CREST: A Joint Framework for Rationalization and Counterfactual Text Generation

Marcos Treviso[1,2]*, **Alexis Ross**[3], **Nuno M. Guerreiro**[1,2], **André F. T. Martins**[1,2,4]
[1]Instituto de Telecomunicações, Lisbon, Portugal
[2]Instituto Superior Técnico & LUMLIS (Lisbon ELLIS Unit), Lisbon, Portugal
[3]Massachusetts Institute of Technology
[4]Unbabel, Lisbon, Portugal

Abstract

Selective rationales and counterfactual examples have emerged as two effective, complementary classes of interpretability methods for analyzing and training NLP models. However, prior work has not explored how these methods can be integrated to combine their complementary advantages. We overcome this limitation by introducing CREST (ContRastive Edits with Sparse raTionalization), a joint framework for selective rationalization and counterfactual text generation, and show that this framework leads to improvements in counterfactual quality, model robustness, and interpretability. First, CREST generates valid counterfactuals that are more natural than those produced by previous methods, and subsequently can be used for data augmentation at scale, reducing the need for human-generated examples. Second, we introduce a new loss function that leverages CREST counterfactuals to regularize selective rationales and show that this regularization improves both model robustness and rationale quality, compared to methods that do not leverage CREST counterfactuals. Our results demonstrate that CREST successfully bridges the gap between selective rationales and counterfactual examples, addressing the limitations of existing methods and providing a more comprehensive view of a model's predictions.

1 Introduction

As NLP models have become larger and less transparent, there has been a growing interest in developing methods for finer-grained interpretation and control of their predictions. One class of methods leverages **selective rationalization** (Lei et al., 2016; Bastings et al., 2019), which trains models to first select *rationales*, or subsets of relevant input tokens, and then make predictions based only on the selected rationales. These methods offer increased interpretability, as well as learning benefits, such as improved robustness to input perturbations (Jain et al., 2020; Chen et al., 2022). Another class of methods generates **counterfactual examples**, or modifications to input examples that change their labels. By providing localized views of decision boundaries, counterfactual examples can be used as explanations of model predictions, contrast datasets for fine-grained evaluation, or new training datapoints for learning more robust models (Ross et al., 2021; Gardner et al., 2020; Kaushik et al., 2020).

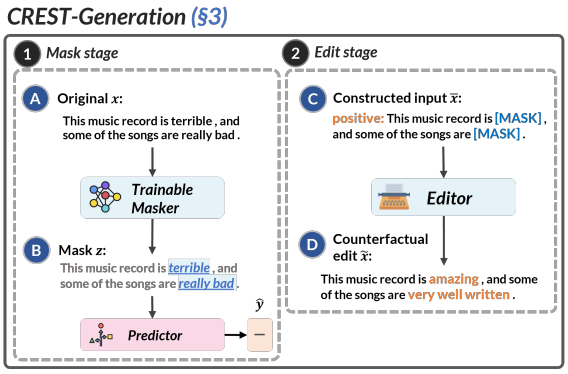

Figure 1: Our generation procedure consists of two stages: (i) a mask stage that highlights relevant tokens in the input through a learnable masker; and (ii) an edit stage, which receives a masked input and uses a masked language model to infill spans conditioned on a prepended label.

This paper is motivated by the observation that selective rationales and counterfactual examples allow for interpreting and controlling model behavior through different means: selective rationalization improves model transparency by weaving interpretability into a model's internal decision-making process, while counterfactual examples provide external signal more closely aligned with human causal reasoning (Wu et al., 2021).

We propose to combine both methods to leverage their complementary advantages. We introduce **CREST (*ContRastive Edits with Sparse raTionalization*)**, a joint framework for rationalization and

*Correspondence to: marcos.treviso@tecnico.pt

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 15109–15126
July 9-14, 2023

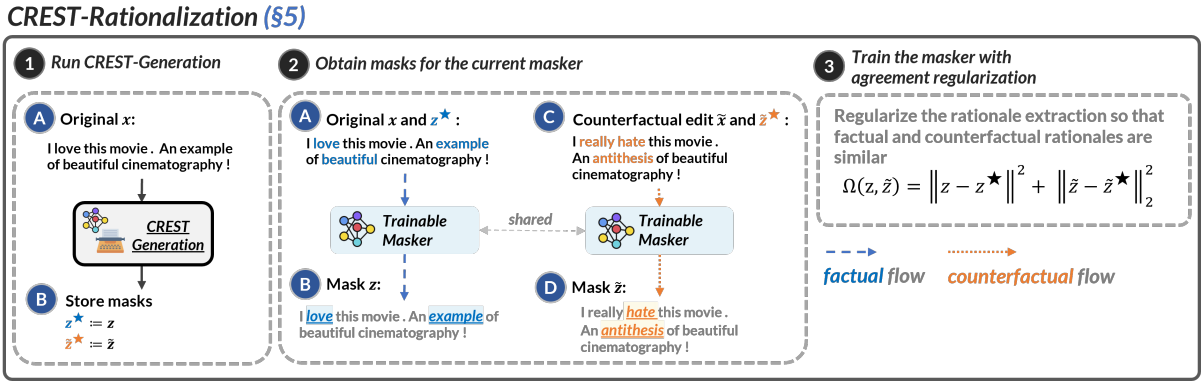

Figure 2: Overview of CREST-Rationalization. We start by passing an input $\boldsymbol{x}$ through CREST-Generation, which yields a counterfactual edit $\tilde{\boldsymbol{x}}$ along side two masks: $\boldsymbol{z}^\star$ for the original input, and $\tilde{\boldsymbol{z}}^\star$ for the counterfactual. Next, we train a new rationalizer (masker) decomposed into two flows: a **factual flow** that takes in $\boldsymbol{x}$ and produces a rationale $\boldsymbol{z}$, and a **counterfactual flow** that receives $\tilde{\boldsymbol{x}}$ and produces a rationale $\tilde{\boldsymbol{z}}$. Lastly, we employ a regularization term $\Omega(\boldsymbol{z}, \tilde{\boldsymbol{z}})$ to encourage agreement between rationales for original and counterfactual examples.

counterfactual text generation. CREST first generates high-quality counterfactuals (Figure 1), then leverages those counterfactuals to encourage consistency across "flows" for factual and counterfactual inputs (Figure 2). In doing so, CREST unifies two key important dimensions of interpretability introduced by Doshi-Velez and Kim (2017, §3.2), forward simulation and counterfactual simulation. Our main contributions are:[1]

- We present **CREST-Generation** (Figure 1), a novel approach to generating counterfactual examples by combining sparse rationalization with span-level masked language modeling (§3), which produces valid, fluent, and diverse counterfactuals (§4, Table 1).
- We introduce **CREST-Rationalization** (Figure 2), a novel approach to regularizing rationalizers. CREST-Rationalization decomposes a rationalizer into factual and counterfactual flows and encourages agreement between the rationales for both (§5).
- We show that CREST-generated counterfactuals can be effectively used to increase model robustness, leading to larger improvements on contrast and out-of-domain datasets than using manual counterfactuals (§6.2, Tables 2 and 3).
- We find that rationales trained with CREST-Rationalization not only are more plausible, but also achieve higher forward and counterfactual simulabilities (§6.3, Table 4).

[1]Code at https://github.com/deep-spin/crest/.

Overall, our experiments show that CREST successfully combines the benefits of counterfactual examples and selective rationales to improve the quality of each, resulting in a more interpretable and robust learned model.

2 Background

2.1 Rationalizers

The traditional framework of rationalization involves training two components cooperatively: the *generator*—which consists of an encoder and an explainer—and the *predictor*. The generator encodes the input and produces a "rationale" (e.g., word highlights), while the predictor classifies the text given only the rationale as input (Lei et al., 2016).

