Interpretation of NLP models through input marginalization

Siwon Kim **Jihun Yi** **Eunji Kim** **Sungroh Yoon***
Data Science and Artificial Intelligence Laboratory
ECE, Interdisciplinary Program in AI, and Institute of Engineering Research
Seoul National University, Seoul 08826, South Korea
{tuslkkk, t080205, kce407, sryoon}@snu.ac.kr

Abstract

To demystify the "black box" property of deep neural networks for natural language processing (NLP), several methods have been proposed to interpret their predictions by measuring the change in prediction probability after erasing each token of an input. Since existing methods replace each token with a predefined value (i.e., zero), the resulting sentence lies out of the training data distribution, yielding misleading interpretations. In this study, we raise the out-of-distribution problem induced by the existing interpretation methods and present a remedy; we propose to marginalize each token out. We interpret various NLP models trained for sentiment analysis and natural language inference using the proposed method.

1 Introduction

The advent of deep learning has greatly improved the performances of natural language processing (NLP) models. Consequently, the models are becoming more complex (Yang et al., 2019; Liu et al., 2019), rendering it difficult to understand the rationale behind their predictions. To use deep neural networks (DNNs) for making high-stakes decisions, the interpretability must be guaranteed to instill the trust in the public. Hence, various attempts have been undertaken to provide an interpretation along with a prediction (Gilpin et al., 2018).

Research in computer vision aims to interpret a target model by measuring attribution scores, i.e., how much each pixel in an input image contributes to the final prediction (Simonyan et al., 2013; Arras et al., 2017; Zeiler and Fergus, 2014; Lundberg and Lee, 2017). Since a pixel of an image corresponds to a token in a sentence, the attribution score of each token can provide an insight into the NLP model's internal reasoning process. A straightforward approach is to ask, "How would the model reaction change if each token was not there?" and

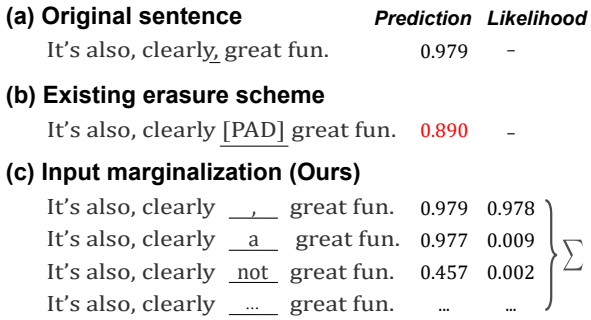

Figure 1: Given the original sentence (a), the existing erasure scheme (b) replaces each token with zero, i.e., [PAD] token. Our method (c) marginalizes each token out considering the likelihoods of candidate tokens.

the change can be measured by the difference in softmax probabilities after *erasing* each token. Li et al. (2016) proposed to erase each token by replacing it with a predefined value, i.e., zero. This became a representative method for interpreting NLP models, followed by several papers using the similar erasure scheme (Feng et al., 2018; Prabhakaran et al., 2019; Jin et al., 2019).

However, such an erasure scheme can cause out-of-distribution (OOD) problem, where the erased sentence deviates from the target model's training data distribution. DNNs tend to assign a lower prediction probability to OOD samples than in-distribution samples (Hendrycks and Gimpel, 2016), as shown in Fig. 1, which results in overestimated contribution of an unimportant token. The OOD problem induced by the existing erasure scheme makes it difficult to identify whether high-scoring tokens actually contribute significantly to the prediction. In computer vision, several studies have highlighted the problem and attempted to address it (Zintgraf et al., 2017; Chang et al., 2018; Yi et al., 2020). To the best of our knowledge, the OOD problem has not been raised in the field of NLP, hence no solution has been suggested yet.

In this study, we ask instead; "How would the model react differently if there were other tokens

*Correspondence to: Sungroh Yoon (sryoon@snu.ac.kr)

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 3154–3167,
November 16–20, 2020.

instead of each token?", as proposed by Chang et al. (2018); Yi et al. (2020). We propose to marginalize each token out to mitigate the OOD problem of the existing erasure scheme. During the marginalization, our method measures the contribution of all probable candidate tokens considering their likelihoods. To calculate the likelihoods, we use the masked language modeling (MLM) of bidirectional encoder representations from transformers (BERT) (Devlin et al., 2019).

Our contributions are as follows:

i) To the best of our knowledge, we first raise the OOD problem that can arise when interpreting NLP models through the existing erasure schemes.

ii) To avoid the OOD problem, we propose a new interpretation method, i.e., input marginalization using MLM for likelihood modeling.

iii) We apply the proposed method to interpret various NLP models and quantitatively verify the correctness of the resulting interpretation.

2 Related Works

2.1 Interpretation of NLP models

Model-aware interpretation methods for DNNs use model information such as gradients. Saliency map (Simonyan et al., 2013) interprets an image classifier by computing the gradient of a target class logit score with respect to each input pixel. Since a token index is not ordinal as an image pixel, the gradient with respect to a token is meaningless. Hence, Li et al. (2016) computed the gradient in an embedding space and Arras et al. (2017) distributed the class score to input embedding dimensions through layer-wise relevance propagation. Both methods sum up the scores of each embedding dimension to provide the attribution score of a token. Because the score can have a negative or positive sign, the sum may offset each other, so the contribution of the token may become zero even if it does contribute to the prediction.

Recently, the attention mechanism has been widely adopted to various NLP tasks (Bahdanau et al., 2014; Vaswani et al., 2017; Zhang et al., 2018) and there have been attempts to use the attention score as an interpretation. (Jain and Wallace, 2019). However, its validity is still controversial (Wiegreffe and Pinter, 2019).

