A Proof of Theorem 1

This appendix provides the proof of theorem 1. First, we will need the following lemma.

Lemma 1.1. *If the predictors have sufficient representation power, we have*

$$\mathcal{L}_p = -H(Y|\boldsymbol{R}), \quad \mathcal{L}_c = -H(Y|\boldsymbol{R}^c). \tag{13}$$

The proof is trivial, noticing cross entropy is upper bounded by entropy.

The following lemmas show that there is a correspondence between the rationale properties in Eqs. (4) and loss terms in Eq. (8).

Lemma 1.2. *A rationalization scheme $\boldsymbol{z}(\boldsymbol{X})$ that satisfies Eq. (4) is the global minimizer of $\mathcal{L}_p$ as defined in Eq. (7).*

Proof. Notice that

$$\mathcal{L}_p = -H(Y|\boldsymbol{R}) = -H(Y|\boldsymbol{r}(\boldsymbol{X})) \geq -H(Y|\boldsymbol{X}). \tag{14}$$

The first equality is given by Lemma 1.1; the second equality is given by Eq. (3). For the inequality, the equality holds if and only if

$$p_Y(\cdot|\boldsymbol{r}(\boldsymbol{X})) = p_Y(\cdot|\boldsymbol{X}), \tag{15}$$

which is Eq. (4). □

Lemma 1.3. *A rationalization scheme $\boldsymbol{z}(\boldsymbol{X})$ that satisfies Eq. (5) is the global minimizer of $\mathcal{L}_g$ as defined in Eq. (9).*

Proof. According to Lemma 1.1, $\mathcal{L}_g$ can be rewritten as

$$\mathcal{L}_g = \max\{H(Y|\boldsymbol{R}) - H(Y|\boldsymbol{R}^c) + h, 0\}, \tag{16}$$

which equals 0 if and only if Eq. (5) holds. □

Lemma 1.4. *A rationalization scheme $\boldsymbol{z}(\boldsymbol{X})$ that satisfies Eq. (6) is the global minimizer of $\mathcal{L}_s$ and $\mathcal{L}_c$ as defined in Eq. (10).*

Proof. The proof is obvious. $\mathcal{L}_s$ and $\mathcal{L}_c$ is 0 if and only if Eq. (6) holds. □

Combining Lemmas 1.2 to 1.4 completes the proof of Theorem 1.

B Experimental Setup of Examples in Table 1 and Degeneration Cases of (Lei et al., 2016)

This section provides the details to obtain the results in Table 1 in the introduction section, where the method of (Lei et al., 2016) generates degenerated rationales.

The method of (Lei et al., 2016) works well in many applications. However, as discussed in Section 1 and 2.2, all the cooperative rationalization approaches may suffer from the problem of degeneration. In this section, we design an experiment to confirm the existence of the problem in the original (Lei et al., 2016) model. We use the same single-beer review constructed from (McAuley et al., 2012), as will be described in Appendix C. Instead of constructing a balanced binary classification task, we set the samples with scores higher than 0.5 as positive examples. On such a task, the prediction model with full inputs achieves 82.3% accuracy on the development set.

During the training of (Lei et al., 2016), we stipulate that the generated rationales are very concise: we punish it when the rationales have more than 3 pieces or more than 20% of the words are generated (both with hinge losses). From the results, we can see that Lei et al. (2016) tends to predict color words, like *dark-brown*, *yellow*, as rationales. This is a clue of degeneration, since most of the appearance reviews start with describing colors. Therefore a degenerated generator can learn to split the vocabulary of colors, and communicate with the predictor by using some of the colors for the positive label and some others for the negative label. Such a learned generator also fails to generalize well, given the significant performance decrease (76.4% v.s. 82.3%). By comparison, our method with three-player game could achieve both higher accuracy and more meaningful rationales.

C Data Construction of the Single-Aspect Beer Reviews

This section describes how we construct the single-aspect review task from the multi-aspect beer review dataset (McAuley et al., 2012).

In many multi-aspect beer reviews, we can see clear patterns indicating the aspect of the following sentences. For example, the sentences starting with “*appearance:*” or “*a:*” are likely to be a review on the appearance aspect; and the sentences

Datasets	# Classes	Vocab Size	# Train	# Dev	# Annotation/Test
Multi-aspect sentiment classification	2	110,985	80,000	10,000	994
Single-aspect sentiment classification	2	12,043	12,000	1,362	1,695
Relation classification	19	23,446	7,000	1,000	2,717

Table 7: Statistics of the datasets used in this paper.

Original Text (positive): *dark-brown/black color with a huge tan head that gradually collapses , leaving thick lacing .*
Rationale from (Lei et al., 2016) (Acc: 76.4%): [*"dark-brown/black color"*] **Rationale from our method** (Acc: 80.4%): [*"huge tan", "thick lacing"*]
Original Text (negative): *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) (Acc: 76.4%): [*"really cloudy lots", "yellow", "no", "no"*] **Rationale from our method** (Acc: 80.4%): [*"cloudy", "lots", "pretty gross", "no lacing"*]

Table 8: An example showing rationales extracted by different models, where (Lei et al., 2016) gives degenerated result.

starting with "*smell:*" or "*nose:*" are likely to be about the aroma aspect.

We then extract all the "*X:*" patterns, and count the frequencies of such patterns, where each X is a word. The patterns "*X:*" with a higher than 400 frequency are kept as **anchor patterns**. The sentences between two anchor patterns "X_1: $\cdots$ X_2" are very likely the review regarding the aspect of X_1. Finally, we extract such review sentences after "*appearance:*" or "*a:*" and before the immediate subsequent anchor patterns as the single-aspect review for the appearance aspect. Each of such instances is regarded as a new single-aspect review. The score of the appearance aspect of the original multi-aspect review is regarded as the score of this new review.