Assume a document $\boldsymbol{x}$ with n tokens as input. The encoder module (enc) converts the input tokens into d-dimensional hidden state vectors $\boldsymbol{H} \in \mathbb{R}^{n \times d}$, which are passed to the explainer (expl) to generate a latent mask $\boldsymbol{z} \in \{0, 1\}^n$. The latent mask serves as the rationale since it is used to select a subset of the input $\boldsymbol{x} \odot \boldsymbol{z}$, which is then passed to the predictor module (pred) to produce a final prediction $\hat{y} \in \mathcal{Y}$, where $\mathcal{Y} = \{1, ..., k\}$ for k-class classification. The full process can be summarized as follows:

$$\boldsymbol{z} = \mathsf{expl}(\mathsf{enc}(\boldsymbol{x}; \phi); \gamma), \quad (1)$$

$$\hat{y} = \mathsf{pred}(\boldsymbol{x} \odot \boldsymbol{z}; \theta), \quad (2)$$

where ϕ, γ, θ are trainable parameters. To ensure that the explainer does not select all tokens (i.e., $z_i = 1, \forall i$), sparsity is usually encouraged in the

rationale extraction. Moreover, explainers can also be encouraged to select contiguous words, as there is some evidence that it improves readibility (Jain et al., 2020). These desired properties may be encouraged via regularization terms during training (Lei et al., 2016; Bastings et al., 2019), or via application of sparse mappings (Treviso and Martins, 2020; Guerreiro and Martins, 2021).

In this work, we will focus specifically on the SPECTRA rationalizer (Guerreiro and Martins, 2021): this model leverages an explainer that extracts a deterministic structured mask $\boldsymbol{z}$ by solving a constrained inference problem with SparseMAP (Niculae et al., 2018). SPECTRA has been shown to achieve comparable performance with other rationalization approaches, in terms of end-task performance, plausibility with human explanations, and robustness to input perturbation (Chen et al., 2022). Moreover, it is easier to train than other stochastic alternatives (Lei et al., 2016; Bastings et al., 2019), and, importantly, it allows for simple control over the properties of the rationales, such as sparsity via its constrained inference formulation: by setting a budget B on the rationale extraction, SPECTRA ensures that the rationale size will not exceed $\lceil Bn \rceil$ tokens.

2.2 Counterfactuals

In NLP, counterfactuals refer to alternative texts that describe a different outcome than what is encoded in a given factual text. Prior works (Verma et al., 2020) have focused on developing methods for generating counterfactuals that adhere to several key properties, including:

- **Validity**: the generated counterfactuals should encode a different label from the original text.
- **Closeness**: the changes made to the text should be small, not involving large-scale rewriting of the input.
- **Fluency**: the generated counterfactuals should be coherent and grammatically correct.
- **Diversity**: the method should generate a wide range of counterfactuals with diverse characteristics, rather than only a limited set of variations.

While many methods for automatic counterfactual generation exist (Wu et al., 2021; Robeer et al., 2021; Dixit et al., 2022), our work is mostly related to MiCE (Ross et al., 2021), which generates counterfactuals in a two stage process that involves masking the top-k tokens with the highest ℓ_1 gradient attribution of a pre-trained classifier, and infilling tokens for masked position with a T5-based model (Raffel et al., 2020). MiCE further refines the resultant counterfactual with a binary search procedure to seek strictly *minimal* edits. However, this process is computationally expensive and, as we show in §4.2, directly optimizing for closeness can lead to counterfactuals that are less valid, fluent, and diverse. Next, we present an alternative method that overcomes these limitations while still producing counterfactuals that are close to original inputs.

3 CREST-Generation

We now introduce CREST (ContRastive Edits with Sparse raTionalization), a framework that combines selective rationalization and counterfactual text generation. CREST has two key components: (i) **CREST-Generation** offers a controlled approach to generating counterfactuals, which we show are valid, fluent, and diverse (§4.2); and (ii) **CREST-Rationalization** leverages these counterfactuals through a novel regularization technique encouraging agreement between rationales for original and counterfactual examples. We demonstrate that combining these two components leads to models that are more robust (§6.2) and interpretable (§6.3). We describe CREST-Generation below and CREST-Rationalization in §5.

Formally, let $\boldsymbol{x} = \langle x_1, ..., x_n \rangle$ represent a factual input text with a label y_f. We define a counterfactual as an input $\tilde{\boldsymbol{x}} = \langle x_1, ..., x_m \rangle$ labeled with y_c such that $y_f \neq y_c$. A counterfactual generator is a mapping that transforms the original text $\boldsymbol{x}$ to a counterfactual $\tilde{\boldsymbol{x}}$. Like MiCE, our approach for generating counterfactuals consists of two stages, as depicted in Figure 1: the mask and the edit stages.

Mask stage. We aim to find a mask vector $\boldsymbol{z} \in \{0, 1\}^n$ such that tokens x_i associated with $z_i = 1$ are relevant for the factual prediction $\hat{y}_f$ of a particular classifier C. To this end, we employ a SPECTRA rationalizer as the **masker**. Concretely, we pre-train a SPECTRA rationalizer on the task at hand with a budget constraint B, and define the mask as the rationale vector $\boldsymbol{z} \in \{0, 1\}^n$ (see §2.1).

Edit stage. Here, we create edits by infilling the masked positions using an **editor** module G, such as a masked language model: $\tilde{\boldsymbol{x}} \sim G_{\text{LM}}(\boldsymbol{x} \odot \boldsymbol{z})$. In order to infill spans rather than single tokens, we follow MiCE and use a T5-based model to infill

	IMDB					SNLI				
Method	**val.** ↑	**fl.** ↓	**div.** ↓	**clo.** ↓	**#tks**	**val.** ↑	**fl.** ↓	**div.** ↓	**clo.** ↓	**#tks**
Chance baseline	50.20	-	-	-	-	52.70	-	-	-	-
References	97.95	66.51	-	-	184.4	96.75	63.52	-	-	7.5
Manual edits	93.44	72.89	81.67	0.14	183.7	93.88	65.25	35.82	0.42	7.7
PWWS	28.07	101.91	74.56	**0.16**	179.0	17.97	160.11	31.81	0.36	6.8
CFGAN	-	-	-	-	-	34.46	155.84	68.94	**0.23**	7.0
PolyJuice	36.69	68.59	56.41	0.45	94.6	41.80	62.62	39.01	0.40	11.6
MiCE (bin. search)	72.13	76.72	73.76	0.20	171.3	76.17	63.94	42.18	0.35	7.9
MiCE (30% mask)	76.80	79.35	49.64	0.39	161.3	77.26	**59.71**	34.08	0.40	8.3
MiCE (50% mask)	83.20	89.92	**20.71**	0.65	115.7	**84.48**	68.32	**24.27**	0.52	7.6
CREST (30% mask)	75.82	67.29	57.58	0.33	180.9	75.45	62.00	41.36	0.29	7.4
CREST (50% mask)	**93.24**	**50.69**	23.08	0.66	193.9	81.23	62.60	30.53	0.41	7.3

Table 1: Intrinsic evaluation of counterfactuals generated by various methods. Validity is computed as the accuracy of an off-the-shelf RoBERTa-base classifier in relation to the gold counterfactual label (not available for PWWS and PolyJuice); fluency is determined by the perplexity score given by GPT-2 large; diversity is computed with self-BLEU; and closeness is reported by the (normalized) edit distance to the factual input. In addition, we report the average number of tokens in the input.

spans for masked positions. During training, we fine-tune the editor to infill original spans of text by prepending gold target labels y_f to original inputs. In order to generate counterfactual edits at test time, we prepend a counterfactual label y_c instead, and sample counterfactuals using beam search.

Overall, our procedure differs from that of MiCE in the mask stage: instead of extracting a mask via gradient-based attributions and subsequent binary search, we leverage SPECTRA to find an optimal mask. Interestingly, by doing so, we not only avoid the computationally expensive binary search procedure, but we also open up new opportunities: as our masking process is differentiable, we can optimize our masker to enhance the quality of both the counterfactuals (§4.2) and the selected rationales (§6.3). We will demonstrate the latter with our proposed CREST-Rationalization setup (§5). All implementation details for the masker and the editor can be found in §B.