Model-agnostic approaches aim to interpret any types of model with no information other than its feed-forward outputs. They observe how much the prediction changes after erasing each unit of input. If it differs significantly, then the unit obtains a high attribution score. In computer vision, the measurement of prediction difference varies from the subtraction of probabilities (Zeiler and Fergus, 2014) to a log-odds probability difference (Zintgraf et al., 2017). In the field of NLP, Li et al. (2016) interpreted NLP models by erasing each dimension of the embedding vector or the token itself, where the erasure was implemented by simply setting the value to a predefined value, i.e., zero. Such an erasure scheme can push the embedding vector or the input out of the training data distribution, thereby resulting in an inaccurate interpretation.

2.2 Interpretation without OOD problem

Several interpretation methods to mitigate the OOD problem have been proposed in computer vision. Zintgraf et al. (2017) proposed to marginalize each pixel out by assuming that the pixel value follows a Gaussian distribution. It had limitations in that the Gaussian distribution differed from the real pixel distribution. Chang et al. (2018) improved it by replacing an image segment with a plausible values generated from a deep generative model. Yi et al. (2020) proposed to adopt an additional DNN to model the pixel distribution, which motivated our work the most.

The method recently proposed by Jin et al. (2019) may appear similar to ours as it marginalizes context words out to obtain the context-free attribution of a token. However, it still cannot overcome the OOD problem because it replaces the token with zero, similar to the existing methods. To the best of our knowledge, no attempt has been undertaken to raise and overcome the OOD problem that arises when interpreting NLP models.

2.3 MLM of BERT

BERT (Devlin et al., 2019), one of the state-of-the-art natural language representations, is trained with two pre-training tasks: MLM and next sentence prediction. The MLM aims to infer the probability of a token to appear in the masked position of an input. As BERT is deeply bidirectional, it can consider the entire context of the sentence which enables the exact likelihood modeling. The likelihoods of the candidate tokens for marginalization are easily and accurately attainable using the MLM of BERT.

3 Methods

We propose input marginalization to mitigate the OOD issue. In the following subsections, we measure the attribution score using the weight of evidence and marginalize over all possible candidate tokens using the MLM of BERT. We extend the method to multi-token cases and introduce adaptively truncated marginalization for an efficient computation. Finally, we propose a new metric, AUC_{rep}, to evaluate the proposed method faithfully. The overall algorithm is provided in Algorithm 1.

3.1 Measurement of model output difference

To measure the changes in the model output, we adopt the widely used weight of evidence (WoE) (Robnik-Šikonja and Kononenko, 2008), which is a log odds difference of prediction probabilities. We define θ as the target model parameter, y_c as a target class to be explained, and $\boldsymbol{x}$ as an input sentence. We introduce $\boldsymbol{x}_{-i}$, i.e., $\boldsymbol{x}$ without i-th token x_i, to quantify the contribution of x_i to predicting y_c. WoE is formulated as follows:

$$\begin{aligned}\text{WoE}_{\theta,i}(y_c|\boldsymbol{x}) = &\log_2(\text{odds}_\theta(y_c|\boldsymbol{x})) \\ &- \log_2(\text{odds}_\theta(y_c|\boldsymbol{x}_{-i})),\end{aligned} \tag{1}$$

where $\text{odds}_\theta(y_c|\boldsymbol{x}) = p_\theta(y_c|\boldsymbol{x})/(1 - p_\theta(y_c|\boldsymbol{x}))$. $p_\theta(y_c|\boldsymbol{x}_{-i})$ captures the notion of the model response without the i-th token. The first term of Eq. 1 can be easily obtained as it is the original prediction probability, while the second term is computed by input marginalization.

3.2 Input marginalization

We rewrite the term $p(y_c|\boldsymbol{x}_{-i})$ of Eq. 1 using marginalization as follows:

$$\begin{aligned}p(y_c|\boldsymbol{x}_{-i}) &= \sum_{\tilde{x}_i \in \mathcal{V}} p(y_c, \tilde{x}_i|\boldsymbol{x}_{-i}) \\ &= \sum_{\tilde{x}_i \in \mathcal{V}} p(y_c|\tilde{x}_i, \boldsymbol{x}_{-i}) \cdot p(\tilde{x}_i|\boldsymbol{x}_{-i}).\end{aligned} \tag{2}$$

Here, $\tilde{x}_i$ is a candidate token that can appear instead of x_i, and $\mathcal{V}$ is a set of vocabulary. $p(y_c|\tilde{x}_i, \boldsymbol{x}_{-i})$ can be easily obtained by a single feed forward to the target model with the i-th token replaced with $\tilde{x}_i$. We compute $p(\tilde{x}_i|\boldsymbol{x}_{-i})$, the likelihood of $\tilde{x}_i$ appearing in the i-th position, by substituting the x_i with the "[MASK]" token and feed forwarding it to BERT. The process of computing the attribution score of a token is repeated for all tokens in the sentence.

Algorithm 1 Input marginalization

Input Target model θ, input $\boldsymbol{x}$, vocabulary $\mathcal{V}$, likelihood threshold σ, and target class y_c
Output Attribution score $\boldsymbol{a}$
for $i = 0$ **to** length($\boldsymbol{x}$) **do**
  $m \leftarrow 0$ ▷ Initialize attribution score
  $\boldsymbol{s} \leftarrow$ copy $\boldsymbol{x}$
  $\boldsymbol{s}_i \leftarrow$ "[MASK]" token
  for all $\tilde{s}_i$ in $\mathcal{V}$ **do**
    $p(\tilde{s}_i|\boldsymbol{s}_{-i}) \leftarrow \text{BERT}_{\text{MLM}}(\boldsymbol{s})$
    if $p(\tilde{s}_i|\boldsymbol{s}_{-i}) > \sigma$ **then**
      $\boldsymbol{s}_i \leftarrow \tilde{s}_i$
      $m \leftarrow m + p(\tilde{s}_i|\boldsymbol{s}_{-i}) \cdot p_\theta(y_c|\boldsymbol{s})$
    end if
  end for
  $\boldsymbol{a}_i = \text{logodds}_\theta(y_c|\boldsymbol{x}) - \text{logodds}_\theta(m)$
    ▷ Prediction difference measurement
end for

