Unsupervised Selective Rationalization with Noise Injection

Adam Storek
Columbia University
astorek@cs.columbia.edu

Melanie Subbiah
Columbia University
m.subbiah@columbia.edu

Kathleen McKeown
Columbia University
kathy@cs.columbia.edu

Abstract

A major issue with using deep learning models in sensitive applications is that they provide no explanation for their output. To address this problem, unsupervised selective rationalization produces rationales alongside predictions by chaining two jointly-trained components, a rationale generator and a predictor. Although this architecture guarantees that the prediction relies solely on the rationale, it does not ensure that the rationale contains a plausible explanation for the prediction. We introduce a novel training technique that effectively limits generation of implausible rationales by injecting noise between the generator and the predictor. Furthermore, we propose a new benchmark for evaluating unsupervised selective rationalization models using movie reviews from existing datasets. We achieve sizeable improvements in rationale plausibility and task accuracy over the state-of-the-art across a variety of tasks, including our new benchmark, while maintaining or improving model faithfulness.[1]

Movie Review:
Ultra low budget but extremely inventive horror film about a group of friends vacationing in a cabin who accidentally awaken an evil force in the woods via the necronomicon, the book of the dead. Bruce Campbell stars as Ash, who eventually becomes the sole survivor and has to battle both the demons from the woods, and his friends who have become demons (including his own girlfriend). **The results shown on screen are amazing considering the film's** tiny budget, constant location changes, and a filming schedule that was sporadic over two years… **Class: Positive**

Figure 1: Example of a rationale selected by BERT-A2R + NI (our model) on the USR Movie Review dataset (our benchmark), which asks models to classify movie reviews as positive or negative.

1 Introduction

With the advent of large pre-trained language models like GPT-3 (Brown et al., 2020), the size and complexity of deep learning models used for natural language processing has dramatically increased. Yet greater performance and complexity can come at the cost of interpretability, masking anything from implementation mistakes to learned bias.

A model architecture that justifies its output by providing relevant subsets of input text as a rationale is therefore desirable (see example in Figure 1). The unsupervised selective rationalization architecture as introduced by Lei et al. (2016) generates rationales alongside predictions by chaining two jointly-trained components, a rationale-generator and a predictor. The generator extracts a rationale: concatenated short and concise spans of the input text that suffice for prediction. The predictor bases its prediction only on this rationale, which encourages **faithfulness**, meaning how much the rationale reveals what parts of the input were important to the model's prediction. In practice, however, the rationale often isn't **plausible**, meaning it can't convince a human of the correct prediction, undermining the architecture's interpretability (Jacovi and Goldberg, 2021; Zheng et al., 2022). Using a high-capacity generator can further degrade plausibility (Yu et al., 2019).

To prevent this effect, we introduce a novel training strategy that leverages online noise injection, based on word-level unsupervised data augmentation (Xie et al., 2020). By definition, if the loss-minimizing generator selects an implausible rationale, then the rationale both (a) offers no plausible connection for a human to the target label and (b) locally improves prediction accuracy. This might include communicating via punctuation (Yu et al., 2019) or subtle input perturbations (Garg and Ramakrishnan, 2020). Our new approach is to inject noise into the generated rationale during training by probabilistically replacing lower-importance words with noise - random words from the vocabulary -

[1] Code and benchmark are available at https://github.com/adamstorek/noise_injection.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 12647–12659
July 9-14, 2023 ©2023 Association for Computational Linguistics

before passing the rationale to the predictor. We observe that this strategy leads to a significant improvement in plausible rationale generation and prediction accuracy without compromising the faithfulness of the architecture. We also show that powerful generators typically interfere with plausible rationale generation but can be effectively deployed when trained with noise injection.

To test our approach, we introduce a new benchmark for unsupervised selective rationalization by integrating existing movie review datasets to replace the retracted canonical beer review dataset (McAuley et al., 2012; McAuley and Leskovec, 2013; Lei et al., 2016).[2] We merge a large IMDb movie review dataset (Maas et al., 2011) for training and validation and a smaller, rationale-annotated movie review dataset (DeYoung et al., 2020; Zaidan and Eisner, 2008; Pang and Lee, 2004) for evaluation. We also evaluate our unsupervised approach on the ERASER Movie Review, MultiRC and FEVER tasks (DeYoung et al., 2020; Khashabi et al., 2018; Thorne et al., 2018).[3]

Our contributions therefore include: 1) characterizing the issue of implausible rationale generation from the perspective of powerful rationale generators, 2) introducing a novel training strategy that limits implausible rationale generation and enables unsupervised selective rationalization models with powerful generators, 3) proposing a new unsupervised rationalization benchmark by repurposing existing movie review datasets, and 4) achieving more plausible rationale generation, with up to a relative 21% improvement in F1 score and a 7.7 point improvement in IOU-F1 score against the baseline model across a number of tasks.