With such an automatically constructed dataset, we form our balanced single-review binary classification tasks (see Section 4.1 and Appendix D), on which our base predictor model (with all the words as inputs) performs an 87.1% on the development set. This is as high as the number we achieved on the multi-aspect task regarding the same aspect (87.6%). This result indicates that the noise introduced by our data construction method is insignificant.

D Data Statistics

Table 7 summarizes the statistics of the three datasets used in the experiments. The single-aspect sentiment classification and the relation classification have randomly held-out development sets from the original training sets.

E Experiment Designs for Human Study

This section explains how we designed the human study.

The goal is to evaluate the unpredictable rates of the input texts after the rationales are removed. To this end, we mask the original texts with the rationales generated by (Lei et al., 2016) and our method. Each rationale word is masked with the symbol '*'. The masked texts from different methods are mixed and shuffled so the evaluators cannot know from which systems an input was generated.

We have two human evaluators who are not the authors of the paper. During evaluation, an evaluator is presented with one masked text and asked to try her/his best to predict the sentiment label of it. If a rationalizing method successfully includes all informative pieces in the rationale, subjects should have around 50% of accuracy in guessing the label.

After the evaluator provides a sentiment label, the subjects are asked to answer the second question about whether the provided text spans are sufficient for them to predict the sentiment. If they believe there are no enough clues and their sentiment classification is based on a random guess, they are instructed to input a *UNK* label as the answer to the second question.

The reason we ask the evaluators to provide predicted labels first is based on the following idea: if the task is directly annotating whether the masked texts are unpredictable, the annotators will tend to label more *UNK* labels to save time. Therefore the ratios of *UNK* labels will be biased. Our experimental design alleviates this problem since the evaluators are always required to try the best to guess the labels first. Therefore they will spend more time thinking about the possible labels, instead of immediately putting a *UNK* label.

On a small subset of 50 examples, the inter-annotator agreement is 76% on the *UNK* labels.

F Additional Experiments on AskUbuntu

Setting Following the suggestion from the reviews, we evaluate the proposed method on the question retrieval task on AskUbuntu (Lei et al., 2016). AskUbuntu is a non-factoid question retrieval benchmark. The goal is to retrieve the most relevant questions from an input question. We use the same data split provided by (Lei et al., 2016).[7]

Specifically, each question consists of two parts, the *question title* and the *question body*. The former summarizes a problem from using Ubuntu, while the latter contains the detailed descriptions. In our experiments, we follow the same setting from (Lei et al., 2016) by only using the question bodies. Different from their work, we do not pre-train an encoder by predicting a question title using the corresponding question body. This is because, the question title can be considered as the rationale of its question body, which might result in potential information leaks to our main rationalization task.

Method We formulate the problem of question retrieval as the pairwise classification task. Given two questions (*i.e.*, a query and a candidate question), we aim to classify them as a positive label if they are relevant and vice-versa. We consider the same generator architecture as used in Section 5 in a siamese setting to extract rationales from questions. The predictor and the complement predictor make the prediction based on the pairwise selected spans. We believe it is the most straightforward way to adapt the proposed framework to the AskUbuntu task. There could be sophisticated and task-specific rationalization approaches to improve the performance on AskUbuntu (*e.g.*, using a ranking model instead of a classification model). However, newly design of introspective modules are also required. We leave these investigations to future works.

Implementation Details We consider the following three-step training strategy: 1) pre-train a classifier with the full text; 2) fix the pre-trained classifier, which is used for both the predictor and the complement predictor in the three-player game approach, and pre-train the rationale generators; and 3) fine-tune all modules end-to-end. This pipeline significantly stabilizes the training and provides better performances.[8] We use the same word embeddings as released by (Lei et al., 2016).

Model	Highlight Percentage	MAP	MAP^c
All	100%	51.55	38.97
Lei2016	20%	43.64	47.84
+minimax	20%	48.58	46.13
Intros	20%	45.08	49.27
+minimax	20%	48.55	48.37

Table 9: Testing MAP on the AskUbuntu dataset. MAP^c refers to the MAP score of the complement predictor. The desired rationalization method will have high MAP and low MAP^c.

Results Table 9 summarizes the results. We observe similar patterns as in previous datasets. The original model from (Lei et al., 2016) fails to maintain the performance compared to the model trained with full texts. Adding the proposed minimax game helps both the (Lei et al., 2016) and the introspection model to generate more informative texts as the rationales, which improves the MAP of the prediction while lowering the complement MAP.

Compared to the other tasks, the complement MAPs on AskUbuntu are relatively large. One reason is that the reported results rely significantly on the three-step training strategy. The best MAP on the development set often occurs after a few epochs of end-to-end training (the third step of our training procedure), which may results in premature training of the generators due to early stop. Another important reason is that there are a larger number of informative words in the questions, which makes it challenging for the generators to include all the useful information.

---

[7] `https://github.com/taolei87/askubuntu`.

[8] One potential reason that the three-step training strategy performs much better than end-to-end training from scratch is that we sample rationales according to the policy $\pi(\cdot)$ during training but take the action with the highest probability during the inference. During the first a few epochs of training, rationale generator almost extracts words at any positions with a probability lower than 0.5. Rationale words are still able to be sampled during training. However, during inference, there are no rationale words selected unless a probability of selection is greater than 0.5. Thus, the MAP on the development set is unchanged at the beginning stage of the training. In other words, there is a risk that the predictor already overfits but we cannot perform early-stopping of the training.