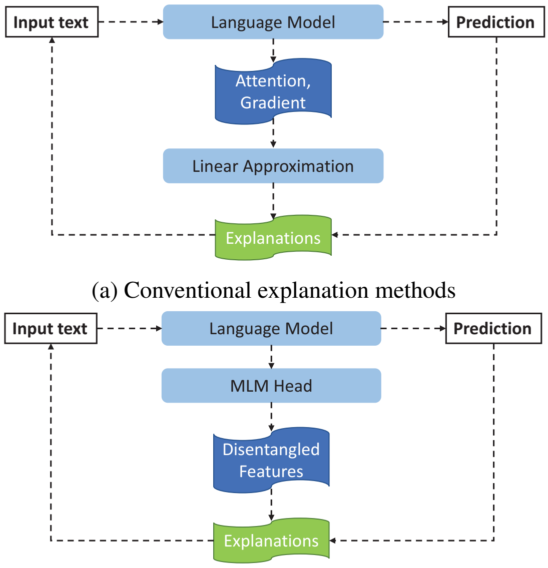
PromptExplainer: Explaining Language Models through Prompt-based Learning

Anonymous ACL submission

Abstract

Pretrained language models have become workhorses for various natural language processing (NLP) tasks, sparking a growing demand for enhanced interpretability and transparency. However, prevailing explanation methods, such as attention-based and gradient-based strategies, largely rely on linear approximations, potentially causing inaccuracies such as accentuating irrelevant input tokens. To mitigate the issue, we develop PromptExplainer, a novel method for explaining language models through prompt-based learning. PromptExplainer aligns the explanation process with the masked language modeling (MLM) task of pretrained language models and leverages the prompt-based learning framework for explanation generation. It disentangles token representations into the explainable embedding space using the MLM head and extracts discriminative features with a verbalizer to generate class-dependent explanations. Extensive experiments demonstrate that PromptExplainer significantly outperforms state-of-the-art explanation methods.



(a) Conventional explanation methods

(b) PromptExplainer

Figure 1: Demonstration of conventional explanation methods and our proposed PromptExplainer. Conventional methods generally apply the linear operation to attentions and/or gradients to generate explanations, while PromtExplainer utilizes MLM head to disentangle token representations to explain language models.

1 Introduction

Recently, pretrained language models (Devlin et al., 2019; Liu et al., 2019; OpenAI, 2022; Touvron et al., 2023) have achieved remarkable success across a wide range of NLP tasks, such as text classification, question answering and machine translation. However, the inherent complexity of these models, often characterized by billions of parameters (Narayanan et al., 2021) and high nonlinearities, makes these models notably opaque and their predictions elusive to users (Ali et al., 2022). Explaining language models is receiving significant attention due to the growing demand for facilitating accountability, transparency, trustworthiness, bias detection and ethical considerations (Bolukbasi et al., 2016; Gonen and Goldberg, 2019; Ali et al., 2022).

Explanation methods generally gain insights into the decision-making process of language models by assessing the significance of each of the input tokens in relation to specific class labels or tokens. Various explainability methods, such as attention-based (Bahdanau et al., 2015; Abnar and Zuidema, 2020) and gradient-based (Wallace et al., 2019; Atanasova et al., 2020; Chefer et al., 2021; Ali et al., 2022) approaches, have been developed. These methods generally employ linear approximation as shown in Figure 1a. For example, the attention-based method, attention rollout (Abnar and Zuidema, 2020), presumes that attention weights for input tokens are linearly combined or propagated across layers to simulate the behavior of transformers. Gradient-based methods (Wallace et al., 2019; Atanasova et al., 2020; Chefer et al., 2021; Ali et al., 2022), on the other hand, explain models by approximating the model's nonlinearity through local linear approximations near specific input tokens, leveraging Taylor's expansion theorem. Nevertheless, the error resulted from linear approximation may be non-negligible when the language model possesses a substantial scale and the task involves considerable complexity. The approximation error can be propagated and magnified across layers. As we will show in this paper, linear approximation may lead to accentuating irrelevant tokens. To avoid using linear approximation, we may have to seek solutions from a different perspective, instead of using the conventional gradient or attention-based methods.

Typically, language models undergo pretraining through the masked language modeling (MLM) task (Devlin et al., 2019; Liu et al., 2019; OpenAI, 2022; Touvron et al., 2023). In this process, the MLM head adeptly captures the complex dependencies among token representations to predict missing words. Aligning NLP tasks with the MLM task and utilizing powerful pretrained components, such as the MLM head, have demonstrated effectiveness in the paradigm of prompt-based learning (Ding et al., 2021; Schick and Schütze, 2021; Cui et al., 2022; Hu et al., 2022). Inspired by these studies, we propose to align the interpretation process with the MLM task to yield more accurate explanations in this paper.

To this end, we propose a novel explanation approach called PromptExplainer: Explaining Language Models through Prompt-based Learning, as illustrated in Figure 1b. This approach adopts prompt-based learning to synchronize the explanation process with the MLM task and capitalize on corresponding components to produce explanations. The PromptExplainer leverages the MLM head to disentangle the token representations into the explainable embedding space whose dimensionality equals the vocabulary size, with each dimension corresponding to a specific token. Additionally, it employs the verbalizer to extract discriminative features relevant to class labels to generate class-dependent explanations.

The proposed PromptExplainer offers several advantages. Firstly, it aligns the explaining process with the pertaining objectives of language models and eliminates the need for linearity assumptions. Secondly, it requires only a few lines of code for implementation and can be seamlessly integrated into existing prompt-based models without any additional parameters. To the best of our knowledge, we are the first to propose the utilization of prompt-based learning to interpret language models. Extensive experiments (in §4) demonstrate that PromptExplainer surpasses state-of-the-art (SOTA) explanation methods by a substantial margin.

