Learning Robust Rationales for Model Explainability: A Guidance-Based Approach

Shuaibo Hu, Kui Yu*

School of Computer and Information, Hefei University of Technology
shuaibohu@mail.hfut.edu.cn, yukui@hfut.edu.cn

Abstract

Selective rationalization can be regarded as a straightforward self-explaining approach for enhancing model explainability in natural language processing tasks. It aims to provide explanations that are more accessible and understandable to non-technical users by first selecting subsets of input texts as rationales and then predicting based on chosen subsets. However, existing methods that follow this select-then-predict framework may suffer from the rationalization degeneration problem, resulting in sub-optimal or unsatisfactory rationales that do not align with human judgments. This problem may further lead to rationalization failure, resulting in meaningless rationales that ultimately undermine people's trust in the rationalization model. To address these challenges, we propose a **G**uidance-based **Rat**ionalization method (**G-RAT**) that effectively improves robustness against failure situations and the quality of rationales by using a guidance module to regularize selections and distributions. Experimental results on two synthetic settings prove that our method is robust to the rationalization degeneration and failure problems, while the results on two real datasets show its effectiveness in providing rationales in line with human judgments. The source code is available at https://github.com/shuaibo919/g-rat.

Introduction

Selective rationalization is a method for explaining the predictions of a machine learning model by selecting a short and coherent part of the input that is sufficient for the prediction when yielding them (Gurrapu et al. 2023). Lei, Barzilay, and Jaakkola (2016) were the first to propose this select-then-predict framework for rationalizing neural predictions in that the selector first selects a rationale that is a subset of the entire input sentence. Then the predictor makes the judge only based on this rationale, as shown in Figure 1.

Based on the select-then-predict framework, many methods have been proposed (Bastings, Aziz, and Titov 2019; Yu et al. 2019; Chang et al. 2020). The rationale learned by these methods is the only available information to the predictor, leading to the predictor overfitting the rationale produced by the selector. That is to say, the whole model can still produce a lower prediction loss even though the quality of selected pieces is not good, and this problem is commonly referred to as rationalization degeneration (Yu et al. 2019; Liu et al. 2022). For example, the case of Figure 2 (Sub-optimal) given the right sentiment prediction but using the rationale missed some critical context. Many approaches (Huang et al. 2021; Yu et al. 2021; Yue et al. 2022; Sha, Camburu, and Lukasiewicz 2022) have been proposed to address the above degeneration issue. The basic idea of these approaches is to restrain the predictor using supplementary modules that utilize full information of inputs such that the predictor does not entirely rely on rationales at training time.

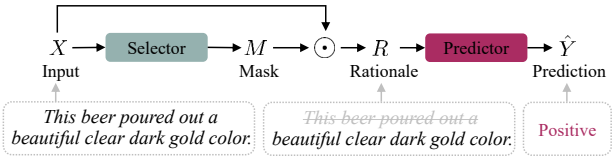

Figure 1: The basic framework for selective rationalization. $X, M, R, \hat{Y}, \odot$ represent the input text, binary mask, rationale, prediction, and element-wise product, respectively.

While these methods can partially alleviate the rationalization degeneration problem, they remain bound to the two-stage framework during training. Consequently, they may be unable to handle more intricate cases, which we named rationalization failure. Unlike the widespread concern about rationalization degeneration, the failure problem of selective rationalization has been largely overlooked in previous works (Zheng et al. 2022). This failure will make the rationale meaningless to users, thus damaging users' trust. Ideally, the selector should determine which words to be selected based on the input semantic information and honestly perform its selection responsibility. However, when a rationalization failure occurs (Zheng et al. 2022; Jacovi and Goldberg 2021), the selector is able to predict and pass its prediction through the binary mask M.

In Figure 2 (Failure), a broken selector always selects the first token of the input when its prediction is positive or the last token of the input when its prediction is negative. As a result, the predictor will produce a lower prediction loss in fitting the particular pattern than the semantics information of rationales, although the pieces selected by this broken selector are meaningless. Can we consider the rationales in this

*Corresponding author.

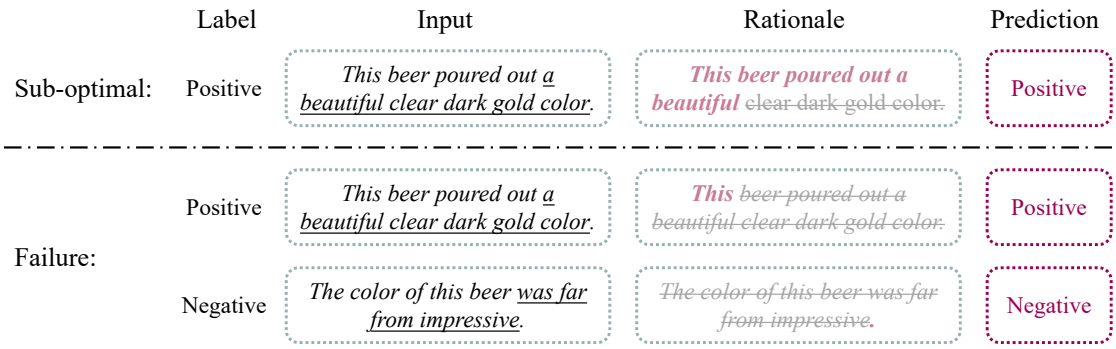

Figure 2: These two toy examples illustrate the degeneration of selective rationalization. The underlined pieces of the text are the gold rationale, and the pieces of tokens in dark red represent the rationales from the selector. In the sub-optimal case, the chosen rationale still contains useful semantic information (the word beautiful) for making accurate predictions, but it may not be the most optimal choice. In the case of failure, the selected rationale conveys the prediction information by choosing the first or last token of the input without contributing any meaningful semantic content to the correct prediction.

case as explanations? Obviously not. It does not provide any valuable information to users. Unfortunately, existing rationalization methods may encounter this failure situation and fail to deal with it. Explicitly modeling rationalization failure is challenging since the prompt from the selector may be more complex than the toy example we mentioned above.