3.3 Multi-token marginalization

We can compute the attribution score for multiple tokens similarly. Let us assume that we wish to measure the joint contribution of two tokens x_i and x_j. Eq. 2 then becomes

$$p(y_c|\boldsymbol{x}_{-i,j}) = \sum_{\tilde{x}_j \in \mathcal{V}} \sum_{\tilde{x}_i \in \mathcal{V}} p(y_c, \tilde{x}_i, \tilde{x}_j|\boldsymbol{x}_{-i,j}). \tag{3}$$

Applying Bayes' theorem, $p(y_c, \tilde{x}_i, \tilde{x}_j|\boldsymbol{x}_{-i,j})$ becomes $p(y_c|\tilde{x}_i, \tilde{x}_j, \boldsymbol{x}_{-i,j}) \cdot p(\tilde{x}_i, \tilde{x}_j|\boldsymbol{x}_{-i,j})$. The latter term of the multiplication can be decomposed into the multiplication of $p(\tilde{x}_i|\boldsymbol{x}_{-i}, \tilde{x}_j)$ and $p(\tilde{x}_j|\boldsymbol{x}_{-i,j})$. Each term can be easily obtained by masking the corresponding position and feed-forwarding it to BERT even when x_i and x_j are distant. For more than two tokens, the attribution scores can be obtained in the similar way.

3.4 Adaptively truncated marginalization

The computational complexity for obtaining an attribution score of one token is $O(|\mathcal{V}|)$, where $|\mathcal{V}|$ is the size of a vocabulary set. For the tokenizer used in BERT, $|\mathcal{V}|$ is greater than 30,000, and the same number of marginalization is required, which is computationally burdensome. For the efficient computation, we propose adaptively truncated marginalization. If the magnitude of $p(\tilde{x}_i|\boldsymbol{x}_{-i})$ is insignificantly small, the contribution of $p(y_c|\tilde{x}_i, \boldsymbol{x}_{-i})$ to the summation in Eq. 2 becomes negligible. Therefore, we marginalize only over candidates whose likelihoods are greater than a likelihood threshold

Figure 2: Interpretation results of the proposed method. "+" and "-" in (a) denote the positive and negative classes of the depicted sentences. "pre" and "hypo" in (b) denote premise and hypothesis of SNLI, respectively. Red and blue colors denote positive and negative contributions to the denoted classes, respectively.

(σ) and normalize the score. Adaptively truncated marginalization approximates Eq. 2 as follows:

$$p(y_c|\boldsymbol{x}_{-i}) \approx \frac{\sum_{\tilde{x}_i \in \tilde{\mathcal{V}}} p(y_c|\tilde{x}_i, \boldsymbol{x}_{-i}) \cdot p(\tilde{x}_i|\boldsymbol{x}_{-i})}{\sum_{\tilde{x}_i \in \tilde{\mathcal{V}}} p(\tilde{x}_i|\boldsymbol{x}_{-i})}, \tag{4}$$

where $\tilde{\mathcal{V}} = \{\, \tilde{x}_i \in \mathcal{V} \mid p(\tilde{x}_i|\boldsymbol{x}_{-i}) > \sigma \,\}$.

Since the likelihood distributions depend on a token's position in the sentence, the number of marginalization varies for every i. We will demonstrate the efficiency of adaptively truncated marginalization and find an optimal σ in Section 4.

3.5 Evaluation of interpretation

Inspired by Petsiuk et al. (2018) and Chang et al. (2018), we propose a metric AUC_{rep} to evaluate interpretation methods for NLP models. Given the attribution scores of a sentence, Petsiuk et al. (2018) plotted a prediction probability curve as pixels filled with zero in the order of importance. If the interpretation is faithful, then the curve will drop rapidly, resulting in a small area under a curve (AUC). However, replacing the token with zero or removing it from a sentence can cause the OOD problem again. Instead, we replace it with a token sampled from the distribution inferred by BERT MLM, as Chang et al. (2018) gradually replaced image segments with a generated sample. As MLM is trained by masking only a part of the input sentence, replacing too many tokens can degrade its modeling performance. Therefore, we calculate the AUC until 20% of the tokens are replaced, and refer to it as AUC_{rep}.

4 Experimental Results

4.1 Experimental setup

To show the model-agnostic and task-agnostic property of our method, we present interpretations of several types of DNNs trained for two tasks: sentiment analysis and natural language inference.

SST-2 For sentiment analysis, we used the Stanford Sentiment Treebank binary classification corpus (SST-2) (Socher et al., 2013), which is a set of movie reviews labeled as positive or negative. We trained an 1-dimensional convolutional neural net-

works (CNNs) and a bidirectional long short-term memory (LSTM) with attention mechanism, and fine-tuned BERT (Devlin et al., 2019).

SNLI For natural language inference, we used the Stanford natural language inference (SNLI) corpus (Young et al., 2014), a collection of pairs of two sentences, premise and hypothesis, annotated with three relationships between them: entailment, contradiction, and neutral. We trained the bidirectional LSTM for SNLI.

The final test accuracy of the target models is provided in Table 1. Note that the architectures of LSTM used for SST-2 and SNLI are distinct. Please refer to the Appendix for the detailed descriptions. Throughout the experiments, we used the same tokenizer as BERT, where the zero-th token is "[PAD]". After training the target models, we interpreted their predictions through the proposed input marginalization. We used pre-trained BERT (Wolf et al., 2019) for likelihood modeling and σ was set to 10^{-5}.