2 Related Work

Existing approaches to explaining language models can be classified into attention-based, gradient-based, and perturbation-based methods. The generated explanations fall into either the class-dependent category (specific to each class label) or the class-agnostic (only based on the input and model) category.

In attention-based methods, utilizing vanilla attention weights in attention modules to interpret model decisions (Bahdanau et al., 2015) is a straightforward approach. However, this method's reliability and effectiveness diminish when applied to Transformer architectures (Wiegreffe and Pinter, 2019), commonly used in language models (Devlin et al., 2019; Liu et al., 2019; OpenAI, 2022; Touvron et al., 2023). To capture Transformers' intricate nonlinearities, attention rollout (Abnar and Zuidema, 2020) linearly combines attention weights across layers. Additionally, attention flow (Abnar and Zuidema, 2020) views attention propagation as a max-flow problem in the pairwise attention graph. Typically, attention-based explanations are considered to be class-agnostic.

Gradient-based methods employ backpropagation gradients to determine the significance of each token. The integrated gradient (Wallace et al., 2019) and input gradients (Atanasova et al., 2020) have been proven effective in various models and domains. Another approach, termed as generic attention explainability (GAE) (Chefer et al., 2021), integrates attention gradients along with gradients from other network components.

It is worth noting that layer-wise relevance propagation (LRP) (Bach et al., 2015) has also been used to measure the relative significance of each token (Voita et al., 2019; Chefer et al., 2021). Ali et al. (2022) discovers that LRP could encounter difficulties in identifying the input feature contributions in Transformers due to the intricate AttentionHeads and LayerNorm. To address the problem, they modify the current propagation rule to adhere to the conservation rule, which mandates that scores assigned to input variables and forming the explanation must sum up to the network's output. LRP-XAI is the SOTA in delivering the most effective class-dependent explanations.

A few perturbation-based methods have been proposed, which utilize the input reductions (Feng et al., 2018; Prabhakaran et al., 2019) to determine the most relevant parts of the input by observing changes in model confidence or Shapley values (Lundberg and Lee, 2017). Contrastive explanations (Lipton, 1990; Jacovi et al., 2021; Yin and Neubig, 2022), which focus on identifying the causal factors influencing a model's output choice between two alternatives, have emerged in the last two years. It is a different task so we do not compare the contrastive methods to our proposed approach.

3 Method

3.1 Overview

Task formulation Interpreting language models involves evaluating token saliency for class-dependent or class-agnostic explanations and highlighting each token's importance for a specific class label or the overall decision process. Our method belongs to the first type that generates class-dependent explanations. Formally, denote $X = (x_1, x_2, ..., x_n)$ as an input sequence of length n, and $C = (c_1, c_2, ..., c_p)$ as the class labels in the dataset. Our objective is to generate an explanation $E_i = (e_1, e_2, ..., e_n)$ that signifies the importance of each token in classifying X into class c_i.

Framework We directly integrate our proposed method within the prompt-based learning framework to explain language models under the classification task. As illustrated in Figure 2, prompt-based learning formulates the text classification task into a masked language modeling problem by enveloping the input sequence X with a template to form a cloze question. The language model (LM) encoder is then used to derive all tokens' representations $H \in \mathbb{R}^{n \times d}$, where d is the dimension. We then utilize the MLM head to project H as the distribution over the vocabulary in the embedding space. Finally, a verbalizer $\mathcal{V}$ is employed to associate certain tokens in the vocabulary with the label space, resulting in predictions and explanations for each class.

3.2 Motivation: MLM head and verbalizer as interpreter expert

In this section, we first demonstrate that the MLM head can project all input token representations as a distribution over the vocabulary in the embedding space. Subsequently, we elucidate why these distributions have the potential to replace traditional attentions or gradients as a new medium for explaining model decisions.

Conventional methodologies allow only the <mask> token to be processed by the MLM head to elucidate sophisticated contextual information and then make predictions. While adept at unraveling complex and agnostic representations, the practicality of utilizing this MLM head to decode unmasked token representations remains an unanswered query. To answer this question, we give a comprehensive analysis and empirical results in Appendix A, with key findings summarized below.

1. **The MLM head exhibits consistent decoding properties for both masked and unmasked token representations**.

2. The MLM head can project all input tokens—both <mask> and unmasked tokens—into **distributions over the vocabulary in the embedding space**, yielding interpretable results that align with model predictions. Specifically, within this space, each dimension corresponds to a unique token in the vocabulary, and the values therein represent the predictive probabilities of all possible tokens at a given position.

3. In the context of MLM, the projected distributions can be understood as representations based on the current token and its surrounding contextual information. These distributions reflect the predictive likelihood of all tokens within the vocabulary. **Consequently, these distributions can be interpreted as the token's contributions to the prediction process**.

