theory (Pearl, 2009) into the proposed method.

In the future, we will focus on developing refined methods for rectifying the model's interest. Additionally, we further intend to explore the potential of influencing model generations by adjusting the degree of the model interest in the input prompts, while maintaining the integrity of the input distribution. For example, improving the decoding strategy for text generation, *i.e.,* incorporating interest patterns into the beam search algorithm. This would involve ranking beams not only by their likelihood but also by how well they align with the interest scores.

8 Ethical Considerations

The significance of the proposed method lies in explaining the behavior and output results of LLMs by quantifying their interest towards input texts. This exploration will help identify and mitigate potential risks associated with using LLMs, thereby supporting ethical considerations. Furthermore, all datasets used in this study are well-established and widely utilized. They have undergone meticulous manual inspection to remove any malicious or offensive content, ensuring the ethical integrity of the research.

Despite the contributions of this paper, there are still potential risks associated with LLMs, such as the generation of harmful or offensive content. To mitigate this issue, it is crucial to control the generation results. The method presented in this paper offers a feasible solution by aligning LLMs' outputs with their interest towards the input texts. This is an area we are actively exploring and will be introduced in our future work.

References

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. 2022. XAI for transformers: Better explanations through conservative propagation. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 435–451. PMLR.

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 *CIKM*, page 2882–2887. Association for Computing Machinery.

Yonatan Belinkov and James R. Glass. 2019. Analysis methods in neural language processing: A survey. In *NAACL-HLT, Volume 1 (Long and Short Papers)*, pages 3348–3354. Association for Computational Linguistics.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Kerui Chen and Xiaofeng Meng. 2020. Interpretation and understanding in machine learning. *Journal of Computer Research and Development*, 57(9):1971–1986.

Huiqi Deng, Na Zou, Mengnan Du, Weifu Chen, Guocan Feng, Ziwei Yang, Zheyang Li, and Quanshi Zhang. 2023. Understanding and unifying fourteen attribution methods with taylor interactions. *CoRR*, abs/2303.01506.

Jacob Devlin, Chang Ming-Wei, Lee Kenton, and Toutanova Kristina. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, Junlang Qian, and Kezhi Mao. 2024. Unveiling and manipulating prompt influence in large language models. In *ICLR*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *EMNLP*, pages 5484–5495. Association for Computational Linguistics.

Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a deep and unified understanding of deep neural models in nlp. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 2454–2463. PMLR.

Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. *AAAI*, 35(14):12963–12971.

Benjamin Hoover, Hendrik Strobelt, and Sebastian Gehrmann. 2020. exBERT: A Visual Analysis Tool to Explore Learned Representations in Transformer Models. In *ACL: System Demonstrations*, pages 187–196. Association for Computational Linguistics.

Maksims Ivanovs, Roberts Kadikis, and Kaspars Ozols. 2021. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognit. Lett.*, 150:228–234.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. 2017. First quora dataset release: Question pairs.

Theo Jaunet, Corentin Kervadec, Romain Vuillemot, Grigory Antipov, Moez Baccouche, and Christian Wolf. 2022. Visqa: X-raying vision and language reasoning in transformers. *IEEE Trans. Vis. Comput. Graph.*, 28(1):976–986.

Ashkan Khakzar, Soroosh Baselizadeh, Saurabh Khanduja, Seong Tae Kim, and Nassir Navab. 2019. Explaining neural networks via perturbing important learned features. *CoRR*, abs/1911.11081.

Bai Li, Zining Zhu, Guillaume Thomas, Yang Xu, and Frank Rudzicz. 2021. How is BERT surprised? layerwise detection of linguistic anomalies. In *ACL-IJCNLP (Volume 1: Long Papers)*, pages 4215–4228. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Clouâtre Louis, Parthasarathi Prasanna, Zouaq Amal, and Chandar Sarath. 2022. Local structure matters most: Perturbation study in NLU. In *Findings of the ACL*, pages 3712–3731. Association for Computational Linguistics.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *NIPS*, pages 4765–4774.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Hosein Mohebbi, Ali Modarressi, and Mohammad Taher Pilehvar. 2021. Exploring the role of BERT token representations to explain sentence probing results. In *EMNLP*, pages 792–806. Association for Computational Linguistics.

Jingcheng Niu, Wenjie Lu, and Gerald Penn. 2022. Does BERT rediscover a classical NLP pipeline? In *COLING*, pages 3143–3153. International Committee on Computational Linguistics.

