Rethinking Cooperative Rationalization: Introspective Extraction and Complement Control

Mo Yu[*] [♡] **Shiyu Chang**[*] [♣♡] **Yang Zhang**[*] [♣♡] **Tommi S. Jaakkola** [♠]
[♡] IBM Research [♣] MIT-IBM Watson AI Lab [♠] MIT
yum@us.ibm.com {shiyu.chang, yang.zhang2}@ibm.com tommi@csail.mit.edu

Abstract

Selective rationalization has become a common mechanism to ensure that predictive models reveal how they use any available features. The selection may be soft or hard, and identifies a subset of input features relevant for prediction. The setup can be viewed as a cooperate game between the selector (aka rationale generator) and the predictor making use of only the selected features. The co-operative setting may, however, be compromised for two reasons. First, the generator typically has no direct access to the outcome it aims to justify, resulting in poor performance. Second, there's typically no control exerted on the information left outside the selection. We revise the overall co-operative framework to address these challenges. We introduce an introspective model which explicitly predicts and incorporates the outcome into the selection process. Moreover, we explicitly control the rationale complement via an adversary so as not to leave any useful information out of the selection. We show that the two complementary mechanisms maintain both high predictive accuracy and lead to comprehensive rationales.[1]

1 Introduction

Rapidly expanding applications of complex neural models also bring forth criteria other than mere performance. For example, medical (Yala et al., 2019) and other high-value decision applications require some means of verifying reasons for the predicted outcomes. This area of self-explaining models in the context of NLP applications has primarily evolved along two parallel tracks. On one hand, we can design neural architectures that expose more intricate mechanisms of reasoning such as module networks (Andreas et al., 2016a,b; Johnson et al., 2017). While important, such approaches may still require adopting specialized designs and architectural choices that do not yet reach accuracies comparable to black-box approaches. On the other hand, we can impose limited architectural constraints in the form of selective rationalization (Lei et al., 2016; Li et al., 2016b; Chen et al., 2018a,b) where the goal is to only expose the portion of the text relevant for prediction. The selection is done by a separately trained model called rationale generator. The resulting text selection can be subsequently used as an input to an unconstrained, complex predictor, *i.e.*, architectures used in the absence of any rationalization.[2] The main challenge of this track is how to properly coordinating the rationale generator with the powerful predictor operating on the selected information during training.

Label: negative **Original Text:** *really cloudy , lots of sediment , washed out yellow color . looks pretty gross , actually , like swamp water . no head , no lacing .* **Rationale from (Lei et al., 2016)**: [*"really cloudy lots", "yellow", "no", "no"*] **Rationale from cooperative introspection model**: [*". looks", "no", "no"*] **Rationale from our full model**: [*"cloudy", "lots", "pretty gross", "no lacing"*]

Table 1: An example of the rationales extracted by different models on the sentiment analysis task of beer reviews (appearance aspect). *Red* words are human-labeled rationales. Details of the experiments can be found in Appendix B.

In this paper, we build on and extend selective rationalization. The selection process can be thought of as a cooperative game between the generator and the predictor operating on the selected,

[*]Authors contributed equally to this work.

[1]The code and data for our method is publicly available at https://github.com/Gorov/three_player_for_emnlp.

[2]Therefore the selective rationalization approach is easier to be adapted to new applications, such as question answering (Yu et al., 2018), image classification (Chen et al., 2018a) and medical image analysis (Yala et al., 2019).

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 4094–4103, Hong Kong, China, November 3–7, 2019.

partial input text. The two players aim for the shared goal of achieving high predictive accuracy, just having to operate within the confines imposed by rationale selection (a small, concise portion of input text). The rationales are learned entirely in an unsupervised manner, without any guidance other than their size, form. An example of ground-truth and learned rationales are given in Table 1.

The key motivation for our work arises from the potential failures of cooperative selection. Since the generator typically has no direct access to the outcome it aims to justify, the learning process may converge to a poorly performing solution. Moreover, since only the selected portion is evaluated for its information value (via the predictor), there is typically no explicit control over the remaining portion of the text left outside the rationale. These two challenges are complementary and should be addressed jointly.

Performance The clues in text classification tasks are typically short phrases (Zaidan et al., 2007). However, diverse textual inputs offer a sea of such clues that may be difficult to disentangle in a manner that generalizes to evaluation data. Indeed, the generator may fail to disentangle the information about the correct label, offering misleading rationales instead. Moreover, as confirmed by the experiments presented in this paper, the collaborative nature of the game may enable the players to select a sub-optimal communication code that does not generalize, rather overfits the training data. Regression tasks considered in prior work typically offer greater feedback for the generator, making it less likely that such communication patterns would arise.

We address these concerns by proposing an *introspective rationale generator*. The key idea is to force the generator to explicitly understand what to generate rationales for. Specifically, we make the label that would be predicted with the full text as an additional input to the generator thereby ensuring better overall performance.