2 Related Work

A major challenge with selective rationalization is that discrete selection of rationale tokens is non-differentiable, making training challenging without additional rationale supervision. Lei et al. (2016) use REINFORCE-style learning (Williams, 1992) to propagate the training signal from the predictor to the generator. Bastings et al. (2019) propose a differentiable approach leveraging the Hard Kumaraswamy Distribution. Yu et al. (2019) strive to improve rationale comprehensiveness. Chang et al. (2020) focus on avoiding spuriously correlated rationales. Yu et al. (2021) tackle the propensity of selective rationalization models to get stuck in local minima. Atanasova et al. (2022) use diagnostics-guided training to improve plausibility.

Our work builds on the previous approaches, since we also frame the generator-predictor interaction as a cooperative game and seek to improve plausibility. The previous approaches have, however, introduced additional training objectives (Atanasova et al., 2022) or involved incorporating a third adversarial (Yu et al., 2019) or cooperative (Yu et al., 2021) component. This increases model complexity significantly, leading to more resource-intensive and/or complicated training. Instead, we demonstrate the effectiveness of online noise injection, a considerably more lightweight approach.

An alternative approach is proposed by DeYoung et al. (2020) who assemble a series of datasets with labeled rationales; this enables fully supervised rationale learning. Given rationale-annotated training sets, Jain et al. (2020) train each model component separately, approaching the accuracy of an entirely black-box model. Although this is a compelling direction, requiring supervision reduces the practical usability of this technique, as many applications lack rationale annotations.

Both unsupervised and supervised selective rationalization approaches generally require a specific token selection strategy to select the output rationale from the generator model (Yu et al., 2021; Jain et al., 2020; Paranjape et al., 2020). No previous work that we are aware of, however, has tried to then modify the output rationale before it is input into the predictor. Using online noise injection to enforce prediction stability is therefore a novel approach that adds greater power to the current architectures and can be easily retrofitted.

3 Implausible Rationale Generation

Previous work has conceptualized the interaction between the generator and the predictor as a cooperative game (Chen et al., 2018a,b; Chang et al., 2019; Yu et al., 2019; Chang et al., 2020; Yu et al., 2021). This repeated sequential game consists of two-round stage games. In the first round, the generator accepts an input sequence $X_{1:T}$ and outputs a rationale selection as a binary mask $M_{1:T} \in \mathcal{M}$ where $\mathcal{M}$ represents the set of all masks such that $X_{1:T} \odot M_{1:T}$ satisfies rationale constraints. In the second round, the predictor accepts an input sequence $X_{1:T} \odot M_{1:T}$ and outputs prediction Y. The

[2] The dataset has been retracted at the request of BeerAdvocate and is no longer in use.

[3] Licensing information can be found in Appendix A.

joint objective is to minimize the loss (see Equation 2) based on the generated mask (see Equation 1):

$$M_{1:T} \leftarrow gen(X_{1:T}; \theta_{gen}), M_{1:T} \in \mathcal{M} \quad (1)$$

$$\min_{\theta_{gen}, \theta_{pre}} \mathcal{L}(pre(X_{1:T} \odot M_{1:T}; \theta_{pre}), \tilde{Y}) \quad (2)$$

For classification, it is customary to minimize the cross-entropy loss $\mathcal{L}_{CE}$. Such a system can be shown to maximize mutual information (MMI) of the rationale with respect to the class label provided sufficient generator and predictor capacity as well as a globally optimal generator (Yu et al., 2021; Chen et al., 2018a):

$$\max_{M_{1:T} \in \mathcal{M}} I(X_{1:T} \odot M_{1:T}; \tilde{Y}) \quad (3)$$

However, this property does not guarantee rationale plausibility.

First, MMI does not protect against spurious correlations (Chang et al., 2020). For example, a pleasant taste is not a good explanation for a positive review of a beer's appearance, although the two aspects are strongly correlated.

Second, MMI does not prevent rationale degeneration if the generator and predictor already contain certain biases, for example from pre-training (Jacovi and Goldberg, 2021).

Third, MMI does not prevent rationale degeneration if the generator and predictor are sufficiently powerful to develop a common encoding. Yu et al. (2019) found that providing the generator with a label predicted by a full-input classifier led the generator to develop a communication scheme with the predictor, including a period for positive and a comma for negative examples. Jacovi and Goldberg (2021) argue that any generator with sufficient capacity to construct a good inner-representation of Y can cause rationale degeneration.

The key underlying cause is that a sufficiently powerful generator is not disincentivized to produce implausible rationales beyond the assumption that generating a plausible rationale should maximize the expected accuracy of the predictor in the current training iteration. However, since the predictor is treated as a black box, this is not guaranteed. On the i-th training iteration, the generator greedily selects a binary mask $M_{1:T}$ that minimizes the expected loss:

$$\arg\min_{M_{1:T} \in \mathcal{M}} \mathbb{E}\left[\mathcal{L}(\widetilde{pre}_i(X_{1:T} \odot M_{1:T}))\right] \quad (4)$$

where $\widetilde{pre}_{G,i}$ represents the generator's learned representation of $pre(\cdot; \theta_{pre})$ from its previous experience interacting with the predictor for $i-1$ iterations in game G. As i increases, the generator learns to leverage deficiencies and biases of the predictor that remain hidden to humans, resulting in rationale plausibility degeneration.

