ER-TEST: Evaluating Explanation Regularization Methods for Language Models

Brihi Joshi[♣*] **Aaron Chan**[♣*] **Ziyi Liu**[♣*]
Shaoliang Nie[♢] **Maziar Sanjabi**[♢] **Hamed Firooz**[♢] **Xiang Ren**[♣]
[♣]University of Southern California [♢]Meta AI
{brihijos, chanaaro, zliu2803, xiangren}@usc.edu
{snie, maziars, mhfirooz}@fb.com

Abstract

By explaining how humans would solve a given task, human rationales can provide strong learning signal for neural language models (NLMs). Explanation regularization (ER) aims to improve NLM generalization by pushing the NLM's machine rationales (*Which input tokens did the NLM focus on?*) to align with human rationales (*Which input tokens would humans focus on?*). Though prior works primarily study ER via in-distribution (ID) evaluation, out-of-distribution (OOD) generalization is often more critical in real-world scenarios, yet ER's effect on OOD generalization has been underexplored. In this paper, we introduce ER-TEST, a framework for evaluating ER models' OOD generalization along three dimensions: unseen datasets, contrast set tests, and functional tests. Using ER-TEST, we comprehensively analyze how ER models' OOD generalization varies with the rationale alignment criterion (loss function), human rationale type (instance-level *vs.* task-level), number and choice of rationale-annotated instances, and time budget for rationale annotation. Across two tasks and six datasets, we show that ER has little impact on ID performance but yields large OOD performance gains, with the best ER criterion being task-dependent. Also, ER can improve OOD performance even with task-level or few human rationales. Finally, we find that rationale annotation is more time-efficient than label annotation for improving OOD performance. Our results with ER-TEST help demonstrate ER's utility and establish best practices for using ER effectively.

1 Introduction

Neural language models (NLMs) have achieved state-of-the-art performance on a broad array of natural language processing (NLP) tasks (Devlin et al., 2018; Liu et al., 2019). Even so, NLMs' reasoning processes are notoriously opaque (Rudin, 2019; Doshi-Velez and Kim, 2017; Lipton, 2018), which has spurred significant interest in designing algorithms to automatically explain NLM behavior (Denil et al., 2014; Sundararajan et al., 2017; Camburu et al., 2018; Rajani et al., 2019; Luo et al., 2021). Most of this work has focused on *rationale extraction*, which explains a NLM's output on a given task instance by highlighting the input tokens that most influenced the output (Denil et al., 2014; Sundararajan et al., 2017; Li et al., 2016; Jin et al., 2019; Lundberg and Lee, 2017; Chan et al., 2022).

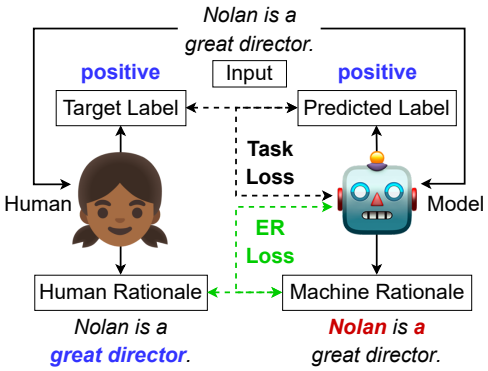

Figure 1: **Explanation Regularization (ER).** ER improves model generalization on NLP tasks by pushing the model's machine rationales (*Which input tokens did the model focus on?*) to align with human rationales (*Which input tokens would humans focus on?*) (Sec. 2).

Recent studies have investigated how *machine rationales* outputted by rationale extraction algorithms can be utilized to improve NLM decision-making (Hase and Bansal, 2021; Hartmann and Sonntag, 2022). Among these prior works, the most common paradigm is *explanation regularization* (ER), which aims to improve NLM by regularizing the NLM to yield machine rationales that align with *human rationales* (Fig. 1) (Ross et al., 2017; Huang et al., 2021; Ghaeini et al., 2019; Zaidan and Eisner, 2008; Kennedy et al., 2020; Rieger et al., 2020; Liu and Avci, 2019). Human rationales can be created by annotating each training instance individually (Lin et al., 2020; Camburu et al., 2018; Rajani et al., 2019) or by applying task-level human priors across all training instances

[*]Equal contribution.

(Rieger et al., 2020; Ross et al., 2017; Liu and Avci, 2019).

Though prior works primarily evaluate ER models' in-distribution (ID) generalization, the results are mixed, and it is unclear when ER is actually helpful. Furthermore, out-of-distribution (OOD) generalization is often more crucial in real-world settings (Chrysostomou and Aletras, 2022; Ruder, 2021), yet ER's impact on OOD generalization has been underexplored (Ross et al., 2017; Kennedy et al., 2020). In particular, due to the lack of unified comparison of different works using ER, little is understood about how OOD performance is affected by major design choices in building an ER pipeline, like the rationale alignment criterion (*i.e.,* loss function), human rationale type (instance-level *vs.* task-level), number and choice of rationale-annotated instances, and time budget for rationale annotation. In light of this, we propose **ER-TEST** [1](Fig. 2), a framework for evaluating ER methods' OOD generalization via: (1) *unseen datasets*, (2) *contrast set tests*, and (3) *functional tests*. For (1), ER-TEST tests ER models' performance on datasets beyond their training distribution. For (2), ER-TEST tests ER models' performance on real-world data instances that are semantically perturbed. For (3), ER-TEST tests ER models' performance on synthetic data instances created to capture specific linguistic capabilities.

Using ER-TEST, we study four questions: (A) Which *rationale alignment criteria* are most effective? (B) Is ER effective with task-level human rationales? (C) How is ER affected by the *number/choice of rationale-annotated instances*? (D) How does ER performance vary with the *rationale annotation time* budget? For two text classification tasks, we show that ER has little impact on ID performance but yields large gains on OOD performance, with the best ER criteria being task-dependent (Sec. 5.2). Furthermore, ER can improve OOD performance even with distantly-supervised (Sec. 5.3) or few (Sec. 5.4) human rationales. Finally, we find that rationale annotation yields more improvements than label annotation, specifically with limited time-budget for annotating (Sec. 5.5). ER-TEST results further show ER's utility and establish best practices for using ER effectively.

[1]Code available at https://github.com/INK-USC/ER-Test.

2 Explanation Regularization (ER)

Given an NLM for an NLP task, the goal of ER is to improve NLM generalization on the task by pushing the NLM's (extractive) machine rationales (*Which input tokens did the NLM focus on?*) to align with human rationales (*Which input tokens would humans focus on?*). The hope is that this inductive bias encourages the NLM to solve the task in a manner that follows humans' reasoning process.

Let $\mathcal{F}$ be an NLM for M-class text classification. $\mathcal{F}$ usually has a BERT-style architecture (Devlin et al., 2018), consisting of a Transformer encoder (Vaswani et al., 2017) followed by a linear layer with softmax classifier. Let $\mathbf{x}_i = [x_i^t]_{t=1}^n$ be the n-token input sequence (*e.g.,* a sentence) for task instance i. Let y_i denote $\mathcal{F}$'s predicted class for $\mathbf{x}_i$. Given $\mathcal{F}$, $\mathbf{x}_i$, and y_i, the goal of rationale extraction is to output machine rationale $\mathbf{r}_i = [r_i^t]_{t=1}^n$, such that each $0 \leq r_i^t \leq 1$ is an *importance score* indicating how strongly token x_i^t influenced $\mathcal{F}$ to predict class y_i (Luo et al., 2021). Let $\mathcal{G}$ denote a rationale extractor, such that $\mathbf{r}_i = \mathcal{G}(\mathcal{F}, \mathbf{x}_i, y_i)$.

$\mathcal{G}$ can also be used to compute machine rationales w.r.t. other classes besides y_i (*e.g.,* target class $\dot{y}_i$). Let $\hat{\mathbf{r}}_i$ denote the machine rationale for $\mathbf{x}_i$ w.r.t. $\dot{y}_i$. Given $\hat{\mathbf{r}}_i$ obtained via $\mathcal{G}$ and $\mathcal{F}$, many works have explored ER, in which $\mathcal{F}$ is regularized such that $\hat{\mathbf{r}}_i$ aligns with human rationale $\dot{\mathbf{r}}_i$ (Zaidan and Eisner, 2008; Lin et al., 2020; Rieger et al., 2020; Ross et al., 2017). $\dot{\mathbf{r}}_i$ can either be human-annotated for individual instances, or generated via human-annotated lexicons for a given task. Typically, $\dot{\mathbf{r}}_i$ is a binary vector, where ones and zeros indicate positive (important) and negative (unimportant) tokens, respectively.

We formalize the ER loss as: $\mathcal{L}_{\text{ER}} = \Phi(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i)$, where Φ is an ER criterion measuring alignment between $\hat{\mathbf{r}}_i$ and $\dot{\mathbf{r}}_i$. Thus, the full learning objective is: $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{ER}}\mathcal{L}_{\text{ER}}$, where $\mathcal{L}_{\text{task}}$ is the task loss (*e.g.,* cross-entropy loss) $\lambda_{\text{ER}} \in \mathbb{R}$ is the *ER strength* (*i.e.,* loss weight) for $\mathcal{L}_{\text{ER}}$. While there are many choices for Φ, it is unclear how Φ impacts training and when certain Φ should be preferred. Also, as a baseline, let $\mathcal{F}_{\text{No-ER}}$ denote an NLM that is trained without ER, such that $\mathcal{L} = \mathcal{L}_{\text{task}}$.

3 ER-TEST

Existing works primarily evaluate ER models via ID generalization (Zaidan and Eisner, 2008; Lin et al., 2020; Rieger et al., 2020; Liu and Avci, 2019;

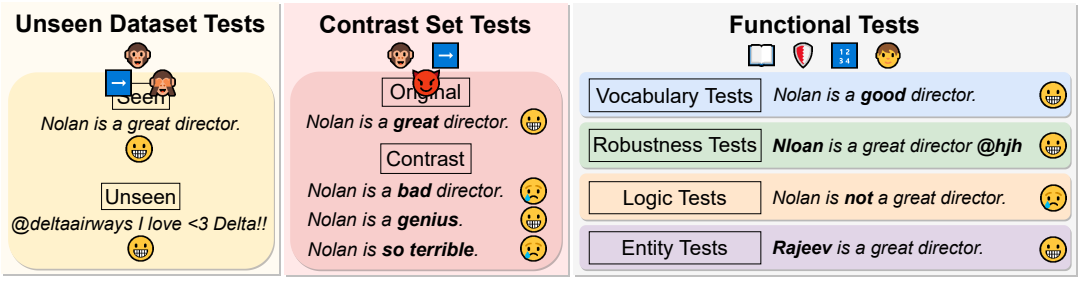

Figure 2: **ER-TEST Framework.** While existing ER works focus on ID generalization, ER-TEST evaluates ER's OOD generalization w.r.t. (A) unseen datasets, (B) contrast set tests, and (C) functional tests (Sec. 3, A.2.3).

Ross et al., 2017; Huang et al., 2021; Ghaeini et al., 2019; Kennedy et al., 2020), though a small number of works have done auxiliary evaluations of OOD generalization (Ross et al., 2017; Kennedy et al., 2020; Rieger et al., 2020). However, these OOD evaluations have been relatively small-scale, only covering a narrow range of OOD generalization aspects, ER criteria, tasks, and datasets. As a result, little is understood about ER's impact on OOD generalization. To address this gap, we propose ER-TEST (Fig. 2), a framework for designing and evaluating ER models' OOD generalization along three dimensions: (1) unseen dataset tests; (2) contrast set tests; and (3) functional tests.

Let $\mathcal{D}$ be an M-class text classification dataset, which we call the ID dataset. Assume $\mathcal{D}$ can be partitioned into training set $\mathcal{D}_{\text{train}}$, development set $\mathcal{D}_{\text{dev}}$, and test set $\mathcal{D}_{\text{test}}$, where $\mathcal{D}_{\text{test}}$ is the ID test set for $\mathcal{D}$. After training $\mathcal{F}$ on $\mathcal{D}_{\text{train}}$ with ER, we measure $\mathcal{F}$'s ID generalization via task performance on $\mathcal{D}_{\text{test}}$ and $\mathcal{F}$'s OOD generalization via (1)-(3).

3.1 Unseen Dataset Tests

First, we evaluate OOD generalization w.r.t. unseen datasets (Fig. 2A). Besides $\mathcal{D}$, suppose we have datasets $\{\tilde{\mathcal{D}}^{(1)}, \tilde{\mathcal{D}}^{(2)}, ...\}$ for the same task as $\mathcal{D}$. Each $\tilde{\mathcal{D}}^{(i)}$ has its own train/dev/test sets and distribution shift from $\mathcal{D}$. After training $\mathcal{F}$ with ER on $\mathcal{D}_{\text{train}}$ and hyperparameter tuning on $\mathcal{D}_{\text{dev}}$, we measure $\mathcal{F}$'s performance on each OOD test set $\tilde{\mathcal{D}}^{(i)}_{\text{test}}$. This tests ER's ability to help $\mathcal{F}$ learn general (*i.e.,* task-level) knowledge representations that can (zero-shot) transfer across datasets.