Table 1: Test accuracy of the target models

Corpus	Model		
	LSTM	BERT	CNN
SST-2	0.7753	0.8578	0.7300
SNLI	0.6314	-	-

4.2 Interpretation results

The interpretation results of the proposed method are shown in Fig. 2. The color indicates the contribution of each token to the final prediction, with blue and red representing a negative and positive contribution, respectively. Its intensity represents the magnitude of the attribution score. More examples are provided in Appendix.

Fig. 2 (a) shows the interpretations of correct predictions for SST-2. The labels are shown in front of the sentences. For predicting the positive class, affirmative tokens such as "brilliant" and "funny" were attributed highly (a1, a9); if they are replaced with other tokens, the prediction probability will decrease significantly. Likewise, negative tokens such as "disappointing" and "suffers" were highlighted for predicting the negative class (a7, a8). If positive and negative tokens appear in one sentence simultaneously, our method successfully assigned the opposite scores to those tokens: positive score to "disappointing" and negative score to "dead brilliant" for predicting negative class (a7).

The interpretations of the LSTM for SNLI are shown in Fig. 2 (b). The sentences were correctly classified to the denoted class. For predicting a class entailment, the token with a similar meaning were assigned high attribution score, such as "swim" (b3). In contrast, tokens that makes two sentences contradicting were highlighted for predicting contradiction, such as "cafe" vs. "bar" (b4).

4.3 Comparison to the existing erasing scheme

In this section, we compare our method with the existing method proposed by Li et al. (2015) through interpretations of models for SST-2. We refer to the existing method as zero erasure throughout the experiments as it replaces tokens with zero.

Qualitative comparison Interpretation results using input marginalization (Marg) and zero erasure (Zero) are depicted in Fig. 3 (a). Fig. 3 (a1-a6) and (a7-a8) were classified to positive and negative class, respectively. As shown in the figure, zero erasure often completely failed to interpret the prediction (a1). Zero erasure also assigned high attribution scores to uninformative tokens such as punctuation and "to" (a3-a6). Our method showed clearer interpretations where unimportant tokens were given low attribution scores, while correctly highlighting the important ones. Moreover, the negative attribution was captured better than zero erasure. For example, in Fig. 3 (a7-a8), the token "bit" reduces the degree of negativity of "disappointing". Compared to potentially more assertive tokens (e.g. "very"), the specific token diluted the negative sentiment of the sentence. The negative contribution of "bit" to predicting the class "negative" was captured only with our method.

Quantitative comparison using $\mathrm{AUC}_{\mathrm{rep}}$ We quantitatively compared our method with zero erasure using the $\mathrm{AUC}_{\mathrm{rep}}$ proposed in Section 3. Another baseline using "[UNK]" token instead of zero was tested to verify that the OOD problem occurs no matter what predefined value is used. We would like to clarify that we did not consider the "[MASK]" token because it is a special token dedicated for the pre-training of BERT. It will obviously cause the OOD problem because it never appears during the training of target classifiers.

The deletion curves in Fig. 3 (b) shows the change in prediction probabilities as tokens with high attribution score are gradually replaced. The

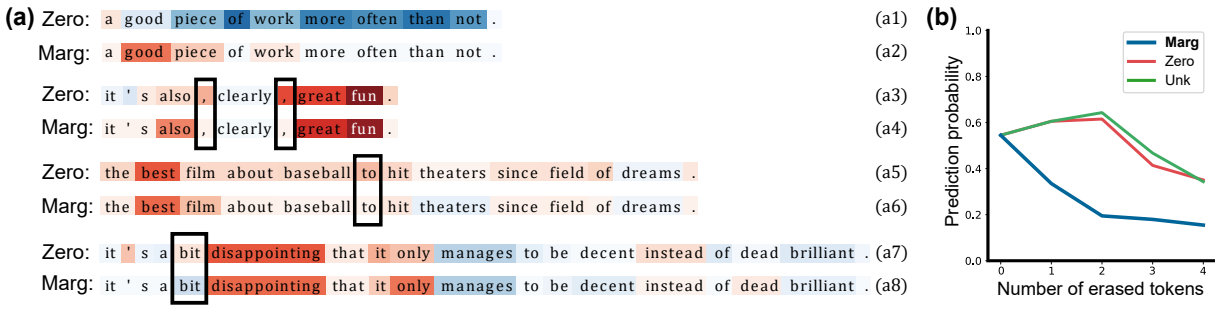

Figure 3: (a) shows examples of interpretations obtained by zero erasure and input marginalization (ours). Red and blue colors denote positive and negative contributions to the predicted classes, respectively. (a1-a6) are correctly classified as positive, and (a7-a8) are correctly classified to negative. (b) shows deletion curves of input marginalization, zero erasure, and "[UNK]" erasure, which are abbreviated as "Marg", "Zero", and "Unk", respectively.

curves show that the deletion curve drawn using our method dropped more rapidly compared to the zero and "[UNK]" erasures. The average AUC_{rep} values for 700 SST-2 sentences are provided in Table 2, and the proposed method showed the lowest AUC_{rep}. This result demonstrates that our method more accurately captures the importance of tokens than the existing erasure scheme.

Table 2: Comparison of AUC_{rep} with the existing erasure scheme (the lower the better).