To address both rationalization degeneration and failure problems, in this paper, we propose a novel method named **G**uidance-based **Rat**ionalization (**G-RAT**) for selective rationalization that contains two modules named the rationale module and the guidance module. The rationale module follows the previous selective rationalization framework, while we use the guidance module to guide the rationale module's selector and predictor simultaneously. The guidance module outputs a weighted score and a prediction distribution. We use the former to regularize the rationale module's selector for dealing with the failure problem and the latter to regularize the rationale module's predictor for dealing with the degeneration problem. Different from the previous rationalization methods, our method takes an important step in coping with rationalization failure. In addition, we have created a new synthetic experiment named Skew-Selector, which simulates a failed selector based on different strengths and can be used to effectively evaluate a rationalization method in preventing the rationalization failure problem. Finally, extensive experiments demonstrate that our approach can produce more informative rationales than existing methods and deal with the failure problem effectively.

Related Work

Model Explainability

Model explainability refers to an understanding and explanation of how a machine learning model works and why it makes specific predictions. This is essential for many reasons, including guaranteeing trust, safety, and accountability in machine learning systems. Current research on model explainability are mainly divided into post-hoc methods and self-explanatory models (Danilevsky et al. 2020; Sun et al. 2021), and our method belongs to the latter.

Existing post-hoc methods aim to interpret a neural network after it has been trained by analyzing how each feature or instance affects the model prediction. Various techniques fall under this category, such as saliency maps (Simonyan, Vedaldi, and Zisserman 2014), LIME (Ribeiro, Singh, and Guestrin 2016), and SHAP (Lundberg and Lee 2017). Recent works (Covert, Lundberg, and Lee 2021; Deng et al. 2023) have also attempted to explain post-hoc approaches in a unified view. Post-hoc methods have limitations. They do not consider the model's structure and require extra computations. They may not be entirely trustworthy in capturing relationships between features and the output (Rudin 2019).

The self-explanatory models focus on building models that are inherently interpretable without the need for external tools. These models can incorporate various types of explanations, such as feature-based explanations (Lei, Barzilay, and Jaakkola 2016; Chen et al. 2018), which select or generate a subset of features that can justify the output; and natural language explanations (Camburu et al. 2018; Kumar and Talukdar 2020; Rajani et al. 2019), which produce textual pieces that can explain the result in a human-readable way.

Selective Rationalization

Selective rationalization aims to construct a self-explanatory model which can provide explanations and predictions simultaneously by extracting important features of inputs. The inputs that are not selected will not have any impact on the prediction. Lei, Barzilay, and Jaakkola (2016) first proposed a select-then-predict framework for rationalization with a reinforce-style training (Williams 1992).

To address the end-to-end optimization problem of the vanilla rationalization framework, Bastings, Aziz, and Titov (2019) proposed the use of a rectified Kumaraswamy distribution to re-parameterize gradient estimates instead of the Bernoulli sampling. Meanwhile, there are other methods available to replace the reinforce-style training in the framework presented by Lei, Barzilay, and Jaakkola, such as the Gumbel-softmax trick (Jang, Gu, and Poole 2016), which has been employed in various works (Bao et al. 2018; Paran-

jape et al. 2020; Sha, Camburu, and Lukasiewicz 2022).

Another series of improvements focus on exploiting the information from the original text to regularize the predictor and improve the quality of the selected rationale. Some researchers have accomplished this by adding extra modules to the rationalization process. Huang et al. (2021) matched the distributions of rationales and input text in both the feature and output spaces. Sha, Camburu, and Lukasiewicz (2022) introduced an adversarial-based technique to make the select-then-predict model learn from an extra predictor. Yue et al. (2022) used the information of non-rationales for extracting rationales. In particular, some work has analyzed the phenomenon of rationalization degeneration following this line of work. For instance, Yu et al. (2021) introduced a soft selection using an attention module to avoid the risks of degeneration from model interlocking. Liu et al. (2022) utilized a shared encoder to keep the selector and the predictor having the same learning speed to alleviate the degeneration problem. The primary goal of these methods is to improve the quality of rationales. Our approach also aligns with this objective but highlights strengthening the robustness to deal with both rationalization degeneration and failure problems.

Methodology

Preliminary

Selective Rationalization Consider a text classification problem, (X, Y), X is the input with n tokens $X = \{x_1, x_2, ..., x_n\}$, and Y is the ground-truth corresponding label from the training set $\mathcal{D}_{tr}$. In the process of selective rationalization, the selector takes X as an input and outputs a binary mask of $M = \{m_1, m_2, ..., m_n\}$, where $m_i \in \{0, 1\}$ indicates whether to select the ith token x_i, then the predictor uses the rationale which is a subset of the input X, $R = M \odot X = \{m_1 x_1, m_2 x_2, ..., m_n x_n\}$ to yield model prediction. Suppose $g(\cdot; \theta_g)$ and $f(\cdot; \theta_f)$ represent the selector and predictor respectively. We then feed the rationale to the predictor to obtain a prediction and calculate the loss $\mathcal{L}_{task}$ for the entire select-then-predict model. Formally, the process of rationale selection and prediction is as follows:

$$\min E_{\substack{X,Y\sim\mathcal{D}_{tr} \\ M\sim g(X,\theta_g)}} [\mathcal{L}_{task}(f(M \odot X; \theta_f), Y)]. \quad (1)$$