3.2 Contrast Set Tests

Second, we evaluate OOD generalization w.r.t. contrast set tests (Fig. 2B). Dataset annotation artifacts (Gururangan et al., 2018) can cause NLMs to learn spurious decision rules that work on the test set but do not capture linguistic abilities that the dataset was designed to assess. Thus, we test $\mathcal{F}$ on contrast sets (Gardner et al., 2020), which are constructed by manually perturbing the test instances of real-world datasets to express counterfactual meanings. Contrast set tests reveal the dataset's intended decision boundaries and if $\mathcal{F}$ has learned undesirable dataset-specific shortcuts. Given $\tilde{\mathcal{D}}^{(i)}_{\text{test}}$, we can convert $\tilde{\mathcal{D}}^{(i)}_{\text{test}}$ to contrast set $\tilde{\mathcal{C}}^{(i)}_{\text{test}}$ using various types of semantic perturbation, such as inversion (*e.g.,* "*big* dog" $\rightarrow$ "*small* dog"), numerical modification (*e.g.,* "*one* dog" $\rightarrow$ "*three* dogs"), and entity replacement (*e.g.,* "good *dog*" $\rightarrow$ "good *cat*"). However, since contrast sets are built from real-world datasets, they provide less flexibility in testing linguistic abilities, as a given perturbation type may not apply to all instances in the dataset. Note that, unlike adversarial examples (Gao and Oates, 2019), contrast sets are not conditioned on $\mathcal{F}$ specifically to attack $\mathcal{F}$.

3.3 Functional Tests

Third, we evaluate OOD generalization w.r.t. functional tests (Fig. 2C). Whereas contrast sets are created by perturbing real-world datasets, functional tests evaluate $\mathcal{F}$ on synthetic datasets, which are manually created via templates to assess specific linguistic abilities (Ribeiro et al., 2020; Li et al., 2020). While contrast set tests focus on semantic abilities, functional tests consider both semantic (*e.g.,* perception of word/phrase sentiment, sensitivity to negation) and syntactic (*e.g.,* robustness to typos or punctuation addition/removal) abilities. Therefore, functional tests trade off data realism for evaluation flexibility. If ER improves $\mathcal{F}$'s functional test performance for a given ability, then ER may be a useful inductive bias for OOD generalization w.r.t. that ability. Across all tasks, ER-TEST contains four general categories of functional tests: Vocabulary, Robustness, Logic, and Entity (Ribeiro et al., 2020). See Sec. A.2.3 for more details.

4 ER-TEST Design Choices

An ER model consists of three key components: rationale alignment criterion, type of human ratio-

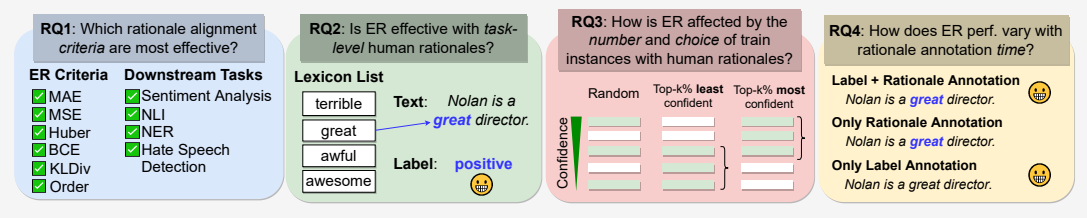

Figure 3: **Experiment Setup**: We use ER-TEST to investigate *four* research questions. (Sec. 5).

nales, and instance selection strategy. ER-TEST evaluates the design choices for each component.

4.1 Rationale Alignment Criteria

Compared to existing works, ER-TEST uses a wider range of rationale alignment criteria to evaluate ER model generalization. This provides a more comprehensive picture of ER's impact on both ID and OOD generalization, helping us understand why and when certain criteria work well. To demonstrate ER-TEST's utility, we consider six representative rationale alignment criteria (*i.e.,* choices of Φ), described below.

Mean Squared Error (MSE) is used in Liu and Avci (2019), Kennedy et al. (2020), and Ross et al. (2017): $\Phi_{\text{MSE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \frac{1}{n}\|\hat{\mathbf{r}}_i - \dot{\mathbf{r}}_i\|_2^2$.

Mean Absolute Error (MAE) is used in Rieger et al. (2020): $\Phi_{\text{MAE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \frac{1}{n}|\hat{\mathbf{r}}_i - \dot{\mathbf{r}}_i|$.

Binary Cross Entropy (BCE) loss is used in Chan et al. (2022) and Chan et al. (2021): $\Phi_{\text{BCE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = -\frac{1}{n}\sum_{t=1}^{n} \dot{r}_i^t \log(\hat{r}_i^t)$.

Huber Loss (Huber, 1992) is a hybrid of MSE and MAE, but is still unexplored for ER. Our experiments use the default $\delta = 1$ (Paszke et al., 2019).

$$\Phi_{\text{Huber}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \begin{cases} \frac{1}{2}\Phi_{\text{MSE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i), & \Phi_{\text{MAE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) < \delta \\ \delta(\Phi_{\text{MAE}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) - \frac{1}{2}\delta), & \text{otherwise} \end{cases} \quad (1)$$

Order Loss Recall that the human rationale $\dot{\mathbf{r}}_i$ labels each token as positive (one) or negative (zero). Whereas other criteria generally push positive/negative tokens' importance scores to be as high/low as possible, order loss (Huang et al., 2021) relaxes MSE to merely enforce that all positive tokens' importance scores are higher than all negative tokens' importance scores. This is especially useful if $\dot{\mathbf{r}}_i$ is somewhat noisy (*e.g.,* some positive-labeled tokens should not really be positive).

$$\Phi_{\text{Order}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \sum_{\dot{r}_i^t=1} \left(\min\left(\frac{\hat{r}_i^t}{\max_{\dot{r}_j^t=0} \hat{r}_j^t} - 1, 0 \right) \right)^2 \quad (2)$$

KL Divergence (KLDiv) is used by Pruthi et al. (2020), Chan et al. (2022), and Chan et al. (2021): $\Phi_{\text{KLDiv}}(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i) = \frac{1}{n}\sum_{t=1}^{n} \dot{r}_i^t \log(\dot{r}_i^t / \hat{r}_i^t)$.

Machine Rationale Extractors We consider three types of machine rationale extractors: *gradient-based*, which computes rationales via $\mathcal{F}(\mathbf{x}_i)$'s gradients (Sundararajan et al., 2017; Sanyal and Ren, 2021; Shrikumar et al., 2017); *attention-based*, which computes rationales via $\mathcal{F}(\mathbf{x}_i)$'s attention weights (Ding and Koehn, 2021; Wiegreffe and Pinter, 2019); and *learned*, which trains a model to compute rationales w.r.t. faithfulness and/or plausibility objectives (Chan et al., 2022). While other rationale extractor types, such as *perturbation-based* (Li et al., 2016), can also be used, we focus on the first three types since they are relatively less compute-intensive. In our experiments, we use Input*Gradient (IxG) (Denil et al., 2014), Attention (Attn) (Ding and Koehn, 2021), and UNIREX (Chan et al., 2022) as representatives for these three types, respectively.

4.2 Types of Human Rationales

Unlike prior works, ER-TEST considers both *instance-level* and *task-level* human rationales.

Instance-Level Rationales Human rationales are often created by annotating each training instance individually (Lin et al., 2020; Camburu et al., 2018; Rajani et al., 2019). For each instance, humans are asked to mark tokens that support the gold label as positive, with the remaining tokens counted as negative. Here, each human rationale is specifically conditioned on the input and gold label for the given instance. However, instance-level rationales are expensive to obtain, given the high manual effort required per instance.

Task-Level Rationales Some works construct distantly-supervised human rationales by applying task-level human priors across all training instances (Kennedy et al., 2020; Rieger et al., 2020; Ross et al., 2017; Liu and Avci, 2019). Given a task-level token lexicon, each instance's rationale is created by marking input tokens present in the lexicon as positive and the rest as negative, or vice versa. Here, rationales are not as fine-grained or tailored for the given dataset, but may provide a more general learning signal for solving the task.

4.3 Instance Selection Strategies

In real-world applications, it is often infeasible to annotate instance-level human rationales $\dot{\mathbf{r}}_i$ for all training instances (Chiang and Lee, 2022; Kaushik et al., 2019). Besides task-level rationales, another approach for addressing this issue could be to annotate only a subset $\mathcal{S}_{\text{train}} \subset \mathcal{D}_{\text{train}}$ of training instances. Given a constant budget of $|\mathcal{S}_{\text{train}}| = \frac{k}{100}|\mathcal{D}_{\text{train}}|$ instances, where $0 < k < 100$, our goal is to select $\mathcal{S}_{\text{train}}$ such that ER with $\mathcal{S}_{\text{train}}$ maximizes $\mathcal{F}$'s task performance. While instance selection via active learning is well-studied for general classification (Schröder and Niekler, 2020), this problem has not been explored in ER. Given non-ER NLM $\mathcal{F}_{\text{No-ER}}$, we use ER-TEST to compare five active-learning-inspired instance selection strategies. Note that these are just basic strategies, used to show ER-TEST's utility. In practice, one could consider more sophisticated strategies that account for other factors like data diversity.

Random Sampling (Rand) constructs $\mathcal{S}_{\text{train}}$ by uniformly sampling $|\mathcal{S}_{\text{train}}|$ instances from $\mathcal{D}$.

Lowest Confidence (LC) selects the $|\mathcal{S}_{\text{train}}|$ instances for which $\mathcal{F}_{\text{No-ER}}$ yields the *lowest* target class confidence probability $\mathcal{F}_{\text{No-ER}}(\dot{y}_i|x_i)$ (Zheng and Padmanabhan, 2002).

Highest Confidence (HC) selects the $|\mathcal{S}_{\text{train}}|$ instances for which $\mathcal{F}_{\text{No-ER}}$ yields the *highest* target class confidence probability $\mathcal{F}_{\text{No-ER}}(\dot{y}_i|x_i)$. This is the opposite of LC.

Lowest Importance Scores (LIS) Given machine rationale $\hat{\mathbf{r}}_i$ for $\mathcal{F}_{\text{No-ER}}$ and $0 < k' < 100$, let $\hat{\mathbf{r}}_i^{(k')}$ denote a vector of the top-$k\%$ highest importance scores in $\hat{\mathbf{r}}_i$. With $r_{\mathcal{S}} = (1/|\hat{\mathbf{r}}_i^{(k')}|)\sum \hat{\mathbf{r}}_i^{(k')}$ as the mean score in $\hat{\mathbf{r}}_i^{(k')}$, LIS selects the $|\mathcal{S}_{\text{train}}|$ instances for which $r_{\mathcal{S}}$ is *lowest*. This is similar to selecting instances with the highest $\hat{\mathbf{r}}_i$ entropy.

Highest Importance Scores (HIS) Given $r_{\mathcal{S}}$, HIS selects the $|\mathcal{S}_{\text{train}}|$ instances for which $r_{\mathcal{S}}$ is *highest*. This is the opposite of LIS.

5 Experiments

Now that ER-TEST lays a foundation for evaluation and design choices, we conduct a systematic study of the ER pipeline through *four* research questions (Fig. 3).

First, we compare different rationale alignment criteria and analyse which performs better for which task. (**RQ1**: Sec. 5.2). Second, we compare ER pipelines with different types of available human rationales: either dense, instance-level vs. distantly-supervised task-level rationales. (**RQ2**: Sec. 5.3) Third, we look into strategies on *how* to select instances on which ER should be applied, given resource constraints on the number of rationale-annotated samples. (**RQ3**: Sec. 5.4). Lastly, we investigate whether ER is worth doing, given a time-budget to obtain rationale-annotated instances, while comparing it methods without ER and the same time-budget to obtain more labelled data. (**RQ4**: Sec. 5.5).

5.1 Tasks and Datasets

ER-TEST uses a diverse set of text classification tasks. We mainly focus on sentiment analysis and natural language inference (NLI), but also consider named entity recognition (NER) and hate speech detection in Appendix A.5.2 First, for sentiment analysis, we use SST (short movie reviews) (Socher et al., 2013; Carton et al., 2020) as the ID dataset. As OOD datasets, we use Yelp (restaurant reviews) (Zhang et al., 2015), Amazon (product reviews) (McAuley and Leskovec, 2013), and Movies (long movie reviews) (Zaidan and Eisner, 2008; DeYoung et al., 2019). Second, for NLI, we use e-SNLI (Camburu et al., 2018; DeYoung et al., 2019) and MNLI (Williams et al., 2017) as the ID and OOD datasets, respectively. See Sec. A.3.1 for more details about datasets, dataset-specific contrast/functional tests, and other tasks.

5.2 *RQ1: Which rationale alignment criteria are most effective?*

Here, we study the effectiveness of different rationale alignment criteria specified in ER-TEST for two tasks - sentiment analysis and NLI.

Setup. Rationale alignment criteria described in Section 4.1 are used to align instance-level rationales for the train set (ID datasets SST and e-SNLI for sentiment analysis and NLI tasks respectively). For the NLM architecture, we use BigBird-Base (Zaheer et al., 2020), in order to handle input sequences of up to 4096 tokens. For all results, we report the mean over three seeds, as well as the standard deviation. We use a learning rate of $2\text{e}{-5}$ and effective batch size of 32. Further implementation details are in Appendix A.3.3.

Results. Tables 1 and 2, and Figure 4 summarize the results for this research question.