In addition to the MLM head, the verbalizer is utilized as another indispensable component for generating language model interpretations. Various verbalizer types, including manual (Schick and Schütze, 2021), soft (Hambardzumyan et al., 2021), prototypical (Cui et al., 2022), and knowledgeable (KPT) (Hu et al., 2022) verbalizers, help pinpoint

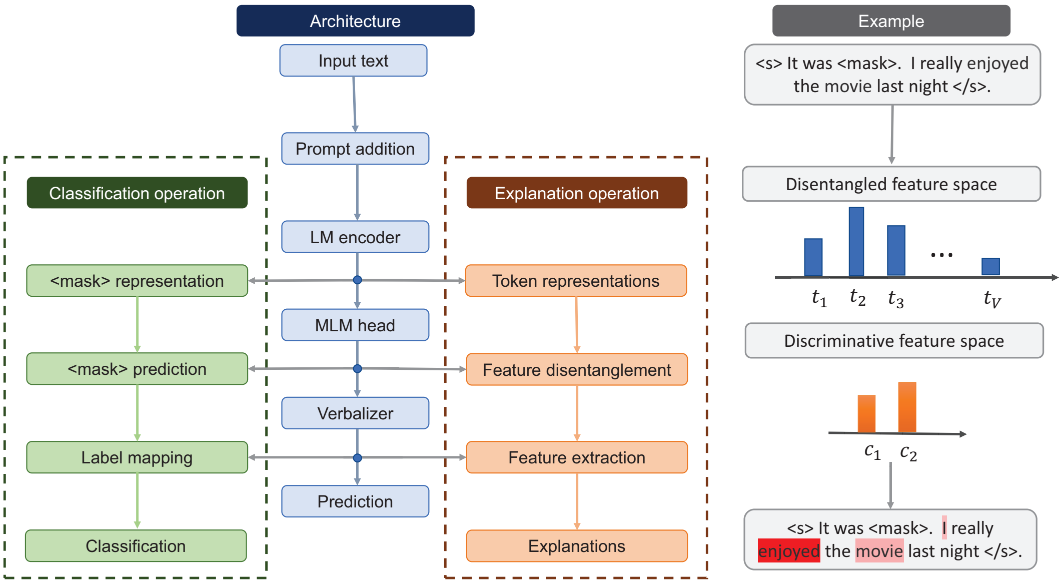


Figure 2: Overview of the classification operation, architecture, PromptExplainer (explanation operation), and an explanation example. The token representations obtained from the language model are disentangled into the explainable embedding space through the MLM head. Subsequently, the verbalizer is employed to extract discriminative features that exhibit a strong correlation with the classification results, enabling the generation of explanations. The given example demonstrates this process, where t_i and c_i denote the i-th disentangled feature and discriminative feature, respectively. A deeper red color indicates a higher explanatory weight.

effective label words to align model outputs with final predictions in prompt-based learning. Thus, the verbalizer is also integral in identifying discriminative vocabulary tokens that ultimately impact model decision-making, aiding in the generation of explanations.

In light of preceding observations and analysis, we articulate two phases of our PromptExplainer: first, utilizing the MLM head to disentangle token representations, and second, employing the verbalizer to extract discriminative features, thereby enabling explanation generation.

3.3 Feature disentanglement

From a feature engineering perspective, the MLM head is pre-trained to project token representations as the token distributions over the vocabulary that exhibits similar characteristics to disentangled features. Firstly, the projected features (i.e., distributions) can be viewed as individual factors, each of which represents a unique token within the vocabulary. Secondly, the features possess semantic interpretability, as each feature signifies the correlation with a predefined token in the vocabulary. Therefore, these projected features can be regarded as disentangled features in an explainable latent space. Formally, the MLM head $\mathcal{M}_h$ projects tokens representations H into the disentangled space by

$$H_V = \mathcal{M}_h(H) \in \mathbb{R}^{n \times V} \quad (1)$$

where V is the vocabulary size.

Two phenomena can be observed in the token distributions over the vocabulary H_V of the unmasked tokens. Firstly, the token with the highest probability is the token itself, which is equivalent to an exam with known answers. This observation also demonstrates that the disentangled features can retain their own information. Secondly, the predicted distribution is not a one-hot distribution; rather, it allows for the presence of certain possibilities for other tokens as well. These probabilities, based on the current token, represent the occurrence of other tokens and can thus be **viewed as contributions of the current token to the occurrence of other tokens**. Hence, the disentangled features function as correlations among tokens, influencing the classification outcomes and facilitating the generation of informative explanations.

3.4 Discriminative feature extraction

In prompt-based text classifiers, a verbalizer is commonly utilized to establish connections between classes and label words. Similarly, the verbalizer $\mathcal{V}$ is also applied to extract discriminative features in H_V. At this stage, the selected features in <mask>

form the model's final predictions, acting as discriminative features that guide its decision-making. Accordingly, we choose these features from all the tokens to generate explanations. Formally, the discriminative features H_D for all the tokens can be obtained by using the verbalizer $\mathcal{V}$:

$$H_D = \mathcal{V}(H_V) \in \mathbb{R}^{n \times p} \quad (2)$$

where p indicates the number of classes and only the features in V that potentially impact the classification are extracted. These extracted logits depict the correlation of each token with the class labels.

3.5 Explanation generation

To determine the contribution of each token to class labels, we begin by applying softmax normalization to derive the correlation between each token and the class labels:

$$H_S = Softmax(H_D) \quad (3)$$

Subsequently, the explanations for class c_i can be acquired by extracting the correlation of each token with the target class using Equation 4.

$$E_i = H_S[:, c_i] \quad (4)$$

3.6 Implementation

Recently, prompt-based learning has become prevalent in executing NLP tasks. Our PromptExplainer, adaptable to most prompt-based learning frameworks, leverages the original pretrained LM head as the MLM head. Given the variance of verbalizers across different prompt-based text classifiers, we directly employ the identical verbalizers from the classifiers to interpret their predictions. Consequently, our PromptExplainer can be seamlessly integrated into existing prompt-based frameworks with only a few lines of code implementing Equations 1 to 4. Detailed instructions and code are available in the supplementary materials.