Rationale quality The cooperative game setup does not explicitly control the information left out of the rationale. As a result, it is possible for the rationales to degenerate as in containing only select words without the appropriate context. In fact, the introspective generator proposed above can aggravate this problem. With access to the predicted label as input, the generator and the predictor can find a communication scheme by encoding the predicted label with special word patterns (*e.g.* highlighting "." for positive examples and "," negative ones). Table 1 shows such degenerate cases for the two cooperative methods.

In order to prevent degenerate rationales, we propose a *three-player game* that renders explicit control over the unselected parts. In addition to the generator and the predictor as in conventional cooperative rationale selection schemes, we add a third adversarial player, called the *complement predictor*, to regularize the cooperative communication between the generator and the predictor. The goal of the complement predictor is to predict the correct label using only words left out of the rationale. During training, the generator aims to fool the complement predictor while still maintaining high accuracy for the predictor. This ensures that the selected rationale must contain all/most of the information about the target label, leaving out irrelevant parts, within size constraints imposed on the rationales.

We also theoretically show that the equilibrium of the three-player game guarantees good properties for the extracted rationales. Moreover, we empirically show that (1) the three-player framework on its own helps cooperative games such as (Lei et al., 2016) to improve both predictive accuracy and rationale quality; (2) by combining the two solutions – introspective generator and the three player game – we can achieve high predictive accuracy and non-degenerate rationales.

2 Problem Formulation

This section formally defines the problem of rationalization, and then proposes a set of conditions that desirable rationales should satisfy, which addresses problems of previous cooperative frameworks. Here are some notations throughout this paper. Bolded upper-cased letters, *e.g.* $\boldsymbol{X}$, denote random vectors; unbolded upper-cased letters, *e.g.* X, denote random scalar variables; bolded lower-cased letters, *e.g.* $\boldsymbol{x}$, denote deterministic vectors or vector functions; unbolded lower-cased letters, *e.g.* x, denote deterministic scalars or scalar functions. $p_X(\cdot|Y)$ denotes conditional probability density/mass function conditional on Y. $H(\cdot)$ denotes Shannon entropy. $\mathbb{E}[\cdot]$ denotes expectation.

2.1 Problem Formulation of Rationalization

The target application here is text classification on data tokens in the form of $\{(\boldsymbol{X}, Y)\}$. Denote

$\boldsymbol{X} = \boldsymbol{X}_{1:L}$ as a sequence of words in an input text with length L. Denote Y as a label. Our goal is to generate a rationale, denoted as $\boldsymbol{r}(\boldsymbol{X}) = \boldsymbol{r}_{1:L}(\boldsymbol{X})$, which is a selection of words in $\boldsymbol{X}$ that accounts for Y. Formally, $\boldsymbol{r}(\boldsymbol{X})$ is a hard-masked version of $\boldsymbol{X}$ that takes the following form at each position i:

$$\boldsymbol{r}_i(\boldsymbol{X}) = \boldsymbol{z}_i(\boldsymbol{X}) \cdot \boldsymbol{X}_i, \tag{1}$$

where $\boldsymbol{z}_i \in \{0,1\}^N$ is the binary mask. Many previous works (Lei et al., 2016; Chen et al., 2018a) follows the above definition of rationales. In this work, we further define the complement of rationale, denoted as $\boldsymbol{r}^c(\boldsymbol{X})$, as

$$\boldsymbol{r}^c_i(\boldsymbol{X}) = (1 - \boldsymbol{z}_i(\boldsymbol{X})) \cdot \boldsymbol{X}_i. \tag{2}$$

For notational ease, define

$$\boldsymbol{R} = \boldsymbol{r}(\boldsymbol{X}), \quad \boldsymbol{R}^c = \boldsymbol{r}^c(\boldsymbol{X}), \quad \boldsymbol{Z} = \boldsymbol{z}(\boldsymbol{X}). \tag{3}$$

2.2 Rationale Conditions

An ideal rationale should satisfy the following conditions.

Sufficiency: $\boldsymbol{R}$ is sufficient to predict Y, *i.e.*

$$p_Y(\cdot|\boldsymbol{R}) = p_Y(\cdot|\boldsymbol{X}). \tag{4}$$

Comprehensiveness: $\boldsymbol{R}^c$ does not contain sufficient information to predict Y, *i.e.*

$$H(Y|\boldsymbol{R}^c) \geq H(Y|\boldsymbol{R}) + h, \tag{5}$$

for some constant h.

Compactness: the segments in $\boldsymbol{X}$ that are included in $\boldsymbol{R}$ should be sparse and consecutive, *i.e.,*

$$\sum_i \boldsymbol{Z}_i \leq s, \quad \sum_i |\boldsymbol{Z}_i - \boldsymbol{Z}_{i-1}| \leq c, \tag{6}$$

for some constants s and c.