4 Evaluating CREST Counterfactuals

This section presents an extensive comparison of counterfactuals generated by different methods.

4.1 Experimental Setting

Data and evaluation. We experiment with our counterfactual generation framework on two different tasks: sentiment classification using IMDB (Maas et al., 2011) and natural language inference (NLI) using SNLI (Bowman et al., 2015). In sentiment classification, we only have a single input to consider, while NLI inputs consist of a premise and a hypothesis, which we concatenate to form a single input. To assess the quality of our automatic counterfactuals, we compare them to manually crafted counterfactuals in the revised IMDB and SNLI datasets created by Kaushik et al. (2020). More dataset details can be found in §A.

Training. We employ a SPECTRA rationalizer with a T5-small architecture as the masker, and train it for 10 epochs on the full IMDB and SNLI datasets. We also use a T5-small architecture for the editor, and train it for 20 epochs with early stopping, following the same training recipe as MiCE. Full training details can be found in §B.3.

Generation. As illustrated in Figure 1, at test time we generate counterfactuals by prepending a contrastive label to the input and passing it to the editor. For sentiment classification, this means switching between positive and negative labels. For NLI, in alignment with Dixit et al. (2022), we adopt a refined approach by restricting the generation of counterfactuals to entailments and contradictions only, therefore ignoring neutral examples, which have a subtle semantic meaning. In contrast, our predictors were trained using neutral examples, and in cases where they predict the neutral class, we default to the second-most probable class.

Baselines. We compare our approach with four open-source baselines that generate counterfactuals: PWWS (Ren et al., 2019), PolyJuice (Wu et al., 2021), CounterfactualGAN (Robeer et al., 2021),[2]

[2]Despite many attempts, CounterfactualGAN did not converge on IMDB, possibly due to the long length of the inputs.

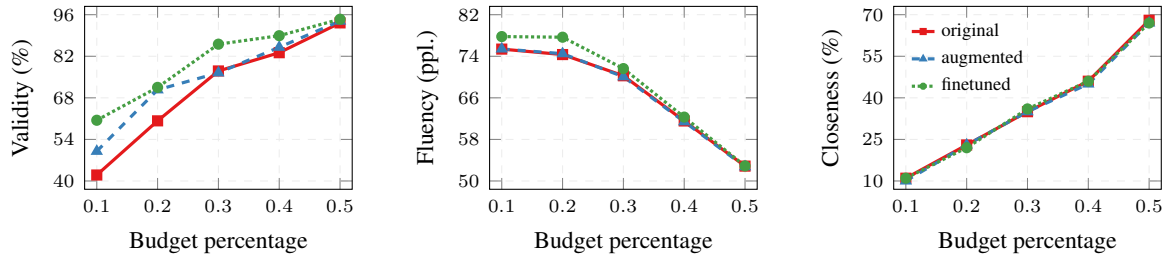

Figure 3: Sparsity analysis of CREST-Generation on IMDB with different budget percentages. The *original* curves show the performance of CREST without any changes, while the *augmented* and *finetuned* curves show the performance of CREST when using manually crafted counterfactuals for data augmentation or finetuning, respectively.

and MiCE (Ross et al., 2021). In particular, to ensure a fair comparison with MiCE, we apply three modifications to the original formulation: (i) we replace its RoBERTa classifier with a T5-based classifier (as used in SPECTRA); (ii) we disable its validity filtering;[3] (iii) we report results with and without the binary search procedure by fixing the percentage of masked tokens.

Metrics. To determine the general **validity** of counterfactuals, we report the accuracy of an off-the-shelf RoBERTa-base classifier available in the HuggingFace Hub.[4] Moreover, we measure **fluency** using perplexity scores from GPT-2 large (Radford et al., 2019) and **diversity** with self-BLEU (Zhu et al., 2018). Finally, we quantify the notion of **closeness** by computing the normalized edit distance to the factual input and the average number of tokens in the document.

4.2 Results

Results are presented in Table 1. As expected, manually crafted counterfactuals achieve high validity, significantly surpassing the chance baseline and establishing a reliable reference point. For IMDB, we find that CREST outperforms other methods by a wide margin in terms of validity and fluency. At the same time, CREST's validity is comparable to the manually crafted counterfactuals, while surprisingly deemed more fluent by GPT-2. Moreover, we note that our modification of disabling MiCE's minimality search leads to counterfactuals that are more valid and diverse but less fluent and less close to the original inputs.

For SNLI, this modification allows MiCE to achieve the best overall scores, closely followed by CREST. However, when controlling for closeness, we observe that CREST outperforms MiCE: at closeness of ∼0.30, CREST (30% mask) outperforms MiCE with binary search in terms of fluency and diversity. Similarly, at a closeness of ∼0.40, CREST (50% mask) surpasses MiCE (30% mask) across the board. As detailed in §C, CREST's counterfactuals are more valid than MiCE's for all closeness bins lower than 38%. We provide examples of counterfactuals produced by CREST and MiCE in Appendix G. Finally, we note that CREST is highly affected by the masking budget, which we explore further next.

Sparsity analysis. We investigate how the number of edits affects counterfactual quality by training maskers with increasing budget constraints (as described in §2.1). The results in Figure 3 show that with increasing masking percentage, generated counterfactuals become less textually similar to original inputs (i.e., less close) but more valid and fluent. This inverse relationship demonstrates that strict minimality, optimized for in methods like MiCE, comes with tradeoffs in counterfactual quality, and that the sparsity budget in CREST can be used to modulate the trade-off between validity and closeness. In Figure 3 we also examine the benefit of manually crafted counterfactuals in two ways: (i) using these examples as additional training data; and (ii) upon having a trained editor, further fine-tuning it with these manual counterfactuals. The results suggest that at lower budget percentages, exploiting a few manually crafted counterfactuals to fine-tune CREST can improve the validity of counterfactuals without harming fluency.

Validity filtering. As previously demonstrated by Wu et al. (2021) and Ross et al. (2022), it is possible to filter out potentially disfluent or invalid counterfactuals by passing all examples to

[3] MiCE with binary search uses implicit validity filtering throughout the search process to set the masking percentage.

[4] `mtreviso/roberta-base-imdb`, `mtreviso/roberta-base-snli`.

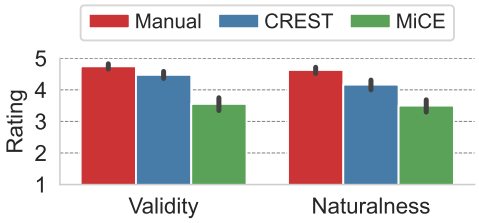

Figure 4: Human study results for counterfactuals produced manually and automatically (CREST and MiCE).

a classifier and discarding the subset with incorrect predictions. In our case, we use the predictor associated with the masker as the classifier. We found find that applying this filtering increases the validity of IMDB counterfactuals from 75.82 to 86.36 with $B = 0.3$, and from 93.24 to 97.36 with $B = 0.5$. For SNLI, validity jumps from 75.45 to 96.39 with $B = 0.3$, and from 81.23 to 96.67 with $B = 0.5$. These results indicate that CREST can rely on its predictor to filter out invalid counterfactuals, a useful characteristic for doing data augmentation, as we will see in §6.2.

4.3 Human Study

We conduct a small-scale human study to evaluate the quality of counterfactuals produced by MiCE and CREST with 50% masking percentage. Annotators were tasked with rating counterfactuals' *validity* and *naturalness* (e.g., based on style, tone, and grammar), each using a 5-point Likert scale. Two fluent English annotators rated 50 examples from the IMDB dataset, and two others rated 50 examples from SNLI. We also evaluate manually created counterfactuals to establish a reliable baseline. More annotation details can be found in §D.