	Interpretation method		
	Zero	Unk	Ours
AUC_{rep}	0.5284	0.5170	**0.4972**

Quantitative comparison using SST-2 tags The SST-2 corpus provides not only sentence-level labels, but also five-class word-level sentiment tags ranging from very negative to very positive. We can verify the validity of the attribution scores by comparing them with the word-level tags. For simplicity, we merged very positive and positive, very negative and negative into positive (pos) and negative (neg), respectively, such that each token is given one tag among three. If a sentence is correctly classified to positive, then three cases exist:

i) pos-tagged word: contributes positively and significantly to the prediction
ii) neut-tagged word: does not contribute much to the prediction
iii) neg-tagged word: contributes negatively to the prediction,

where neut denotes neutral.

To assess if our method can assign high score to case i), we measured the intersection of tokens (IoT) between pos-tagged tokens and highly attributed tokens in one sentence, motivated by intersection of union (IoU) which is a widely used interpretation evaluation metric in the vision field (Chang et al., 2018). IoT is defined as $|P \cap T| \,/\, |P|$, where P denotes a set of + tagged tokens, and T denotes a set of top-10 highly attributed tokens. The average IoT for 100 sentences was 0.72 and 0.64 for our method and zero erasure, respectively. This demonstrates that the tokens assigned with the highest attribution scores by our methods are likely to have a significant impact on the sentiment annotation.

A faithful interpretation method is expected to assign a small attribution score for the tokens belonging to ii). For 500 interpretations, the average attribution score of the neutral words was 0.053 and 0.175 with our method and zero erasure, respectively. With our method, the candidate tokens inducing the OOD problem like zero have an insignificant effect on the final attribution score because they are assigned relatively low likelihoods.

4.4 Additional analysis using input marginalization

The experimental results above demonstrates that our method can provide faithful interpretations. It thus can be used to analyze DNNs. First, we can compare the rationale of various models by analyzing their interpretations. Fig. 4 (a1-a4) show the interpretations of SST-2 samples correctly classified to positive by both BERT and LSTM. BERT tended to focus more on affirmative tokens such as "full" and "memorable" (a2) and successfully identified the role of the token "but" (a4) where the sentiment is reversed after it from negative to positive. Fig. 4 (a5-a8) show the interpretations of samples that are labeled as positive but misclassified as negative by

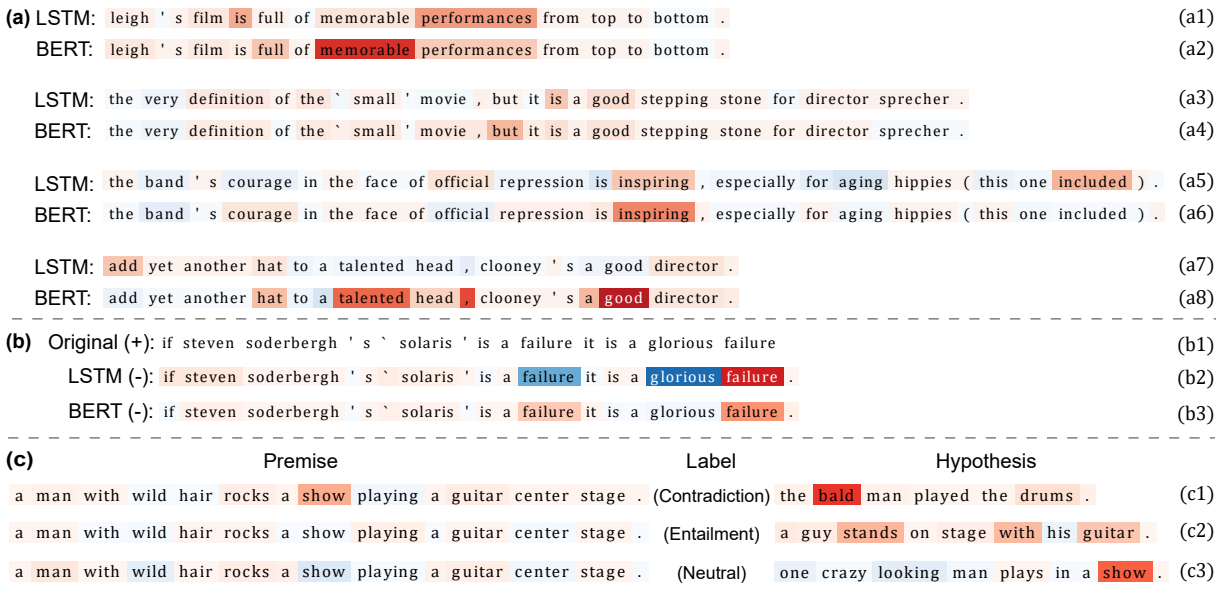

Figure 4: Interpretation results using input marginalization. Red and blue colors denote positive and negative contributions to the predicted classes. (a) shows interpretations of SST-2 predictions. (a1-a6) are correctly classified to positive, and (a7-a8) are correctly classified to negative. (b) shows positive sentences which are misclassified to negative by both LSTM and BERT. (c) shows the interpretations of SNLI predictions.

LSTM. The decisions of LSTM were significantly influenced by the word "included" and "add" (a5, a7). In contrast, BERT correctly classified them as positive by focusing on "inspiring" and "good".

Our method enables debugging the model by analyzing the misclassification case. Fig. 4 (b) shows the sentences whose true labels are positive but incorrectly classified as negative by both models. We measured the attribution score with respect to the negative class. For both models, the word "failure" was assigned significantly high attribution score indicating that both models failed to recognize the overall positive sentiment of the sentence by focusing on the negativity inherent in the word.

Fig. 4 (c) shows different attribution scores assigned to the same premise when the hypothesis changes. It is shown that the tokens in the hypotheses received higher scores than those in the premises. In fact, they obtained attribution scores twice as high as those in the premises for 500 interpretations. We can potentially conclude that the model was trained to pay more attention to hypothesis, since the SNLI corpus consists of repetitive premises and varying hypotheses. Moreover, (c1) shows that even if there are two contradictory word pairs, "wild hair"-"bald" and "guitar"-"drum", the model focused more on the the former. Our method allows potential model debugging when the interpretation turns out to be counterintuitive.