4 Experiments

Following previous research (Schnake et al., 2022; Ali et al., 2022), we evaluate the PromptExplainer's effectiveness based on qualitative and quantitative explanation faithfulness experiments. Four text classification datasets, diverse templates and verbalizers are utilized in the experiments. We adopt RoBERTa-large (Liu et al., 2019) as our primary model, owing to its widespread use in prompt-based learning and superior performance in text classification (Ding et al., 2021; Schick and Schütze, 2021; Cui et al., 2022; Hu et al., 2022). We also provide experimental results on BERT (Devlin et al., 2019) in Appendix B to verify PromptExplainer's performance on various language models.

Dataset	# Class	Test Size	Template
AG's News	4	7600	A <mask> news: x
DBPedia	14	70000	[Topic : <mask>] x
Yahoo	10	60000	A <mask> question: x
IMDB	2	25000	It was <mask>. x

Table 1: The statistics and templates of each dataset. x indicates the input text.

4.1 Verbalizer

In our main experiments, which involve both quantitative and qualitative evaluations, we use current SOTA verbalizer KPT (Hu et al., 2022), which integrates label words from external resources. The model parameters precisely adhere to the recommendations in KPT. We report the results using the tuned language model in the 5-shot setting [1]. For detailed model parameters, please refer to (Hu et al., 2022).

4.2 Datasets and templates

We conduct experiments to assess various explanation methods on three topic classification datasets: AG's News (Zhang et al., 2015), DBPedia (Lehmann et al., 2015), Yahoo (Zhang et al., 2015); and one sentiment classification dataset: IMDB (Maas et al., 2011). We adopt commonly used templates in previous studies to perform prompt addition. Detailed information on the datasets and templates is shown in Table 1.

4.3 Baselines

We compare our proposed PromptExplainer with SOTA explanation methods, including both gradient-based and attention-based approaches.

We average the attention to <mask> across different heads in the last layer (**A-Last**) (Hollenstein and Beinborn, 2021) and also consider the attention **Rollout**(Abnar and Zuidema, 2020), which highlights the layerwise structure of deep Transformer models beyond raw attention head analysis.

[1] Prompt-based classifiers are extensively utilized in low-data regimes, such as few-shot settings. With a mere 5% difference in classification accuracy between 1-shot and 20-shot as illustrated in KPT, we only report explanation results for 5-shot trained models for each dataset. The results and patterns are similar for other shots, such as 10-shot and 20-shot. We run experiments using 24GB NVIDIA A5000.

We further evaluate **Gradient × Input (GI)**, as employed in (Denil et al., 2014; Shrikumar et al., 2017; Atanasova et al., 2020). Another competitive baseline, **Generic Attention Explainability (GAE)** (Chefer et al., 2021), integrates attention gradients with gradients from other network segments. **LRP-XAI** (Ali et al., 2022), designed to ensure that LRP-based methods adhere to the conservation axiom by altering propagation in layer normalization and attention heads, is the current SOTA.

4.4 Quantitative evaluation

Method	AG's News	DBPedia	Yahoo	IMDB
A-Last	71.5	78.0	42.0	84.9
Rollout	63.0	65.8	35.1	77.1
GI	69.3	70.7	37.6	78.1
GAE	72.6	79.9	43.7	86.0
LRP-XAI	71.2	78.6	43.3	87.6
PromptExplainer	**76.5**	**82.6**	**46.0**	**87.8**

Table 2: Activation probability (%). A higher probability is better and indicates that adding the most relevant nodes strongly activates the correct model prediction.

Method	AG's News	DBPedia	Yahoo	IMDB
A-Last	0.265	0.308	0.536	0.167
Rollout	0.415	0.468	0.684	0.192
GI	0.274	0.298	0.553	0.251
GAE	0.260	0.277	0.509	0.152
LRP-XAI	0.253	0.290	0.542	0.181
PromptExplainer	**0.231**	**0.242**	**0.500**	**0.143**

Table 3: Pruning MSE. A lower MSE is better and indicates that removing less relevant nodes has little effect on the model prediction.

Following previous research (Schnake et al., 2022; Ali et al., 2022), we validate various explanation techniques using an input perturbation strategy, prioritizing the most or least significant input tokens. Our evaluation of explanatory faithfulness encompasses two tasks, each correspondingly evaluated using specific metrics: activation probability and pruning mean squared error (MSE):

- Activation Task: All input tokens are initially removed. Tokens are then progressively added (10% interval), ordered from most to least relevant. The ground-truth class's output probability, namely the **activation probability**, is observed. A higher activation score means a more accurate explanation.

- Pruning Task: All the input tokens are retained initially. Tokens are then successively removed (10% interval) in order from least to most relevant. The **pruning mean squared error (MSE)** between the predictions of the unpruned model and the pruned outputs is calculated. A lower MSE value means a more faithful explanation.

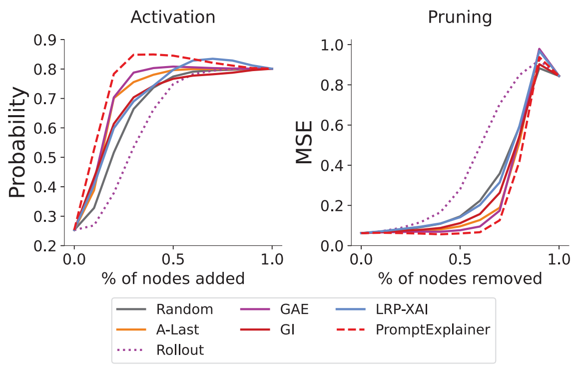


Figure 3: Evaluation of explanations using input perturbations on AG's News

Note, in the activation task, we begin with a sentence comprised solely of <unk> tokens. Conversely, in the pruning task, we progressively substitute tokens with <unk> tokens. These evaluation settings align with those used in prior studies (Schnake et al., 2022; Ali et al., 2022). To ensure a fair comparison, we employ the official codes of the baselines and subsequently generate explanations using the attentions and/or gradients from the same trained prompt-based model.