Regularizing for Shortness and Coherence We expect that the selector $g(\cdot; \theta_g)$ selects short and fluent rationales in practice. To achieve this goal, Lei, Barzilay, and Jaakkola constrain the rationales to use an additional regularizer $\mathcal{L}_s = \lambda_1 \sum_{i=0}^{n} |M_i| + \lambda_2 \sum_{i=1}^{n} |M_i - M_{i-1}|$ with respect to the selections where the first term penalizes the number of selections, and the second one encourages continuity of selections. The specific form may be a different subject to the different architecture of the rationale model (Bastings, Aziz, and Titov 2019; Huang et al. 2021; Sha, Camburu, and Lukasiewicz 2022). For example, many methods replace the first term of the above regularizer with $\lambda_1 |\alpha - \frac{1}{n} \sum_{i=0}^{n} M_i|$, where the $\alpha \in [0, 1]$ explicitly specifies the degree of sparsity. So Eq.(1) can be rewritten as:

$$\min E_{\substack{X,Y\sim\mathcal{D}_{tr} \\ M\sim g(X,\theta_g)}} [\mathcal{L}_{task}(f(M \odot X; \theta_f), Y) + \mathcal{L}_s]. \quad (2)$$

Guidance-Based Rationalization

Overall Pipeline As shown in Figure 3, our proposed G-RAT consists of two modules: a rationale module and a guidance module. The rationale module is based on the standard select-then-predict framework, and the guidance module is an end-to-end predictor. In addition to generating a prediction distribution $p(H)$, the guidance module will also generate a weighted score α, which is the weight used to multiply with the representation of each token before the guidance module generates $p(H)$, so it can reflect the contribution of tokens to the prediction to some extent. Then, we incorporate the guidance module into our training process and use the guidance model's two outputs to regularize the rationale module, which can also be seen as a process of knowledge distillation. Once training is completed, we only keep the rationale module for predicting and providing rationales.

Regularizing Selections for Failure Alleviation As discussed previously, rationalization failure means that the selector and predictor cooperate by selecting meaningless tokens to encode class information instead of selecting those that truly explain the prediction. This happens because (1) there are no restrictions on M to prevent it from encoding class information, and (2) the predictor also has the ability to directly use each token's position in the rationale R for prediction, such as the failure example in Figure 2. Therefore, the rationalization failure can be alleviated if we can restrict or directly destroy the ability of M to contain the class information. Based on this idea, we propose a new strategy to alleviate this problem. Specifically, we make some noise disturbances on M to make the predictor unable to get a higher prediction accuracy when it uses the positional information, thus forcing the predictor to focus on the semantic information instead of the positional information in R. However, it is well-known that selective rationalization schemes are hard to train (Yu et al. 2019; Bastings, Aziz, and Titov 2019). Adding noise directly to disturb M may lead to the instability of the rationale $M \odot X$ and make the training process even harder to converge.

To tackle this issue, we incorporate noise into the guidance model's weighted score α, which is then used to guide learning M instead of directly disturbing M. This weighted score α represents the weights of each token in the input text, as we mentioned previously, so it can guide the selector to choose the tokens that contributed to the prediction. In addition, the weighted score α is a vector with continuous values, and introducing noise to it has less influence on the convergence of the training process than introducing noise to the binary mask M. With the weighted score α, the rationale module's selector can be constrained not to encode the class information into M, reducing the possibility of falling into rationalization failure. Specifically, to achieve this goal, our guidance module takes the original text X as the input and consists of two primary cells: $h(\cdot; \theta_h)$ and $p(\cdot; \theta_p)$. For each $X \in \mathcal{D}_{tr}$, the first cell $h(\cdot; \theta_h)$ generates a score α and hidden states H, while the second cell $p(\cdot; \theta_p)$ takes H to output its prediction $p(H)$ which will used to deal with the rationalization degeneration problem in the next section.

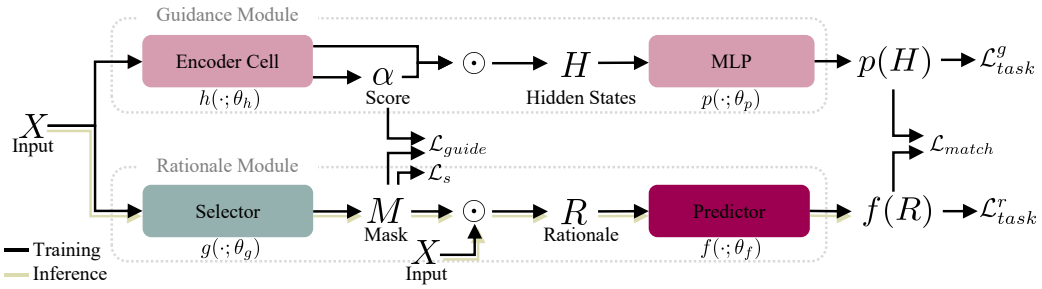

Figure 3: The proposed rationalization architecture. The guidance module regularizes the selector and the predictor using two loss terms $\mathcal{L}_{guide}$, $\mathcal{L}_{match}$. The rationale module is trained separately from the guidance module and only uses the guidance module as guidance. Only the rationale module is used at inference time to obtain the prediction and the rationale.