4.5 Effect of language modeling

In Eq. 2, an exact modeling of the likelihood $p(\tilde{x}_i|\boldsymbol{x}_{-i})$ is important for the accurate calculation of the attribution scores. Hence, the high agreement between the modeled and the real-world distributions will result in a more accurate interpretation. We analyzed the effect of the likelihood modeling capability on the accuracy of interpretation results. We tested three additional likelihood modeling: uniform distribution, prior probability, and fine-tuned BERT MLM.

Uniform $p(\tilde{x}_i|\boldsymbol{x}_{-i}) = 1/|\mathcal{V}| = 1/30522$ in the case of BERT tokenizer.

Prior $p(\tilde{x}_i|\boldsymbol{x}_{-i}) = p(\tilde{x}_i)$, defined by counting the frequency of each token in the training data.

Fine-tuned MLM We fine-tuned the MLM of BERT with the SST-2 dataset for two epochs (MLM_{fine}).

Using each likelihood distribution modeling, we interpreted the BERT classifier trained for the SST-2 corpus. The results are shown in Fig. 5. The uniform distribution failed to provide an accurate interpretation. The result with prior probability modeling appeared slightly clearer, but was still misleading. MLM_{pre} successfully highlighted important tokens, but it assigned high scores to tokens that were not expected to contribute significantly to predicting the sentiment of a movie review (e.g., "film" and "movie" marked with box). MLM_{fine} yielded

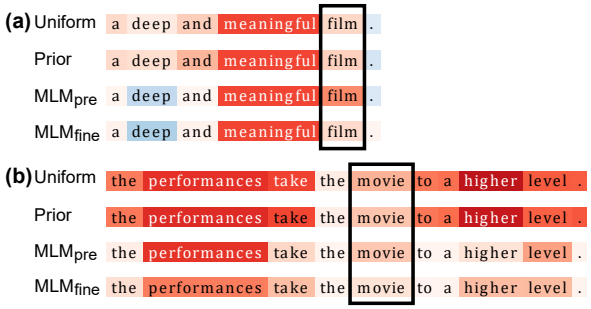

Figure 5: Interpretation results using different likelihood modelings. Each sentence is correctly classified to positive. Red and blue colors denote positive and negative contributions to the predicted classes.

the most reasonable interpretation, where the attribution score of "film" and "movie" was reduced from 0.256 and 0.631 to 0.007 and 0.321, respectively, compared to MLM_{pre}. We can expect the interpretation results to become more plausible as the likelihood modeling improves.

4.6 Ablation study on adaptively truncated marginalization

We introduced adaptively truncated marginalization in Section 3.4 for a faster computation. The full marginalization over all possible tokens yields the most exact attribution scores. Thus, we searched for an optimal threshold σ of adaptively truncated marginalization that reduces the computational complexity while maintaining a high correlation to the scores from full marginalization. We measured the correlation using the Pearson correlation coefficients. Furthermore, we tested fixed truncation, which calculates top-n likely candidates without considering the varying likelihoods depending on the position.

Table 3 shows the Pearson correlation coefficient and the average number of marginalization under various thresholds. $\sigma = 10^{-5}$ and 10^{-6} showed very similar interpretations to the full marginalization while the average marginalization number reduced to 2.5% and 10.4%, respectively, compared to 30,522 of the full marginalization. We regarded $\sigma = 10^{-5}$, which showed a lower number of marginalization under a similar correlation, as the optimal value. The fixed truncation showed a lower correlation under the similar average number of marginalization. The computational complexity can be further reduced by accepting a slight loss in the accuracy.

Table 3: The Pearson correlation with full marginalization and the average number of marginalization under various thresholds. σ: likelihood threshold, n: marginalization number threshold for fixed truncation.

		Corr	Avg #
Threshold	$\sigma = 10^{-6}$	**0.9999**	3,186
	$\sigma = 10^{-5}$	**0.9999**	791
	$\sigma = 10^{-4}$	0.9988	171
	$\sigma = 10^{-3}$	0.9928	33
	$n = 10^{3}$	0.9958	1,000
	$n = 10^{2}$	0.9823	100

5 Conclusion

Interpretability is becoming more important owing to the increase in deep learning in NLP. Hence, several interpretation methods have been proposed, and we reviewed their limitations throughout the paper. Among them, we focused on the OOD problem arising from the widely used zero erasure scheme, which results in misleading interpretation. To the best of our knowledge, neither the OOD problem has been raised in interpreting NLP models nor the attempt to resolve it has been undertaken. Our proposed input marginalization, which can mitigate the OOD problem, can result in a faithful interpretation, thereby enabling better understanding of "black box" DNNs.

The scope of this study was primarily focused on interpreting DNNs for sentiment analysis and natural language inference. Regarding the model-agnostic and task-agnostic properties of our method, they are applicable to any types of NLP model for various tasks, such as neural machine translation and visual question answering. It will be meaningful to interpret the state-of-the-art models such as XLNet (Yang et al., 2019) and ELECTRA (Clark et al., 2019). In addition, as experimentally analyzed, the interpretation result of our method is affected by the likelihood modeling performance. We can expect even more faithful interpretation if the modeling performance improves.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science and ICT) [2018R1A2B3001628], the Brain Korea 21 Plus Project in 2020, and Hyundai Motor Company.

References

Leila Arras, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. Explaining recurrent neural network predictions in sentiment analysis. *EMNLP 2017*, page 159.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. 2018. Explaining image classifiers by counterfactual generation.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

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.

Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. 2018. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE.

Dan Hendrycks and Kevin Gimpel. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Xisen Jin, Junyi Du, Zhongyu Wei, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194*.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*.

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. In *Advances in neural information processing systems*, pages 4765–4774.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Vitali Petsiuk, Abir Das, and Kate Saenko. 2018. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. *arXiv preprint arXiv:1910.04210*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.

Marko Robnik-Šikonja and Igor Kononenko. 2008. Explaining classifications for individual instances. *IEEE Transactions on Knowledge and Data Engineering*, 20(5):589–600.

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

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Jihun Yi, Eunji Kim, Siwon Kim, and Sungroh Yoon. 2020. Information-theoretic visual explanation for black-box classifiers.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Luisa M Zintgraf, Taco S Cohen, Tameem Adel, and Max Welling. 2017. Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.

Appendix

A1 Details of the experimental setup

A1.1 Dataset description

Stanford Sentiment Treebank (SST-2) SST-2 corpus (Socher et al., 2013) contains 11,855 sentences of movie review. Each sentence is labeled with five sentiments: very positive, positive, neutral, negative, very negative. We merged very positive and positive into positive and very negative and negative into negative, thereby resulting in binary class.

Stanford Natural Language Inference (SNLI) SNLI corpus (Young et al., 2014) contains 570k English sentence pairs annotated with one of three relationships: entailment, contradiction, and neutral. We used the dataset with the same split provided by Torchtext (Paszke et al., 2019). The dataset distributions used to train the target models are as below:

Table A1: Dataset distribution

	Train	Dev	Test	Total
SST-2	6,920	872	1,821	9,613
SNLI	549,367	9,842	9,824	569,033

A1.2 Target model architecture

We trained three models for SST-2 and one model for SNLI. For all models, we set an embedding dimension to 100 and activation function to ReLU. An output dimension was set to two and three for SST-2 and SNLI, respectively. We note that we did not conduct extensive hyper-parameter search because the accuracy is not important for evaluating our method.

LSTM for SST-2 Bidirectional LSTM with attention mechanism consists of an embedding layer, two bidirectional LSTM layers with a hidden dimension of 200, and a fully connected (FC) layer. The dropout rate for the embedding layer, LSTM, and the fully connected layer was 0.3, 0.5, and 0.5, respectively.

BERT We fine-tuned BERT-base classifier whose pre-trained weights are provided by Wolf et al. (2019) for five epochs.

CNN CNN consists of an embedding layer, three convolution layers and an FC layer. Each convolution layer contains 100 filters with size three, four, and five, respectively. The dropout rate for the convolution layer was set to 0.5.

LSTM for SNLI LSTM for SNLI comprises an embedding layer, a projection layer, a bidirectional LSTM layer, and four FC layers. The projection layer is an FC layer with an output dimension of 300. The encoder consists of one bidirectional LSTM layer with a hidden dimension of 300. Both premise and hypothesis are encoded with the same encoder and concatenated before the FC layer.

A1.3 Computing infrastructure

For all experiments, we used Ubuntu 16.04 on Intel i7-9800X(3.8GHz, 16.5MB) with four GTX Titan X Pascal (12GB).

A2 Interpretation results

A2.1 Statistical comparison

Pearson correlation coefficient and Spearman rank order correlation between the attribution scores from our method and zero erasure are shown in the table below.

Table A2: Dataset distribution

	Pearson	Spearman
Corr coef	0.5877	0.5383

A2.2 Additional comparison

We additionally compared our method to existing interpretation methods for further comparison, including DeepLIFT (DLIFT) (Shrikumar et al., 2017), Integrated Gradients (IG) (Sundararajan et al., 2017), gradient Shapley (SHAP) (Lundberg and Lee, 2017), and LIME (Ribeiro et al., 2016) We used a widely used open source interpretable AI library, captum (Kokhlikyan et al., 2020). The table below shows that our method exhibited the lowest $\mathrm{AUC}_{\mathrm{rep}}$ among various baselines (lower is better).