Table 2 and Table 3 present the average results on various datasets for the activation and pruning tasks, respectively. It can be observed that our proposed PromptExplainer substantially surpasses other baselines by a significant margin. The underperformance of Rollout and GI indicates the ineffectiveness of its presumed linear attention and weight propagation across the 24 layers in RoBERTa.

Figure 3 illustrates the activation and pruning curves for the AG's News dataset. From the activation curve, it is evident that the performance of PromptExplainer, LRP-XAI, and GAE starts to

Figure 4: Visualization of the attribution scores assigned to each word in a sentence from the Yahoo dataset with the label "artist". The intensity of the red color deepens as the explanatory weight increases, highlighting the significance of each word.

decline after a specific point. This is because most of the discriminative tokens are included at that point. As additional tokens are added, they may be misleading and introduce noise to the model, thereby inducing a performance drop. The inflection point's occurrence substantiates the explanation's faithfulness. Regarding the pruning curve, PromptExplainer consistently achieves the lowest MSE in most cases, further corroborating its effectiveness. The improvement brought by PromptExplainer can be attributed to the effective alignment with the MLM objective and utilization of the robust MLM head, which allows for a deeper understanding of the language model's behavior.

4.5 Qualitative evaluation

In this subsection, we will qualitatively examine the explanations generated by different methods. Figure 4 illustrates the extracted explanations using various methods. In the provided sentence, two keywords are directly linked to the class label "artist". The first keyword is the name of the singer, "Ivan Parker", whom the RoBERTa-large model recognizes as an artist. Several methods, including A-Last, Rollout, LRP-XAI, and PromptExplainer, are capable of identifying this information. Regarding the second keyword, "singer", which demonstrates the highest correlation with the "artist" label, only our proposed PromptExplainer is able to recognize it. It is also important to mention that most baseline methods often prioritize the inserted template, overlooking the practical meaning conveyed by the sentence. We provide additional examples in Appendix C to verify the PromptExplainer's superiority in capturing, identifying, and recognizing essential keywords for accurate classification and analysis purposes.

4.6 Effects of prompt templates and verbalizers

To verify the applicability of PromptExplainer to other prompt-based learning frameworks, we conduct supplementary experiments. The variations among different prompt-based models mainly lie in their templates and verbalizers. Therefore, we examine the performance of PromptExplainer across different templates and verbalizers to validate its generalization capability.

4.6.1 Different template results

Template ID	Template
1	A <mask> news: x
2	x This topic is about <MASK>.
3	[Category : <MASK>] x
4	[Topic : <MASK>] x

Table 4: Different templates for AG's News. x indicates the input text.

We carry out experiments on AG's News using various templates presented in Table 4 to assess the generated explanations by PromptExplainer. It is important to mention that all templates yield comparable classification accuracy, ensuring a fair comparison. The activation and pruning results are displayed in Table 5. Every template contains distinct words. Template 2 differs in its position compared to the other templates. Activation probability and MSE show slight variations among templates. These results demonstrate PromptExplainer's robustness, indicating its successful application to diverse prompt-based learning frameworks with varying templates.

Template ID	1	2	3	4
Activation probability	76.5	75.8	76.6	76.2
Pruning MSE	0.231	0.241	0.224	0.235

Table 5: Experimental results of different templates on AG's News.

4.6.2 Different verbalizer results

In our previous experiments, we mainly use the KPT verbalizer. This study evaluates PromptExplainer against other advanced verbalizers to gauge its effectiveness: (1) manual verbalizer (Ding et al., 2021) that relies on manually chosen label words for each class. The number of label words is set to 1, 10, and 30; (2) prototypical verbalizer (Cui et al., 2022), which constructs verbalizers automatically by learning class prototypes from training data.

Table 6 and Table 7 display the results obtained with different verbalizers. PromptExplainer demonstrates its effectiveness and wide applicability by achieving the best results in most cases. When employing a manual verbalizer with a single word per class, PromptExplainer ranks second. However, by augmenting the number of label words (e.g., 10 or 30 per class), PromptExplainer emerges as the top performer. The performance of PromptExplainer improves as the number of label words per class increases. This phenomenon can be attributed to the fact that disentangled features may contain not only token-label correlation but also other factors, such as position and syntactic information. By expanding the label words for each class, the diversity of word part-of-speech (POS) is enhanced, thereby reducing biases that arise from syntactic and positional factors.

Verbalizer	**Manual-1**	**Manual-10**	**Manual-30**	**Prototypical**	**KPT**
A-Last	68.9	73.4	61.7	66.9	71.5
Rollout	60.5	62.4	54.1	60.3	63.0
GI	65.3	70.0	58.7	64.4	69.3
GAE	69.4	74.5	62.5	67.1	72.6
LRP-XAI	**70.7**	73.5	62.3	69.1	71.2
PromptExplainer	69.6	**76.2**	**64.8**	**70.7**	**76.5**

Table 6: Activation probability (%) using various verbalizers.

Verbalizer	**Manual-1**	**Manual-10**	**Manual-30**	**Prototypical**	**KPT**
A-Last	0.447	0.289	0.361	0.482	0.265
Rollout	0.623	0.482	0.490	0.614	0.415
GI	0.468	0.340	0.384	0.510	0.274
GAE	**0.439**	0.298	0.348	0.476	0.260
LRP-XAI	0.445	0.314	0.368	0.478	0.253
PromptExplainer	0.442	**0.278**	**0.345**	**0.438**	**0.231**

Table 7: Pruning MSE using various verbalizers.