Cheonbok Park, Jaegul Choo, Inyoup Na, Yongjang Jo, Sungbok Shin, Jaehyo Yoo, Bum Chul Kwon, Jian Zhao, Hyungjong Noh, and Yeonsoo Lee. 2019. San-vis: Visual analytics for understanding self-attention networks. In *IEEE Visualization Conference*, pages 146–150. IEEE.

Judea Pearl. 2009. *Causality: Models, Reasoning and Inference*, 2nd edition. Cambridge University Press, New York.

Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. RISE: randomized input sampling for explanation of black-box models. In *BMVC*, page 151. BMVA Press.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. In *EMNLP-IJCNLP*, pages 5739–5744. Association for Computational Linguistics.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *COLING*, pages 1–14. International Committee on Computational Linguistics.

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, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392. The Association for Computational Linguistics.

Qudrat E. Alahy Ratul, Edoardo Serra, and Alfredo Cuzzocrea. 2021. Evaluating attribution methods in machine learning interpretability. In *IEEE Big Data*, pages 5239–5245. IEEE.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *ACM SIGKDD*, pages 1135–1144. ACM.

Jaspreet Singh, Jonas Wallat, and Avishek Anand. 2020. Bertnesia: Investigating the capture and forgetting of knowledge in BERT. In *BlackboxNLP Workshop@EMNLP*, pages 174–183. Association for Computational Linguistics.

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 *EMNLP*, pages 1631–1642. ACL.

Ionut-Teodor Sorodoc, Kristina Gulordava, and Gemma Boleda. 2020. Probing for referential information in language models. In *ACL*, pages 4177–4189, Online. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Judith van Stegeren and Mariët Theune. 2019. Narrative Generation in the Wild: Methods from NaNoGenMo. In *Proceedings of the Second Workshop on Storytelling*, pages 65–74, Florence, Italy. Association for Computational Linguistics.

Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. 2023. Attentionviz: A global view of transformer attention. *Preprint*, arXiv:2305.03210.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. 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. *CoRR*, abs/2205.01068.

Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Trans. Intell. Syst. Technol.*

A Proof of Adaptive Estimation

A.1 Multivariate Gaussian Distribution

Supposing a random vector $\boldsymbol{x} \in \mathbb{R}^d$ is Gaussian-distributed $\boldsymbol{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, its probability density function (PDF) could be defined as[7]:

$$p(\boldsymbol{x}) = \frac{1}{\sqrt[2]{(2\pi)^d \det \boldsymbol{\Sigma}}} \exp(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})) \tag{5}$$

where d is the dimension of the vector; det indicates the determinant. exp refers to the natural exponential function. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ denote the d-dimensional mean vector and the $d \times d$ covariance matrix, respectively. For calculating the Shannon Entropy of a multivariate Gaussian distribution, it could be expressed as:

$$H(\mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{d}{2} \ln{(2\pi e)} + \frac{1}{2} \ln \det \boldsymbol{\Sigma} \tag{6}$$

A.2 Conditional Entropy

Following transformation in Guan et al. (2019), the conditional entropy $H(\delta(s_i)|\varphi(s_i))$ in Eq. 4 can be re-written as:

$$\begin{aligned} H(\delta(s_i)|\varphi(s_i)) &= H(\varphi(\delta(s_i))|\varphi(s_i)) \\ &= \sum_{j=1}^{|s_i|} H(\varphi(\delta(w_{ij}))|\varphi(s_i)) \\ &= \sum_{j=1}^{|s_i|} p(\varphi(s_i)) p(\varphi(\delta(w_{ij}))|\varphi(s_i)) \cdot \\ &\quad \ln p(\varphi(\delta(w_{ij}))|\varphi(s_i)) \\ &= \sum_{j=1}^{|s_i|} p(\varphi(s_i|\delta(w_{ij}))|\varphi(s_i)) \\ &\quad \ln p(\varphi(s_i|\delta(w_{ij}))|\varphi(s_i)) \end{aligned} \tag{7}$$

where the conditional distribution $p(\varphi(s_i|\delta(w_{ij}))|\varphi(s_i))$ represents the probability of perturbed word features given the original sentence representation and is equivalent to $p(\tilde{\boldsymbol{w}_{ij}}|\boldsymbol{s}_i)$ under the specified model and dataset. Additionally, this conditional distribution of perturbed word feature $p(\tilde{\boldsymbol{w}_{ij}}|\boldsymbol{s}_i)$ is characterized by the noise distribution $p(\boldsymbol{w}_{ij}|\boldsymbol{s}_i) = p(\boldsymbol{\epsilon}_{ij})$. For the noise distribution, it could be re-written as $\boldsymbol{\epsilon}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \sigma_{ij}\boldsymbol{I})$, where $\boldsymbol{I}$ denotes the identity matrix and σ_{ij}, representing the noise magnitude, could be further defined as $1/\rho_{ij}$. Hence, the conditional entropy in Eq. 7 could be reformulated:

[7] https://en.wikipedia.org/wiki/Multivariate_normal_distribution

$\mathcal{M}$	110-125M	340-355M	774M	1.3-1.5B
LR	$1e^{-5}$	$5e^{-6}$	$3e^{-6}$	$1e^{-6}$
$\mathcal{D}$	SST-2	QNLI	QQP	
#Epoch	5	8	10	
#Steps	1000	1500	5000	
#Train	67,349	104,743	363,846	
#Valid	872	5,463	40,430	

Table 2: Fine-tuning and data details. *LR* refers to the learning rate. Across all models, several parameters share uniform settings, including "*Learning Rate Schedule* = *Linear*", "*Optimizer* = *AdamW*", "*batch size* = 32", "*Seed* = 42" and "*Evaluation Strategy* = *Steps*".

$$\begin{aligned} H(\delta(s_i)|\varphi(s_i)) &= \sum_{j=1}^{|s_i|} H[\mathcal{N}_d(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij})] \\ &= \sum_{j=1}^{|s_i|} \{\frac{d}{2} \ln{(2\pi e)} + \frac{1}{2} \ln{(\det \boldsymbol{\Sigma}_{ij})}\} \\ &= \sum_{j=1}^{|s_i|} \{\frac{d}{2} \ln{(2\pi e)} + \frac{1}{2} \ln{(\frac{1}{\rho_{ij}})^d} \det \boldsymbol{I}\} \\ &= \sum_{j=1}^{|s_i|} \frac{d}{2} \ln{(2\pi e)} - \sum_{j=1}^{|s_i|} \frac{d}{2} \ln \rho_{ij} \end{aligned} \tag{8}$$

where $\frac{1}{\rho_{ij}}$ signifies the noise level, which can be further explained by the information transformation theory in the interpretable machine learning (*i.e.,* a large $\frac{1}{\rho_{ij}}$ indicates that a substantial portion of input information is disregarded.). This implies that the more a word captivates the model's interest, the less susceptible it is to noise, thereby ensuring the transmission of more pertinent information to subsequent layers. Consequently, based on the result of Eq. 8, Eq. 4 could be reformulated as:

$$\begin{aligned} \mathcal{J}(\boldsymbol{\epsilon}_i) &= \mathbb{E}[||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2] \\ &\quad - \lambda H(\varphi(s_i|\delta(s_i))|\varphi(s_i)) \\ &= \mathbb{E}[||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2] \\ &\quad - \lambda\{-\sum_{j=1}^{|s_i|} \frac{d}{2} \ln \rho_{ij} + \overbrace{\sum_{j=1}^{|s_i|} \frac{d}{2} \ln{(2\pi e)}}^{constant}\} \\ &= \mathbb{E}[||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2] + \frac{d \cdot \lambda}{2} \sum_{j=1}^{|s_i|} \ln \rho_{ij} \end{aligned} \tag{9}$$

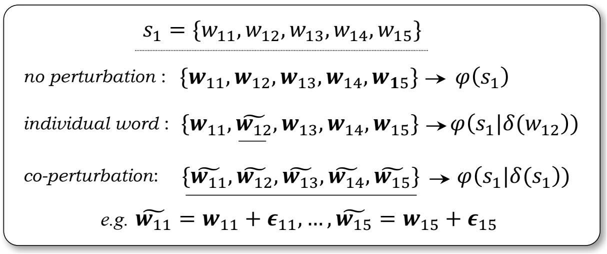


Figure 6: The co-perturbation strategy introduces noises into all words simultaneously. The model's interest is discerned via the adaptive algorithm.

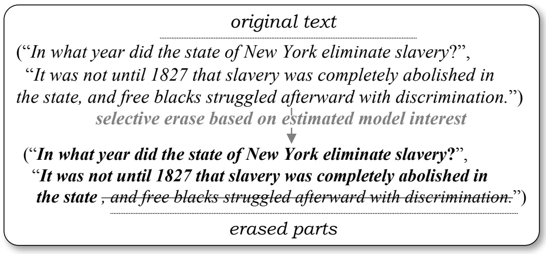


Figure 7: An illustrative example of the content erasure. During the process, the erased contents depend on the estimated model interest.