Here is an explanation of what each of these conditions means. The sufficiency condition is the core one of a legitimate rationale, which essentially stipulates that the rationale maintains the same predictive power as $\boldsymbol{X}$ to predict Y. The compactness condition stipulates that the rationale should be continuous and should not contain more words than necessary. For example, without the compactness condition, a trivial solution to Eq. (4) would be $\boldsymbol{X}$ itself. The first inequality in Eq. (6) constrains the sparsity of rationale, and the second one constrains the continuity. The comprehensiveness condition requires some elaboration, which we will discuss in the next subsection.

2.3 Comprehensiveness and Degeneration

There are two justifications of the comprehensiveness condition. First, it regulates the information outside the rationale, so that the rationale contains all the relevant and useful information, hence the name comprehensiveness. Second and more importantly, we will show there exists a failure case, called degeneration, which can only be prevented by the comprehensiveness condition.

Degeneration refers to the situation where, rather than finding words in $\boldsymbol{X}$ that explains Y, $\boldsymbol{R}$ attempts to encode the probability of Y using trivial information, *e.g.* punctuation and position. Consider the following toy example of binary classification ($Y \in \{0,1\}$), where $\boldsymbol{X}$ can always perfectly predict Y. Then the following rationale satisfies the sufficiency and compactness: $\boldsymbol{R}$ includes only the first word of $\boldsymbol{X}$ when $Y = 0$, and only the last word when $Y = 1$. It is obvious that this $\boldsymbol{R}$ is sufficient to predict $\boldsymbol{Y}$ (by looking at if the first word or the last word is chosen), and thus satisfies the sufficiency condition. Apparently this $\boldsymbol{R}$ is perfectly compact (only one word). However, this rationale does not provide a valid explanation.

Theoretically, any previous cooperative framework may suffer from the above ill-defined problem, if the generator has the potential to accurately guess Y with sufficient capacity.[3] This problem happens because they have no control of the words unselected by $\boldsymbol{R}$. Intuitively, in the presence of degeneration, some key predictors in $\boldsymbol{X}$ will be left unselected by $\boldsymbol{R}$. Thus by looking at the predictive power of $\boldsymbol{R}^c$, we can determine if degeneration occurs. Specifically, when degeneration is present, all the useful information is left unselected by $\boldsymbol{R}$, and so $H(Y|\boldsymbol{R}^c)$ is low. That is why the lower bound in Eq. (5) rules out the degeneration cases.

3 The Proposed Three-Player Models

This section introduces our new rationalization solutions, which is theoretically guaranteed to be able to find the rationales that satisfy Eqs. (4-6).

3.1 The Basic Three-Player Model

This section introduces our basic three-player model. The model consists of three players: a **rationale generator** that generates the rationale $\boldsymbol{R}$ and its complement $\boldsymbol{R}^c$ from text, a **predictor** that predicts the probability of Y based on $\boldsymbol{R}$, a **com-**

[3] We show such cases of (Lei et al., 2016) in Appendix B.

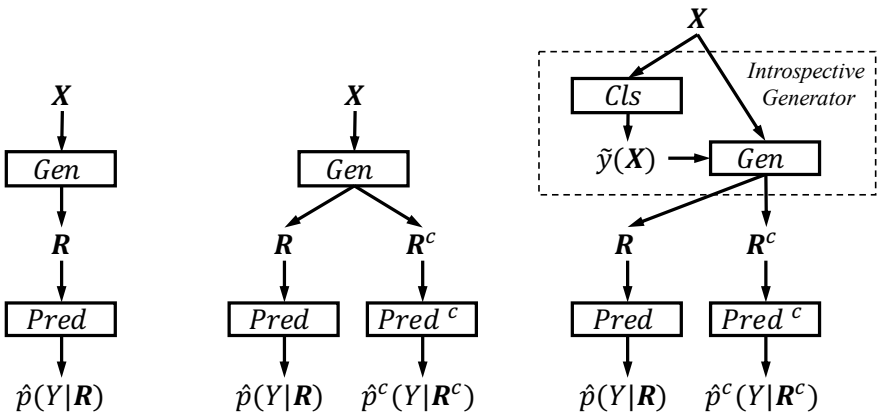

Figure 1: Illustration of different rationalization frameworks. (a) The cooperative framework from (Lei et al., 2016), which consists of two players, *i.e.,* a generator (*Gen*) and a predictor (*Pred*). (b) A straightforward extension of the model (a) with a compliment predictor (*Pred* ^{c}). The generator plays a cooperative game with the predictor and plays a mini-max game with the compliment predictor. (c) The introspective three-player framework. The introspective module first predicts the possible outcome $\tilde{y}$ based on the full texts x and then generate rationales using both x and $\tilde{y}$. The third framework (c) is a special case of (b). Such an inductive bias in model design preserves the predictive performance.

plement predictor that predicts the probability of Y based on $\boldsymbol{R}^c$.