4.7 Other analysis

Significance of this study: While large language models (LLMs) have recently garnered significant attention, conventional LMs like BERT and RoBERTa remain indispensable for classification tasks. This is primarily due to two key reasons. Firstly, LLMs typically demand substantial computing resources or incur high API costs, resulting in slower response times compared to traditional LMs. Secondly, certain open-sourced LLMs still underperform RoBERTa in classification tasks. For instance, in a 1-shot text classification task on AG's News, BLOOM-176B (Scao et al., 2022), LLaMA-33B (Touvron et al., 2023), and LLaMA-65B (Touvron et al., 2023) achieved accuracies of 79.6%, 76.4%, and 76.8%, respectively (Ma et al., 2023), whereas RoBERTa, as reported in 2022 (Hu et al., 2022), achieved 83.7%. These figures underscore the significance of conventional language models, emphasizing the need to understand these models further and thus the importance of our proposed PromptExplainer.

Extension to LLMs: Our proposed PromptExplainer primarily leverages the concept of using MLM head to interpret token representations in the vocabulary space. However, it cannot be directly used to interpret autoregressive LLMs. This limitation arises from the fact that traditional LMs are based on masked language modeling, while autoregressive LLMs rely on next-word prediction. Consequently, the representations projected by the MLM head in RoBERTa reflect the probability of the current token based on bidirectional contextual information, whereas LLMs' LM head representations signify the probability of the next token based on all preceding tokens. This disparity hinders the direct application of PromptExplainer to LLMs. Nevertheless, the concept of using the LM head to interpret LLMs holds promise and is a potential avenue for future research, which we leave as future work.

5 Conclusion

In this paper, we present PromptExplainer, a method for explaining language models through prompt-based learning. Our approach aligns the interpreting process with the MLM objective and leverages the MLM head to disentangle token representations, creating an explainable feature space. We then utilize the verbalizer to extract discriminative features to generate explanations. Extensive experiments demonstrate the superior performance of PromptExplainer. In future work, we intend to extend the core concept of PromptExplainer, which involves leveraging the LM head to provide explanations for model decisions, to LLMs such as GTPX (OpenAI, 2022).

6 Limitations

There are several limitations in our work. Firstly, the disentangled features encompass not only the correlation with label words but also other information, such as positional and syntactic information, which could impact the token-label correlation, therefore affecting the explanation faithfulness, as discussed in §4.6.2. How to effectively distill the explanatory information from these disentangled features poses an important question. Additionally, as discussed in §4.7, when adapting the PromptExplainer concept for autoregressive LLMs, certain modifications are necessary due to differences in their pretraining objectives.

Ethics Statement

This work introduces PromptExplainer, a method designed to explain language models using prompt-based learning. It requires only a few lines of code for implementation and can be seamlessly integrated into existing prompt-based models. All experiments conducted in this study utilize publicly available datasets and codes. To facilitate future reproduction without unnecessary energy consumption, we will make our codes openly accessible.

References

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

Ameen Ali, Thomas Schnake, Oliver Eberle, Grégoire Montavon, Klaus-Robert Müller, and Lior Wolf. 2022. XAI for transformers: Better explanations through conservative propagation. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 435–451. PMLR.

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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

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

Ganqu Cui, Shengding Hu, Ning Ding, Longtao Huang, and Zhiyuan Liu. 2022. Prototypical verbalizer for prompt-based few-shot tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7014–7024, Dublin, Ireland. Association for Computational Linguistics.

Misha Denil, Alban Demiraj, and Nando De Freitas. 2014. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

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.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 609–614, Minneapolis, Minnesota. Association for Computational Linguistics.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. WARP: Word-level Adversarial ReProgramming. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference* *on Natural Language Processing (Volume 1: Long Papers)*, pages 4921–4933, Online. Association for Computational Linguistics.

Nora Hollenstein and Lisa Beinborn. 2021. Relative importance in sentence processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 141–150, Online. Association for Computational Linguistics.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2022. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2225–2240, Dublin, Ireland. Association for Computational Linguistics.

Alon Jacovi, Swabha Swayamdipta, Shauli Ravfogel, Yanai Elazar, Yejin Choi, and Yoav Goldberg. 2021. Contrastive explanations for model interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1597–1611, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Peter Lipton. 1990. Contrastive explanation. *Royal Institute of Philosophy Supplements*, 27:247–266.

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. *arXiv preprint arXiv:1907.11692*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. 2023. Fairness-guided few-shot prompting for large language models. In *Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.

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.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '21, New York, NY, USA. Association for Computing Machinery.

OpenAI. 2022. Chatgpt. `https://openai.com`. Version used: GPT-3.5.

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.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Thomas Schnake, Oliver Eberle, Jonas Lederer, Shinichi Nakajima, Kristof T. Schütt, Klaus-Robert Müller, and Grégoire Montavon. 2022. Higher-order explanations of graph neural networks via relevant walks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7581–7596.

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.

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

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.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

A Analysis: How Can MLM Head Deocde Token Representations?

In this section, we explore if the MLM head can decode unmasked token representations and analyze the characteristics of these decoded representations, providing the theoretical groundwork for our proposed PromptExplainer.

Homogeneity of <mask> token and unmasked tokens. All input tokens, including the <mask> token and unmasked tokens, are encoded within the same latent space and processed by identical attention blocks within the language model. Consequently, in the feature space, the encoded <mask> representation and all other unmasked tokens coexist within the same space, demonstrating homogeneity.

While residing in the same latent space, the meaningfulness of employing the MLM head to decode unmasked representations raises questions. To address this, we visualize results to gain insights into the decoding impact of the MLM head on unmasked token representations.