The study results, depicted in Figure 4, show that humans find manual counterfactuals to be more valid and natural compared to automatically generated ones. Furthermore, CREST's counterfactuals receive higher ratings for validity and naturalness compared to MiCE, aligning with the results obtained from automatic metrics.

5 CREST-Rationalization

Now that we have a method that generates high-quality counterfactual examples, a natural step is to use these examples for data augmentation. However, vanilla data augmentation does not take advantage of the paired structure of original/contrastive examples and instead just treats them as individual datapoints. In this section, we present CREST's second component, CREST-Rationalization (illustrated in Figure 2), which leverages the relationships between factual and counterfactual inputs through a SPECTRA rationalizer with an **agreement regularization** strategy, described next.

5.1 Linking Counterfactuals and Rationales

We propose to incorporate counterfactuals into a model's functionality by taking advantage of the fully differentiable rationalization setup. Concretely, we decompose a rationalizer into two flows, as depicted in Figure 2: a **factual flow** that receives factual inputs $\boldsymbol{x}$ and outputs a factual prediction $\hat{y}$, and a **counterfactual flow** that receives counterfactual inputs $\tilde{\boldsymbol{x}}$ and should output a counterfactual prediction $\tilde{y} \neq \hat{y}$. As a by-product of using a rationalizer, we also obtain a factual rationale $\boldsymbol{z} \in \{0,1\}^n$ for $\boldsymbol{x}$ and a counterfactual rationale $\tilde{\boldsymbol{z}} \in \{0,1\}^m$ for $\tilde{\boldsymbol{x}}$, where $n = |\boldsymbol{x}|$ and $m = |\tilde{\boldsymbol{x}}|$.

Training. Let $\Theta = \{\phi, \gamma, \theta\}$ represent the trainable parameters of a rationalizer (defined in §2.1). We propose the following loss function:

$$\mathcal{L}(\Theta) = \mathcal{L}_f(y_f, \hat{y}(\Theta)) + \alpha \mathcal{L}_c(y_c, \tilde{y}(\Theta)) + \lambda \Omega(\boldsymbol{z}(\Theta), \tilde{\boldsymbol{z}}(\Theta)), \quad (3)$$

where $\mathcal{L}_f(\cdot)$ and $\mathcal{L}_c(\cdot)$ represent cross-entropy losses for the factual and counterfactual flows, respectively, and $\Omega(\cdot)$ is a novel penalty term to encourage factual and counterfactual rationales to focus on the same positions, as defined next. $\alpha \in \mathbb{R}$ and $\lambda \in \mathbb{R}$ are hyperparameters.

Agreement regularization. To produce paired rationales for both the factual and counterfactual flows, we incorporate regularization terms into the training of a rationalizer to encourage the factual explainer to produce rationales similar to those originally generated by the *masker* $\boldsymbol{z}^\star$, and the counterfactual explainer to produce rationales that focus on the tokens modified by the *editor* $\tilde{\boldsymbol{z}}^\star$. We derive the ground truth counterfactual rationale $\tilde{\boldsymbol{z}}^\star$ by aligning $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$ and marking tokens that were inserted or substituted as 1, and others as 0. The regularization terms are defined as:

$$\Omega(\boldsymbol{z}, \tilde{\boldsymbol{z}}) = \|\boldsymbol{z}(\Theta) - \boldsymbol{z}^\star\|_2^2 + \|\tilde{\boldsymbol{z}}(\Theta) - \tilde{\boldsymbol{z}}^\star\|_2^2. \quad (4)$$

To allow the counterfactual rationale $\tilde{\boldsymbol{z}}$ to focus on all important positions in the input, we adjust the budget for the counterfactual flow based on the length of the synthetic example produced by the counterfactual generator. Specifically, we multiply the budget by a factor of $\frac{\|\tilde{\boldsymbol{z}}^\star\|_0}{\|\boldsymbol{z}^\star\|_0}$.

Setup	IMDB	rIMDB	cIMDB	RotTom	SST-2	Amazon	Yelp
F	91.1 ± 0.3	91.4 ± 0.8	88.5 ± 0.9	76.5 ± 1.6	79.8 ± 1.6	86.0 ± 0.7	88.5 ± 0.7
With data augmentation:							
$F + C_H$	90.9 ± 0.5	**92.9** ± 0.9	**90.4** ± 1.6	76.6 ± 1.5	80.7 ± 1.3	86.3 ± 1.0	89.1 ± 1.2
$F + C_{S,V}$	91.0 ± 0.2	91.2 ± 1.0	89.3 ± 0.8	76.8 ± 0.9	79.3 ± 0.3	85.2 ± 0.9	88.0 ± 1.0
$F + C_S$	90.8 ± 0.2	91.6 ± 1.3	89.2 ± 0.4	76.7 ± 1.0	80.6 ± 0.6	86.4 ± 0.6	89.1 ± 0.5
With agreement regularization:							
F & $C_{S,V}$	90.7 ± 0.5	92.2 ± 0.7	88.9 ± 1.0	76.3 ± 1.4	80.2 ± 1.3	86.3 ± 0.7	88.9 ± 0.7
F & C_S	**91.2** ± 0.5	**92.9** ± 0.5	89.7 ± 1.1	**77.3** ± 2.3	**81.1** ± 2.4	**86.8** ± 0.8	**89.3** ± 0.7

Table 2: Accuracy of SPECTRA trained on IMDB and evaluated on in-domain, contrast, and out-of-domain datasets. We present mean and std. values across five random seeds. Values in **bold**: top results; underlined: second-best.

6 Exploiting Counterfactuals for Training

In this section, we evaluate the effects of incorporating CREST-generated counterfactuals into training by comparing a vanilla data augmentation approach with our CREST-Rationalization approach. We compare how each affects model robustness (§6.2) and interpretability (§6.3).

6.1 Experimental Setting

We use the IMDB and SNLI datasets to train SPECTRA rationalizers with and without counterfactual examples, and further evaluate on in-domain, contrast and out-of-domain (OOD) datasets. For IMDB, we evaluate on the revised IMDB, contrast IMDB, RottenTomatoes, SST-2, Amazon Polarity, and Yelp. For SNLI, we evaluate on the Hard SNLI, revised SNLI, break, MultiNLI, and Adversarial NLI. Dataset details can be found in §A. To produce CREST counterfactuals, which we refer to as "synthetic", we use a 30% masking budget as it provides a good balance between validity, fluency, and closeness (*cf.* Figure 3). We tune the counterfactual loss (α) and agreement regularization (λ) weights on the dev set. We report results with $\alpha = 0.01$ and $\lambda = 0.001$ for IMDB, and $\alpha = 0.01$ and $\lambda = 0.1$ for SNLI.

6.2 Robustness Results

Tables 2 and 3 show results for counterfactual data augmentation and agreement regularization for IMDB and SNLI, respectively. We compare a standard SPECTRA trained on factual examples (F) with other SPECTRA models trained on augmentated data from human-crafted counterfactuals ($F + C_H$) and synthetic counterfactuals generated by CREST ($F + C_S$), which we additionally post-process to drop invalid examples ($F + C_{S,V}$).

Discussion. As shown in Table 2, CREST-Rationalization (F & C_S) consistently outperforms vanilla counterfactual augmentation ($F + C_S$) on all sentiment classification datasets. It achieves the top results on the full IMDB and on all OOD datasets, while also leading to strong results on contrastive datasets—competitive with manual counterfactuals ($F + C_H$). When analyzing the performance of CREST-Rationalization trained on a subset of valid examples (F & $C_{S,V}$) versus the entire dataset (F & C_S), the models trained on the entire dataset maintain a higher level of performance across all datasets. However, when using counterfactuals for data augmentation, this trend is less pronounced, especially for in-domain and contrastive datasets. In §E, we explore the impact of the number of augmented examples on results and find that, consistent with previous research (Huang et al., 2020; Joshi and He, 2022), augmenting the training set with a small portion of valid and diverse synthetic counterfactuals leads to more robust models, and can even outweigh the benefits of manual counterfactuals.