This internal process can be formalized as follows:

$$\begin{aligned} &\hat{H} = h(X;\theta_h) + e(X), \quad \alpha = softmax(\hat{H}W_1 + \epsilon), \\ &H = (\alpha \odot \hat{H})W_2, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \end{aligned} \tag{3}$$

Here e is the embedding layer, W_1 and W_2 represent two distinct linear layers. The variable ϵ is random noise sampled from a multivariate Gaussian distribution. Since the score α generated by the guidance module reflects the contribution of each token with regard to the whole input, its value does not match the binary situation in the rationale module. Therefore, before using this score to guide the selection of the rationale module, we scale each element of α by $\min(\alpha_i/\bar{\alpha}, 1.0)$ and we marked it as a parametric function $\hat{\alpha}(\cdot;\theta_{\hat{\alpha}})$ for simplicity. Then we use $\hat{\alpha}(X;\theta_{\hat{\alpha}})$ as the supervision to calculate its cross-entropy with the selector's output $g(X;\theta_g)$, which regularizes the rationale selection:

$$\mathcal{L}_{guide} = CrossEntropy(g(X;\theta_g), \hat{\alpha}(X;\theta_{\hat{\alpha}})). \tag{4}$$

Matching Predictions for Degeneration Reduction In the previous section, the weighted score α is used to alleviate the rationalization failure problem, allowing our method to repair the broken selector. Unfortunately, it cannot repair the inadequate or sub-optimal rationales that will occur in the rationalization degeneration situations since the score α can only partially reflect the contribution of each token to the prediction. However, we know that the best rationale can make the same prediction as the original input, which means that the prediction distribution based on the best rationale should be consistent with the one based on the original input. Based on this idea, we match the two prediction distributions of the guidance module and the rationale module as consistently as possible since the guidance module makes predictions based on the original input X to meet the above requirements. Then we compute the Jensen-Shannon divergence between the two predictive distributions and use it as a regularization term on the rationale-predictor:

$$\mathcal{L}_{match} = JS(f(M \odot X;\theta_f) || p(h(X;\theta_h) \odot \alpha;\theta_p)), \tag{5}$$

here, the prediction distribution of the rationale module is represented by $f(M \odot X;\theta_f)$ where the rationale $R = M \odot X$ is the input. Similarly, the part of the guidance module is represented by $p(h(X;\theta_h) \odot \alpha;\theta_p)$ where the hidden states $H = h(X;\theta_h) \odot \alpha$ are the input. The distribution alignment enforces the outputs of the rationale module and guidance module to have the minimum distance. It can be seen as a way of preserving the information from the original input into the rationales and can help avoid selecting irrelevant features that may lead to rationalization degeneration.

Switching of Regularization Terms in Training We face a new concern when incorporating Eq.(4) into our training as a regularization term. It's possible that the rationale-selector could become overly dependent on the weighted score α of the guidance module, which could negatively impact the overall performance. Therefore, we use a coefficient τ to control this external guidance, which will decay from 1 to 0 in the training process. Another interesting fact is that Eq.(5) as a regularization term only makes sense after the guidance module converges, so we reuse the above coefficient to smoothly switch the above two regularizers and combine the two terms as a one $\mathcal{L}_{g\&m}$:

$$\mathcal{L}_{g\&m} = \tau\lambda_{guide}\mathcal{L}_{guide} + (1-\tau)\lambda_{match}\mathcal{L}_{match}, \tag{6}$$

here, λ_{guide}, λ_{match} are used to control the constraint strength of $\mathcal{L}_{guide}$, $\mathcal{L}_{match}$ respectively. For simplicity, We linearly decay this coefficient τ until it reaches 0 in training.

Training and Inference By combining all the above equations, we divide the total losses into two parts: the loss of the rationale module and the loss of the guidance module. The objective of our rationale module is as follows:

$$\min E_{\substack{X,Y\sim\mathcal{D}_{tr} \\ M\sim g(X,\theta_g)}} [\mathcal{L}^r_{task} + \mathcal{L}_s + \mathcal{L}_{g\&m}], \tag{7}$$

where $\mathcal{L}^r_{task}$ uses the superscript r to indicate that this is the task loss for the rationale module $\mathcal{L}_{task}(f(M \odot X;\theta_f), Y)$. Since the guidance module does not have the sampling operation $M \sim g(X,\theta_g)$, its objective is to directly minimize the difference between the predicted and the true labels:

$$\min E_{X,Y\sim\mathcal{D}_{tr}}[\mathcal{L}^g_{task}], \tag{8}$$

where $\mathcal{L}^g_{task}$ indicates that this is the task loss for the guidance module $\mathcal{L}_{task}(p(h(X;\theta_h) \odot \alpha;\theta_p), Y)$.

During the training phase, we use the Adam (Kingma and Ba 2015) optimizer alternatively to optimize the above two

Setting		RNP*				FR*				G-RAT			
		Acc	P	R	F1	Acc	P	R	F1	Acc	P	R	F1
Smell	Skew-10	82.6	68.5	63.7	61.5	87.1	73.9	71.7	72.8	85.5	84.8	63.9	**72.9**
	Skew-15	80.4	54.5	51.6	49.3	86.1	71.3	68.0	69.6	85.6	84.9	63.3	**72.5**
	Skew-20	76.8	10.8	10.8	11.0	85.5	72.3	69.0	70.6	84.6	85.3	63.3	**72.7**
Palate	Skew-10	77.3	5.6	7.4	5.5	75.8	54.6	61.2	57.7	81.0	63.4	59.8	**61.7**
	Skew-15	77.1	1.2	2.5	1.3	81.7	51.0	58.4	54.5	82.0	63.5	59.5	**61.4**
	Skew-20	75.6	0.4	1.4	0.6	83.1	48.0	58.9	52.9	85.5	69.9	51.1	**59.0**

Table 1: Results of Skew-Predictor settings. " * " represents the results from Liu et al. (2022).