ID Generalization. We observe (Table 1) that using ID task performance, it is difficult to distinguish

Methods	Sentiment Analysis				NLI	
	In-Distribution	Out-of-Distribution			In-Distribution	Out-of-Distribution
	SST	Amazon	Yelp	Movies	e-SNLI	MNLI
None	94.22 (±0.77)	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)	76.18 (±1.28)	46.15 (±4.38)
IxG+MSE	94.29 (±0.05)	90.58 (±0.77)	92.17 (±0.64)	90.00 (±5.63)	78.98 (±1.00)	54.23 (±2.67)
IxG+MAE	94.11 (±0.38)	92.02 (±0.25)	94.55 (±0.30)	95.50 (±1.32)	78.77 (±1.01)	52.41 (±4.50)
IxG+BCE	94.15 (±0.53)	90.70 (±1.19)	91.82 (±2.30)	92.00 (±6.98)	79.07 (±0.83)	53.68 (±4.15)
IxG+Huber	94.19 (±0.19)	90.43 (±1.45)	92.38 (±2.11)	91.83 (±3.75)	78.99 (±0.81)	53.97 (±3.11)
IxG+Order	94.37 (±0.11)	89.47 (±2.71)	87.95 (±6.36)	84.50 (±10.15)	79.11 (±0.87)	55.26 (±3.56)
IxG+KLDiv	94.62 (±0.61)	91.63 (±0.51)	93.55 (±1.69)	93.00 (±2.18)	73.68 (±4.77)	46.57 (±1.35)
Attn+MSE	94.71 (±0.75)	91.88 (±0.53)	94.70 (±0.18)	95.83 (±1.15)	76.04 (±0.43)	48.60 (±2.55)
Attn+KLDiv	94.29 (±0.65)	91.43 (±0.71)	94.58 (±0.51)	96.67 (±0.76)	77.35 (±0.59)	49.66 (±2.47)
UNIREX	93.30 (±1.06)	83.27 (±6.43)	92.30 (±1.36)	93.00 (±0.50)	72.23 (±0.97)	42.92 (±3.35)

Table 1: **RQ1: Which rationale alignment criteria are most effective?**: Metrics displayed are Accuracy (↑) for sentiment analysis and Macro F1 (↑) for NLI. Cells highlighted in blue show significant improvement over the *None* baseline. ($p < 0.05$)

between different rationale alignment criteria, as all of them yield about the same task performance as the None baseline (for both SST and eSNLI).

Unseen Datasets. For sentiment analysis, *MAE* yields significant gains over all other rationale alignment criteria. However, despite performing best on SST, *Order* performs much worse than all other rationale alignment criteria. For NLI, *Order* loss leads to the highest performance with the MNLI dataset. Overall, OOD task performance is much better than ID at distinguishing between ER criteria, especially showing ER's improvement over None.

ER Criteria	Contrast Set					
	Sentiment Analysis			NLI		
	Original	Contrast	Δ	Original	Contrast	Δ
None	88.39 (±2.05)	85.11 (±2.72)	-3.28	46.15 (±4.38)	43.73 (±2.81)	-2.42
IxG+MSE	88.11 (±2.33)	86.07 (±2.48)	-2.04	54.23 (±2.67)	51.95 (±1.21)	-2.28
IxG+MAE	91.12 (±0.59)	89.82 (±1.20)	-1.30	52.41 (±4.50)	52.02 (±1.49)	-0.39
IxG+BCE	89.55 (±1.42)	87.30 (±4.03)	-2.25	53.68 (±4.15)	52.37 (±1.42)	-1.31
IxG+Huber	89.20 (±1.67)	86.13 (±1.74)	-3.07	53.97 (±3.11)	52.32 (±1.04)	-1.65
IxG+Order	86.00 (±5.27)	83.40 (±6.16)	-2.60	55.26 (±3.56)	52.78 (±0.74)	-2.48

Table 2: **RQ1 - Contrast Set Tests**: Δ (↓) is the difference in performance of $\mathcal{F}$ between the contrast and original set. A value farther from 0 suggests that $\mathcal{F}$ has learnt shortcuts specific a dataset, which are not generalizable. Cells highlighted in pink show least drop in performance with a contrast set.

Contrast Set Tests. We observe (Table 2) the drop in performance (Δ) for sentiment analysis and NLI when using a contrast set designed for the given dataset. *MAE* leads to the least drop in performance and all methods apart from *Order* yield lower drops than None. All of them also have a higher performance on the original and contrast sets. For sentiment analysis, we observe that *Order* has the highest variance, and for NLI, it has the highest drop in performance. We believe that some of it can be attributed to the soft-ranking that is imposed by *Order*, which may be indifferent towards minor label-changing edits, that is observed by the contrast sets.

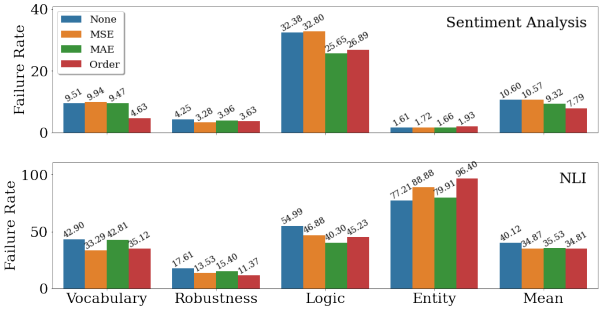

Figure 4: **RQ1 - Funtional Tests:** Shown here are Failure Rates (out of 100) (↓) for four functional tests (See. Table 11). Each rationale alignment criterion corresponds to the IxG rationale extractor.

Functional Tests. Figure 4 demonstrates *failure rates* on functional tests. We observe that apart from the entity-based tests, rationale alignment criteria generally have lower failure rates than *None*. Generally, all methods perform well on robustness-based tests, as they have lower failure rates, with order loss having the least. What is important to note is the significant improvement by Order loss in vocabulary-based tests than *None*, even though all of the methods are exposed to the same training set instances. We hypothesize that the biases induced by ER alleviates the shortcuts learnt by *None*, also demonstrated by an overall lower failure rate of rationale alignment criteria.

5.3 *RQ2: Is ER effective with distantly supervised human rationales?*

As described in Section 4.2, obtaining instance-level rationales are expensive to obtain. In this research question, we compare and contrast differences between instance-level and task-level human rationales with ER-TEST, on the Sentiment Analysis task.

Setup. In this RQ, we use the same setup for instance-level rationales as described in RQ 1 (Sec.

5.2). For generating task-level rationales, we merge the AFINN (Nielsen, 2011) and SenticNet (Cambria et al., 2020) lexicons.

Rationale Type	Criteria	Sentiment Analysis			
		In-Distribution	Out-of-Distribution		
		SST	Amazon	Yelp	Movies
None	-	94.22 (±0.77)	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)
Instance-level	IxG+MAE	94.69 (±0.93)	91.28 (±0.74)	93.28 (±2.16)	94.83 (±2.08)
	IxG+Huber	94.27 (±0.84)	91.17(±1.50)	93.40 (±1.45)	91.17 (±3.33)
Task-level	IxG+MAE	94.53 (±0.60)	**92.02 (±0.45)**	**94.10 (±0.91)**	**95.83 (±1.26)**
	IxG+Huber	93.81 (±0.47)	91.05 (±1.45)	93.88 (±0.41)	94.00 (±0.50)

Table 3: **RQ2**: Instance-level vs. Task-level rationales. (Sec. 5.3). All values are Accuracy (↑).

Results. We show ID and OOD performance of instance and task-based rationales in Table 3. Although ID performance (SST) is comparable for both rationale types, task-level rationales lead to minor improvements in OOD cases (Amazon, Yelp, Movies). We believe one of the reasons this boost in performance is observed is because the lexicon list used to generate rationales are task-specific (for sentiment analysis) and dataset-agnostic, and contain more general sentiment-related terms that are also present in OOD datasets, unlike instance-based rationales that contain nuances. However, having general lexicons that are not nuanced prohibits task-based models to generalise to harder instances which may not be resolved by lexicons themselves. This is observed by poor performance of task-based rationales in both contrast set and functional tests, when compared to instance-based rationales. (More in Sec. A.5.1)

5.4 *RQ3: How is ER affected by the number/choice of training instances with human rationales?*

So far, we have looked into ER when rationales (whether instance- or task-based) are available for all of the instances. However, it is important to determine *which* instances should be prioritized for human rationales annotation when annotation resources are limited. In this RQ, we investigate the performance of the different instance-selection methods outlined in Section 4.3, under varying resource constraints.

Setup. For experiments in this RQ, we compare the best performing rationale alignment criterion (IxG+MAE) and vary the instance budget (k) between 5/15/50/100%. (More in Sec. A.6.1)

Results. Table 4 demonstrates that importance-score based instance selection strategies (LIS, HIS) generally yield better improvements in OOD

k%	Method	Sentiment Analysis			
		In-Distribution	Out-of-Distribution		
		SST	Amazon	Yelp	Movies
None	-	94.22 (±0.77)	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)
100	-	94.11 (±0.38)	92.02 (±0.25)	94.55 (±0.30)	95.50 (±1.32)
5	Random	94.36 (±0.05)	91.57 (±0.10)	93.36 (±0.15)	92.39 (±2.50)
	LC	93.14 (±1.97)	90.72 (±0.43)	93.50 (±0.53)	93.17 (±1.26)
	HC	94.32 (±0.42)	91.57 (±0.19)	93.03 (±0.81)	91.33 (±3.09)
	LIS	93.92 (±1.07)	92.42 (±0.48)	94.28 (±0.31)	96.50 (±1.5)
	HIS	93.94 (±0.83)	90.58 (±0.95)	91.47 (±2.37)	92.00 (±4.58)
50	Random	93.47 (±0.02)	90.28 (±1.42)	91.85 (±2.11)	89.78 (±5.68)
	LC	87.07 (±5.15)	78.82 (±20.68)	77.73 (±26.53)	76.67 (±19.08)
	HC	92.93 (±0.17)	92.15 (±0.36)	94.48 (±0.94)	91.00 (±6.50)
	LIS	93.17 (±0.55)	90.60 (±0.25)	92.72 (±0.53)	93.50 (±0.87)
	HIS	94.23 (±0.65)	88.85 (±2.67)	91.47 (±1.47)	93.67 (±1.89)

Table 4: **Sample Selection Methods**: Settings mentioned in blue are significantly better than None settings. (See Appendix 17 for more details)

datasets when compared to label-confidence based strategies (LC, HC). Furthermore, certain instance selection criterion (like LC) perform significantly worse as the instance budget is increased. Zooming-into LIS, we compare it to random sampling and Non-ER settings in Figure 5. We can observe that for low resource cases (5/50%), LIS leads to similar OOD performance to $k = 100\%$ (using all samples for ER), and is always greater than Random/Non-ER. This also shows that carefully selecting a small subset of samples for rationale annotation can yield same benefits like that of annotating all the samples, with a lower annotation cost, and significant improvements over Random/No-ER.

5.5 *RQ4: How is ER affected by the time taken to annotate human rationales?*

So far, we explored questions surrounding ER that assumed the ease of obtaining rationale-annotated instances. However, obtaining rationales for ER are not only tedious, but also time-consuming. In this RQ, we benchmark the time efficiency of ER through the lens of time taken to collect such data, when compared to collecting labelled data without rationales.

Setup. Our setup comprises of two steps - firstly, estimating the time taken to annotate one instance using Amazon Mechanical Turk (MTurk), followed by using these estimates to create training sets with varying time budgets. On MTurk[2] (details in A.7), we devise *three* tasks – one where the annotators have to first select a sentiment for an instance, and then highlight rationale tokens that support their selected sentiment (***Label + Expl***), one where they have to highlight the rationales given a ground truth sentiment (***Only Expl***) and one baseline task where they only have to label an instance with a sentiment

[2] https://www.mturk.com/

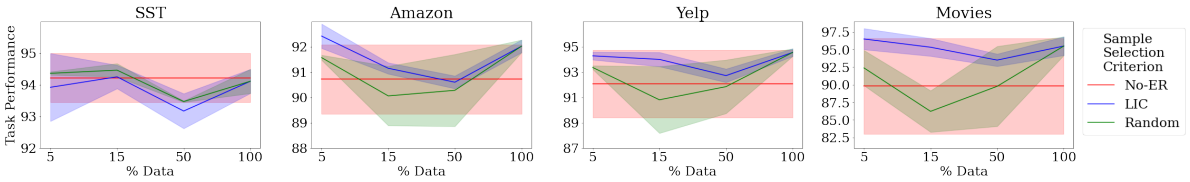

Figure 5: **RQ3: How is ER affected by the number/choice of training instances with human rationales?**: Task Performance (Accuracy) vs. % of rationale-annotated data for different sample selection criteria on four sentiment analysis datasets.

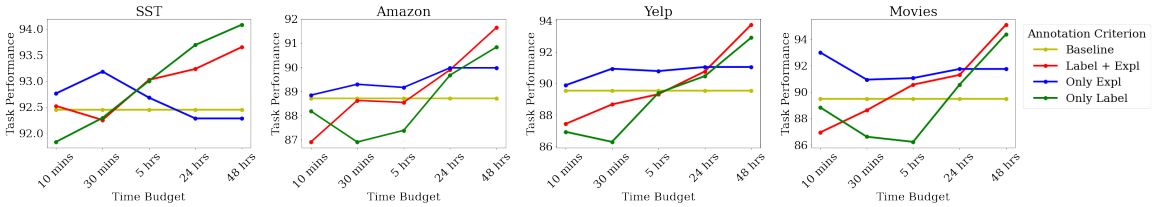

Figure 6: **RQ4: How is ER affected by the time taken to annotate human rationales?**: Task Performance (Accuracy) vs. time budget for rationale annotation, for each kind of annotation strategy, for each of the four sentiment analysis datasets. There are 1000 instances in the baseline training set, and 1 hr of annotation corresponds to 42, 98 and 77 instances each for ***Label + Expl***, ***Only Expl*** and ***Only Label*** annotation tasks respectively. Annotation is done on ID dataset (SST) only.