4 Online Noise Injection

We propose a strategy that disrupts the generator's learned representation of the predictor $\widetilde{pre}_{G,i}$ for all games $G \in \mathcal{G}$, thereby making it harder for the generator to learn to exploit quirks of the predictor. We use online noise injection, which probabilistically perturbs unimportant words in a rationale sequence X of length T (see Algorithm 1).

```
Algorithm 1: Noise Injection.
  Input: input text $X_{1:T}$; binary mask $M_{1:T}$
  Data: set of documents $\mathcal{D}$; vocabulary $\mathcal{V}$
  $R_{1:T} \leftarrow X_{1:T} \odot M_{1:T}$;
  $R^*_{1:T} \leftarrow R_{1:T}$;
  forall $r_i \in R_{1:T}$ do
      $p_i$ = ProbOfReplacement$_{\mathcal{D}}(r_i)$;
      $replace \leftarrow Binomial(1, p_i)$;
      if $replace$ then
          $r^*_i \leftarrow$ SampleFromVocab$_{\mathcal{D};\mathcal{V}}()$;
      end
  end
  return perturbed rationale $R^*_{1:T}$
```

If the generator attempts to generate an implausible rationale during training iteration i, it strategically includes unimportant words from the input text in the generated rationale, relying on the predictor to pick up on the bias. By subtly perturbing the rationale - replacing the unimportant words - noise injection disrupts this attempt, and the predictor does not respond to the generator-injected bias favorably as expected by the generator. The generator is therefore forced to unlearn/reset its representation $\widetilde{pre}_{G,i}$ of the predictor and reassess its strategy, learning that generating implausible rationales is ineffective. Across any two stages i, j of game G, noise injection therefore keeps the learned representations of the predictor more consistent:

$$\forall G \in \mathcal{G}, \forall i, j \in stages(G), \widetilde{pre}_{G,i}(\cdot) \approx \widetilde{pre}_{G,j}(\cdot) \quad (5)$$

We implement the ProbOfReplacement and SampleFromVocab functions by adapting a strategy that probabilistically replaces words with small TF*IDF, originally proposed for unsupervised data augmentation by Xie et al. (2020). We precompute the probability of replacement of each word $w_i \in d$ in each document $d \in \mathcal{D}$ as its normalized TF*IDF score multiplied by the document length and a hyperparameter representing the magnitude of augmentation p:

$$\frac{w_{max} - \mathit{TF*IDF}(w_i)}{\sum_{w \in d} w_{max} - \mathit{TF*IDF}(w)} p|d| \tag{6}$$

$$w_{max} = \max_{w \in d} \mathit{TF*IDF}(w) \tag{7}$$

We use these precomputed probabilities to sample which words to replace as shown in Algorithm 1. The words are replaced with random words from the vocabulary $\mathcal{V}$. Nonetheless, we also strive to prevent sampling "keywords" from the vocabulary - words that are highly indicative of a label - to avoid confusing the predictor. We compute the sampling probability of w_i as its normalized ATF*IDF, where ATF corresponds to term frequency macro-averaged over $\mathcal{D}$:

$$\frac{w^*_{max} - \mathit{ATF*IDF}(w_i)}{\sum_{w \in d} w^*_{max} - \mathit{ATF*IDF}(w)} \tag{8}$$

$$w^*_{max} = \max_{w \in d} \mathit{ATF*IDF}(w) \tag{9}$$

5 Model

Our baseline model builds on the A2R architecture by Yu et al. (2021) who improve training stability by using an auxiliary predictor connected directly to the generator via an attention layer - this allows for gradients to flow. A2R selects top-$\frac{k}{2}$ bigrams with the highest attention scores from the generator as the rationale and input for the second predictor, with k corresponding to the number of rationale tokens selected as a fraction of the size of the input text. The two components minimize their separate criteria as well as the Jensen-Shannon divergence of their predictions Y^a and Y^r for the attention-based predictor and the rationale-based predictor, respectively. A2R's generator consists of a fixed GloVe (Pennington et al., 2014) embedding layer and a linear token scoring layer.

To take full advantage of our noise injection strategy, we replace the limited-capacity generator with BERT (Devlin et al., 2019). This allows us to use a simpler attention-based predictor than A2R (see Figure 2). To further manifest the efficacy of noise injection, we opt for a top-k unigram selection strategy which offers less regularization compared to a bigram selection strategy. Selecting unigrams is more challenging because it allows the model to select uninformative stopwords like "a" or "the".

Our architecture is shown in Figure 2. Both the selection strategy and the noise injection are model-external and untrained. As in Yu et al. (2021), the attention-based (see Equation 10) and the rationale-based (see Equation 11) components are trained using identical objectives - minimizing the sum of the cross-entropy loss and the Jensen-Shannon divergence of the two predictors:

$$\mathcal{L}_a = \mathcal{L}_{CE}(Y^a, \tilde{Y}) + \lambda JSD(Y^a, Y^r) \tag{10}$$

$$\mathcal{L}_r = \mathcal{L}_{CE}(Y^r, \tilde{Y}) + \lambda JSD(Y^a, Y^r) \tag{11}$$

We refer to our model as BERT-A2R and add +NI when noise injection is used during training.