Setting		Appearance					Smell					Palate				
		Acc	S	P	R	F1	Acc	S	P	R	F1	Acc	S	P	R	F1
Skew-55	Re-RNP	85.6	19.3	75.4	73.5	74.4	84.1	14.7	67.7	56.4	61.6	59.1	12.8	0.2	0.2	0.2
	FR	82.8	20.6	78.8	81.6	80.2	86.1	14.9	75.0	64.3	69.3	73.3	11.2	39.4	20.0	26.5
	G-RAT	81.8	19.3	82.9	81.0	**81.9**	88.6	14.8	80.4	69.7	**74.7**	82.2	14.3	60.9	63.3	**62.1**
Skew-60	Re-RNP	85.7	20.1	72.6	74.6	73.6	81.1	13.9	70.2	55.9	62.3	85.9	15.4	0.2	0.2	0.2
	FR	83.3	20.7	79.8	83.2	81.5	84.1	14.6	77.4	65.6	71.1	67.5	12.3	0.2	0.2	0.2
	G-RAT	86.2	19.7	82.7	82.5	**82.6**	82.8	13.9	82.4	67.3	**74.1**	87.5	14.1	59.8	63.2	**61.5**
Skew-65	Re-RNP	83.3	19.6	72.2	73.4	72.8	82.2	17.6	45.0	46.8	45.9	83.7	15.1	0.1	0.2	0.2
	FR	83.3	18.8	75.0	73.1	74.0	80.9	18.0	41.4	44.2	42.8	85.7	12.5	0.3	0.2	0.3
	G-RAT	86.0	19.8	80.8	81.8	**81.3**	84.3	13.8	82.3	66.6	**73.6**	83.7	15.2	56.7	62.9	**59.6**
Skew-70	Re-RNP	81.5	21.4	64.0	70.9	67.2	81.9	16.7	43.4	42.8	43.1	84.0	14.9	0.2	0.2	0.2
	FR	77.6	19.1	71.6	71.5	71.5	81.2	18.3	36.8	40.5	38.5	86.9	13.7	0.4	0.4	0.4
	G-RAT	90.1	19.0	81.1	79.3	**80.2**	87.5	14.1	79.6	65.5	**71.9**	83.9	17.5	45.4	56.7	**50.4**
Skew-75	Re-RNP	86.1	20.7	68.9	73.7	71.2	79.4	14.7	2.1	1.8	1.9	80.8	15.0	0.2	0.2	0.2
	FR	79.7	19.4	68.4	69.8	69.0	81.8	14.4	2.0	1.7	1.8	83.7	14.2	0.2	0.2	0.2
	G-RAT	83.2	19.2	79.0	77.6	**78.3**	82.5	17.7	57.6	59.3	**58.5**	82.6	17.9	0.9	1.3	**1.1**
Skew-80	Re-RNP	83.3	19.2	69.9	69.7	69.8	81.0	15.4	2.6	2.3	2.4	81.3	15.2	0.2	0.2	**0.2**
	FR	84.4	18.7	68.2	66.5	67.3	79.1	14.6	0.2	0.2	0.2	83.9	14.8	0.3	0.2	**0.2**
	G-RAT	80.1	18.9	74.6	72.8	**73.7**	85.2	17.1	5.5	6.1	**5.8**	86.9	16.4	0.2	0.3	**0.2**

Table 2: Results of Skew-Selector settings. In the three aspects, the precise k of different skew thresholds are $\{55.0, 55.5, 55.2\}$, $\{60.2, 60.0, 60.6\}$, $\{65.0, 65.5, 65.6\}$, $\{70.3, 70.5, 70.1\}$, $\{75.1, 75.2, 75.1\}$, and $\{80.0, 80.1, 81.2\}$ respectively.

objectives on our training dataset $\mathcal{D}_{tr}$. This allows us to leverage the guidance module as a regularizer to guide the rationale module without the latter's objective affecting its parameters. During the inference phase, we only execute the rationale module to obtain the final prediction results and the rationales $R = M \odot X$, consistent with previous rationalization methods, as shown in Figure 3. More detailed settings on training and hyperparameters can be found in Appendix.

Experiments

Experimental Settings

Datasets Following the work of Huang et al. (2021) and Liu et al. (2022), we consider two widely used datasets for selective rationalization. 1) **BeerAdvocate** (McAuley, Leskovec, and Jurafsky 2012) contains more than 220,000 beer reviews, where users rate different aspects of beer from 0 to 5 stars. We replicate the pre-process of Lei, Barzilay, and Jaakkola (2016) where the dataset has been decorrelated into three aspects. 2) **HotelReview** (Wang, Lu, and Zhai 2010) is another multi-aspect dataset similar to BeerAdvocate. It contains reviews of hotels and ratings in different aspects. For a fair comparison, we consider both the above beer and hotel reviews prediction as a binary classification task as Bao et al. (2018) did, while other settings followed the previous works (Huang et al. 2021; Yu et al. 2021). The Appendix has pre-processing settings details.