(***Only Label***). For each of the above tasks, through our MTurk experiments, our 178 Turkers yielded mean/std times of $140.56s \pm 8.45$ (***Only Label***), $110.31s \pm 3.21$ (***Only Expl***), and $263.10s \pm 7.31$ (***Label+Expl***). By filtering out 'cheaters' who submitted empty/low-effort responses, we achieved high inter-annotator agreement. For ***Only Label*** and ***Label+Expl***, the Fleiss' kappa scores were 0.74 and 0.70. For ***Only Expl*** and ***Label+Expl***, the rationale overlap rates (Zaidan and Eisner, 2008) were 0.78 and 0.66. We replicated this experiment in a small-scale study with nine CS students and observed similar trends.

Using these time estimates, we devise three experiment settings. Given a baseline labelled training set $\mathcal{D}_{base}$ of 1000 instances and a time budget $\mathcal{T}$, we can - 1) Add human rationales for a subset $\mathcal{S}^{\mathcal{T}}_{expl}$ of $\mathcal{D}_{base}$, 2) Add new instances $\mathcal{D}^{\mathcal{T}}_{label}$ with only labels to $\mathcal{D}_{base}$, or 3) Add new instances $\mathcal{D}^{\mathcal{T}}_{label+expl}$ with labels and rationales to $\mathcal{D}_{base}$. Note that, the ***number*** of new instances added in each of the above experiment settings depend on the time taken to annotate the ***Only Expl***, ***Only Label*** and ***Label + Expl*** tasks respectively.

Results. As we can observe in Figure 6, when provided with a lower time budget for annotation ($\leq$ 5 hours), annotating rationales for existing instances in the training set (***Only Expl***) yield improvements over adding new instances with labels (and rationales), in all of the OOD datasets. However, their performance saturates over time. When provided with a higher time-budget, adding new instances with both labels and their rationales (***Label + Expl***) is better than only adding labelled data without rationales (***Only Label***). This is even though ***Label + Expl*** takes the most amount of time to annotate, and thus fewer instances with these annotations would be added with a given time budget. In general, we observe that about 24 hours of ***Only Label*** annotation yields the same OOD performance with just 30 mins of ***Only Expl*** annotation. This validates that ER not only provides improvements in generalization, but also does it in a time- (and cost-) efficient manner.

6 Related Work

Explanation-Based Learning Many methods have been proposed for explanation-based learning (Hase and Bansal, 2021; Hartmann and Sonntag, 2022), especially using human explanations (Tan, 2022). ER, which is based on machine-human rationale alignment, is a common paradigm for learning from human explanations. In ER, the human rationale can be obtained by annotating each instance individually (Zaidan and Eisner, 2008; Lin et al., 2020; Camburu et al., 2018; Rajani et al., 2019; DeYoung et al., 2019) or by applying domain-level lexicons across all instances (Rieger et al., 2020; Ross et al., 2017; Ghaeini et al., 2019; Kennedy et al., 2020; Liu and Avci, 2019). Existing choices of rationale alignment criteria include MSE (Liu and Avci, 2019; Kennedy et al., 2020; Ross et al., 2017), MAE (Rieger et al., 2020), BCE (Chan et al., 2021), order loss (Huang et al., 2021), and KL divergence (Chan et al., 2021). Beyond ER, there are other ways to learn from explanations. Lu et al. (2022) used human-in-the-loop feedback on machine rationales for data augmentation. Meanwhile, Ye and Durrett (2022) used machine rationales to

calibrate black box models and improve their performance on low-resource domains.

Evaluating ER Existing works have primarily evaluated ER models via ID generalization (Zaidan and Eisner, 2008; Lin et al., 2020; Huang et al., 2021), which only captures one aspect of ER's impact. Meanwhile, a few works have considered auxiliary evaluations — *e.g.,* machine-human rationale alignment (Huang et al., 2021; Ghaeini et al., 2019), task performance on unseen datasets (Ross et al., 2017; Kennedy et al., 2020), social group fairness (Rieger et al., 2020; Liu and Avci, 2019). Carton et al. (2022) showed that maximizing machine-human rationale alignment does not always improve task performance, while human rationales vary in their ability to provide useful information for task prediction.

7 Conclusion and Future Work

In this work, we study explanation regularization (ER) – aligning machine rationales with human rationales, in detail. We propose ER-TEST, that evaluates ER's OOD generalization along three pillars - unseen datasets, contrast set tests and functional tests, and uses it to investigate four research questions surrounding the choice of the rationale alignment criterion, type of human rationale, choice of and time taken to obtain rationale-annotation instances. Although ER shows minor impact on ID task performance, improvements on OOD datasets is significant. Furthermore, ER not only works well with dense, instance-level human rationales, but also with distantly supervised task-level rationales. Lastly, ER is shown to provide benefits even with limited number of rationale-annotated instances, or within time constraints for rationale annotation. In future, we aim to study ER as a tool for improving human-in-the-loop (HITL) debugging of NLMs. Furthermore, currently ER-TEST is only defined for extractive rationales. Human feedback for free-text machine rationales is also a promising extension for ER-TEST.

8 Acknowledgments

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via Contract No. 2019-19051600007, DARPA MCS program under Contract No. N660011924033, NSF IIS 2048211, and gift awards from Google and Amazon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. We would also like to thank all of our collaborators at the USC INK Research Lab for their constructive feedback on this work.

9 Limitations

Some tests defined by ER-TEST might not be applicable for all NLP tasks. It is difficult to define tests like contrast set tests for certain NLP tasks like NER. Furthermore, currently these tests have been picked up from their respective releases, however, they are extremely tedious to design and generate for new tasks and datasets.

Current work simulates human rationales in the ER pipeline. ER is meant to align machine-generated rationales with human-rationales. However in our current work, we use human rationales that are pre-annotated as part of the datasets we use. This simulation of live human feedback is used in rationale alignment criterion. We believe this limitation can be easily addressed, by collecting human-in-the-loop rationale annotations.

Current work assumes ER pipelines to be offline in nature. Fine-tuning strategies have shown to distort the underlying data distribution (Kumar et al., 2022), therefore, once $\mathcal{F}$ undergoes ER, its machine rationales can differ from before. Currently, ER is being studied in an offline manner – once human rationales are collected, they are used to update model weights. However, what is more effective is to study the effect of ER when applied incrementally in an online manner, thus improving rationale alignment.

10 Ethics Statement

Data. All the datasets that we use in our work are released publicly for usage and have been duly attributed to their original authors.

User Study. As part of our user study conducted in Section 5.5, we collected information about the time taken to annotate rationales and labels for instances. We provide the instructions given to MTurkers in Appendix A.7, along with screenshots

of the UI displayed to them. Further details about the task setup and results are provided in Section 5.5. Each task is setup in a manner that ensure that the annotators receive compensation that is above minimum wage.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Erik Cambria, Yang Li, Frank Z. Xing, Soujanya Poria, and Kenneth Kwok. 2020. Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20, page 105–114, New York, NY, USA. Association for Computing Machinery.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *arXiv preprint arXiv:1812.01193*.

Samuel Carton, Surya Kanoria, and Chenhao Tan. 2022. What to learn, and how: Toward effective learning from rationales. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1075–1088, Dublin, Ireland. Association for Computational Linguistics.

Samuel Carton, Anirudh Rathore, and Chenhao Tan. 2020. Evaluating and characterizing human rationales. *arXiv preprint arXiv:2010.04736*.

Aaron Chan, Maziar Sanjabi, Lambert Mathias, Liang Tan, Shaoliang Nie, Xiaochang Peng, Xiang Ren, and Hamed Firooz. 2022. Unirex: A unified learning framework for language model rationale extraction. *International Conference on Machine Learning*.

Aaron Chan, Jiashu Xu, Boyuan Long, Soumya Sanyal, Tanishq Gupta, and Xiang Ren. 2021. Salkg: Learning from knowledge graph explanations for commonsense reasoning. *Advances in Neural Information Processing Systems*, 34.

Cheng-Han Chiang and Hung-yi Lee. 2022. Re-examining human annotations for interpretable nlp.

George Chrysostomou and Nikolaos Aletras. 2022. An empirical study on explanations in out-of-domain settings.

Ona de Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. 2018. Hate speech dataset from a white supremacy forum. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 11–20, Brussels, Belgium. Association for Computational Linguistics.

Misha Denil, Alban Demiraj, and Nando De Freitas. 2014. Extraction of salient sentences from labelled documents. *arXiv preprint arXiv:1412.6815*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. Eraser: A benchmark to evaluate rationalized nlp models. *arXiv preprint arXiv:1911.03429*.

Shuoyang Ding and Philipp Koehn. 2021. Evaluating saliency methods for neural language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5034–5052, Online. Association for Computational Linguistics.

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

Hang Gao and Tim Oates. 2019. Universal adversarial perturbation for text classification.

Matt Gardner, Yoav Artzi, Victoria Basmova, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. *arXiv preprint arXiv:2004.02709*.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Reza Ghaeini, Xiaoli Z Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. Saliency learning: Teaching the model where to pay attention. *arXiv preprint arXiv:1902.08649*.

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.

Mareike Hartmann and Daniel Sonntag. 2022. A survey on improving NLP models with human explanations. In *Proceedings of the First Workshop on Learning with Natural Language Supervision*, pages 40–47, Dublin, Ireland. Association for Computational Linguistics.

Peter Hase and Mohit Bansal. 2021. When can models learn from explanations? a formal framework for understanding the roles of explanation data. *arXiv preprint arXiv:2102.02201*.

Quzhe Huang, Shengqi Zhu, Yansong Feng, and Dongyan Zhao. 2021. Exploring distantly-labeled rationales in neural network models. *arXiv preprint arXiv:2106.01809*.

Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer.

Xisen Jin, Francesco Barbieri, Brendan Kennedy, Aida Mostafazadeh Davani, Leonardo Neves, and Xiang Ren. 2021. On transferability of bias mitigation effects in language model fine-tuning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3770–3783, Online. Association for Computational Linguistics.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2019. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *arXiv preprint arXiv:1911.06194*.

Erik Jones, Robin Jia, Aditi Raghunathan, and Percy Liang. 2020. Robust encodings: A framework for combating adversarial typos. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2752–2765, Online. Association for Computational Linguistics.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*.

Brendan Kennedy, Mohammad Atari, Aida M Davani, Leigh Yeh, Ali Omrani, Yehsong Kim, Kris Coombs, Shreya Havaldar, Gwenyth Portillo-Wightman, Elaine Gonzalez, and et al. 2018. Introducing the gab hate corpus: Defining and applying hate-based rhetoric to social media posts at scale.

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. *arXiv preprint arXiv:2005.02439*.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution.

Chuanrong Li, Lin Shengshuo, Zeyu Liu, Xinyi Wu, Xuhui Zhou, and Shane Steinert-Threlkeld. 2020. Linguistically-informed transformations (LIT): A method for automatically generating contrast sets. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 126–135, Online. Association for Computational Linguistics.

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

Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. 2020. Triggerner: Learning with entity triggers as explanations for named entity recognition. *arXiv preprint arXiv:2004.07493*.

Zachary C Lipton. 2018. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Frederick Liu and Besim Avci. 2019. Incorporating priors with feature attribution on text classification. *arXiv preprint arXiv:1906.08286*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jinghui Lu, Linyi Yang, Brian Namee, and Yue Zhang. 2022. A rationale-centric framework for human-in-the-loop machine learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6986–6996, Dublin, Ireland. Association for Computational Linguistics.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777.

Siwen Luo, Hamish Ivison, Caren Han, and Josiah Poon. 2021. Local interpretations for explainable natural language processing: A survey. *arXiv preprint arXiv:2103.11072*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Ninareh Mehrabi, Thamme Gowda, Fred Morstatter, Nanyun Peng, and Aram Galstyan. 2020. *Man is to Person as Woman is to Location: Measuring Gender Bias in Named Entity Recognition*, page 231–232. Association for Computing Machinery, New York, NY, USA.

Shubhanshu Mishra, Sijun He, and Luca Belli. [link].

F. Å. Nielsen. 2011. Afinn.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Danish Pruthi, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C Lipton, Graham Neubig, and William W Cohen. 2020. Evaluating explanations: How much do explanations from the teacher aid students? *arXiv preprint arXiv:2012.00893*.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of nlp models with checklist. *arXiv preprint arXiv:2005.04118*.

Laura Rieger, Chandan Singh, William Murdoch, and Bin Yu. 2020. Interpretations are useful: penalizing explanations to align neural networks with prior knowledge. In *International conference on machine learning*, pages 8116–8126. PMLR.

Andrew Slavin Ross, Michael C Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. *arXiv preprint arXiv:1703.03717*.

Sebastian Ruder. 2021. Challenges and Opportunities in NLP Benchmarking. http://ruder.io/nlp-benchmarking.

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

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10285–10299, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Christopher Schröder and Andreas Niekler. 2020. A survey of active learning for text classification using deep neural networks.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Aarne Talman and Stergios Chatzikyriakidis. 2018. Testing the generalization power of neural network models across nli benchmarks.

Chenhao Tan. 2022. On the diversity and limits of human explanations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2173–2188, Seattle, United States. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Tianlu Wang, Xuezhi Wang, Yao Qin, Ben Packer, Kang Li, Jilin Chen, Alex Beutel, and Ed Chi. 2020. CAT-gen: Improving robustness in NLP models via controlled adversarial text generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5141–5146, Online. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Xi Ye and Greg Durrett. 2022. Can explanations be useful for calibrating black box models? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6199–6212, Dublin, Ireland. Association for Computational Linguistics.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. In *NeurIPS*.

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.

Matthew D Zeiler and Rob Fergus. 2013. Visualizing and understanding convolutional networks.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Zhiqiang Zheng and B. Padmanabhan. 2002. On active learning for data acquisition. In *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pages 562–569.