6 USR Movie Review Dataset

Previous work on unsupervised selective rationalization used a decorrelated subset of the BeerAdvocate review dataset (McAuley et al., 2012) as preprocessed by Lei et al. (2016). The dataset has recently been removed at the request of BeerAdvocate and is therefore inaccessible to the scientific community. BeerAdvocate reviews consists of 80,000 labeled reviews without rationales for training/validation and ∼1,000 labeled reviews with token-level annotated rationales for testing. Alternative datasets either include rationale labels for the entire dataset (DeYoung et al., 2020) or do not provide rationale labels altogether (e.g. Maas et al. (2011)). Moreover, large datasets such as MultiRC or FEVER tend to provide sentence-level rationales compared to BeerAdvocate token-level rationales. We thus repurpose existing movie review datasets to recreate a task similar to beer review, enabling new work on unsupervised selective rationalization to evaluate their performance against models designed for beer review. We merge a smaller ERASER Movie Review dataset (DeYoung et al., 2020; Zaidan and Eisner, 2008; Pang and Lee, 2004) that has full token-level rationale annotations with the lower-cased Large Movie Review Dataset (Maas et al., 2011) which has no rationale annotations.

The movie review task is similar to the binarized beer review task as used in Chang et al. (2019); Yu et al. (2019); Chang et al. (2020); Yu et al. (2021);

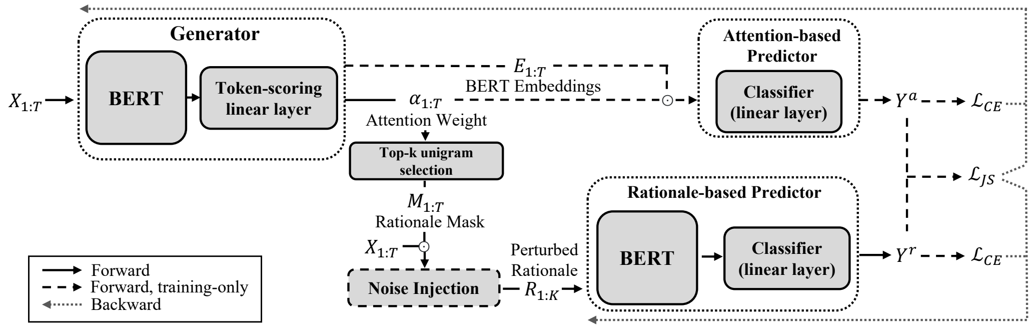

Figure 2: BERT-A2R + NI architecture. We replaced the generator's fixed GloVe (Pennington et al., 2014) embedding layer used in A2R with BERT-base. The original A2R uses a fixed GloVe embedding layer, GRU (Cho et al., 2014), and a linear classifier pipeline for each predictor. For the attention-based predictor, we remove the GloVe-GRU pipeline and instead reuse the generator's BERT embeddings. For the rationale-based predictor, we replace the GloVe-GRU pipeline with another BERT-base. Both A2R and BERT-A2R feed the masked input text directly into the predictor. To add noise injection during training, we first feed the masked input text into the noise injection component. This component is disabled during evaluation.

IMDb Movie Review 1:
Jim Carrey shines in this beautiful movie. This is now one of my favorite movies. I read all about the making and I thought it was incredible how they did it. I can't wait till this comes out on dvd. I saw this in theaters so many times, I can't even count how times I've seen it. **Class: Positive**

ERASER Movie Review 1:
"Party Camp," is one of **the most mindnumbingly brainless** comedies I've seen in awhile. A late **rip-off** of the "Meatballs" series, the film follows a group of young camp counselors at camp chipmunk. That's really about all that can be said about the "plot" because **nothing much happens**, except that the main character, wise-cracking Jerry (Andrew Ross), has the hots for a cute blonde (Kerry Brennan), and there is a big contest in the climax. How fun! **Class: Negative**

ERASER Movie Review 2:
Absolute Power, the new film produced and directed by Clint Eastwood, attempts to be a thriller set in the world of hypocritical presidents and their murderous political staff. It is about **as thrilling as a lecture on the mating habits of the South American grasshopper. One can only wonder how an utterly absurd script like the one written by William Goldman could have ever interested** Eastwood. **Not only is the plot unbelievable and contrived, but even the writing itself lacks any consistency or intelligence.** **Class: Negative**

Figure 3: Examples from the USR Movie Review Dataset. Note that compared to ERASER reviews, IMDb reviews tend to be shorter; ERASER reviews vary in length dramatically. Furthermore, ERASER rationale annotations are often inconsistent: the rationale for review 1 contains only very short spans, whereas the rationale for review 2 spans a few sentences.

both are binary sentiment classification tasks based on English user reviews. However, human rationale annotations of Eraser Movie Review are less coherent and consistent than beer review (see Figure 3) and lack single-aspect labels comparable to beer review's appearance, aroma, and taste labels. Moreover, movie review annotations tend to be over-complete (Yu et al., 2021): the same relevant information is often repeated many times in each review. This new task therefore also evaluates previous models' robustness to a subtle distribution shift, an increasingly important consideration for real-world systems.