Comparison Methods We compare our approach to several influential selective rationalization methods and the latest improvement work: **RNP** (Lei, Barzilay, and Jaakkola 2016), **DMR** (Huang et al. 2021), **A2R** (Yu et al. 2021), and **FR** (Liu et al. 2022), all these methods are detailed in the section of related work. In addition, our re-implementation of RNP using the straight-through trick (Bengio, Léonard, and Courville 2013) and Gumbel-softmax (Jang, Gu, and Poole 2016) for reparameterization is called **Re-RNP**. To ensure the comparability of the results, we follow the commonly used setting in our implementations. For all the implemented methods, we replicate the setting that most pre-

Methods	Appearance					Smell					Palate				
	Acc	S	P	R	F1	Acc	S	P	R	F1	Acc	S	P	R	F1
RNP*	84.0	18.7	72.0	72.7	72.3	85.2	15.1	59.0	57.2	58.1	90.0	13.4	63.1	68.2	65.5
Re-RNP	85.6	17.9	72.5	68.3	70.3	84.8	16.1	56.9	60.9	58.8	77.2	17.3	45.7	56.8	50.6
DMR*	-	18.2	71.1	70.2	70.7	-	15.4	59.8	58.9	59.3	-	11.9	53.2	50.9	52.0
A2R*	83.9	18.4	72.7	72.3	72.5	86.3	15.4	63.6	62.9	63.2	81.2	12.4	57.4	57.3	57.4
FR*	87.2	18.4	82.9	82.6	82.8	86.6	15.0	74.7	72.1	73.4	89.7	12.1	67.8	66.2	**67.0**
G-RAT	88.4	18.5	84.8	83.2	**84.0**	87.8	15.5	79.1	74.3	**76.6**	88.3	12.3	63.4	67.2	65.2
G-RAT$_{-\mathcal{L}_{guide}}$	85.4	18.9	80.8	82.1	81.4	83.9	15.4	64.3	66.3	65.3	85.7	13.4	63.5	57.3	60.2
G-RAT$_{-\mathcal{L}_{match}}$	84.6	19.0	82.9	79.7	81.2	84.7	15.7	78.8	61.7	69.2	81.3	14.2	61.0	68.1	64.4

(a) Beer Reviews

Methods	Location					Service					Cleanliness				
	Acc	S	P	R	F1	Acc	S	P	R	F1	Acc	S	P	R	F1
RNP*	97.5	8.8	46.2	48.2	47.1	97.5	11.0	34.2	32.9	33.5	96.0	10.6	29.1	34.6	31.6
Re-RNP	97.6	9.9	43.8	44.4	44.1	96.9	11.8	40.0	35.2	37.5	96.4	11.3	29.4	31.8	30.5
DMR*	-	10.7	47.5	60.1	53.1	-	11.6	43.0	43.6	43.3	-	10.3	31.4	36.4	33.7
A2R*	87.5	8.5	43.1	43.2	43.1	96.5	11.4	37.3	37.2	37.2	94.5	8.9	33.2	33.3	33.3
FR*	93.5	9.0	55.5	58.9	57.1	94.5	11.5	44.8	44.7	44.8	96.0	11.1	34.9	43.4	38.7
G-RAT	97.4	10.1	56.1	59.3	**57.6**	97.9	12.1	48.8	44.1	**46.3**	95.9	11.9	41.4	37.3	**39.2**
G-RAT$_{-\mathcal{L}_{guide}}$	98.0	9.5	53.5	47.3	50.2	98.3	12.4	41.3	39.0	40.1	96.1	10.3	38.0	33.3	35.5
G-RAT$_{-\mathcal{L}_{match}}$	98.0	9.7	58.3	53.5	55.8	98.1	12.8	43.1	46.2	44.6	95.8	11.3	40.9	35.9	38.2

(b) Hotel Reviews

Table 3: Results on beer and hotel review prediction tasks. " * " represents the results from Yu et al. (2021) and Liu et al. (2022).

vious works have adopted (Yu et al. 2021; Liu et al. 2022) by using the 100-dimension Glove (Pennington, Socher, and Manning 2014) as the embedding, GRU as the encoder cell. Experiments are all conducted on a single Tesla A100 GPU.

Evaluation Metrics Following the work of Chang et al. (2020) and Yu et al. (2021), we mainly focus on the quality of the rationales, which is measured by the overlap between the model-selected and human-annotated tokens using the token-level precision, recall, and F1-score denoted as **P**, **R**, and **F1** respectively. The best results of the F1-score are emphasized in **bold**. **S** refers to the average proportion of selected tokens in the original text. *Acc* refers to the predictive accuracy, with all methods getting similar values.

Synthetic Experiments

To show that our G-RAT is more robust to the rationalization degeneration and failure problem, we conduct two synthetic experiments in the BeerAdvocate dataset. These synthetic experiments were first proposed in A2R (Yu et al. 2021) and later improved in FR (Liu et al. 2022). We mainly refer to the experimental settings in FR and then compare our G-RAT with it and our direct baseline RNP.

Skew-Predictor Yu et al. (2021) pre-trained the predictor separately using only the first sentence of the input text in some aspects of BeerAdvocate. As a result, the over-fitted predictor in the first sentence will pass an incorrect gradient to the selector, thus simulating a rationalization degeneration situation. Thus, we initialize the predictor with intentionally misleading pre-trained parameters before training the selector and predictor cooperatively. We replicate their experiment as Liu et al. (2022) did and use skew-k to represent the skew threshold, where k represents the number of epochs for which the predictor was pre-trained.

Skew-Selector This is our newly designed synthetic experiment with a stronger setup than Liu et al. (2022) did. we pre-train the selector separately using the text classification label as the label of mask only in the first token position while keeping the regularizer of shortness and coherence $\mathcal{L}_s$. As a result, the predictor will be able to know the category of a rationale merely through whether the first token is selected as a part of the rationale or not and may overfit this positional bias. Compared with the setup in FR, our synthetic experiment considers the regularizer $\mathcal{L}_s$, which makes it closer to the rationalization failure that may occur in real training, and we conduct experiments on all three aspects of BeerAdvocate not only in the palate aspect. In detail, we pre-trained a broken selector with $k = \{55, 60, 65, 70, 75, 80\}$ on each of the three aspects, and we predefined the sparsity with $\{15\%, 10\%, 10\%\}$ respectively. Since the accuracy increases rapidly in the first few epochs, the skew threshold k here is the task accuracy, not the number of epochs.