Figure 1(b) illustrates the basic three-player model. Compared with (Lei et al., 2016), as shown in Figure 1(a), the three-player model introduces an additional complement predictor, which plays a minimax game in addition to the cooperative game in (Lei et al., 2016). For clarity, we will describe the game backward, starting from the two predictors followed by the generator.

Predictors: The predictor estimates probability of Y conditioned on $\boldsymbol{R}$, denoted as $\hat{p}(Y|\boldsymbol{R})$. The complement predictor estimates probability of Y conditioned on $\boldsymbol{R}^c$, denoted as $\hat{p}^c(Y|\boldsymbol{R})$. Both predictors are trained using the cross entropy loss, *i.e.*

$$\begin{aligned}\mathcal{L}_p &= \min_{\hat{p}(\cdot,\cdot)} -H(p(Y|\boldsymbol{R}); \hat{p}(Y|\boldsymbol{R})) \\ \mathcal{L}_c &= \min_{\hat{p}^c(\cdot,\cdot)} -H(p(Y|\boldsymbol{R}^c); \hat{p}^c(Y|\boldsymbol{R}^c)),\end{aligned} \tag{7}$$

where $H(p;q)$ denotes the cross entropy between p and q. $p(\cdot|\cdot)$ denotes the empirical distribution. It is worth emphasizing that $\mathcal{L}_p$ and $\mathcal{L}_c$ are both functions of the generator.

Generator: The generator extracts $\boldsymbol{R}$ and $\boldsymbol{R}^c$ by generating the rationale mask, $\boldsymbol{z}(\cdot)$, as shown in Eqs. (1-2). Specifically, $\boldsymbol{z}(\cdot)$ is determined by minimizing the weighted combination of four losses:

$$\min_{\boldsymbol{z}(\cdot)} \mathcal{L}_p + \lambda_g \mathcal{L}_g + \lambda_s \mathcal{L}_s + \lambda_c \mathcal{L}_c, \tag{8}$$

where $\mathcal{L}_g$ encourages the gap between $\mathcal{L}_p$ and $\mathcal{L}_c$ to be large, *i.e.*

$$\mathcal{L}_g = \max\{\mathcal{L}_p - \mathcal{L}_c + h, 0\}. \tag{9}$$

It stipulates the comprehensiveness property of the rationale (Eq. (5)). Intuitively, if the complement rationale is less informative of Y than the rationale, then $\mathcal{L}_c$ should be larger than $\mathcal{L}_p$.

$\mathcal{L}_s$ and $\mathcal{L}_c$ impose the sparsity and continuity respectively, which correspond to Eq. (6):

$$\begin{aligned}\mathcal{L}_s &= \max\Big\{\sum_i \boldsymbol{Z}_i - s, 0\Big\}, \\ \mathcal{L}_c &= \sum_i \max\{||\boldsymbol{Z}_i - \boldsymbol{Z}_{i-1}|| - c, 0\}.\end{aligned} \tag{10}$$

From Eq. (7), we can see that the generator plays a *cooperative game* with the predictor, because both tries to maximize the predictive performance of $\boldsymbol{R}$. On the other hand, the generator plays an *adversarial game* with the complement predictor, because the latter tries to maximize the predictive performance of $\boldsymbol{R}^c$, but the former tries to reduce it. Without the complement predictor, and thus the loss $\mathcal{L}_g$, the framework reduces to the method in (Lei et al., 2016).

Training During training, the three players perform gradient descent steps with respect to their own losses. For the generator, Since $\boldsymbol{z}(\boldsymbol{X})$ is a set of binary variables, we cannot apply the regular gradient descent algorithm. Instead we will use policy gradient (Williams, 1992) to optimize the models. We maximize the reward that is defined as the negative loss in Eq. (8). In order to have bounded rewards for training stability, the negative losses $\mathcal{L}_p$ and $\mathcal{L}_c$ are replaced with accuracy.

Theoretical guarantees The proposed framework is able to obtain a rationale that simultane-

ously satisfies the conditions in Eqs. (4) to (6), as stated in the following theorem:

Theorem 1. *A rationalization scheme $\boldsymbol{z}(\boldsymbol{X})$ that simultaneously satisfies Eqs.* (4)-(6) *is the global optimizer of Eq.* (8)*.*

The proof is given in Appendix A. The basic idea is that there is a correspondence between each term in Eq. (8) and each of the properties Eqs. (4)-(6). The minimization of each loss term is equivalent to satisfying the corresponding property.