Examining the results for NLI in Table 3, we observe that both counterfactual augmentation and agreement regularization interchangeably yield top results across datasets. Remarkably, in contrast to sentiment classification, we achieve more substantial improvements with agreement regularization models when these are trained on valid counterfactuals, as opposed to the full set.

Overall, these observations imply that CREST-Rationalization is a viable alternative to data augmentation for improving model robustness, especially for learning contrastive behavior for sentiment classification. In the next section, we explore the advantages of CREST-Rationalization for improving model interpretability.

6.3 Interpretability Analysis

In our final experiments, we assess the benefits of our proposed regularization method on model inter-

Setup	SNLI	SNLI-h	rSNLI	break	MNLI-m	MNLI-mm	ANLI
F	86.6 ± 0.2	73.7 ± 0.2	71.1 ± 0.8	69.5 ± 1.5	**64.6** ± 1.1	65.9 ± 0.9	32.6 ± 0.7
With data augmentation:							
$F + C_H$	86.6 ± 0.3	74.9 ± 1.1	**72.4** ± 0.3	70.1 ± 1.9	64.2 ± 0.9	65.8 ± 0.9	31.8 ± 0.4
$F + C_{S,V}$	86.5 ± 0.3	**75.8** ± 1.2	71.8 ± 1.0	69.1 ± 2.0	64.4 ± 0.3	65.9 ± 0.4	32.2 ± 0.2
$F + C_S$	86.6 ± 0.3	74.7 ± 1.1	71.6 ± 0.8	**71.2** ± 1.4	64.5 ± 0.4	**66.4** ± 0.6	32.2 ± 1.0
With agreement regularization:							
F & $C_{S,V}$	**86.8** ± 0.1	75.3 ± 0.8	66.8 ± 0.7	68.2 ± 2.1	**64.6** ± 0.7	66.1 ± 0.6	**32.8** ± 0.6
F & C_S	86.6 ± 0.1	75.5 ± 1.3	67.0 ± 1.3	69.9 ± 1.7	64.2 ± 1.1	66.0 ± 0.7	32.5 ± 0.5

Table 3: Accuracy of SPECTRA trained on SNLI and evaluated on in-domain, contrast, and out-of-domain datasets. We present mean and std. values across five random seeds. Values in **bold**: top results; underlined: second-best.

pretability. We evaluate effects on rationale quality along three dimensions: plausibility, forward simulability, and counterfactual simulability.

Plausibility. We use the MovieReviews (DeYoung et al., 2020) and the e-SNLI (Camburu et al., 2018) datasets to study the human-likeness of rationales by matching them with human-labeled explanations and measuring their AUC, which automatically accounts for multiple binarization levels.[5]

Forward simulability. Simulability measures how often a human agrees with a given classifier when presented with explanations, and many works propose different variants to compute simulability scores in an automatic way (Doshi-Velez and Kim, 2017; Treviso and Martins, 2020; Hase et al., 2020; Pruthi et al., 2022). Here, we adopt the framework proposed by Treviso and Martins (2020), which views explanations as a message between a classifier and a linear student model, and determines simulability as the fraction of examples for which the communication is successful. In our case, we cast a SPECTRA rationalizer as the classifier, use its rationales as explanations, and train a linear student on factual examples of the IMDB and SNLI datasets. High simulability scores indicate more understandable and informative explanations.

Counterfactual simulability. Building on the manual simulability setup proposed by Doshi-Velez and Kim (2017), we introduce a new approach to automatically evaluate explanations that interact with counterfactuals. Formally, let C be a classifier that when given an input $\boldsymbol{x}$ produces a prediction $\hat{y}$ and a rationale $\boldsymbol{z}$. Moreover, let G be a pre-trained counterfactual editor, which receives $\boldsymbol{x}$ and $\boldsymbol{z}$ and produces a counterfactual $\tilde{\boldsymbol{x}}$ by infilling spans on positions masked according to $\boldsymbol{z}$ (e.g., via masking).

[5]We determine the explanation score for a single word by calculating the average of the scores of its word pieces.

We define *counterfactual simulability* as follows:

$$\frac{1}{N}\sum_{n=1}^{N}[[C(\boldsymbol{x}_n) \neq C(G(\boldsymbol{x}_n \odot \boldsymbol{z}_n))]], \quad (5)$$

where $[[\cdot]]$ is the Iverson bracket notation. Intuitively, counterfactual simulability measures the ability of a rationale to change the label predicted by the classifier when it receives a contrastive edit with infilled tokens by a counterfactual generator as input. Therefore, a high counterfactual simulability indicates that the rationale $\boldsymbol{z}$ focuses on the highly contrastive parts of the input.

Results. The results of our analysis are shown in Table 4. We observe that plausibility can substantially benefit from synthetic CREST-generated counterfactual examples, especially for a rationalizer trained with our agreement regularization, which outperforms other approaches by a large margin. Additionally, leveraging synthetic counterfactuals, either via data augmentation or agreement regularization, leads to a high forward simulability score, though by a smaller margin—within the standard deviation of other approaches. Finally, when looking at counterfactual simulability, we note that models that leverage CREST counterfactuals consistently lead to better rationales. In particular, agreement regularization leads to strong results on both tasks while also producing more plausible rationales, showing the efficacy of CREST-Rationalization in learning contrastive behavior.

7 Related Works

Generating counterfactuals. Existing approaches to generating counterfactuals for NLP use heuristics (Ren et al., 2019; Ribeiro et al., 2020), leverage plug-and-play approaches to controlled generation (Madaan et al., 2021), or, most relatedly, fine-tune language models to

Setup	Sentiment Classification			Natural Language Inference		
	Plausibility	**F. sim.**	**C. sim.**	**Plausibility**	**F. sim.**	**C. sim.**
F	0.6733 ± 0.02	91.70 ± 0.92	81.18 ± 2.79	0.7735 ± 0.00	59.26 ± 0.41	70.01 ± 0.44
With data augmentation:						
$F + C_H$	0.6718 ± 0.04	91.44 ± 1.46	80.53 ± 4.17	0.7736 ± 0.01	59.51 ± 0.86	69.90 ± 0.57
$F + C_S$	0.6758 ± 0.01	91.68 ± 0.59	84.54 ± 1.09	0.7779 ± 0.00	**59.54** ± 0.08	**70.76** ± 0.54
With agreement regularization:						
F & C_S	**0.6904** ± 0.02	**91.93** ± 0.83	**86.43** ± 1.56	**0.7808** ± 0.00	59.31 ± 0.20	70.69 ± 0.29

Table 4: Interpretability analysis of rationalizers trained with CREST-generated counterfactuals, either with data augmentation or agreement regularization. Plausibility represents matching with human rationales, whereas F. sim. and C. sim. represent forward and counterfactual simulability. **Bold**: top results; underlined: second-best.

generate counterfactuals (Wu et al., 2021; Ross et al., 2021, 2022; Robeer et al., 2021). For instance, PolyJuice (Wu et al., 2021) finetunes a GPT-2 model on human-crafted counterfactuals to generate counterfactuals following pre-defined control codes, while CounterfactualGAN (Robeer et al., 2021) adopts a GAN-like setup. We show that CREST-Generation outperforms both methods in terms of counterfactual quality. Most closely related is MiCE (Ross et al., 2021), which also uses a two-stage approach based on a masker and an editor to generate counterfactuals. Unlike MiCE, we propose to relax the minimality constraint and generate masks using selective rationales rather than gradients, resulting not only in higher-quality counterfactuals, but also in a fully-differentiable set-up that allows for further optimization of the masker. Other recent work includes Tailor (Ross et al., 2022), a semantically-controlled generation system that requires a human-in-the-loop to generate counterfactuals, as well as retrieval-based and prompting approaches such as RGF (Paranjape et al., 2022) and CORE (Dixit et al., 2022).