Datasets	**QNLI**		**SST-2**		**QQP**	
Settings	*no* ρ	*with* ρ (↑)	*no* ρ	*with* ρ (↑)	*no* ρ	*with* ρ (↑)
BERT-110M	0.8960	+1.086 (0.0201)	0.9151	+2.903 (0.0667)	0.8886	+1.173 (0.0258)
BERT-340M	0.9010	+1.474 (0.0320)	0.9241	+6.121 (0.0110)	0.8983	+1.364 (0.0301)
GPT2-124M	0.8832	+1.003 (0.0327)	0.9025	+1.003 (0.1250)	0.8911	+.9530 (0.0220)
GPT2-355M	0.9002	+1.525 (0.0522)	0.9381	+2.612 (0.2501)	0.8986	+1.364 (0.0311)
GPT2-774M	0.9108	+2.159 (0.0784)	0.9415	+3.397 (0.3125)	0.8937	+1.772 (0.0410)
GPT2-1.5B	0.9145	+3.493 (0.1242)	0.9541	+4.076 (0.3750)	0.9023	+2.037 (0.0468)
OPT-125M	0.8878	+1.396 (0.0496)	0.9059	+2.251 (0.1110)	0.8980	+1.106 (0.0246)
OPT-350M	0.9029	+1.233 (0.0435)	0.9243	+5.158 (0.2778)	0.9016	+1.347 (0.0301)
OPT-1.3B	0.9165	+3.097 (0.1056)	0.9564	+8.680 (0.4430)	0.9100	+1.936 (0.0427)

Table 3: Accuracy (#Correct/#Total) of several PLMs on three datasets, where *no* ρ and *with* ρ denote the results of before and after the interest rectification, respectively. The performance improvement in the first column of "*with* ρ" is the ratio of improvement over the entire validation set and should be scaled by e^{-3}. The improvement in parentheses refers to the results on the common "error" parts.

A.3 MLE Loss

In Eq. 4, the first term $\mathbb{E}[||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2]$ can be interpreted as the Maximum Likelihood Estimation (MLE) of the noise. To substantiate this interpretation, we may postulate that $\varphi(\delta(s_i))|\varphi(s_i) \sim \mathcal{N}(\varphi(s_i), \boldsymbol{\Sigma}_s = \sigma_s^2 \boldsymbol{I})$ follows a Gaussian distribution. As such, we can obtain:

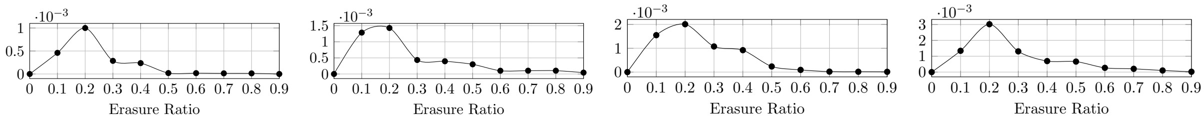
$$
\begin{aligned}
\text{argmax}_{\{\rho i1,\ldots,\rho i|s_i|\}} &= \ln \prod_j p(\varphi(\delta(w_{ij}))|\varphi(s_i)) \\
&\approx \text{argmax}_{\{\rho i1,\ldots,\rho i|s_i|\}} \ln p(\varphi(\delta(s_i))|\varphi(s_i)) \\
&= \text{argmax}_{\{\rho i1,\ldots,\rho i|s_i|\}} \{-\ln (\sqrt{(2\pi)^d|\boldsymbol{\Sigma}|} \\
&- \frac{1}{2}(\varphi(s_i|\delta(s_i)) - \varphi(s_i))^\top \boldsymbol{\Sigma}_s^{-1} \\
&\qquad (\varphi(s_i|\delta(s_i)) - \varphi(s_i))\} \\
&= \text{argmin}_{\{\rho i1,\ldots,\rho i|s_i|\}} \frac{||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2}{2\sigma_s^{2d}} \\
&= \text{argmin}_{\{\rho i1,\ldots,\rho i|s_i|\}} ||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2
\end{aligned} \tag{10}
$$

From Eq. 10, it can be drawn that the minimization of $||\varphi(s_i) - \varphi(s_i|\delta(s_i))||^2$ could be treated as the MLE of the model interest $\{\rho_{i1}, \ldots, \rho_{i|s_i|}\}$.