Table A3: Quantitative comparison to the existing methods

Methods				
DLIFT	IG	SHAP	LIME	Ours
0.5261	0.5387	0.5588	0.6618	**0.4971**

A2.3 Interpretation results of SST-2

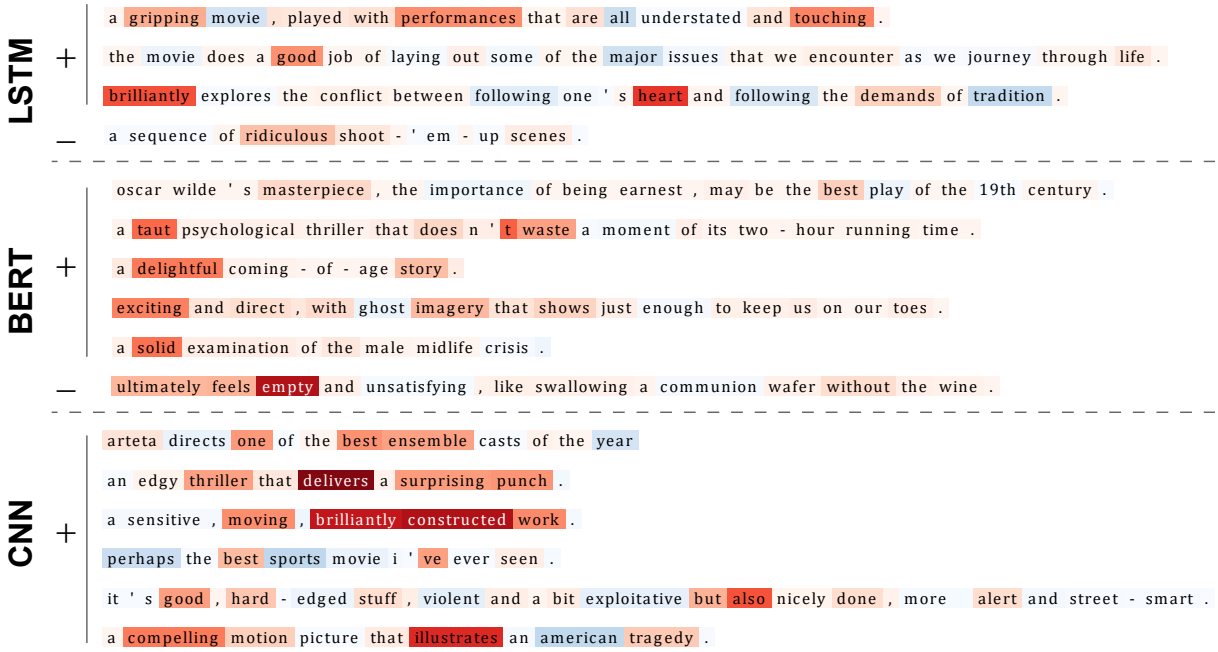

Figure A1: Interpretations of SST-2 trained models using input marginalization (ours). The characters before sentences denote the target class of the interpretation: + denotes positive and - denotes negative. Red and blue color denote a positive and negative contribution to the predicted class. All sentences were correctly classified.

A2.4 Interpretation results of SNLI

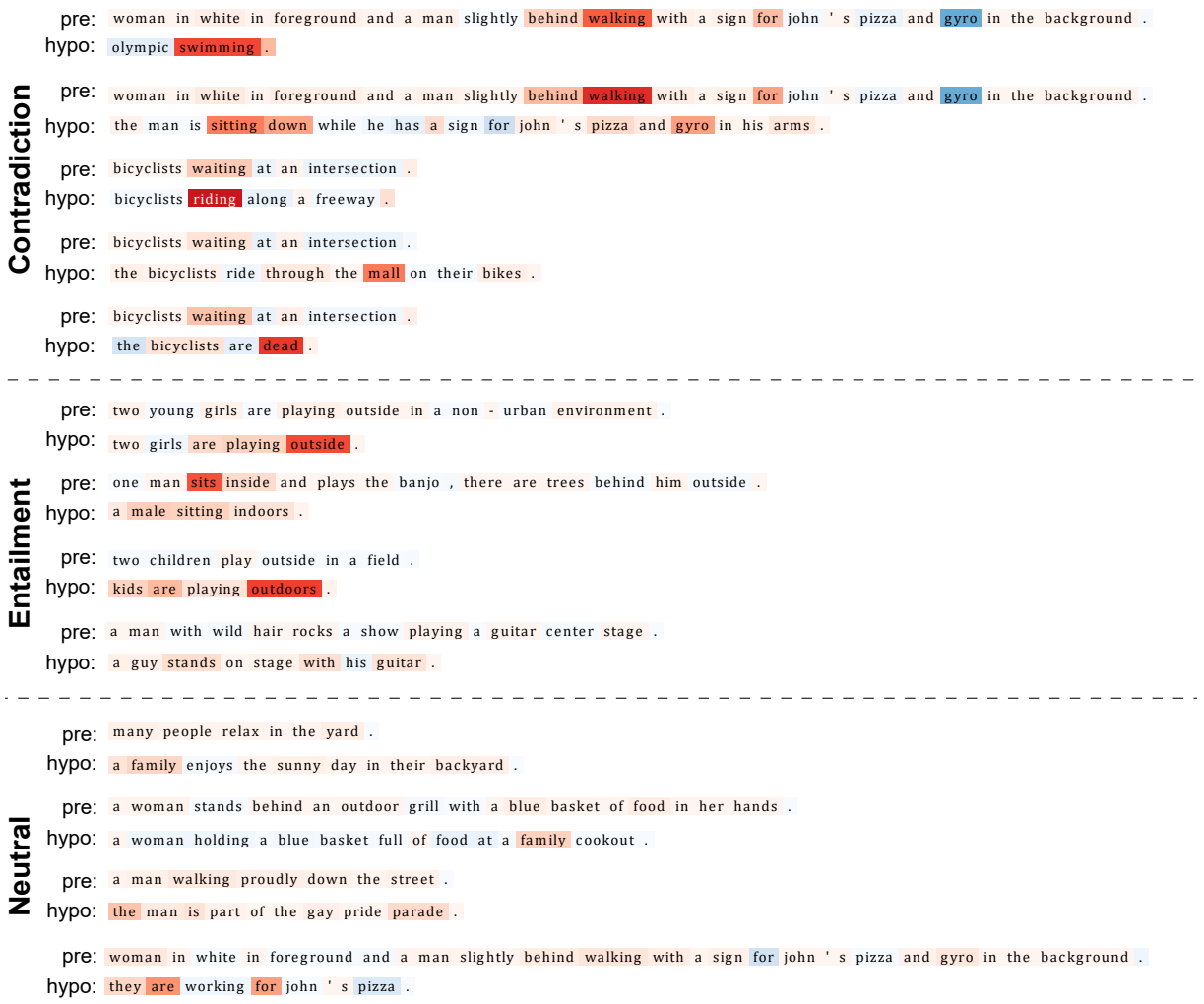

Figure A2: Interpretations of models trained for SNLI using input marginalization (ours). Red and blue color denote a positive and negative contribution to the predicted class. All sentences were correctly classified.

A2.5 Comparison to zero erasure

Figure A3: Interpretation results of zero erasure and input marginalization (ours). Red and blue color denote a positive and negative contribution to the predicted class, respectively. Zero in the figure denotes the existing zero erasure and Marg denotes the proposed input marginalization.