The reviews from the ERASER Dataset were collected and processed by Pang and Lee (2004) from the IMDb archive of the rec.arts.movies.reviews newsgroup, whereas the Large Movie Review Dataset was scraped from the IMDb website by Maas et al. (2011). In order to avoid overlap between the train and test sets, we looked for similarity by searching for matches between lower-cased, break-tag-free, stop-word-free, lemmatized sentences which spanned at least 5 tokens to avoid generic matches such as "would not recommend" or "great film !". We discovered no overlap between the datasets. We use 40,000 reviews from the Large Movie Review Dataset for training and the remaining 10,000 reviews for validation. We then test our model on the 2,000 annotated examples from ERASER Movie Review.

7 Experimental setup

Metrics We evaluate generated rationales across several datasets using different metrics that capture faithfulness and plausibility. Faithfulness captures the extent to which the generated rationales truly explain the model's output. For faithfulness, we use comprehensiveness and sufficiency metrics (DeYoung et al., 2020). A rationale is *comprehensive* if it extracts all the information contained in the input text that is relevant for prediction and *sufficient* if it contains enough relevant information to make an accurate prediction. The comprehensiveness score measures the difference between the model's predictions on the entire input text and the input text without the selected rationale (higher is better), whereas the sufficiency score measures the difference between the model's predictions on the entire input text and just on the rationale (lower is better).

For plausibility, we use standard alignment metrics in reference to the human-annotated rationales: precision, recall, and F1 score as well as IOU-F1 score (referred to as IOU in tables) with partial match threshold 0.1 (DeYoung et al., 2020; Paranjape et al., 2020). We use token-level metrics for Movie Review which offers token-level annotations and sentence-level metrics for MultiRC and FEVER which provide only sentence-level annotations. Finally, we report prediction accuracy for the overall classification task. All results are averaged across 5 random seeds and reported as the mean with standard deviation in parentheses.

Implementation Our BERT-A2R models are trained for a maximum of 20 epochs for ERASER Movies and 5 epochs for every other dataset, keeping the checkpoint with the lowest validation loss. All BERT-A2R variants use uncased BERT-base, A2R closeness parameter $\lambda = 0.1$, and the selection strategy of picking the top $k = 20\%$ of the highest attention-scoring tokens for movie review or sentences for MultiRC and FEVER. We compute sentence-level scores by taking sentence-level averages of token scores. For optimization, we used Adam (Kingma, D.P. et al., 2015) with learning rate 2e-5 and batch size 16. Noise injection level p was set to 0.2 for USR and ERASER Movie review, 0.3 for MultiRC, and 0.05 for FEVER. This was determined based on our hyperparameter search. All of the models were trained on a single machine equipped with a 12-core processor, 64 GB of RAM, and a GPU with 24 GB of VRAM. [4]

8 Results

8.1 Does noise injection improve selective rationalization?

Model	**Acc.**	**F1**
Hard-Kuma (2019)	-	27.0
BERT Sparse IB (2020)	84.0	27.5
A2R (2021)	-	34.9
BERT-A2R (Ours)	84.0 (2.9)	36.4 (2.8)
BERT-A2R + **NI** (Ours)	**85.7** (2.7)	**38.6** (0.6)

Table 1: Results on ERASER Movie Review (without rationale supervision). +**NI** indicates using noise injection. We only report Accuracy and F1 to match published results on this benchmark and dashes indicate where the original paper did not publish this metric.

To compare against previous published results, we trained a BERT-A2R model on the ERASER Movie Review dataset with and without noise injection and compared our numbers to published results from the best unsupervised selective rationalization systems on this benchmark (see Table 1). All models were trained without rationale supervision. We see that our model with noise injection improves on both the classification task accuracy and the rationale F1 score relative to previous systems. Note that noise injection improves the F1 score more than the introduction of BERT to A2R.

We then train BERT-A2R models with and without noise injection on the MultiRC and FEVER benchmarks (see Table 2) as well as on our new USR Movie Review benchmark (see Table 3). Again, our noise injection training strategy achieves statistically significant improvements in rationale alignment with human annotations ($p < 0.01$ on the MultiRC and USR Movies, $p < 0.05$ on the FEVER, and $p < 0.1$ on ERASER Movies), achieving up to a relative 21% improvement in F1 score over our already performant baseline. The plausibility improvement applies for both token-level and sentence-level extraction tasks and across all metrics. Prediction accuracy also improves across all tasks except FEVER. Noise injection also does not seem to have a negative impact on model faithfulness. On ERASER benchmarks, neither comprehensiveness nor sufficiency worsen dramatically, and in the case that one score worsens, the other score tends to remain stable or even improve. On USR movie review, we see an improvement in both faithfulness scores from using noise injection.

[4] Training details can be found in Appendix B.