A Appendix

A.1 Section 2 Appendix

$\mathcal{G}$ first computes raw importance scores $\mathbf{s}_i \in \mathbb{R}^n$, then normalizes $\mathbf{s}_i$ as probabilities $\mathbf{r}_i$ using the sigmoid function.

Broadly, heuristic $\mathcal{G}$'s can either be a gradient-based, that assign importance scores based on gradient changes in $\mathcal{F}$ (Sundararajan et al., 2017; Sanyal and Ren, 2021; Shrikumar et al., 2017), sampling-based, that assign important scores based on the neighbours/context of a given token (Zeiler and Fergus, 2013; Jin et al., 2019), or attention-based, that use the attention-scores or a function of them to assign importance scores. (Ding and Koehn, 2021).

A.2 Section 3 Appendix

A.2.1 ID Generalization

While ER-TEST's main focus is on evaluating OOD generalization, ER-TEST also considers ID generalization as a baseline evaluation. Let $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}_{i=1}^{N}$ be a M-class text classification dataset, where $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ are the text inputs, $\mathcal{Y} = \{\dot{y}_i\}_{i=1}^{N}$ are the target classes, and N is the number of instances $(\mathbf{x}_i, \dot{y}_i)$ in $\mathcal{D}$. We call $\mathcal{D}$ the ID dataset. Assume $\mathcal{D}$ can be partitioned into train set $\mathcal{D}_{\text{train}}$, dev set $\mathcal{D}_{\text{dev}}$, and test set $\mathcal{D}_{\text{test}}$, where $\mathcal{D}_{\text{test}}$ is an ID test set for $\mathcal{D}$. After using ER to train $\mathcal{F}$ on $\mathcal{D}_{\text{train}}$, we measure $\mathcal{F}$'s task performance on the ID test set $\mathcal{D}_{\text{test}}$. Note that this is a standard protocol used by existing works to evaluate ER models (Zaidan and Eisner, 2008; Rieger et al., 2020; Liu and Avci, 2019; Ross et al., 2017; Huang et al., 2021; Ghaeini et al., 2019; Kennedy et al., 2020)

A.2.2 Contrast Sets

Given $\tilde{\mathcal{D}}_{\text{test}}^{(i)}(j)$ (a j^{th} instance belonging to an OOD test set $\tilde{\mathcal{D}}_{\text{test}}^{(i)}$), a perturbation function $\beta_p^{(i)}$ is applied to that instance, where p denotes the kind of perturbation taking effect, and it often changes the target label for that instance. For example, p can signify semantic (*e.g.,* changing *tall* to *short*), numeral (*e.g.,* changing *one dog* to *three dogs*), or entities (*e.g.,* changing *dogs* to *cats*). Each perturbation type is specific to the dataset it is being created for, so that instance labels are changed in a meaningful manner. The resulting set of instances $\mathcal{C}^{\langle\rangle} = \beta_p^{(i)}(\tilde{\mathcal{D}}_{\text{test}}^{(i)}(j))\forall j, p$ are termed as a *contrast set* for that dataset. Based on the way they are created, contrast sets are a property of the dataset, and are not created to explicitly challenge $\mathcal{F}$ (unlike adversarial examples (Gao and Oates, 2019)).

A.2.3 Functional Tests

Vocabulary Tests Vocabulary tests are used to evaluate $\mathcal{F}$'s capability to address changes in the vocabulary of the text, and is particularly diverse w.r.t the parts-of-speech it caters to. For example, certain vocabulary tests evaluate the relationship (taxonomy) between different nouns in a sentence, whereas some swap the modifiers or the verbs present in a sentence in a meaningful manner based on the task at hand, to capture $\mathcal{F}$'s targeted performance towards such changes (Ribeiro et al., 2020).

Robustness Tests Robustness tests evaluate $\mathcal{F}$'s behavior under character-level edits to words in a sentence, keeping the rest of the context same so as to not change the overall prediction. They include testing against typos as well as contractions in words, as well as addition of tokens that are irrelevant for the downstream task (like URLs or gibberish like Twitter handles). (Jones et al., 2020; Wang et al., 2020)

Logic Tests Testing $\mathcal{F}$'s reasoning capabilities towards logical changes in a sentence is also important to evaluate its reliance on shortcut-patterns. These tests perturb sentences in a logical manner (by adding or removing negations, or purposefully

inducing contradictions) that also change the target label in the same manner. (Talman and Chatzikyriakidis, 2018; McCoy et al., 2019)

Entity Tests For certain tasks, named entities like numbers, locations and proper nouns are not relevant for predicted a target label, and are often a source of gender or demographic biases (Mishra et al.; Mehrabi et al., 2020). Entity tests measure $\mathcal{F}$'s sensitivity towards changes in named entities such that the overall context as well as the task label remains the same (Ribeiro et al., 2020).

A.3 Section 4 Appendix

A.3.1 Tasks and Datasets

To evaluate ER models, ER-TEST considers a diverse set of sequence and token classification tasks. For each, task ER-TEST provides one ID dataset (annotated with human rationales) and multiple OOD datasets. Compared to prior works, ER-TEST's task/dataset diversity enables more extensive analysis of ER model generalization.

First, we have sentiment analysis, using SST (movie reviews) (Socher et al., 2013; Carton et al., 2020) as the ID dataset. For OOD datasets, we use Yelp (restaurant reviews) (Zhang et al., 2015), Amazon (product reviews) (McAuley and Leskovec, 2013), and Movies (movie reviews) (Zaidan and Eisner, 2008; DeYoung et al., 2019). Movies' inputs are much longer than the other three datasets'. For contrast set tests, we use an OOD contrast set for sentiment analysis released by the authors of the original paper (Gardner et al., 2020), which are created for the Movies dataset. Furthermore, for functional tests, we use an OOD test suite (flight reviews) from the `CheckList` (Ribeiro et al., 2020) which contains both template-based instances to test linguistic capabilities, as well as real-world data (tweets).

Second, we have natural language inference (NLI), using e-SNLI (Camburu et al., 2018; DeYoung et al., 2019) as the ID dataset. For the OOD dataset, we use MNLI (Williams et al., 2017). e-SNLI contains only image captions, while MNLI contains both written and spoken text, covering various topics, styles, and formality levels. For NLI, we also use an OOD contrast set created for the MNLI dataset (Li et al., 2020). Functional tests for NLI are generated from the `AllenNLP` test suite (Gardner et al., 2017) for textual entailment.

Third, we have named entity recognition (NER), using CoNLL-2003 (Sang and De Meulder, 2003; Lin et al., 2020) as the ID dataset. For the OOD dataset, we use OntoNotes v5.0 (Pradhan et al., 2013). CoNLL-2003 contains only Reuters news stories, while OntoNotes v5.0 contains text from newswires, magazines, telephone conversations, websites, and other sources.

A.3.2 Intrinsic Evaluation of ER

ER in general is sensitive to certain hyperparameters for yielding meaningful training curves and actually attaining alignment between machine and human rationales. Due to a large set of tunable hyperparameters, running all configurations of ER are not feasible. Therefore, we intrinsically evaluate hyperparameter configurations by assessing the loss curves (which model alignment between machine and human rationales) w.r.t different hyperparameters values. We observe that the acceptable band of learning rates for ER is very narrow, and we use $2\text{e}{-5}$ in all of our experiments. Furthermore, we also observe that setting $\lambda_{\text{ER}} = 1$ and $\gamma_{\text{ER}} = 100$ yields the most drop in the loss curves while training, so we use these hyperparameters for the rest of our experiments. We detail these experiments in Appendix A.3.3.

A.3.3 Intrinsic Evaluation: evaluating ER's sensitivity to hyperparameters

When using ER to train $\mathcal{F}$, it is important to assess whether ER exhibits expected training behavior, orthogonally to task performance. If ER improves task performance, this kind of analysis can help us better understand ER's effectiveness. Conversely, if ER does not improve task performance, such analysis can help us identify the problem.

Let $\gamma_{\text{ER}} > 0$ be the *rationale scaling factor*, used to scale $\hat{\mathbf{s}}_i$ prior to sigmoid normalization. If the magnitudes of the $\hat{\mathbf{s}}_i$ scores are lower, then the $\hat{\mathbf{r}}_i$ scores will be closer to 0.5 (*i.e.,* lower confidence). However, scaling $\hat{\mathbf{s}}_i$ by $\gamma_{\text{ER}} > 1$ will increase the magnitude of $\hat{\mathbf{s}}_i$, yielding $\hat{\mathbf{r}}_i$ scores closer to 0 or 1 (*i.e.,* higher confidence).

Motivated by this, ER-TEST's intrinsic evaluation is based on *machine-human rationale alignment*, captured by the ER loss $\mathcal{L}_{\text{ER}} = \Phi(\hat{\mathbf{r}}_i, \dot{\mathbf{r}}_i)$. When using ER, we should generally expect the ER loss to decrease as $\mathcal{F}$ is trained. In practice, this may not always be the case, even when ER leads to slightly higher task performance (which is likely a mirage caused by lucky random seeds)! That is, by definition, non-decreasing ER loss signals ineffective ER usage, since the machine rationales

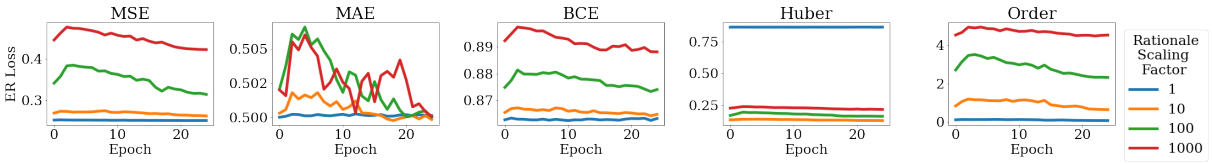

Figure 7: **ER Loss Curves (Rationale Scaling Factor).** For rationale extractor, we use IxG

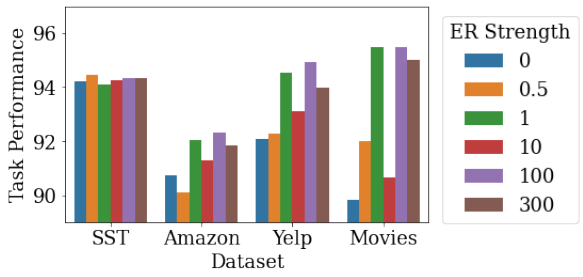

Figure 8: **ER Strength *vs.* Task Performance.** For various combinations of sentiment analysis dataset and ER strength, we plot task performance using IxG+MAE.

	ER Strength				
ER criteria	0.5	1	10	100	300
IxG+MSE	0.91	**1.52**	1.41	1.29	1.35
IxG+MAE	1.89	**2.01**	1.72	1.80	1.74
IxG+BCE	1.99	**2.17**	1.65	1.65	1.75
IxG+Huber	1.85	2.09	2.24	2.27	**2.40**
IxG+Order	2.15	2.40	1.60	**2.53**	1.89

Table 5: **Relative Decrease in ER Loss.** For various ER strengths, we report the percentage decrease in ER train loss (on SST), from max point to min point.

are not becoming more similar to the human rationales. This can stem from a number of issues: *e.g.,* poor choice of ER criteria Φ, improper ER strength λ_{ER}, improper rationale scaling factor γ_{ER}, noisy human rationale $\dot{\mathbf{r}}_i$, insufficient $\mathcal{F}$ capacity. Thus, we measure machine-human rationale alignment as the first step in diagnosing such issues. Let *ER loss curve* denote a chart which plots $\mathcal{L}_{\text{ER}}$ *vs.* the number of train epochs. For each combination of ER criteria Φ and some training configuration, we plot ER loss curves for the training set. Each component of our intrinsic evaluation varies a different hyperparameter in the training configuration: (A) ER strength λ_{ER}; (B) rationale scaling factor γ_{ER}; and (C) learning rate α. In contrast, prior works do not explore the relationship between $\mathcal{L}_{\text{ER}}$ and these training variables (Huang et al., 2021; Ghaeini et al., 2019).

For intrinsic evaluation, we use ER strength $\lambda_{\text{ER}} = 1$, rationale scaling factor $\gamma_{\text{ER}} = 1$, and learning rate $\alpha = 2\text{e}{-5}$, unless otherwise specified. As a proof of concept, we focus on SST here, but plan to add other datasets in future work.

A.3.4 Misc. Details

All models are trained on GeForce GTX 1080 Ti and Quadro RTX 6000 GPUs. All implementations are done using the HuggingFace API (Wolf et al., 2019).

A.3.5 ER Strength

Fig. 10 displays the ER loss curves for different ER strengths $\lambda_{\text{ER}} = [0.5, 1, 10, 100, 300]$, on SST using MAE. Among the λ_{ER} values, we see that $\lambda_{\text{ER}} = 1$ yields ER loss curves with the greatest decrease (Table 5), signaling good ER optimization.

A.3.6 Rationale Scaling Factor

Fig. 7 displays the ER loss curves for different rationale scale factors $\gamma_{\text{ER}} = [1, 10, 100, 1000]$, on SST. Among the four γ_{ER} values, we see that $\gamma_{\text{ER}} = 100$ yields ER loss curves with the greatest decrease (Table 6), signaling good ER optimization. Meanwhile, although ER works use $\gamma_{\text{ER}} = 1$ by default, we see that $\gamma_{\text{ER}} = 1$ yields nearly flat ER loss curves for all five Φ choices. This suggests poor ER optimization. Based on these results, we fix $\gamma_{\text{ER}} = 100$ for all experiments (Sec. 5), thus greatly reducing the hyperparameter search space.