B Baselines

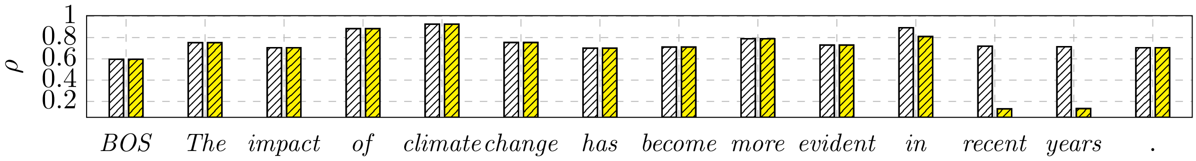
In this paper, we adopt four input attribution methods to verify the effectiveness of the proposed method:

- LRP (Layer-wise Relevance Propagation) is an input attribution method that helps interpret neural network predictions. It traces the contributions of each neuron back through the layers to the input features, assigning relevance scores that indicate the importance of each feature for the final prediction.
- SHAP (SHapley Additive exPlanations) is based on the Shapley values from cooperative game theory. It assigns an importance value to each feature by considering the contribution of each feature to the model's predictions across all possible combinations of features.



(a) GPT-2 124M (b) GPT-2 355M (c) GPT-2 774M (d) GPT-2 1.5B

Figure 8: Performance improvement across various erasure ratios according to the model interest.



(a) Model interest towards the prompt "*The impact of climate change is becoming more evident in recent years.*" based on Llama3-8B-Instruct, where the white bars denote the original results and the yellow ones indicate the adjusted results.

> *Rising temperatures, changing precipitation patterns, and increased frequency and severity of extreme weatre affecting ecosystems, human health, and the economy.* ***Climate change is also exacerbating existing social and econoties, disproportionately affecting vulnerable populations such as low-income communities, indigenous peoples, and smaveloping states.*** *To mitigate the effects of climate change, it is essential to reduce greenhouse gas emissions andto a low-carbon economy. This can be achieved through a combination of measures, including: 1. Transitioning to regy sources, such as solar and wind power, to reduce dependence on fossil fuels. 2. Improving energy efficiency in bunsportation, and industry. 3. Electrifying transportation and promoting the use of electric vehicles.4. ...*

> *Rising temperatures, changing precipitation patterns, and increased frequency and severity of extreme weather events are just a few examples of the many ways in which climate change is affecting our planet.* ***The average global temperature has risen by about 1℃ since the late 1800s, and is projected to continue to rise by another 2-5℃ by the end of this century if greenhouse gas emissions continue to increase.*** *This rise in temperature is causing a range of problems, including more frequent and severe heatwaves, droughts, and storms.*

(b) Generated texts of Llama3-8B-Instruct based on the original (top) and adjusted (bottom) model interest. The "*generate()*" is invoked with "*max_length=150*, *num_beam_groups=1*, *do_sample=False*, *num_beams=3*".

Figure 9: The estimated model interest and the generated texts of Llama3-8B-Instruct model.

- LIME (Local Interpretable Model-agnostic Explanations) creates a local surrogate model, typically a simple linear model, that approximates the behavior of the complex model around the instance being explained.

- RISE (Randomized Input Sampling for Explanation) is particularly designed for image classification and generates binary masks and applies them to the inputs, recording the model's predictions for each masked input. By aggregating these results, RISE creates an importance map that shows which parts of the input are most influential in the model's decision.

- TDD (Token Distribution Dynamics) projects input tokens into the embedding space and then estimates their significance based on distribution dynamics over the vocabulary.

C RQ3: Applications

In addition to the performance improvement shown in Table 1, we also investigate the impact of the proportion of erased contents on model performance improvement, with results presented in Figure 8. It can be seen that the model is sensitive to the proportion of erased text, and as the proportion of erasure increases, the model performance also improves in a reasonable range. When the proportion reaches a certain level (about 0.2 in the GPT-2 124M model), the performance improvement achieves its peak, but further erasure leads to diminishing returns and eventually a dramatic decline in performance (close to 0). One possible reason is that excessive erasure could compromise the semantic coherence of the original texts, inadvertently resulting in misleading outputs. A moderate erasure, conversely, strikes a more effective balance between maintaining semantic integrity and emphasizing crucial information.

For the application of adjusting text generation, we provide the details of the case in Section 5.4.2. Figure 9a illustrates the estimated model's interest towards the prompt based on the original outputs, where "BOS" denotes the special token (*e.g.*, "⟨|begin_of_text|⟩" for Llama3-8B-Instruct). After adjusting the interest, the model generates more coherent content aligned with the desired context (see Figure 9b).