We first wrap the input sentence "I really enjoy this movie"with a template "It was <mask>", which is widely used in prompt-based learning. Subsequently, we feed this constructed sentence into RoBERTa-large to observe how its representations evolve across the various layers. Specifically, we input all token representations, including both the <mask> token and unmasked tokens, into the MLM head for projection into the embedding space. The resulting distribution over the vocabulary signifies the likelihood of filling in the respective positions. We then identify the token with the maximum probability at each position. These results are visually depicted in Figure 5a.

Firstly, it is noteworthy that all token representations can be effectively decoded into meaningful predictions by the MLM head. For instance, the representation of "movie"can be projected as "comic"and "film"in intermediate layers. Concerning the <mask> token, it is amenable to projection as "superb"and "fun"in the intermediate layers through the MLM head.

Secondly, the predictive probability for unmasked tokens in the final layer is consistently accurate, meaning that the tokens with the highest probability consistently correspond to the input tokens themselves. This discovery underscores the fact that each token's representation inherently contains self-information and can be successfully comprehended by the MLM head.

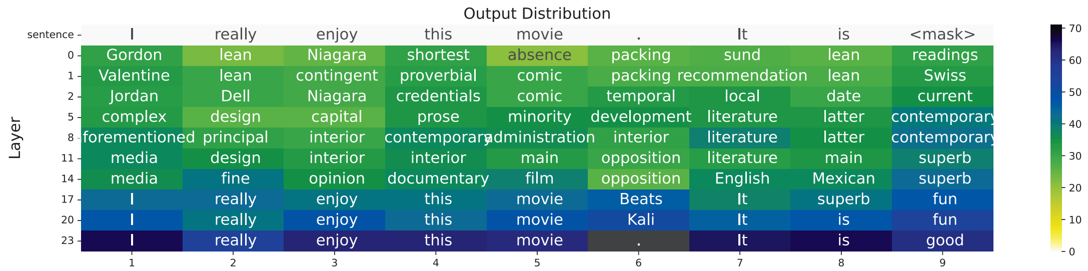
Thirdly, we proceed to visualize the ranking of the ultimately-predicted (target) token by the MLM head at each layer, as illustrated in Figure 5b. It becomes evident that the ranking of the target token progressively ascends through the layers as the MLM decoding process advances. This progression follows an approximately monotonic pattern.

Expanding on this, the projected distribution for each token shares the same dimensionality as the vocabulary size. Each dimension corresponds to a unique token in the vocabulary, with its value representing the probability of occurrence. This underscores the interpretability of the embedding space.

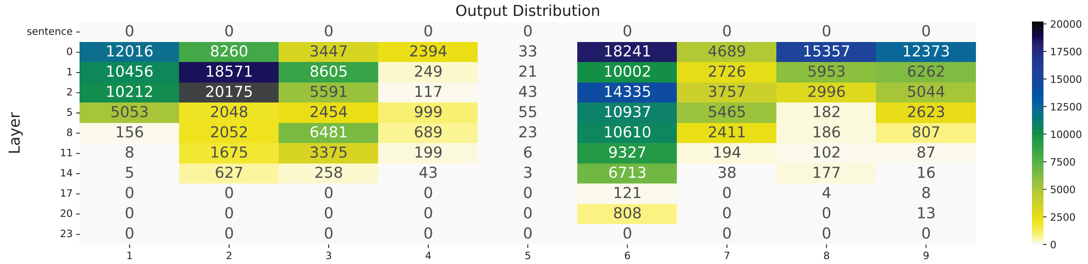
In line with the MLM objective, the distribution at a specific position can be primarily attributed to the inclusion of the input token at that position. Consequently, this distribution can be leveraged to assess the individual contribution of each input token to the overall predictive likelihood across the entire vocabulary.

Drawing from the preceding analysis, we can succinctly summarize our key findings as follows:

1. **The MLM head exhibits consistent decoding properties for both masked and un-**



(a) Visualization of MLM-decoded token with the maximum probability at each layer.



(b) Visualization of the ranking of the target token at each layer.

Figure 5: Visualization of using the MLM head to decode all input tokens at each layer.

masked token representations.

2. The MLM head can project all input tokens—both <mask> and unmasked tokens—into **distributions over the vocabulary in the embedding space**, yielding interpretable results that align with model predictions. Specifically, within this space, each dimension corresponds to a unique token in the vocabulary, and the values therein represent the predictive probabilities of all possible tokens at a given position.

3. In the context of MLM, the projected distributions can be understood as representations based on the current token and its surrounding contextual information. These distributions reflect the predictive likelihood of all tokens within the vocabulary. **Consequently, these distributions can be interpreted as the token's contributions to the prediction process**.

B Experiments on BERT-large

Table 8 and Table 9 present the results on various datasets for the activation and pruning tasks on BERT, respectively. It can be observed that our proposed PromptExplainer substantially outperfroms other baselines by a significant margin on BERT.

Method	**AG's News**	**DBPedia**	**Yahoo**	**IMDB**
A-Last	59.7	75.5	36.4	67.6
Rollout	50.0	66.2	28.2	64.1
GI	51.8	61.6	28.0	59.9
GAE	63.4	76.1	37.2	72.4
LRP-XAI	58.3	73.4	32.0	68.6
PromptExplainer	**65.1**	**79.2**	**38.6**	**74.4**

Table 8: Activation probability (%) on BERT. A higher probability is better and indicates that adding the most relevant nodes strongly activates the correct model prediction.