A.3.7 Learning Rate

Here, we obtain similar conclusions, with $\alpha = 2\text{e}{-5}$ yielding the best ER loss curves. Fig. 11 displays the ER loss curves for different learning rates $\alpha = [2\text{e}{-6}, 2\text{e}{-5}, 2\text{e}{-4}]$. Among the three learning rates, we see that $\alpha = 2\text{e}{-5}$ yields the most steadily decreasing ER loss curves.

A.3.8 ER performance with different hyperparameters

ER Strength *vs.* Task Performance To measure ER's impact on task performance, we plot $\mathcal{F}$'s task performance as a function of ER strength λ_{ER}. This is conducted for ID test sets.

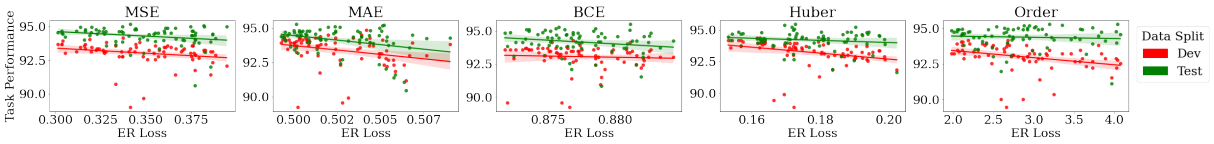

Figure 9: **Task Performance vs. ER Loss**. Here we use IxG as rationale extractor

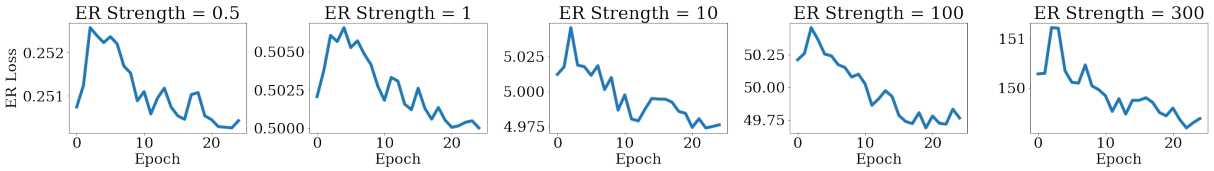

Figure 10: **ER Loss Curves (ER Strength).** Here, we use the MAE criterion and IxG as rationale extractor

	Rationale Scaling Factor			
ER criteria	1	10	100	1000
IxG+MSE	0.69	4.60	**18.35**	11.41
IxG+MAE	0.04	0.40	**1.29**	1.17
IxG+BCE	0.10	0.34	0.90	**1.03**
IxG+Huber	0.10	7.75	**16.67**	9.30
IxG+Order	7.21	9.38	**47.97**	**1.89**

Table 6: **Relative Decrease in ER Loss.** For various ER rationale scaling factors, we report the percentage decrease in ER train loss (on SST), from max point to min point.

	Dev		Test	
ER criteria	Slope (↓)	R^2 (↑)	Slope (↓)	R^2 (↑)
IxG+MSE	-7.48	0.050	-6.75	0.059
IxG+MAE	**-128.60**	0.083	**-133.03**	**0.110**
IxG+BCE	-17.48	0.003	-56.30	0.040
IxG+Huber	-23.59	0.091	-8.40	0.022
IxG+Order	-0.49	**0.101**	-0.085	0.004

Table 7: **ER Loss *vs.* Task Performance.** We summarize the line plots in Fig. 9 (ER Loss *vs.* Task Performance), using slope and R^2 score (Sec. A.3.8). Ideally, Fig. 9's lines would have *low slope* and *high* R^2, indicating that ER helps improve task performance. We see that MAE yields the best ER results.

For each sentiment analysis dataset, Fig. 8 shows task performance for ER strengths $\lambda_{\text{ER}} = [0, 0.5, 1, 10, 100, 300]$, using MAE. Note that $\lambda_{\text{ER}} = 0$ is equivalent to training the NLM without ER (*i.e.,* None in Table 1). For the ID dataset (SST), we see that all ER strengths yield very similar task performance, suggesting that ER has little effect on ID task performance. However, for the OOD datasets (Amazon, Yelp, Movies), task performance generally increases as λ_{ER} increases, showing ER's positive impact on NLM generalization. Overall, based on OOD task performance, we find that $\lambda_{\text{ER}} = [1, 100]$ are the best ER strengths. This aligns with the results of Sec. A.3.5.

	Sentiment Analysis (Out-of-Domain)			
ER criteria	Amazon	Yelp	Movies	Mean
None	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)	90.87 (± 3.60)
IxG+MAE ($\lambda_{\text{ER}} = 0.5$)	90.12 (±2.98)	92.27 (±3.29)	92.00 (±5.68)	91.46 (±0.91)
IxG+MAE ($\lambda_{\text{ER}} = 1$)	92.02 (±0.25)	94.55 (±0.30)	95.50 (±1.32)	94.02 (±2.15)
IxG+MAE ($\lambda_{\text{ER}} = 10$)	91.27 (±0.28)	93.10 (±1.08)	90.67 (±3.79)	91.68 (±1.06)
IxG+MAE ($\lambda_{\text{ER}} = 100$)	**92.33 (±0.28)**	**94.92 (±0.56)**	**95.50 (±0.50)**	**94.25 (±1.89)**
IxG+MAE ($\lambda_{\text{ER}} = 300$)	91.83 (±0.42)	93.97 (±1.28)	95.00 (±0.50)	93.60 (±1.74)
IxG+MAE ($\gamma_{\text{ER}} = 1$)	90.63 (±1.88)	92.32 (±2.23)	88.67 (±4.25)	90.54 (±2.22)
IxG+MAE ($\gamma_{\text{ER}} = 10$)	**92.30 (±1.21)**	93.01 (±2.14)	**96.83 (±1.04)**	**94.07 (±3.89)**
IxG+MAE ($\gamma_{\text{ER}} = 100$)	92.02 (±0.25)	**94.55 (±0.30)**	95.50 (±1.32)	94.02 (±2.15)
IxG+MAE ($\gamma_{\text{ER}} = 1000$)	90.47 (±2.06)	92.80 (±2.90)	92.67 (±6.25)	91.98 (±1.14)
IxG+MAE ($\alpha = 2\text{e}{-4}$)	89.35 (±2.85)	91.23 (±2.84)	93.00 (±2.65)	91.19 (±2.22)
IxG+MAE ($\alpha = 2\text{e}{-5}$)	**92.02 (±0.25)**	**94.55 (±0.30)**	**95.50 (±1.32)**	**94.02 (±2.15)**
IxG+MAE ($\alpha = 2\text{e}{-6}$)	88.60 (±1.60)	83.27 (±6.49)	81.17 (±6.93)	84.34 (±9.70)

Table 8: **Task Performance *vs.* {ER Strength (λ_{ER}), Rationale Scaling Factor (γ_{ER})}.** Higher values are better.

ER Loss *vs.* Task Performance To measure ER's impact on task performance, we plot $\mathcal{F}$'s task performance as a function of ER loss $\mathcal{L}_{\text{ER}}$. This is conducted for both ID and OOD test sets.

Fig. 9 displays the SST results for ID task performance (accuracy) *vs.* ER loss. For a given ER criterion, each point in the corresponding scatter plot represents the checkpoint at some train epoch of the ER-trained model, evaluated on either the dev set or test set (yielding two point sets). Fitting each point set with linear regression, we find that there is an inverse relationship between task performance and ER loss. In other words, higher machine-human rationale alignment (*i.e.,* low ER loss) corresponds to higher task performance, which validates the usage of ER to improve generalization. Table 7 displays the slopes and R^2 scores of the lines in Fig. 9. The slope indicates the strength of the relationship between machine-human rationale alignment and task performance (lower is better), while the R^2 score indicates how accurately each line fits its corresponding data points. Among the five ER criteria, across dev and test, we find that MAE has the lowest slopes and highest R^2 scores overall, suggesting that using ER with MAE is most effective.

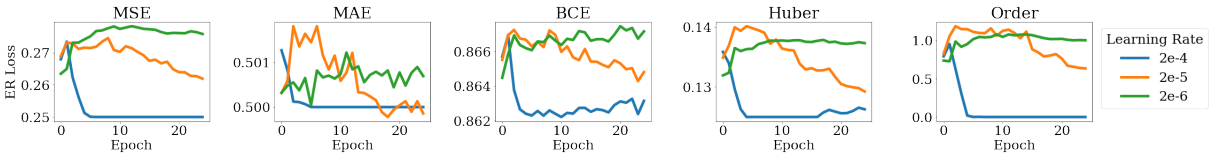

Figure 11: **ER Loss Curves (Learning Rate)**. Here we use IxG as rationale extractor

ER criteria	Sentiment Analysis			
	In-Domain	Out-of-Domain		
	SST	Amazon	Yelp	Movies
None	0.00 (±0.00)	0.00 (±0.00)	0.00 (±0.00)	0.00 (±0.00)
IxG+MSE	0.32 (±1.05)	**-1.25 (±1.20)**	**-2.33 (±4.64)**	-6.50 (±40.66)
IxG+MAE	-0.09 (±0.24)	-0.58 (±3.45)	-0.94 (±11.21)	**-7.00 (±40.66)**
IxG+BCE	**-0.16 (±0.33)**	0.46 (±4.11)	0.96 (±26.99)	0.16 (±47.72)
IxG+Huber	0.12 (±0.42)	0.19 (±2.25)	-1.05 (±4.11)	-4.33 (±37.72)
IxG+Order	1.90 (±1.38)	6.98 (±3.87)	19.86 (±45.54)	21.66 (±35.72)

Table 9: **ID/OOD Opportunity Cost.** Lower values are better.

Percentage of Dev Instances in *incor→cor* Group, Binned by $\mathcal{F}_{\text{No-ER}}$ Target Class Confidence									
0.0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
22.85	26.00	40.38	49.20	28.78	0.00	0.00	0.00	0.00	0.00

Table 10: **Change in Target Class Confidence.** For bins where $\mathcal{F}_{\text{No-ER}}$'s target class confidence is low, there is a higher percentage of instances that are predicted incorrectly/correctly without/with ER. This suggests that instances with low target class confidence are more likely to benefit from ER.

ER Opportunity Cost An ER-trained NLM $\mathcal{F}_{\text{task, ER}}$ and a non-ER-trained NLM $\mathcal{F}_{\text{task, No-ER}}$ are likely to yield different outputs given the same inputs. Let $\mathcal{D}^{+}_{\text{ER}} \subseteq \mathcal{D}$ and $\mathcal{D}^{+}_{\text{No-ER}} \subseteq \mathcal{D}$ denote the sets of instances predicted correctly by $\mathcal{F}_{\text{task, ER}}$ and $\mathcal{F}_{\text{task, No-ER}}$, respectively. Ideally, we would have $\mathcal{D}^{+}_{\text{No-ER}} \subset \mathcal{D}^{+}_{\text{ER}}$. This means there is no *opportunity cost* in using ER, as ER increases the number of correct instances without turning any previously-correct incorrect. However, this may not necessarily be the case, so we measure ER's opportunity cost as follows. Let $n^{+}_{\text{ER}} = |\mathcal{D}^{+}_{\text{ER}} \backslash (\mathcal{D}^{+}_{\text{ER}} \cap \mathcal{D}^{+}_{\text{No-ER}})|$ be the number of instances predicted correctly by $\mathcal{F}_{\text{task, ER}}$, but not by $\mathcal{F}_{\text{task, No-ER}}$. Let $n^{+}_{\text{No-ER}} = |\mathcal{D}^{+}_{\text{No-ER}} \backslash (\mathcal{D}^{+}_{\text{No-ER}} \cap \mathcal{D}^{+}_{\text{ER}})|$ be the number of instances predicted correctly by $\mathcal{F}_{\text{task, No-ER}}$, but not by $\mathcal{F}_{\text{task, ER}}$. Then, the opportunity cost of using ER is defined as:

$$o_{\text{ER}} = \frac{n^{+}_{\text{No-ER}} - n^{+}_{\text{ER}}}{|\mathcal{D}|} \tag{3}$$

In practice, instead of defining o_{ER} for all of $\mathcal{D}$, we only consider test sets $\mathcal{D}_{\text{test}}$ and $\tilde{\mathcal{D}}_{\text{test}}$.

Change in Target Class Confidence Let $\mathcal{F}_{\text{No-ER}}$ and $\mathcal{F}_{\text{ER}}$ denote non-ER-trained (vanilla) and ER-trained NLMs, respectively. For each test instance, we plot $\mathcal{F}_{\text{No-ER}}$'s predicted target class confidence probability *vs.* $\mathcal{F}_{\text{ER}}$'s. Each point in the plot is color-coded by whether ER changes the prediction from correct to incorrect, changes the prediction from incorrect to correct, keeps the prediction as correct, or keeps the prediction as incorrect. The purpose of this plot is to visualize how individual instances' predictions are affected by ER. We conduct this for ID dev sets.