3.2 The Introspection Generator

As discussed in Section 1, in the existing generator-predictor framework, the generator may fail to disentangle the information about the correct label, offering misleading rationales instead. To address this problem, we propose a new generator module, called the **Introspective Generator**, which explicitly predicts the label before making rationale selections.

Figure 1(c) illustrates the model with the introspection generator. Specifically, the improved generator still fits into the basic three-player framework in Section 3.1. The only difference is in how the generator generates the mask $\boldsymbol{z}(\boldsymbol{X})$, which now breaks down into two steps.

First, the module uses a *regular classifier* that takes the input $\boldsymbol{X}$ and predicts the label, denoted $\tilde{y}(\boldsymbol{X})$. For classification tasks, we use the maximum likelihood estimate, *i.e.*

$$\tilde{y}(\boldsymbol{X}) = \underset{y}{\operatorname{argmax}}\ \tilde{p}(Y = y|\boldsymbol{X}), \tag{11}$$

where $\tilde{p}(Y = y|\boldsymbol{X})$ is the predicted probability by maximizing the cross entropy, which is pretrained.

Second, a label-aware rationale generator generates the binary mask of the rationales, *i.e.*

$$z(\boldsymbol{X}) = \tilde{z}(\boldsymbol{X}, \tilde{y}(\boldsymbol{X})). \tag{12}$$

Note that $\tilde{y}$ is a function of $\boldsymbol{X}$, so the entire introspective generator is essentially a function of $\boldsymbol{X}$. In this way, all the formulations in Section 3.1 and the Theorem 1 still hold for the three-player game with the introspective generator.

In the implementation, the classifier can use the same architecture like the predictor and the complement predictor. The generator is of the same architecture in Section 3.1, but with the additional input of $\hat{y}(\boldsymbol{X})$.

Remark on degeneration Obviously, when working in a cooperative game, the introspection generator will make the degeneration problem more severe: when the classifier $\tilde{p}(\cdot|\boldsymbol{X})$ becomes sufficiently accurate during training, the generator only needs to encode the information of $\tilde{y}$ into $\boldsymbol{R}$. Therefore our three-player game, while improving over any existing generator-predictor frameworks on its own, is critical for the introspective model.

4 Experimental Settings

4.1 Datasets

We construct three text classification tasks, including two sentiment classification tasks from the BeerAdvocate review dataset (McAuley et al., 2012)[4], and a more complex relation classification task from SemEval 2010 Task 8 (Hendrickx et al., 2009). Table 2 gives examples of the above tasks. Finally, as suggested by an anonymous reviewer, we evaluate on the text matching benchmark AskUbuntu, following Lei et al. (2016). The experimental setting and results are reported in Appendix F.

Multi-aspect beer review This is the same data used in (Lei et al., 2016). Each review evaluates multiple aspects of a kind of beer, including appearance, smell, palate, and an overall score. For each aspect, a rating $\in$ [0,1] is labeled. We limit ourselves to the appearance aspect only and use a threshold of 0.6 to create balanced binary classification tasks for each aspect. Then, the task becomes to predict the appearance aspect of a beer based on multi-aspect text inputs. The advantage of this dataset is that it enables automatic evaluation of rationale extraction. The dataset provides sentence-level annotations on about 1,000 reviews, where each sentence is labeled by the aspect it covers. Note that in this dataset, each aspect is often described by a single sentence with clear polarity. Thus, a generator can select a sentence based on the topic distribution of words. The selected sentence often has very high overlap with the ground truth annotations and also contains sufficient information for predicting the sentiment. This characteristic of the dataset makes it a relatively easy task, and thus we further consider two more challenging tasks.

Single-aspect beer review To construct a more challenging task, for each review, we extract the

[4]http://snap.stanford.edu/data/web-BeerAdvocate.html

Task	Label	Input Texts
Multi-Aspect Beer Review	positive (regarding the aspect of *appearance*)	***Clear, burnished copper-brown topped by a large beige head that displays impressive persistance and leaves a small to moderate amount of lace in sheets when it eventually departs***. *The nose is sweet and spicy and the flavor is malty sweet, accented nicely by honey and by abundant caramel/toffee notes. There's some spiciness (especially cinnamon), but it's not overdone. The finish contains a moderate amount of bitterness and a trace of alcohol. The mouthfeel is exemplary* $\cdots$
Single-Aspect Beer Review	positive	*appearance : dark-brown/black color with a* ***huge tan head*** *that gradually collapses , leaving* ***thick lacing*** .
Relation Classification	Message-Topic(e1,e2)	*It was a friendly* ***call***$_{e_1}$ ***to remind*** *them* ***about the bill***$_{e_2}$ *and make sure they have a copy of the invoice*

Table 2: Example of the three tasks used for evaluation. The ***red bold*** words are sample rationales (created by the authors for illustration purpose only for the last two tasks).

sentences that are specifically about the appearance aspect from the aforementioned BeerAdvocate dataset[5], and the task is to predict the sentiment of the appearance only on the extracted sentences. The details of the dataset construction can be found in Appendix C. This new dataset is obviously more challenging in terms of generating meaningful rationales. This is because the generator is required to select more find-grained rationales. Since there are no rationale annotations, we rely on subjective evaluations to test the quality of the extracted rationales.