		Task	**Plausibility**				**Faithfulness**	
Dataset	**Model**	**Acc.**	**P**	**R**	**F1**	**IOU**	**Com ↑**	**Suf ↓**
MultiRC	BA2R	66.1 (1.9)	18.5 (1.6)	21.9 (2.2)	19.3 (1.8)	n/a	**-.01** (.01)	**-.02** (.02)
	BA2R+**NI**	**66.4** (0.8)	**22.6** (1.2)	**26.9** (1.8)	**23.8** (1.4)	n/a	**-.01** (.01)	**-.02** (.02)
FEVER	BA2R	**82.1** (3.2)	36.3 (0.6)	44.0 (0.3)	36.7 (0.5)	n/a	**.02** (.01)	**-.01** (.02)
	BA2R+**NI**	78.2 (1.9)	**39.0** (2.5)	**47.2** (2.9)	**39.5** (2.5)	n/a	**.02** (.00)	.00 (.00)
Movies	BA2R	84.0 (2.9)	36.3 (2.8)	36.5 (2.8)	36.4 (2.8)	30.9 (3.9)	**.02** (.02)	**-.04** (.02)
	BA2R+**NI**	**85.7** (2.7)	**38.5** (0.6)	**38.7** (0.6)	**38.6** (0.6)	**34.4** (2.2)	**.05** (.02)	-.02 (.01)

Table 2: Results on ERASER benchmark datasets. **P**, **R**, and **F1** are sentence-level for MultiRC and FEVER, since they use sentence-level rationale annotations, and token-level for Movie Review, as it uses token-level annotations. **IOU** is only sensible to use for token-level rationale annotations.

8.2 How does the noise injection level *p* affect model performance?

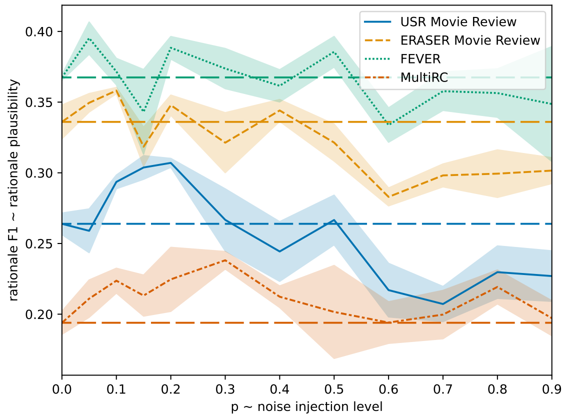

Figure 4: BERT-A2R + NI rationale F1 on the test set with varying noise injection level p. Error bands show ±1 standard error. Long-dash lines indicate the no noise injection baselines ($p = 0$) for each dataset.

We train variants of BERT-A2R+NI with different levels of p to examine what noise level is optimal for different datasets (see Figure 4). We average results across 5 seeds but there is still some noise given that the methodology injects noise into the process. It appears that in all cases noise injection seems to degrade performance once p becomes too high as we would expect since too much noise prevents useful signal from getting through. The optimal p varies depending on the task. Rationale alignment performance on FEVER peaks at just $p = 0.05$. The optimum for ERASER and USR Movie Review is at $p = 0.1$ and $p = 0.2$, respectively. The best performance on MultiRC was achieved at $p = 0.3$. There are numerous factors that might interact with noise injection to cause this behavior: task-specific demands, sentence vs. token-level rationale annotations, and the suitability of other training parameters. These interactions might be complex, especially with training strategies that dynamically adjust p during training. We leave exploration of these factors for future work.

8.3 Does noise injection enable the use of powerful high-capacity rationale generators?

For this experiment, we train BERT-A2R with fixed or trainable BERT weights in the generator, with or without noise injection, and evaluate on our new USR Movie Review benchmark (see Table 3). The version with fixed BERT weights in the generator has much less trainable capacity and cannot learn a task-specific text representation, whereas the generator with trainable BERT weights can potentially learn much better rationales or degrade to implausible rationales.

We find that the tuned generator trained with noise injection achieves superior performance across all the rationalization metrics without compromising prediction accuracy (2.8 improvement in rationale F1 score and a 7.7 improvement in rationale IOU-F1 score relative to the fixed setting). In contrast, the tuned generator without noise injection training performed the worst in all rationale metrics as well as prediction accuracy. Noise injection with a fixed generator results in a minor improvement in both plausibility metrics and prediction accuracy. We can therefore observe not only that noise injection allows us to leverage the power of a tunable BERT model in the generator that previously would have resulted in performance degradation, but also that the benefits of noise injection are greater with a powerful high-capacity generator model.

Finally, the addition of noise injection training also slightly improves comprehensiveness for both fixed and tuned generators while improving suffi-

	Task	Plausibility				Faithfulness	
Model	**Acc.**	**P**	**R**	**F1**	**IOU**	**Com ↑**	**Suf ↓**
fixed gen. weights	85.0 (0.8)	21.9 (0.4)	47.4 (0.8)	30.0 (0.5)	29.9 (0.6)	.02 (.00)	-.02 (.00)
fixed gen. weights + **NI**	85.8 (1.1)	22.3 (0.4)	48.2 (0.9)	30.5 (0.6)	30.7 (0.8)	.03 (.01)	-.01 (.00)
tuned gen. weights	82.4 (8.6)	20.2 (2.2)	43.7 (4.7)	27.6 (3.0)	29.1 (5.3)	.03 (.02)	-.03 (.03)
tuned gen. weights + **NI**	**87.9** (1.8)	**24.4** (0.6)	**52.7** (1.3)	**33.3** (0.8)	**38.4** (1.9)	**.04** (.01)	**-.04** (.02)

Table 3: Results on USR Movie Review using fixed or trainable BERT weights in the BERT-A2R generator.

ciency for the tuned generator.