Training with counterfactuals. Existing approaches to training with counterfactuals predominantly leverage data augmentation. Priors works have explored how augmenting with both manual (Kaushik et al., 2020; Khashabi et al., 2020; Huang et al., 2020; Joshi and He, 2022) and automatically-generated (Wu et al., 2021; Ross et al., 2022; Dixit et al., 2022) counterfactuals affects model robustness. Unlike these works, CREST-Rationalization introduces a new strategy for training with counterfactuals that leverages the paired structure of original and counterfactual examples, improving model robustness and interpretability compared to data augmentation. Also related is the training objective proposed by Gupta et al. (2021) to promote consistency across pairs of examples with shared substructures for neural module networks, and the loss term proposed by Teney et al. (2020) to model the factual-counterfactual paired structured via gradient supervision. In contrast, CREST can be used to *generate* paired examples, can be applied to non-modular tasks, and does not require second-order derivatives.

Rationalization. There have been many modifications to the rationalization setup to improve task accuracy and rationale quality. Some examples include conditioning the rationalization on pre-specified labels (Yu et al., 2019), using an information-bottleneck formulation to ensure informative rationales (Paranjape et al., 2020), training with human-created rationales (Lehman et al., 2019), and replacing stochastic variables with deterministic mappings (Guerreiro and Martins, 2021). We find that CREST-Rationalization, which is fully unsupervised, outperforms standard rationalizers in terms of model robustness and quality of rationales.

8 Conclusions

We proposed CREST, a joint framework for selective rationalization and counterfactual text generation that is capable of producing valid, fluent, and diverse counterfactuals, while being flexible for controlling the amount of perturbations. We have shown that counterfactuals can be successfully incorporated into a rationalizer, either via counterfactual data augmentation or agreement regularization, to improve model robustness and rationale quality. Our results demonstrate that CREST successfully bridges the gap between selective rationales and counterfactual examples, addressing the limitations of existing methods and providing a more comprehensive view of a model's predictions.

Limitations

Our work shows that CREST is a suitable framework for generating high-quality counterfactuals and producing plausible rationales, and we hope that CREST motivates new research to develop more robust and interpretable models. We note, however, two main limitations in our framework. First, our counterfactuals are the result of a large language model (T5), and as such, they may carry all the limitations within these models. Therefore, caution should be exercised when making statements about the quality of counterfactuals beyond the metrics reported in this paper, especially if these statements might have societal impacts. Second, CREST relies on a rationalizer to produce highlights-based explanations, and therefore it is limited in its ability to answer interpretability questions that go beyond the tokens of the factual or counterfactual input.

Acknowledgments

This work was supported by the European Research Council (ERC StG DeepSPIN 758969), by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), by P2020 project MAIA (LISBOA-01-0247- FEDER045909), by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (NextGenAI, Center for Responsible AI), and by contract UIDB/50008/2020. We are grateful to Duarte Alves, Haau-Sing Lee, Taisiya Glushkova, and Henrico Brum for the participation in human evaluation experiments.

References

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.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. In *Advances in Neural Information Processing Systems 31*, pages 9539–9549. Curran Associates, Inc.

Howard Chen, Jacqueline He, Karthik Narasimhan, and Danqi Chen. 2022. Can rationalization improve robustness? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3792–3805, Seattle, United States. 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.

Tanay Dixit, Bhargavi Paranjape, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. CORE: A retrieve-then-edit framework for counterfactual data generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2964–2984, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Finale Doshi-Velez and Been Kim. 2017. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608v2*.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Nuno M. Guerreiro and André F. T. Martins. 2021. SPECTRA: Sparse structured text rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6534–6550, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nitish Gupta, Sameer Singh, Matt Gardner, and Dan Roth. 2021. Paired examples as indirect supervision in latent decision models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5774–5785, Online and

Punta Cana, Dominican Republic. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana. Association for Computational Linguistics.

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4351–4367, Online. Association for Computational Linguistics.

William Huang, Haokun Liu, and Samuel R. Bowman. 2020. Counterfactually-augmented SNLI training data does not yield better generalization than unaugmented data. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 82–87, Online. Association for Computational Linguistics.

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.

Nitish Joshi and He He. 2022. An investigation of the (in)effectiveness of counterfactually augmented data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3668–3681, Dublin, Ireland. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2020. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*.

Daniel Khashabi, Tushar Khot, and Ashish Sabharwal. 2020. More bang for your buck: Natural perturbation for robust question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 163–170, Online. Association for Computational Linguistics.

Eric Lehman, Jay DeYoung, Regina Barzilay, and Byron C. Wallace. 2019. Inferring which medical treatments work from reports of clinical trials. 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 3705–3717, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

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.

Nishtha Madaan, Inkit Padhi, Naveen Panwar, and Diptikalyan Saha. 2021. Generate your counterfactuals: Towards controlled counterfactual generation for text. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13516–13524.

Vlad Niculae, Andre Martins, Mathieu Blondel, and Claire Cardie. 2018. Sparsemap: Differentiable sparse structured inference. In *International Conference on Machine Learning*, pages 3799–3808. PMLR.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

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.

Bhargavi Paranjape, Matthew Lamm, and Ian Tenney. 2022. Retrieval-guided counterfactual generation for QA. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1670–1686, Dublin, Ireland. Association for Computational Linguistics.

Danish Pruthi, Rachit Bansal, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C.

Lipton, Graham Neubig, and William W. Cohen. 2022. Evaluating explanations: How much do explanations from the teacher aid students? *Transactions of the Association for Computational Linguistics*, 10:359–375.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Shuhuai Ren, Yihe Deng, Kun He, and Wanxiang Che. 2019. Generating natural language adversarial examples through probability weighted word saliency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1085–1097, Florence, Italy. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Marcel Robeer, Floris Bex, and Ad Feelders. 2021. Generating realistic natural language counterfactuals. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3611–3625, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexis Ross, Ana Marasović, and Matthew Peters. 2021. Explaining NLP models via minimal contrastive editing (MiCE). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3840–3852, Online. Association for Computational Linguistics.

Alexis Ross, Tongshuang Wu, Hao Peng, Matthew Peters, and Matt Gardner. 2022. Tailor: Generating and perturbing text with semantic controls. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3194–3213, Dublin, Ireland. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Damien Teney, Ehsan Abbasnedjad, and Anton van den Hengel. 2020. Learning what makes a difference from counterfactual examples and gradient supervision. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 580–599. Springer.

Marcos Treviso and André F. T. Martins. 2020. The explanation game: Towards prediction explainability through sparse communication. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 107–118, Online. Association for Computational Linguistics.

Sahil Verma, John Dickerson, and Keegan Hines. 2020. Counterfactual explanations for machine learning: A review. *arXiv preprint arXiv:2010.10596v3*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6707–6723, Online. Association for Computational Linguistics.

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Texygen: A benchmarking platform for text generation models. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, page 1097–1100, New York, NY, USA. Association for Computing Machinery.

A Datasets

The revised IMDB and SNLI datasets, which we refer to as rIMDB and rSNLI respectively, were created by Kaushik et al. (2020). They contain counterfactuals consisting of revised versions made by humans on the Amazon's Mechanical Turk crowdsourcing platform. For both datasets, the authors ensure that (a) the counterfactuals are valid; (b) the edited reviews are coherent; and (c) the counterfactuals do not contain unnecessary modifications. For SNLI, counterfactuals were created either by revising the premise or the hypothesis. We refer to (Kaushik et al., 2020) for more details on the data generation process. Table 5 presents statistics for the datasets used for training models in this work.

	Train		Val.		Test	
Dataset	**docs**	**tks**	**docs**	**tks**	**docs**	**tks**
IMDB	22.5K	6M	2.5K	679K	25K	6M
rIMDB	3414	629K	490	92K	976	180K
SNLI	549K	12M	10K	232K	10K	231K
rSNLI	4165	188K	500	24K	1000	48K

Table 5: Datasets statistics.