Multi-class relation classification To show the generalizability of our proposed approaches to other NLP applications, we further consider the SemEval 2010 Task 8 dataset (Hendrickx et al., 2009). Given two target entities e_1 and e_2 in a sentence, the goal is to classify the relation type (with directions) between the two entities (including None if there is no relation). Similar to the single-aspect beer review, it is also a fine-grained rationalization task. The major difference is that the number of class labels in this task is much larger, and we hope to investigate its effects on degeneration and performance downgrade.

4.2 Implementation Details

For both the generators and the two predictors, we use bidirectional LSTMs with hidden dimension 400. In the introspection generator, the classifier consists of the same bidirectional LSTM, and $\boldsymbol{z}(\boldsymbol{X}, \tilde{y})$ is implemented as an LSTM sequential labeler with the label $\tilde{y}$ transformed to an embedding vector that serves as the initial hidden states of the

[5]We will release the single-aspect dataset.

Model	10% Highlighting Acc	10% Highlighting Prec	20% Highlighting Acc	20% Highlighting Rec
Random	65.70	17.75	72.31	19.84
Lei2016	82.05	86.14	83.88	79.98
+minimax	83.45	86.54	84.25	85.16
Intros	87.10	68.37	87.50	59.63
+minimax	86.16	85.67	86.86	79.40

Table 3: Main results of binary classification on multi-aspect beer reviews. The *Acc* column shows the accuracy of prediction. The *Prec* and *Rec* are precision and recall on the extracted rationales compared to the human annotations. The accuracy of sentiment prediction with the full texts is **87.59**.

LSTM. For the relation classification task, since the model needs to be aware of the two target entities, we add the relative position features following Nguyen and Grishman (2015). We map the relative position features to learnable embedding vectors and concatenate them with word embeddings as the inputs to the LSTM encoder of each player. All hyper-parameters are tuned on the development sets according to predictive accuracy. In other words, all the models are tuned **without** seeing any rationale annotations.

5 Experiments

5.1 Multi-Aspect Beer Review

Table 3 summarizes the main results on the multi-aspect beer review task. In this task, a desirable rationale should be both appearance-related and sufficient for sentiment predictions. Since the average sparsity level of human-annotated rationales is about 18%, we consider the following evaluation settings. Specifically, we compare the generated rationales to human annotations by measuring the precision when extracting 10% of words and the

recall for 20%[6]. In addition to the precision/recall, we also report the predictive accuracy of the extracted rationales on predicting the sentiment.

When only 10% of the words are used, Lei et al. (2016) has a significant performance downgrade compared to the accuracy when using the whole passage (82.05 v.s. 87.59). With the additional third player added, the accuracy is slightly improved, which validates that controlling the unselected words improves the robustness of rationales. On the other hand, our introspection models are able to maintain higher predictive accuracy (86.16 v.s. 82.05) compared to (Lei et al., 2016), while only sacrificing a little loss on highlighting precision (0.47% drop).

Similar observations are made when 20% of the words required to highlight with one exception. Comparing the model of (Lei et al., 2016) with and without the proposed mini-max module, there is a huge gap of more than 5% on recall of generated rationales. This confirms the motivation that the original cooperative game tends to generate less comprehensive rationales, where the three-player framework controls the unselected words to be less informative so the recall is significantly improved.

It is worth mentioning that when a classifier is trained with randomly highlighted rationales (*i.e.* random dropout (Srivastava et al., 2014) on the inputs), it performs significantly worse on both predictive accuracy and highlighting qualities. This confirms that extracting concise and sufficient rationales is not a trivial task. Moreover, our re-implementation of (Lei et al., 2016) for the original regression task achieves 90.1% precision when highlighting 10% words, which suggest that rationalization for binary classification is more challenging compared to the regression where finer supervision is available.

In summary, our three-player framework consistently improves the quality of extracted rationales on both of the original (Lei et al., 2016) and the introspective framework. Particularly, without controlling the unselected words, the introspection model experiences a serious degeneration problem as expected. With the three-player framework, it manages to maintain both high predictive accuracy and high quality of rationalization.

[6]Previous work in (Lei et al., 2016) only evaluates the precision for the 10% extraction.