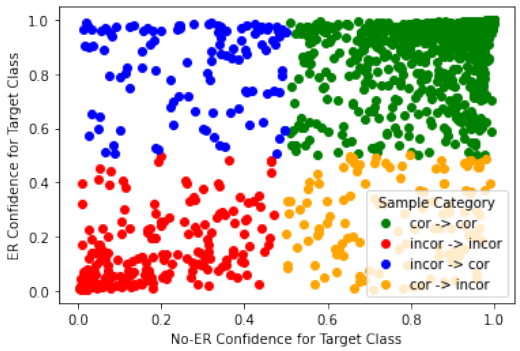

Figure 12: **Change in Target Class Confidence**

We consider ER with the MAE criterion, trained/evaluated on SST (via dev ID task performance). Fig. 12 visualizes how ER changes each dev instance's target class confidence as a result of ER, color-coding each point w.r.t. how ER changes the model's predicted class for this point. Among instances for which $\mathcal{F}_{\text{No-ER}}$'s target class confidence is low, there is a higher percentage of instances that are predicted incorrectly/correctly without/with ER (*i.e., incor→cor*). This suggests that, for $\mathcal{F}_{\text{No-ER}}$, instances with low target class confidence are more likely to benefit from ER (Table 10). Also, based on the T-test, target class confidence scores are significantly higher ($p < 0.005$) with ER than without.

Table 9 displays the opportunity cost results for sentiment analysis. Generally, the opportunity cost results mirror the task performance results in Table 1, such that the methods with highest task performance tend to have the lowest opportunity cost. However, using opportunity cost, the variance is very high for OOD datasets, making it difficult

to compare methods. In future work, we plan to modify the opportunity cost metrics to better accommodate OOD settings.

A.3.9 Efficient hyperparameter tuning with ER-TEST

In intrinsic evaluation (Sec. A.3.2), we used ER loss curves as priors for selecting three key ER hyperparameters (*i.e.,* ER strength λ_{ER}, rationale scaling factor γ_{ER}, learning rate α). In Sec. 5, we assumed a tuning budget that allows only one value for each of λ_{ER}, γ_{ER}, and α. By not tuning these hyperparameters, we greatly reduced our hyperparameter search space. Since ER has little effect on ID task performance, tuning based on ID task performance is unlikely to have helped anyway. ER works better on OOD data, but it also does not make sense to tune based on OOD task performance (otherwise, it would not be OOD). Though the ER hyperparameters chosen via intrinsic evaluation generally improved OOD task performance, we seek to verify their effectiveness compared to other possible hyperparameter values.

In Table 8, we report sentiment analysis OOD (Amazon, Yelp, Movies) task performance, while varying each of the three hyperparameters. We include a Mean column, which averages the Amazon/Yelp/Movies columns. Our hyperparameters chosen via ER loss curves are highlighted in blue. For λ_{ER}, 1 (ours) and 100 yield very similar Mean results, while considerably beating the other three values. For γ_{ER}, we see the same trend for 100 (ours) and 10. For α, $2\text{e}{-5}$ (ours) vastly outperforms other values in all columns. These results validate the utility of ER-TEST's intrinsic evaluation for low-resource ER hyperparameter tuning.

A.4 RQ1

NER Results We also have named entity recognition (NER) task, using CoNLL-2003 (Sang and De Meulder, 2003; Lin et al., 2020) as the ID dataset. For the OOD dataset, we use OntoNotes v5.0 (Pradhan et al., 2013). CoNLL-2003 contains only Reuters news stories, while OntoNotes v5.0 contains text from newswires, magazines, telephone conversations, websites, and other sources. In Table 15, we display the ID and OOD results of NER. In ID, we see more variance in task performance among ER criteria, although the variance is still quite small among the best methods (MSE, MAE, Huber). Here, MAE yields the highest task performance, while BCE yields the lowest by far. In OOD, MAE still performs best, while MSE and Huber are competitive.

Functional Tests We provide details for different functional tests listen in Section 3.3. We breakdown each subcategory of functional tests and show performances of different ER criteria on those individual tests. For functional tests on the sentiment analysis task, refer to Table 11. NLI functional tests are listed in Table 14.

A.5 RQ2

A.5.1 Lexicon-matching

Let $\mathcal{L}_{\mathcal{D}}$ be a list of lexicons curated by human annotators, specific to a given dataset $\mathcal{D}$. Let $l(\cdot)$ be an indicator function that searches for a given lexicon list in all the tokens of an instance, and returns a binary representation of the same size as the instance with 1s in places with lexicon matches (0 otherwise). Therefore, we can obtain distantly-supervised human rationales $\dot{\mathbf{r}}_i = l(\mathcal{L}_{\mathcal{D}}, x_i)$ and apply rationale alignment criteria as described in Section 4.1.

Each lexicon is matched to n-grams(uni-/bi-/tri-grams), which leads to 93% of the train set instances to be matched. Additionally, since we combined two lexicons as resources, there are words appeared as positive and negative. We maintain lexicon overlapping with different sentiment polarities when matching with tokens. For equal comparison, we use instance-based rationales on the same train subset. We also run contrast set tests and functional tests on both lexicon-based and instance-based methods. The results are shown in Table 12 and Table 13.

A.5.2 Hate Speech Detection Tests

Task-Level Rationales For example, Kennedy et al. (2020) used a "blacklist" lexicon to distantly supervise human rationales for the hate speech detection task. In the past, hate speech detection models were largely oversensitive to certain group identifier words (*e.g.,* "black", "Muslim", "gay"), almost always predicting hate speech for text containing these words. To address this, they first manually annotated a lexicon of group identifiers that should be ignored for hate speech detection. Then, for all training instances, they automatically marked only tokens belonging to the lexicon as negative (and the rest as positive). By using these human rationales for ER, they trained the NLM to be less biased w.r.t. these

Capability	Test Type	ER criteria					
		None	IxG+MSE	IxG+MAE	IxG+BCE	IxG+Huber	IxG+Order
Vocabulary	Sentiment-laden words in context	1.20 (±0.74)	0.60 (±0.16)	1.27 (±0.84)	1.00 (±0.86)	1.13 (±0.50)	0.80 (±0.28)
	Change Neutral words with BERT	5.59 (±0.16)	5.13 (±0.90)	5.40 (±0.28)	5.67 (±0.68)	5.67 (±0.74)	5.60 (±1.63)
	Intensifiers	2.13 (±1.63)	1.80 (±0.16)	1.40 (±0.16)	2.67 (±0.77)	2.67 (±0.96)	1.60 (±0.65)
	Reducers	23.85 (±7.18)	35.00 (±46.01)	27.38 (±5.95)	25.00 (±25.00)	17.46 (±13.65)	0.77 (±0.43)
	Add +ve phrases	1.40 (±0.28)	2.33 (±1.84)	0.67 (±0.50)	1.27 (±1.00)	2.33 (±1.76)	2.07 (±1.52)
	Add -ve phrases	22.86 (±7.43)	14.80 (±1.40)	20.67 (±4.07)	17.40 (±3.64)	20.67 (±3.35)	16.93 (±1.91)
Robustness	Adding Random URLs and Handles	9.80 (±0.48)	7.27 (±2.23)	9.07 (±1.80)	7.87 (±2.76)	10.27 (±0.9)	9.6 (±2.47)
	Punctuations	3.93 (±0.89)	1.93 (±0.41)	3.00 (±1.02)	2.87 (±0.19)	3.80 (±0.28)	2.67 (±0.34)
	Typos	2.60 (±0.90)	2.53 (±0.82)	2.60 (±0.57)	3.13 (±0.90)	2.60 (±0.75)	2.00 (±0.86)
	2 Typos	3.93 (±0.65)	3.87 (±1.24)	4.27 (±0.5)	4.13 (±1.2)	4.6 (±0.43)	3.33 (±0.25)
	Contractions	1.00 (±0.00)	0.80 (±0.33)	0.87 (±0.25)	0.80 (±0.43)	0.47 (±0.09)	0.53 (±0.50)
Logic	Negatives	5.20 (±2.75)	4.27 (±1.65)	4.47 (±3.07)	4.47 (±1.75)	3.93 (±1.57)	5.67 (±1.68)
	Non-negatives	59.73 (±9.48)	59.00 (±15.81)	37.47 (±10.41)	63.27 (±17.61)	59.07 (±14.97)	45.87 (±24.13)
	Negation of positive with neutral stuff in the middle	32.2 (±14.65)	35.13 (±1.91)	35.00 (±16.52)	19.00 (±8.66)	40.93 (±4.31)	29.13 (±10.60)
Entity	Change Names	0.70 (±0.14	1.91 (±0.71)	1.11 (±0.51)	0.81 (±0.14)	1.61 (±0.62)	1.91 (±1.51)
	Change Locations	3.33 (±0.74)	2.73 (±1.15)	3.40 (±0.86)	3.07 (±1.79)	3.00 (±0.33)	3.20 (±1.57)
	Change Numbers	0.80 (±0.00)	0.53 (±0.34)	0.47 (±0.41)	0.60 (±0.33)	0.60 (±0.43)	0.67 (±0.81)

Table 11: **Functional Tests:** Sentiment Analysis

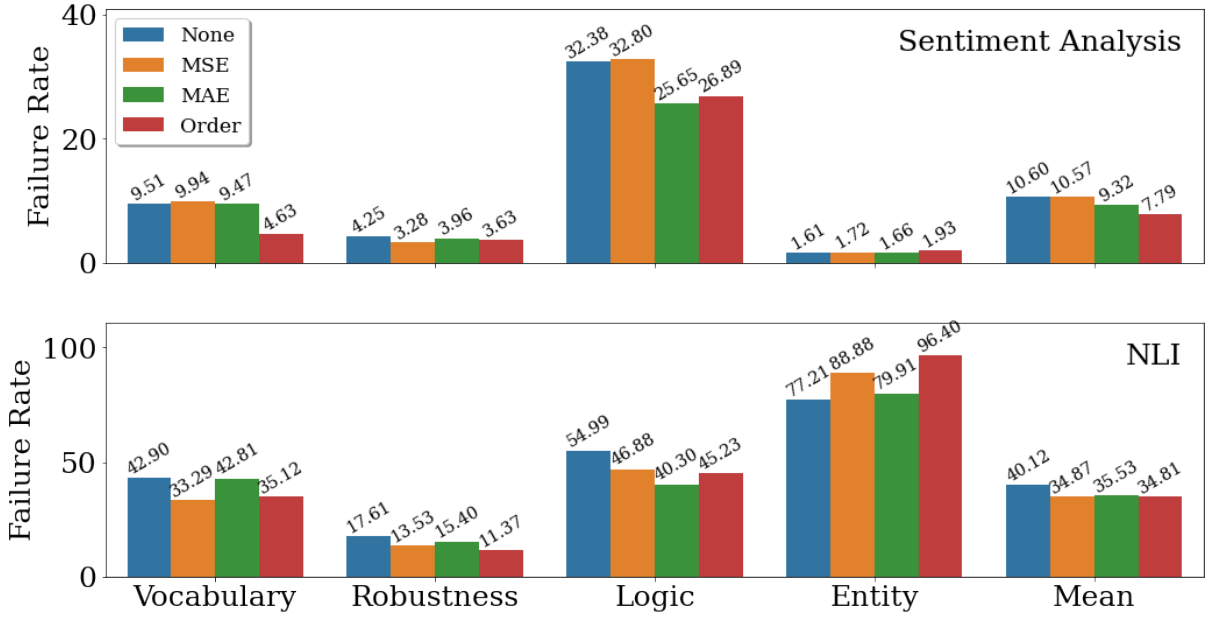

Figure 13: **Functional Tests' Failure Rates (lower the better):** We plot the failure rates of the four functional tests (vocab., robust., logic, entity) as described in Section 3.3, as well as the overall failure rate on all of the tests combined (mean). Each of the values are out of 100, but plotted accordingly for visible comparison. Here we use IxG as rationale extractor.

Method	ER Criteria	Sentiment Analysis		
		Original	Contrast	Delta
Lexicon	IxG+MAE	91.46 (±0.72)	89.82 (±2.46)	-1.64
	IxG+Huber	90.64 (±1.25)	88.52 (±2.25)	-2.12
Instance	IxG+MAE	91.12 (±0.59)	89.82 (±1.20)	-1.3
	IxG+Huber	89.20 (±1.67)	86.13 (±1.74)	-3.15

Table 12: **Contrast Set Tests:** Lexicon-based VS Instance-based. We use

Capability	Vocabulary	Robustness	Logic	Entity	Overall
Lexicon	10.11	4.21	32.27	2.28	12.22
Instance	11.90	3.574	30.2	1.86	11.89

Table 13: **Functional Tests:** Lexicon-based VS Instance-based. Here we use MAE criterion and IxG as rationale extractor.

group identifiers. For the purpose of our study, we use the lexicons as used by (Jin et al., 2021) to generate distantly-supervised rationales for the Stormfromt (Stf) dataset (de Gibert et al., 2018). Each instance in the Stf dataset is matched to one or more lexicons by simple character-level matching, and the rationales are generated as described above. We train $\mathcal{F}$ with the Stf dataset. We report all accuracies in Table 16. As it was observed in Section 5.2, ER does not lead to a significant improvement in performance for the Stf test set. However, it is important to note that "blacklisting" group identifier lexicons does