Additionally, we incorporate various contrastive and out-of-domain datasets to evaluate our models. For IMDB, we use the contrast IMDB (Gardner et al., 2020), RottenTomatoes (Pang and Lee, 2005), SST-2 (Socher et al., 2013), Amazon Polarity and Yelp (Zhang et al., 2015). For SNLI, we evaluate on the Hard SNLI (Gururangan et al., 2018), break (Glockner et al., 2018), MultiNLI (Williams et al., 2018), and Adversarial NLI (Nie et al., 2020). We refer to the original works for more details.

B CREST Details

B.1 Masker

For all datasets, the masker consists of a SPECTRA rationalizer that uses a T5-small encoder as the backbone for the encoder and predictor (see §2.1). Our implementation is derived directly from its original source (Guerreiro and Martins, 2021). We set the maximum sequence length to 512, truncating inputs when necessary. We employ a contiguity penalty of 10^{-4} for IMDB and 10^{-2} for SNLI. We train all models for a minimum of 3 epochs and maximum of 15 epochs along with early stopping with a patience of 5 epochs. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4} and weight decay of 10^{-6}.

B.2 Editor

For all datasets, CREST and MiCE editors consist of a full T5-small model (Raffel et al., 2020), which includes both the encoder and the decoder modules. We use the T5 implementation available in the *transformers* library (Wolf et al., 2020) for our editor. We train all models for a minimum of 3 epochs and maximum of 20 epochs along with early stopping with a patience of 5 epochs. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-4} and weight decay of 10^{-6}. For both CREST and MiCE, we generate counterfactuals with beam search with a beam of size 15 and disabling bigram repetitions. We post-process the output of the editor to trim spaces and repetitions of special symbols (e.g., `</s>`).

B.3 SPECTRA rationalizers

All of our SPECTRA rationalizers share the same setup and training hyperparameters as the one used by the masker in §4, but were trained with distinct random seeds. We tuned the counterfactual loss weight α within $\{1.0, 0.1, 0.01, 0.001, 0.0001\}$, and λ within $\{1.0, 0.1, 0.01, 0.001\}$ for models trained with agreement rationalization. More specifically, we performed hyperparameter tuning on the validation set, with the goal of maximizing in-domain accuracy. As a result, we obtained $\alpha = 0.01$ and $\lambda = 0.001$ for IMDB, and $\alpha = 0.01$ and $\lambda = 0.1$ for SNLI.

C Validity vs. Closeness

To better assess the performance of CREST and MiCE by varying closeness, we plot in Figure 5 binned-validity scores of CREST and MiCE with 30% masking on the revised SNLI dataset. Although CREST is deemed less valid than MiCE overall (*cf.* Table 1), we note that CREST generates more valid counterfactuals in lower minimality ranges. This provides further evidence that CREST remains superior to MiCE on closeness intervals of particular interest for generating counterfactuals in an automatic way.

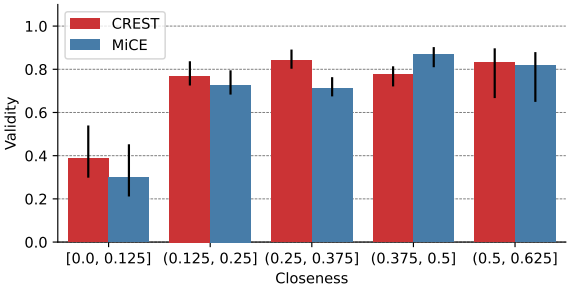

Figure 5: Validity by binned closeness ranges for MiCE (30% masking) and CREST (30% masking). At lower closeness ranges, CREST produces more valid counterfactuals than does MiCE.

D Human Annotation

The annotation task was conducted by four distinct individuals, all of whom are English-fluent PhD students. Two annotators were employed for IMDB and two for SNLI. The annotators were not given any information regarding the methods used to create each counterfactual, and the documents were presented in a random order to maintain source anonymity. The annotators were presented with the reference text and its corresponding gold label. Subsequently, for each method, they were asked to assess both the validity and the naturalness of the resulting counterfactuals using a 5-point Likert scale. We provided a guide page to calibrate the annotators' understating of validity and naturalness prior the annotation process. We presented hypothetical examples with different levels of validity and naturalness and provided the following instructions regarding both aspects:

- "If every phrase in the text unequivocally suggests a counterfactual label, the example is deemed fully valid and should receive a top score of 5/5."

- "If the counterfactual text aligns with the style, tone, and grammar of real-world examples, it's considered highly natural and deserves a score of 5/5."

We measure inter-annotator agreement with a normalized and inverted Mean Absolute Difference (MAD), which computes a "soft" accuracy by averaging absolute difference ratings and normalizing them to a 0-1 range. We present the annotation results in Table 6. Our results show that humans agreed more on manual examples than on automatic approaches. On the other hand, for SNLI, annotators assigned similar scores across all methods. In terms of overall metrics, including validity, naturalness, and agreement, the scores were lower for IMDB than for SNLI, highlighting the difficulty associated with the generation of counterfactuals for long movie reviews.

	IMDB			SNLI		
Method	v	n	r_o	v	n	r_o
Manual	4.60	4.36	0.83	4.89	4.90	0.95
MiCE	2.76	2.29	0.71	4.35	4.71	0.94
CREST	4.06	3.44	0.76	4.89	4.89	0.96
Overall	3.81	3.36	0.77	4.71	4.83	0.95

Table 6: Annotation statistics. v and n represent the averaged validity and naturalness scores, whereas r_o is the relative observed agreement computed with a normalized and inverted MAD.

Setup	Data size	RotTom	SST-2	Amazon	Yelp
F	100%	76.5 ± 1.6	79.8 ± 1.6	86.0 ± 0.7	88.5 ± 0.7
With data augmentation:					
$F + C_H$	+8%	76.6 ± 1.5	80.7 ± 1.3	86.3 ± 1.0	89.1 ± 1.2
$F + C_{S,V}$	+1%	77.2 ± 1.1	80.5 ± 0.5	86.1 ± 0.2	88.8 ± 0.3
$F + C_{S,V}$	+2%	76.2 ± 1.2	80.8 ± 0.8	86.7 ± 0.5	89.6 ± 0.5
$F + C_{S,V}$	+4%	**77.7** ± 0.8	80.8 ± 0.7	**87.0** ± 0.6	**89.8** ± 0.6
$F + C_{S,V}$	+8%	76.6 ± 2.2	80.2 ± 1.7	86.1 ± 0.9	88.2 ± 1.0
$F + C_{S,V}$	+85%	76.8 ± 0.9	79.3 ± 0.3	85.2 ± 0.9	88.0 ± 1.0
$F + C_S$	+100%	76.7 ± 1.0	80.6 ± 0.6	86.4 ± 0.6	89.1 ± 0.5
With agreement regularization:					
F & $C_{S,V}$	85%	76.3 ± 1.4	80.2 ± 1.3	86.3 ± 0.7	88.9 ± 0.7
F & C_S	100%	77.3 ± 2.3	**81.1** ± 2.4	86.8 ± 0.8	89.3 ± 0.7

Table 7: OOD accuracy of SPECTRA rationalizers with different portions of augmented counterfactuals. **Bold**: top results; underlined: second-best.

Annotation interface. Figure 6 shows a snapshot of the interface used for the annotation, which is publicly available at https://www.github.com/mtreviso/TextRankerJS.

E Counterfactual Data Augmentation Analysis

Previous studies on counterfactual data augmentation have found that model performance highly depends on the number and diversity of augmented samples (Huang et al., 2020; Joshi and He, 2022). To account for this, we investigate the effect of adding increasingly larger portions of CREST counterfactuals for data augmentation on the IMDB dataset. Our findings are summarized in Table 7.