Results In the above Skew-Selector setting, the $h(\cdot;\theta_h)$ of the guidance module and the $g(\cdot;\theta_g)$ of the rationale module in our method are all initialized with the parameters of the skew-selector for the fair comparison. The experimental results of these two synthetic experiments are shown in Table 1 and Table 2. We first found that all methods achieve high prediction accuracy at different skew thresholds. How-

ever, with increasing the skew threshold k, RNP fails to find human-annotated rationales, and its F1-score rapidly decreases in the palate aspect of the Skew-Predictor and Skew-Selector settings. FR shows its ability to alleviate the degeneration problem in the Skew-Predictor setting, but its F1-score declines at the lowest threshold (Skew-55) of the palate aspect in the Skew-Selector setting. In contrast, our method is robust in dealing with this situation of deliberately initializing misleading parameters. It shows consistent performance under different threshold k on two skew settings, and even in the most challenging palate aspect, it still works well until in the skew-75 setting. We also found that all the methods achieve a high F1-score in the appearance aspect under the Skew-Selector setting. This was because the first token we used to convey the class information overlapped with the description of the appearance (the first sentence is usually about the appearance aspect of the beer in the BeerAdvocate dataset). Finally, these results show that our method can prevent the rationalization degeneration and failure problems. This is a significant step towards solving a critical issue in the existing rationalization framework, which can enhance people's trust in model prediction, especially in some safety-critical application scenarios.

Results on Real-World Settings

Reviews Prediction Table 3 (a) and Table 3 (b) show the main results compared to previous methods on two review prediction benchmarks. All comparative experiments used the settings mentioned in the previous section and selected the same pre-defined sparsity in the regularization term $\mathcal{L}_s$. Regarding the F1 score, our approach outperforms the best available methods in five aspects of the two datasets. In particular, compared to our direct re-implementation baseline Re-RNP, we get significant improvements in terms of F1 score in several aspects (i.e., Appearance, Smell, Palate, Location). The results of synthetic experiments have shown that G-RAT can improve the robustness of the rationalization method by introducing two regularizers $\mathcal{L}_{guide}$ and $\mathcal{L}_{match}$, and the improvement on this real-world settings proves that these two regularizers can also improve the overall performance. We also conducted the following ablation studies further to verify these two regularizers' influence.

Ablation Study 1) We removed the regularizer $\mathcal{L}_{guide}$ and report the results in the line G-RAT$_{-\mathcal{L}_{guide}}$, and 2) we removed the regularizer $\mathcal{L}_{match}$ and report the results in the line G-RAT$_{-\mathcal{L}_{match}}$. In Table 3, we can see that removing any regularizer will lead to a decline in the F1 score in all aspects. This shows that these two regularizers can cooperate with each other and help improve the rationale quality.

Parameters Sensitivity Analysis

In the previous setup, we set $\lambda_{guide} = 5.0$ and $\lambda_{match} = 1.0$. This is an empirical choice because these two regularizers can have a similar scale as the task loss $\mathcal{L}_{task}$. To gain an insight into the effects of selecting different λ, we further conduct experiments varying λ_{guide} and λ_{match}.

Sensitivity of λ_{guide} We repeat the Skew-Selector experiment with the skew thresholds of $k = 70$ and $k = 65$ on the three aspects while setting $\lambda_{match} = 0$. We disregard the gradually decaying τ to observe the effect of $\mathcal{L}_{guide}$ without potential interference. Figure 4 describes the results under the value of λ_{guide} ranging from 0.1 to 30. It is observed that $\mathcal{L}_{guide}$ performs well when λ_{guide} is within the range of $[2, 10]$. If λ_{guide} is smaller than this range, $\mathcal{L}_{guide}$ struggles to repair the broken selector, whereas if λ_{guide} is too large, it restricts the rationale-selector's ability excessively.

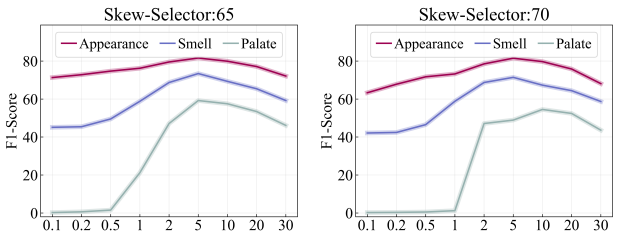

Figure 4: Analysis of the sensitivity of λ_{guide}

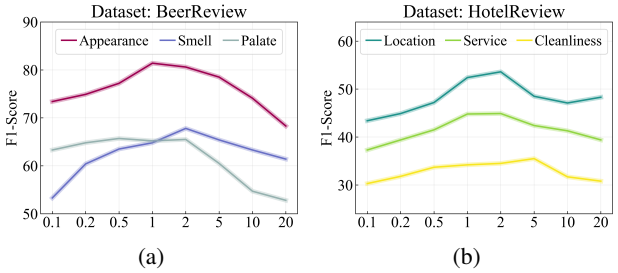

Figure 5: Analysis of the sensitivity of λ_{match}

Sensitivity of λ_{match} Similar to the previous analysis, we set λ_{guide} to 0 to exclude interference, and then we re-run G-RAT on the beer and hotel review tasks, with the value of λ_{match} varying from 0.1 to 20. Figure 5 shows that the performance on both datasets presents a shape of low on both sides and high in the middle which shows that λ_{match} performs well in the wide range $[0.5, 2]$.