Model	Single-Aspect Beer		Relation	
	Acc	Acc^c	Acc	Acc^c
All (100%)	87.1	51.3	77.8	4.9
Random	74.3	83.1	51.8	64.9
Lei2016	81.4	74.6	73.5	36.8
+minimax	82.9	73.2	75.0	30.9
Intros	87.0	85.8	75.2	56.6
+minimax	85.3	73.6	76.2	29.1

Table 4: Predictive accuracy on the single-aspect beer appearance review (left) and the relation classification (right). Acc^c refers to the accuracy of the complement predictor. We restrict the extracted rationales to be on average eight words and four continuous pieces for the beer review and six words and three pieces for relation classification. A desired rationalization method will have high Acc and low Acc^c.

5.2 Single-Aspect Beer Review

In this section, we evaluate the proposed methods on the more challenging single-aspect beer dataset. Similar to previous experiments, we force all the methods to have comparable highlighting ratio and continuity constraints for fair evaluation. The highlighting ratio is determined from human estimation on a small set of data. We report two classification results, which are the accuracy of the predictor and complement predictor. For both cooperative methods, *i.e.* (Lei et al., 2016) with and without introspection , we train an independent extra predictor on the unselected words from the generator, which does not affect the training of the generator-predictor framework.

From the left part of Table 4, we observe that the original model of (Lei et al., 2016) has a hard time maintaining the accuracy compared to the classifier trained with full texts. Transforming it into a three-player game helps to improve the performance of the evaluation set while lower the accuracy of the complement predictor. Since the extracted rationales are in a similar length, these results suggest that learning with a three-player game successfully enforces the generator not to leave informative words unselected. Similar to the multi-aspect results, the cooperative introspection model suffers from the degeneration problem, which produces a high complement accuracy. The three-player game yields a more comprehensive rationalization with small losses in accuracy.

Human evaluation We further conduct subjective evaluations by comparing the original model of (Lei et al., 2016) with our introspective three-player model. We mask the extracted rationales and present the unselected words only to the hu-

Model	%UNK	Acc	$\text{Acc}_{\text{w/o UNK}}$
Lei2016	43.5	63.5	69.0
Intros+minimax	**54.0**†	**58.0**	**66.3**

Table 5: Subjective evaluations on the task of controlling the unselected rationale words. Acc denotes the accuracy in guessing sentiment labels. $\text{Acc}_{\text{w/o UNK}}$ denotes the sentiment accuracy for these samples that are not selected as "UNK" for the secondary task. † denotes p-value < 0.005 in t-test. A desired rationalization method achieves high "UNK" rate and performance randomly for the Acc predictions.

man evaluators. The evaluators are asked to predict the sentiment label as their first task. If a rationalizing method successfully includes all informative pieces in the rationale, subjects should have around 50% of accuracy in guessing the label. In addition, after providing the sentiment label, subjects are then asked to answer a secondary question, which is whether the provided text spans are sufficient for them to predict the sentiment. If they believe there are not enough clues and their sentiment classification is based on a random guess, they are instructed to select unknown (denoted as "UNK") as the answer to the second question. Appendix E elaborates why we design subjective evaluations in such a way in more details.

Table 5 shows the performance of subjective evaluations. Looking at the first column of the table, our model is better in confusing human, which gives a higher rate in selecting "UNK". It confirms that the three-player introspective model selects more comprehensive rationales and leave less informative texts unattended. Furthermore, the results also show that human evaluators offer worse sentiment predictions on the proposed approach, which is also desired and expected.

5.3 Relation Classification

The predictive performances on the relation classification task are shown in the right part of the Table 4. We observe consistent results as in previous datasets. Clearly, the introspective generator helps the accuracy and the three-player game regularize the complement of the rationale selections.

Examples of the extracted rationales For relation classification, it is difficult to conduct subjective evaluations because the task requires people to have sufficient knowledge of the schema of relation annotation. To further demonstrate the quality of generated rationales, we provide some illustrative examples. Since there is a rich form of supervised signal, *i.e.,* the number of class labels is large, the chance of any visible degeneration of the Lei et al. (2016)'s model should be low. However, we still spot quite a few cases. In the first example, Lei et al. (2016) fails to highlight the second entity while ours does. In the second example, the introspective three-player model selects more words than (Lei et al., 2016). In this case, the two entities themselves suffice to serve as the rationales. However, our model preserves the words like "working". This problem might due to the bias of the dataset. For example, some words that are not relevant to the target entities may still correlate with the labels. In the case, our model will pick these words as a part of the rationale.

Original Text: *of the hundreds of strains of avian influenza a* ***viruses***$_{e_1}$ *, only four have caused human* ***infections***$_{e_2}$ *: h5n1 , h7n3 , h7n7 , and h9n2* **Label:** *Cause-Effect*(e_1,e_2)
Lei et al. (2016): [***viruses***$_{e_1}$ *, only four have caused*] **Our Intros+minimax**: [***viruses***$_{e_1}$], [*four have caused human* ***infections***$_{e_2}$]
Original Text: *i spent a year working for a* ***software***$_{e_1}$ ***company***$_{e_2}$ *to pay off my college loans* **Label:** *Product-Producer*(e_1,e_2)
Lei et al. (2016): [*a* ***software***$_{e_1}$ ***company***$_{e_2}$ *to*] **Our Intros+minimax**: [*working for a* ***software***$_{e_1}$ ***company***$_{e_2}$], [*loans*]

Table 6: Illustrative examples of generated rationales on the relation classification task. Entities are shown in ***bold***.