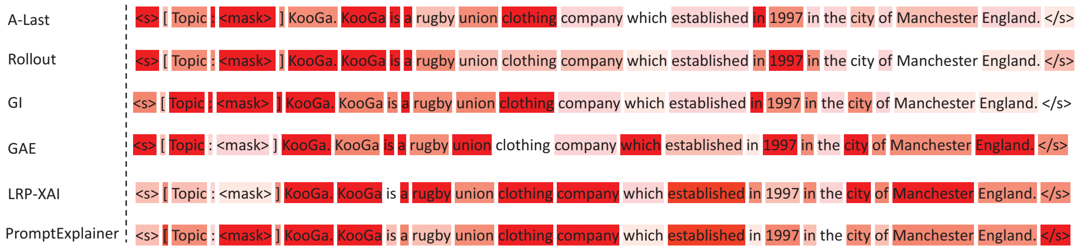
C Additional Qualitative Results

The keywords associated with the class "company" in Figure 6a are "Kooga", "clothing company ", and "established". Among the methods used, only LRP-XAI and PromptExplainer accurately identify all three keywords. Moving on to the second example presented in Figure 6b, the terms "Inc" and "company" are directly associated with its label "company". In this case, only GI and PromptExplainer successfully grasp these two keywords. Regarding the third example in Figure 6c, where the key phrase "photographer and author" plays a crucial role in classifying the sentence as "artist", PromptExplainer is the sole method that notices and comprehends the significance of the entire

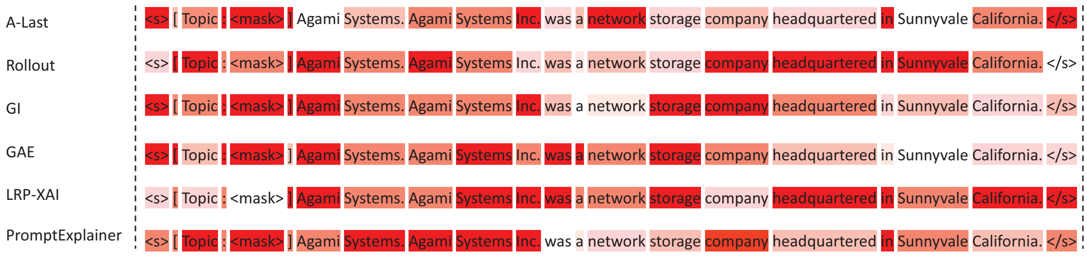
Method	AG's News	DBPedia	Yahoo	IMDB
A-Last	0.343	0.260	0.573	0.250
Rollout	0.512	0.502	0.684	0.247
GI	0.418	0.386	0.638	0.289
GAE	0.291	0.268	0.561	0.210
LRP-XAI	0.347	0.278	0.592	0.239
PromptExplainer	**0.274**	**0.247**	**0.534**	**0.186**

Table 9: Pruning MSE on BERT. A lower MSE is better and indicates that removing less relevant nodes has little effect on the model prediction.

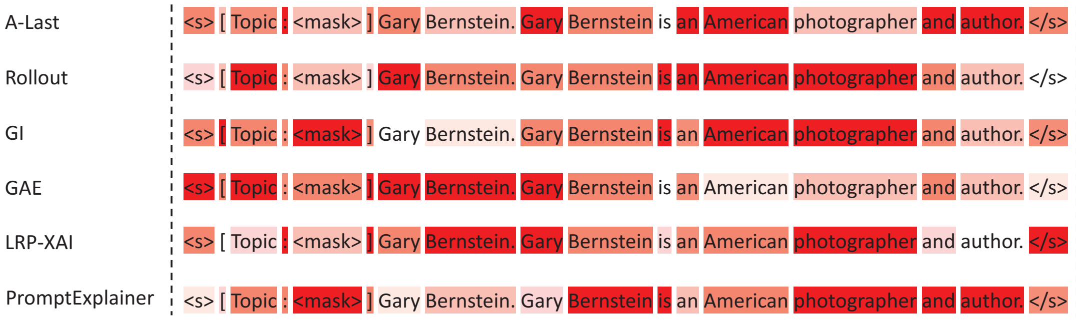
phrase. Lastly, considering the final example illustrated in Figure 6d, the keywords “member” and “Ohio House of Representatives” allow for the classification of this example as “politics”. Remarkably, only LRP-XAI and PromptExplainer exhibit the capability to recognize these two keywords. In summary, these four examples collectively serve as compelling evidence of the remarkable effectiveness of our proposed PromptExplainer.



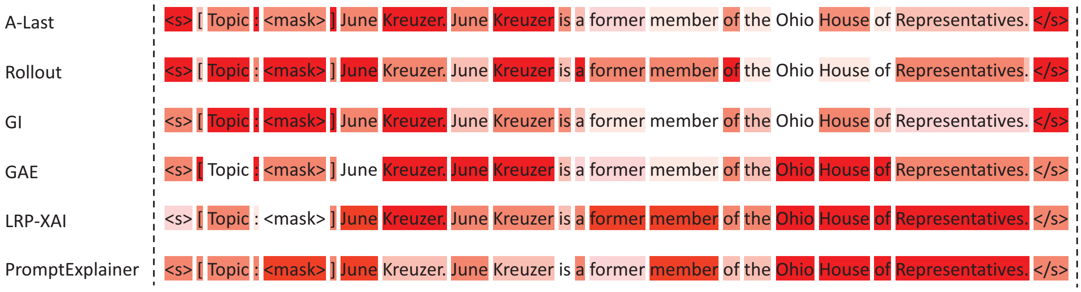
(a) Visualization of the attribution scores assigned to each word in a sentence tagged with "company".



(b) Visualization of the attribution scores assigned to each word in a sentence tagged with "company".



(c) Visualization of the attribution scores assigned to each word in a sentence tagged with "artist".



(d) Visualization of the attribution scores assigned to each word in a sentence tagged with "politics".

Figure 6: Examples for qualitative results.