Conclusion and Future Work

In this paper, we have thoroughly discussed the rationalization degeneration and failure problems that may arise in the existing rationalization methods. Therefore, we proposed G-RAT, a robust guidance-based rationalization approach that utilizes a guidance module to regulate the process of selective rationalization. Through quantitatively testing and analyzing G-RAT in both real-world and synthetic settings, the results showed that it outperformed existing methods in rationale quality and robustness. G-RAT helps to promote trustworthiness in model explainability. Moving forward, we plan to explore the feasibility of rationalizing large language models, such as incorporating abstract rationales or human feedback, and further study other ways to address the rationalization degeneration and failure problems.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2020AAA0106100 and the National Natural Science Foundation of China under Grant 62376087.

References

Bao, Y.; Chang, S.; Yu, M.; and Barzilay, R. 2018. Deriving Machine Attention from Human Rationales. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1903–1913. Brussels, Belgium: Association for Computational Linguistics.

Bastings, J.; Aziz, W.; and Titov, I. 2019. Interpretable Neural Predictions with Differentiable Binary Variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. arXiv:1308.3432.

Camburu, O.-M.; Rocktäschel, T.; Lukasiewicz, T.; and Blunsom, P. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. *Neural Information Processing Systems*.

Chang, S.; Zhang, Y.; Yu, M.; and Jaakkola, T. 2020. Invariant Rationalization. In III, H. D.; and Singh, A., eds., *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1448–1458. PMLR.

Chen, J.; Song, L.; Wainwright, M.; and Jordan, M. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 883–892. PMLR.

Covert, I. C.; Lundberg, S.; and Lee, S.-I. 2021. Explaining by removing: A unified framework for model explanation. *The Journal of Machine Learning Research*, 22(1): 9477–9566.

Danilevsky, M.; Qian, K.; Aharonov, R.; Katsis, Y.; Kawas, B.; and Sen, P. 2020. A Survey of the State of Explainable AI for Natural Language Processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 447–459. Suzhou, China: Association for Computational Linguistics.

Deng, H.; Zou, N.; Du, M.; Chen, W.; Feng, G.; Yang, Z.; Li, Z.; and Zhang, Q. 2023. Understanding and Unifying Fourteen Attribution Methods with Taylor Interactions. arXiv:2303.01506.

Gurrapu, S.; Kulkarni, A.; Huang, L.; Lourentzou, I.; and Batarseh, F. A. 2023. Rationalization for explainable NLP: a survey. *Frontiers in Artificial Intelligence*, 6.

Huang, Y.; Chen, Y.; Du, Y.; and Yang, Z. 2021. Distribution matching for rationalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 13090–13097.

Jacovi, A.; and Goldberg, Y. 2021. Aligning faithful interpretations with their social attribution. *Transactions of the Association for Computational Linguistics*, 9: 294–310.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA.

Kumar, S.; and Talukdar, P. 2020. NILE: Natural Language Inference with Faithful Natural Language Explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8730–8742.

Lei, T.; Barzilay, R.; and Jaakkola, T. 2016. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 107–117. Austin, Texas: Association for Computational Linguistics.

Liu, W.; Wang, H.; Wang, J.; Li, R.; Yue, C.; and Zhang, Y. 2022. FR: Folded Rationalization with a Unified Encoder. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 6954–6966. Curran Associates, Inc.

Lundberg, S. M.; and Lee, S.-I. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

McAuley, J.; Leskovec, J.; and Jurafsky, D. 2012. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, 1020–1025. IEEE.

Paranjape, B.; Joshi, M.; Thickstun, J.; Hajishirzi, H.; and Zettlemoyer, L. 2020. An Information Bottleneck Approach for Controlling Conciseness in Rationale Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1938–1952.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Rajani, N. F.; McCann, B.; Xiong, C.; and Socher, R. 2019. Explain Yourself! Leveraging Language Models for Commonsense Reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4932–4942.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 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*, 1135–1144.

Rudin, C. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5): 206–215.

Sha, L.; Camburu, O.-M.; and Lukasiewicz, T. 2022. Rationalizing Predictions by Adversarial Information Calibration. *Artificial Intelligence*, 103828.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep inside convolutional networks: visualising image classification models and saliency maps. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ICLR.

Sun, X.; Yang, D.; Li, X.; Zhang, T.; Meng, Y.; Qiu, H.; Wang, G.; Hovy, E.; and Li, J. 2021. Interpreting Deep Learning Models in Natural Language Processing: A Review. arXiv:2110.10470.

Wang, H.; Lu, Y.; and Zhai, C. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 783–792.

Williams, R. J. 1992. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning*, 229–256.

Yu, M.; Chang, S.; Zhang, Y.; and Jaakkola, T. 2019. Rethinking Cooperative Rationalization: Introspective Extraction and Complement Control. 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)*, 4094–4103. Hong Kong, China: Association for Computational Linguistics.

Yu, M.; Zhang, Y.; Chang, S.; and Jaakkola, T. 2021. Understanding Interlocking Dynamics of Cooperative Rationalization. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 12822–12835. Curran Associates, Inc.

Yue, L.; Liu, Q.; Du, Y.; An, Y.; Wang, L.; and Chen, E. 2022. DARE: Disentanglement-Augmented Rationale Extraction. *Advances in Neural Information Processing Systems*, 35: 26603–26617.

Zheng, Y.; Booth, S.; Shah, J.; and Zhou, Y. 2022. The Irrationality of Neural Rationale Models. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, 64–73. Seattle, U.S.A.: Association for Computational Linguistics.