6 Related Work

Model interpretability Besides the two major categories of self-explaining models discussed in Section 1, model interpretability is widely studied in the general machine learning field. For example, evaluating feature importance with gradient information (Simonyan et al., 2013; Li et al., 2016a; Sundararajan et al., 2017) or local perturbations (Kononenko et al., 2010; Lundberg and Lee, 2017); and interpreting deep networks by locally fitting interpretable models (Ribeiro et al., 2016; Alvarez-Melis and Jaakkola, 2018).

Besides selective rationalization, the cooperative game has been studied in the latter direction above (Lee et al., 2018). It has also been applied to a relevant problem on summarization (Arumae and Liu, 2019), where the selected summary should be sufficient for answering questions related to the document. In this problem, the sum-

mary is a special type of rationale of a document. Another related concurrent work (Bastings et al., 2019) proposes differentiable solution to optimize the cooperative rationalization method.

Game-theoretical methods Though not having been explored much for self-explaining models, the minimax game setup has been widely used in many machine learning problems, such as self-playing for chess games (Silver et al., 2017), generative models (Goodfellow et al., 2014) and many tasks that can be formulated as multi-agent reinforcement learning (Busoniu et al., 2006). Our three-player game framework also shares a similar idea with (Zhang et al., 2017; Zhao et al., 2017), which aim to learn domain-invariant representations with both cooperative and minimax games.

7 Conclusion

We proposed a novel framework for improving the predictive accuracy and comprehensiveness of the selective rationalization methods. This framework (1) addresses the degeneration problem in previous cooperative frameworks by regularizing the unselected words via a three-player game; and (2) augments the conventional generator with introspection, which can better maintain the performance for down-stream tasks. Experiments with both automatic evaluation and subjective studies confirm the advantage of our proposed framework.

References

David Alvarez-Melis and Tommi S Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. *arXiv preprint arXiv:1806.07538*.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016a. Learning to compose neural networks for question answering. In *Proceedings of NAACL-HLT*, pages 1545–1554.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016b. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

Kristjan Arumae and Fei Liu. 2019. Guiding extractive summarization with question-answering rewards. 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 2566–2577.

Joost Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. *arXiv preprint arXiv:1905.08160*.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. 2006. Multi-agent reinforcement learning: A survey. In *2006 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–6. IEEE.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018a. Learning to explain: An information-theoretic perspective on model interpretation. In *International Conference on Machine Learning*, pages 882–891.

Jianbo Chen, Le Song, Martin J Wainwright, and Michael I Jordan. 2018b. L-Shapley and C-Shapley: Efficient model interpretation for structured data. *arXiv preprint arXiv:1808.02610*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 94–99. Association for Computational Linguistics.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Inferring and executing programs for visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2989–2998.

Igor Kononenko et al. 2010. An efficient explanation of individual classifications using game theory. *Journal of Machine Learning Research*, 11(Jan):1–18.

Guang-He Lee, David Alvarez-Melis, and Tommi S Jaakkola. 2018. Game-theoretic interpretability for temporal modeling. *arXiv preprint arXiv:1807.00130*.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016a. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691.

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

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.

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

Thien Huu Nguyen and Ralph Grishman. 2015. Combining neural networks and log-linear models to improve relation extraction. *arXiv preprint arXiv:1511.05926*.

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.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815*.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

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. JMLR. org.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*.

Adam Yala, Constance Lehman, Tal Schuster, Tally Portnoi, and Regina Barzilay. 2019. A deep learning mammography-based model for improved breast cancer risk prediction. *Radiology*, page 182716.

Mo Yu, Shiyu Chang, and Tommi S Jaakkola. 2018. Learning corresponded rationales for text matching.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using annotator rationales to improve machine learning for text categorization. In *Human language technologies 2007: The conference of the North American chapter of the association for computational linguistics; proceedings of the main conference*, pages 260–267.

Yuan Zhang, Regina Barzilay, and Tommi Jaakkola. 2017. Aspect-augmented adversarial networks for domain adaptation. *Transactions of the Association for Computational Linguistics*, 5:515–528.

Mingmin Zhao, Shichao Yue, Dina Katabi, Tommi S Jaakkola, and Matt T Bianchi. 2017. Learning sleep stages from radio signals: A conditional adversarial architecture. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4100–4109. JMLR. org.