Human-annotated:
With the exception of Don Knotts as the annoying "tv repairman" **the film is cast perfectly**:

BERT-A2R:
With the exception of Don Knotts as the annoying "tv repairman" **the film is cast perfectly**:

BERT-A2R + NI:
With the exception of Don Knotts as the annoying "tv repairman" **the** film **is cast perfectly**: **Class: Positive**

Figure 5: An occasional failure case of noise injection training - omitting frequently used words in movie reviews, such as "film".

8.4 What errors do the models make?

Human-annotated:
Proof of Life, Russell Crowe's one-two punch of a deft kidnap and rescue thriller, **is one of those rare gems. A taut drama laced with strong and subtle acting, an intelligent script, and masterful directing**, together it **delivers something virtually unheard of** in the film industry these days, **genuine motivation in a story that rings true**.

BERT-A2R:
Proof of Life, Russell Crowe**'s** one-two punch of a deft kidnap and rescue thriller, **is one of those rare gems. A taut** drama **laced with** strong **and** subtle acting**, an intelligent** script, and masterful directing, **together it delivers something** virtually unheard of in the film industry these days, genuine motivation **in a** story **that rings** true.

BERT-A2R + NI:
Proof of Life, Russell Crowe's one-two punch of a deft kidnap and rescue thriller, **is one of those rare gems. A taut drama laced with strong and subtle acting, an intelligent script, and masterful directing, together it delivers something** virtually unheard of in the film industry these days, **genuine motivation in a** story **that rings true**.

Class: Positive

Figure 6: This review shows the benefits of BERT-A2R + NI's propensity to highlight longer rationale spans where the baseline selects only single words.

For our qualitative analysis we randomly selected 20 reviews to evaluate the effect of adding noise injection to BERT-A2R during training. From this review sample, we include examples that we believe are characteristic for the behavior we observed. First, a BERT-A2R trained with noise injection tends to select longer spans of text as rationales (see Figure 6, 7), generally without sacrificing precision compared to the baseline. Selecting continuous rationales greatly improves readability and human-alignment as noted by Lei et al. (2016).

Human-annotated:
The movie's running time is under two hours, but it seems like it is well over it. **There's just not enough humor to speed things along, and not enough meaning to propel any drama.**

BERT-A2R:
The movie's running time is under two hours, but it seems **like** it is well over **it**. **There**'s **just not enough humor to** speed things along, and **not enough meaning to** propel **any drama.**

BERT-A2R + NI:
The movie's running time is under two hours, **but it seems like it is well over it. There's just not enough humor to speed things along, and not enough meaning to propel any drama.** **Class: Negative**

Figure 7: BERT-A2R + NI produces a more continuous and readable rationale, but it also includes a not-so-relevant part of the previous sentence.

We also observed that BERT-A2R + NI occasionally fails to select generic words such as "film" that, nevertheless, form a part of the rationale (see Figure 5). This could be a downside to our noise injection strategy, since the model will learn to ignore words with low TF*IDF even though they are relevant in a minority of cases. A potential remedy might be to use task-specific heuristics to generate probability of replacement information instead of the general low TF*IDF strategy. We leave this for future work.

Conclusion

In this paper, we investigate a major obstacle of unsupervised selective rationalization frameworks, where the generator has a tendency to learn to generate implausible rationales: rationales that lack a convincing explanation of the correct prediction. We explain the generator's propensity towards degeneration in terms of a flawed incentive structure, characterizing unsupervised selective rationalization as a sequential, repeated cooperative game. Through this lens, we propose a novel training strategy that penalizes implausible rationale generation, thereby realigning the incentive structure with the objective to generate plausible rationales. Using a new benchmark for unsupervised selective rationalization, we show that our noise injection approach is beneficial for training high-capacity generators, outperforming the current state of the art models.

Limitations

One of the main limitations of the noise injection training strategy is that statistics used to determine probability of replacement and sampling probability are token-specific. Although this works well on languages with limited morphology such as English, inflected languages like Czech that rely on declension and conjugation might require a lemma-based strategy or a different technique altogether. Furthermore, the model extracts a rationale of fixed length k, proportional to the length of the input text. Nevertheless, input text might include more or less information relevant to the class label; a sparsity objective as proposed by Paranjape et al. (2020) could remedy this issue. Lastly, injecting noise during training sometimes leads to more unpredictable training runs.

Additional model limitations are connected to using BERT. Despite its performance and fast training, using BERT limits the scalability to long text due to the 512-token limitation; nevertheless, tasks involving long text might be able to leverage specialized approaches such as Beltagy et al. (2020). Likewise, BERT renders BERT-A2R about 20 times larger than the GRU-based A2R, requiring greater GPU resources.