Discussion. We find that incorporating human-crafted counterfactuals ($F + C_H$) improves SPECTRA performance on all OOD datasets. On top

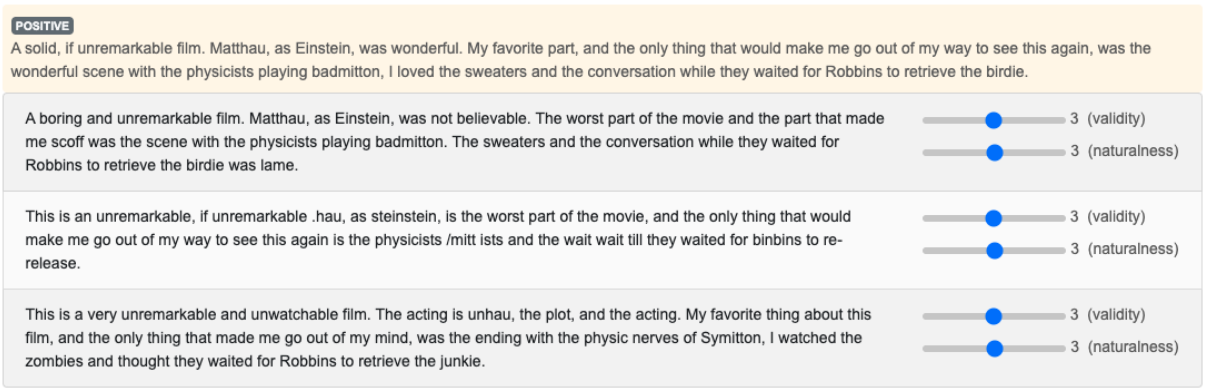

Figure 6: Snapshot of the annotation interface.

of that, we note that using a small proportion (4% of the full IMDB) of valid CREST counterfactuals ($F + C_{S,V}$) through data augmentation also leads to improvements on all datasets and outweighs the benefits of manual counterfactuals. This finding confirms that, as found by PolyJuice (Wu et al., 2021), synthetic counterfactuals can improve model robustness. Conversely, as the number of augmented counterfactuals increases (85%), the performance on OOD datasets starts to decrease, which is also consistent with the findings of Huang et al. (2020). When augmenting the entire training set we obtain an increase of accuracy, suggesting that the counterfactual loss weight (α) was properly adjusted on the validation set. Finally, we observe that while applying CREST-Rationalization only on valid examples (F & $C_{S,V}$) degrades performance, applying CREST-Rationalization on all paired examples (F & C_S) maintains a high accuracy on OOD datasets and concurrently leads to strong results on in-domain and contrast datasets (see Table 2).

F Computing infrastructure

Our infrastructure consists of four machines with the specifications shown in Table 8. The machines were used interchangeably and all experiments were carried in a single GPU.

GPU	CPU
4 × Titan Xp - 12GB	16 × AMD Ryzen - 128GB
4 × GTX 1080Ti - 12GB	8 × Intel i7 - 128GB
3 × RTX 2080Ti - 12GB	12 × AMD Ryzen - 128GB
3 × RTX 2080Ti - 12GB	12 × AMD Ryzen - 128GB

Table 8: Computing infrastructure.

G Examples of Counterfactuals

Table 9 shows examples of counterfactuals produced by MiCE and CREST with 30% masking.

	Sentiment Classification:
Input:	If you haven't seen this, it's terrible. It is pure trash. I saw this about 17 years ago, and I'm still screwed up from it.
MiCE:	If you haven't seen this, it's a great movie. I saw this about 17 years ago, and I'm still screwed up from it.
CREST:	If you haven't seen this movie, it's worth seeing. It's very funny. I saw it about 17 years ago, and I'm still screwed up from it.
Input:	Touching; Well directed autobiography of a talented young director/producer. A love story with Rabin's assassination in the background. Worth seeing !
MiCE:	Watching abiography of a very young writer/producer. A great story of Rabin's assassination in the background! Worth seeing!!
CREST:	This is the worst film of a talented young director/producer. And Rabin's assassination in the background is even worse!
Input:	A solid, if unremarkable film. Matthau, as Einstein, was wonderful. My favorite part, and the only thing that would make me go out of my way to see this again, was the wonderful scene with the physicists playing badmitton, I loved the sweaters and the conversation while they waited for Robbins to retrieve the birdie.
MiCE:	This is an unremarkable, if unremarkable .hau, as steinstein, is the worst part of the movie, and the only thing that would make me go out of my way to see this again is the physicists /mitt ists and the wait wait till they waited for binbins to re-release.
CREST:	This is a very unremarkable and unwatchable film. The acting is unhau, the plot, and the acting. My favorite thing about this film, and the only thing that made me go out of my mind, was the ending with the physic nerves of Symitton, I watched the zombies and thought they waited for Robbins to retrieve the junkie.
Input:	I saw this film earlier today, and I was amazed at how accurate the dialog is for the main characters. It didn't feel like a film - it felt more like a documentary (the part I liked best). The leading ladies in this film seemed as real to me as any fifteen year-old girls I know. All in all, a very enjoyable film for those who enjoy independent films.
MiCE:	I saw this film earlier today, and I was amazed at how bad the film is for the sake of a film - it feels more like thanthe part I played in this film. To me - fifteen year-old s I don't know. All in all this is a bad film for those who like independent films :
CREST:	I saw this movie earlier today, and I was surprised at how bad it is for the first time. It's not a good movie - it's just a bad movie (the only thing I can say about it). The acting is awful to me as any fifteen year-old as I can. All in all, the movie is a waste of time for me.
	Natural Language Inference:
Prem:	A large group of people walking in a busy city at night.
Hyp:	People are outside in a park.
MiCE:	People are walking in a city at night
CREST:	People walking in a city.
Prem:	Players from two opposing teams wearing colorful cleats struggle to gain control over a ball on an AstroTurf field.
Hyp:	The players are playing a sport.
MiCE:	The players are playing chess at home
CREST:	The players are sitting on a couch.
Prem:	A woman is in the middle of hitting a tennis ball.
Hyp:	A woman is playing tennis.
MiCE:	A woman is playing basketball at home
CREST:	A woman is playing basketball.
Prem:	Two boys with blond-hair, wearing striped shirts on a bed.
Hyp:	Children playing in the park.
MiCE:	Children are on the bed.
CREST:	Boys are on the bed.
Prem:	Bubbles surround a statue in the middle of a street.
Hyp:	There are bubbles around the statue.
MiCE:	There are bubbles surround the statue.
CREST:	Bubbles are in the ocean.
Prem:	A young girl is standing in a kitchen holding a green bib.
Hyp:	A boy is playing with a firetruck.
MiCE:	A child is in a fire place
CREST:	A girl is holding a bib.

Table 9: Examples of original inputs from the IMDB and SNLI datasets followed by synthetic counterfactuals produced by MiCE and CREST with 30% masking.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Final section (9)

☑ A2. Did you discuss any potential risks of your work?
Final section (9)

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

☑ A4. Have you used AI writing assistants when working on this paper?
ChatGPT, mostly to rephrase some sentences by following the prompt "rewrite this sentence in a better, more fluent, way (keep tone)".

B ☑ Did you use or create scientific artifacts?

Section 1 (footnote).

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1 for datasets, and Appendix B for the model architecture / implementation.

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

☐ 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)?
Not applicable. Left blank.

☐ 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?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ 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 A.

C ☑ Did you run computational experiments?

Sections 4 and 6, and Appendix C.

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

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Sections 4 and 6, and Appendix B.

☑ 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?
Sections 4 and 6, and Appendix C.

☑ 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.)?
For some metrics, yes (simulability in section 6).

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

Section 4.3 and Appendix D

☑ 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.?
Appendix D

☒ 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)?
Considering the simplicity of the study, we found this to be unnecessary.

☒ 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?
Considering the simplicity of the study, we found this to be unnecessary.

☒ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Considering the simplicity of the study, we found this to be unnecessary.

☒ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Considering the simplicity of the study, we found this to be unnecessary.