The dataset also comes with a few limitations. As Yu et al. (2021) note, some reviews contain many clear explanations for the target label, decreasing the need for the generator to include all relevant explanations in the rationale. Similarly, the sparsity of human-annotated rationales can be inconsistent across reviews: as shown in Figure 3, some rationales include long, generous spans of text that contain irrelevant information, whereas other rationales consist of merely the most important phrases.

Ethics Statement

We believe that improving the effectiveness and efficiency of unsupervised selective rationalization in the context of large pre-trained models such as BERT (Devlin et al., 2019) can help uncover and mitigate their learned bias as well as any implementation mistakes. Enabling models to produce plausible faithful rationales increases transparency, improving the end-user's understanding of the model's prediction and allowing AI practitioners to make more informed ethical choices in deploying models.

Acknowledgments

This research was conducted with funding from the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001120C0123. The views, opinions, and findings expressed are those of the authors and do not represent the official views or policies of the U.S. Department of Defense or the U.S. Government.

References

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2022. Diagnostics-guided explanation generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10445–10453.

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable Neural Predictions with Differentiable Binary Variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. ArXiv:2004.05150 [cs].

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec

Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2019. A game theoretic approach to class-wise selective rationalization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Shiyu Chang, Yang Zhang, Mo Yu, and Tommi Jaakkola. 2020. Invariant Rationalization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1448–1458. PMLR.

Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018a. Learning to explain: An information-theoretic perspective on model interpretation. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 883–892. PMLR.

Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. 2018b. L-shapley and c-shapley: Efficient model interpretation for structured data.

Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

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

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A Benchmark to Evaluate Rationalized NLP Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

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

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to Faithfully Rationalize by Construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Kingma, D.P., Ba, L.J., and Amsterdam Machine Learning lab (IVI, FNWI). 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*. arXiv.org.

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, Austin, Texas. Association for Computational Linguistics.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

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. ISSN: 2374-8486.

Julian John McAuley and Jure Leskovec. 2013. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web - WWW '13*, pages 897–908, Rio de Janeiro, Brazil. ACM Press.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Bhargavi Paranjape, Mandar Joshi, John Thickstun, Hannaneh Hajishirzi, and Luke Zettlemoyer. 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)*, pages 1938–1952, Online. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc V. Le. 2020. +unsupervised data augmentation for consistency training. In *NeuRIPS*.

Mo Yu, Shiyu Chang, Yang Zhang, and Tommi Jaakkola. 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)*, pages 4094–4103, Hong Kong, China. Association for Computational Linguistics.

Mo Yu, Yang Zhang, Shiyu Chang, and Tommi S. Jaakkola. 2021. Understanding Interlocking Dynamics of Cooperative Rationalization.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 31–40, Honolulu, Hawaii. Association for Computational Linguistics.

Yiming Zheng, Serena Booth, Julie Shah, and Yilun Zhou. 2022. The irrationality of neural rationale models. In *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*, pages 64–73, Seattle, U.S.A. Association for Computational Linguistics.

Appendix

A Licensing

Model	**License**
A2R	MIT License
HF BERT-base-uncased	Apache 2.0
NLTK "popular"	Apache 2.0

Table 4: Listing of model licenses.

Dataset	**License**
FEVER	Apache License 2.0
MultiRC	Apache License 2.0
Movies	Apache License 2.0
IMDb Movies	None, to our knowledge
USR Movies	MIT License

Table 5: Listing of dataset licenses.

B Training Details

Total estimated GPU hours spent on training: 500. BERT-A2R has 109484547 parameters.

Dataset	**Train**	**Val**	**Test**
FEVER	97957	6122	6111
MultiRC	24029	3214	4848
Movies	1600	200	200
USR Movies	40000	10000	2000

Table 6: Dataset details: Number of examples.

Dataset	**Train**	**Test**
FEVER	150 min	150 s
MultiRC	70 min	90 s
Movies	17 min	15 s
USR Movies	110 min	70 s

Table 7: Dataset details: BERT-A2R runtime.

Dataset	**LR**	**BS**	**#E**	**P**
FEVER	2e-5	16	5	2
MultiRC	2e-5	16	5	n/a
Movies	2e-5	16	20	5
USR Movies	2e-5	16 (64)	5 (10)	2 (n/a)

Table 8: BERT-A2R Training parameters by dataset. **LR**, **BS**, **#E** and **P** stand for learning rate, batch size, number of epochs, and patience. Parameters in parentheses are for fixed BERT generator training.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
No section number but directly following conclusion.

☒ A2. Did you discuss any potential risks of your work?
We do not see that our work introduces any new risks over the already published and publicly available previous work. In fact, we believe that better rationale plausibility improves interpretability and fairness, thereby reducing the risk that black-box models pose to the general public, especially to the historically disadvantaged groups.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

5, 8

☑ B1. Did you cite the creators of artifacts you used?
1-8

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
5-8

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We haven't collected any data ourselves.

☒ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We haven't collected any data ourselves.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix B; For each split of each dataset, we included the number of examples.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

C ☑ Did you run computational experiments?

7-8

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
7, Appendix B

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
7-8

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
7-8

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
7, Appendix A

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.