Capability	Test Type	ER criteria					
		None	IxG+MSE	IxG+MAE	IxG+BCE	IxG+Huber	IxG+Order
Vocabulary	Antonym in Hypothesis	71.66 (±20.98)	64.77 (±21.97)	84.55 (±11.53)	65.88 (±21.40)	74.77 (±20.41)	62.55 (±13.16)
	Synonym in Hypothesis	32.61 (±7.41)	24.11 (±7.62)	30.11 (±6.42)	25.88 (±6.86)	30.77 (±7.07)	29.27 (±6.95)
	Supertype in Hypothesis	24.44 (±15.95)	11.00 (±3.62)	13.77 (±6.71)	9.31 (±5.90)	8.77 (±8.06)	13.55 (±7.10)
Robustness	Punctuation	14.55 (±4.13)	9.44 (±2.79)	11.33 (±1.63)	8.11 (±1.19)	10.00 (±2.58)	9.88 (±2.51)
	Typo	15.88 (±3.44)	10.22 (±3.04)	12.33 (±1.63)	9.66 (±2.10)	10.88 (±2.68)	10.77 (±2.52)
	2 Typos	15.33 (±3.68)	9.77 (±1.81)	12.00 (±1.76)	9.44 (±2.31)	11.11 (±2.99)	10.00 (±2.66)
	Contractions	24.69 (±6.98)	24.69 (±8.72)	25.92 (±9.07)	22.22 (±9.07)	25.92 (±7.40)	14.81 (±5.23)
Logic	Negation in the Hypothesis	50.88 (±32.25)	27.77 (±37.24)	9.77 (±15.66)	41.33 (±41.54)	15.22 (±28.77)	18.44 (±23.21)
	Induce Contradiction	99.88 (±0.31)	98.54 (±3.78)	91.69 (±20.37)	98.65 (±2.56)	98.42 (±4.44)	99.88 (±0.31)
	Same Premise and Hypothesis	14.22 (±8.63)	14.33 (±10.14)	19.44 (±12.12)	18.16 (±12.69)	14.38 (±9.23)	17.38 (±10.16)
Entity	Switch one Entity in the Hypothesis	77.21 (±39.57)	88.88 (±24.11)	79.91 (±22.20)	85.18 (±30.04)	83.83 (±24.25)	96.40 (±4.85)

Table 14: **Functional Tests:** NLI

Methods	NER	
	In-Distribution	Out-of-Distribution
	CoNLL-2003	OntoNotes v5.0
None	77.24 (±0.20)	20.78 (±0.41)
IxG+MSE	78.02 (±0.69)	21.60 (±0.46)
IxG+MAE	**78.34 (±0.81)**	**21.73 (±0.31)**
IxG+BCE	64.53 (±13.22)	17.32 (±3.59)
IxG+Huber	77.83 (±1.09)	21.38 (±0.16)
IxG+Order	72.62 (±5.01)	19.14 (±1.75)

Table 15: **ID/OOD Task Performance on NER (Instance-Based Human Rationales).**

ER Criteria	Hate Speech Detection					
	In-Distribution		Out-of-Distribution			
	Stf		HatEval		GHC	
	Accuracy ↑	FPRD ↓	Accuracy ↑	FPRD ↓	Accuracy ↑	FPRD ↓
None	89.50 (±0.20)	1.11 (±0.58)	63.68 (±0.78)	1.64 (±0.66)	89.43 (±0.98)	1.09 (±0.12)
IxG+MSE	89.46 (±0.21)	2.18 (±0.47)	64.30 (±1.52)	1.99 (±0.26)	88.19 (±0.62)	1.50 (±0.10)
IxG+MAE	**89.59 (±0.06)**	1.39 (±0.62)	63.30 (±0.49)	1.80 (±0.59)	88.07 (±1.66)	1.43 (±0.24)
IxG+BCE	89.42 (±0.71)	1.87 (±0.45)	63.54 (±0.57)	1.87 (±0.45)	88.99 (±0.83)	1.36 (±0.58)
IxG+Huber	89.50 (±0.51)	1.90 (±0.35)	**64.85 (±1.50)**	2.11 (±0.27)	87.77 (±1.21)	1.84 (±0.34)
IxG+Order	89.21 (±1.18)	**0.56 (±0.09)**	64.46(±1.18)	**0.92 (±0.92)**	**92.84 (±0.46)**	**0.59 (±0.25)**

Table 16: **ID/OOD Task Performance (Distantly-supervised Human Rationales)**: Higher values for accuracy and lower values for FPRD are considered better. All models displayed are trained on the ID dataset (Stf) with distantly supervised rationales (for ER criteria) and no rationales (for None) and evaluated on ID and OOD test splits.

not lead to a drop in ID performance either. Benefits of "blacklisting" are then observed in OOD generalization. We evaluate ER methods on OOD hate speech detection datasets like HatEval (Barbieri et al., 2020) and Gab Hate Corpus (GHC) (Kennedy et al., 2018). All of the datasets contain binary labels for hateful and non-hateful content. The Stf dataset is collected from a white-supremacist forum, whereas HatEval instances are tweets and GHC instances are taken from the Gab forum. Table 16 shows that while the improvements in HatEval are not significant, there are significant accuracy improvements for the GHC test set, which are due to the Order ER criterion.

Fairness Tests In addition to generic performance metrics like accuracy, we also measure group identifier bias (against the groups detailed by group identifier lexicons) by evaluating the False Positive Rate Difference (FPRD) as shown by (Jin et al., 2021). FPRD is computed as $\sum_z |\text{FPR}_z - \text{FPR}_{\text{overall}}|$, where FPR_z is the false positive rate of all of the test instances mentioning group identifier z, and $\text{FPR}_{\text{overall}}$ is the false positive rate of all the test instance. Essentially, FPRD evaluates if $\mathcal{F}$ is more biased against a given group identifier z, than all of the groups. A lower FPRD value indicates less biased against the listed group identifiers by $\mathcal{F}$.

Table 16 lists the FPRD values of all the ER criteria in ID and OOD datasets. While all other criteria suffer with higher bias than None, we observe that Order criterion consistently leads to the least bias, both in-distribution and out-of-distribution. Furthermore, the reduction in bias is significant when compared to None. Interestingly, Order ER criterion was initially conceived for distantly-supervised rationales (Huang et al., 2021), and the authors of the original paper also demonstrated experiments with rationales generated from lexicons where Order criterion leads to improvements. Our observations are in-line with theirs, and we also show its benefit in reducing bias in $\mathcal{F}$.

A.6 RQ3

A.6.1 Details for Instance Prioritisation Experiments

In this section, we provide further implementation details for confidence-based instance prioritisation experiments as described in Section 4.3.

Given that we have 3-seed runs for the None model in Table 1, we extract the confidence scores based on the given metric (LC/HC/LIS/HIS), and then average these confidence/importance scores

across the 3 seed runs to obtain a single score for every instance. This process is done for training set instances only. This is followed by ranking each instance by the aggregated confidence metric and selecting the top $k\%$ of samples from this ranking. For experiments with random sampling based prioritisation, we generate 3 random subsets selected in a uniform manner.

While training in this setting, we ensure that within each batch, certain (one third to be specific) set of instances have available rationales. For these instances, we calculate the ER loss $\mathcal{L}_{\text{ER}}$, whereas, for the rest of the instances in the batch, we compute the task loss $\mathcal{L}_{task}$. All prioritisation settings are trained with 3 different model seeds and the aggregated results for ID and OOD datasets are shown in Table 4.

k (in %)	Selection Method	Sentiment Analysis			
		In-Distribution	Out-of-Distribution		
		SST	Amazon	Yelp	Movies
None	-	94.22 (±0.77)	90.72 (±1.36)	92.07 (±2.66)	89.83 (±6.79)
100	-	94.11 (±0.38)	92.02 (±0.25)	94.55 (±0.30)	95.50 (±1.32)
5	Random	94.36 (±0.05)	91.57 (±0.10)	93.36 (±0.15)	92.39 (±2.50)
	LC	93.14 (±1.97)	90.72 (±0.43)	93.50 (±0.53)	**93.17 (±1.26)**
	HC	94.32 (±0.42)	91.57 (±0.19)	93.03 (±0.81)	91.33 (±3.09)
	LIS	93.92 (±1.07)	92.42 (±0.48)	94.28 (±0.31)	96.50 (±1.5)
	HIS	93.94 (±0.83)	90.58 (±0.95)	91.47 (±2.37)	92.00 (±4.58)
15	Random	94.46 (±0.21)	90.06 (±1.17)	90.81 (±2.63)	86.22 (±2.94)
	LC	93.48 (±0.80)	90.12 (±2.66)	90.90 (±5.30)	83.67 (±14.02)
	HC	94.39 (±0.27)	90.38 (±1.12)	93.48 (±0.64)	91.33 (±5.11)
	LIS	94.25 (±0.37)	91.15 (±0.22)	94.00 (±0.56)	95.33 (±1.26)
	HIS	94.47 (±0.22)	91.13 (±0.60)	92.67 (±0.98)	93.50 (±3.12)
50	Random	93.47 (±0.02)	90.28 (±1.42)	91.85 (±2.11)	89.78 (±5.68)
	LC	87.07 (±5.15)	78.82 (±20.68)	77.73 (±26.53)	76.67 (±19.08)
	HC	92.93 (±0.17)	92.15 (±0.36)	94.48 (±0.94)	91.00 (±6.50)
	LIS	93.17 (±0.55)	90.60 (±0.25)	92.72 (±0.53)	93.50 (±0.87)
	HIS	94.23 (±0.65)	88.85 (±2.67)	91.47 (±1.47)	93.67 (±1.89)

Table 17: **Instance Prioritisation Methods (with ID/OOD Performance)**: All values are accuracy (higher the better) on sentiment analysis. None corresponds to models trained without ER, where $k = 100\%$ corresponds to no annotation budget. Each of the $k = [5, 15, 50]\%$ have 3 instance prioritisation methods. $\square$ corresponds to cases where HC and Random are significantly similar and greater than LC. $*$ corresponds to cases where HC is significantly greater than Random and greater than LC. $\bullet$ corresponds to cases where all the three methods are significantly similar.$\diamond$ and $\star$ correspond to cases where the 100% ER setup is significantly similar and greater than None respectively. All tests are conducted with $(p < 0.05)$.

A.7 RQ4

A.7.1 MTurk Annotation

In this section, we demonstrate the MTurk experiment setup. Each MTurk annotator is paid minimum wage. Figures 14, 15 and 16 demonstrate UIs used by MTurk annotators for time-budget experiments.

A.7.2 Task Setup

Each task is timed and have the same set of 200 instances to be annotated. Each instance is annotated by three annotators.

Using the annotations we receive, we aggregate the time taken across all annotators and instances to obtain a rough time estimate taken to annotate *one* instance for a given task.

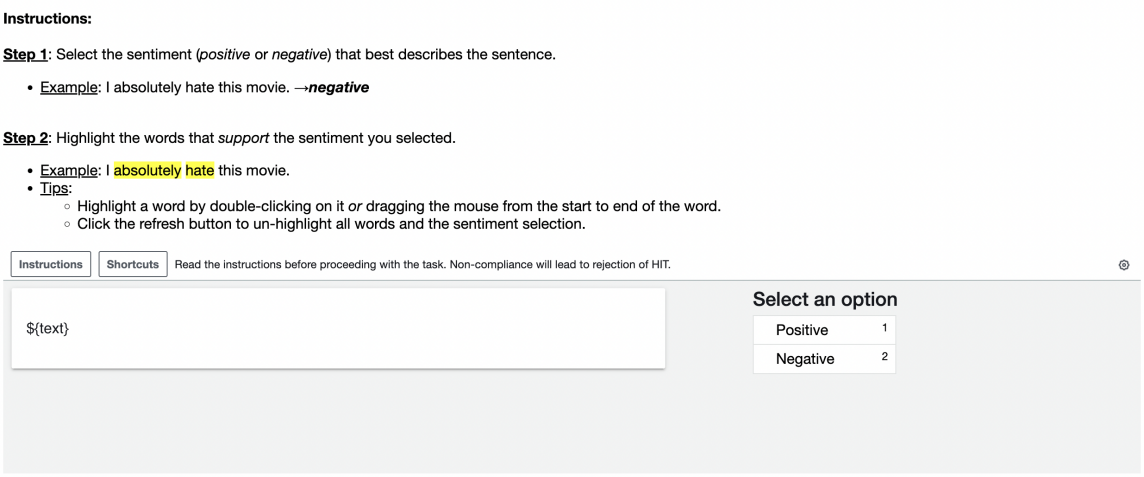

Figure 14: **Label + Expl:** Instructions and setup for Label + Expl annotation

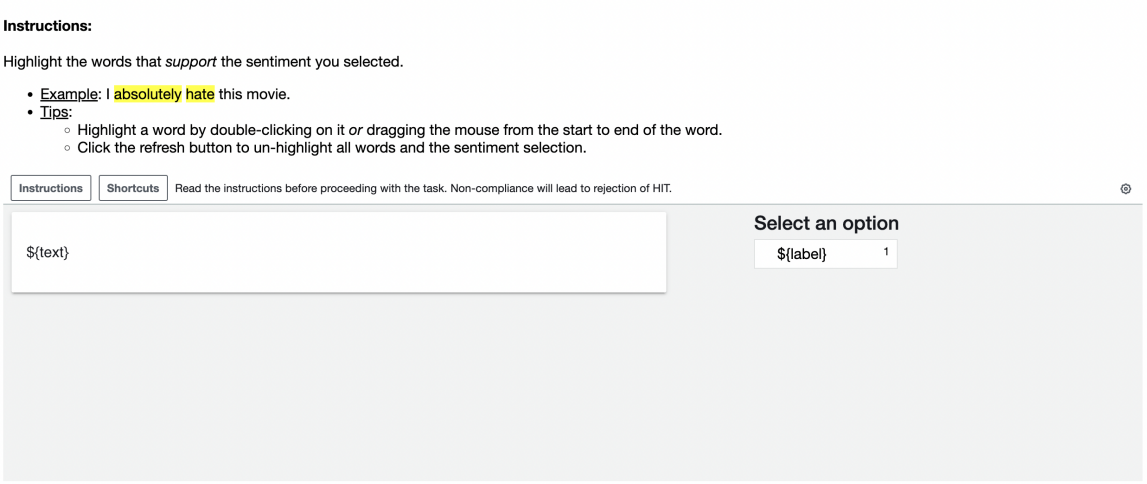

Figure 15: **Only Expl:** Instructions and setup for Only Expl annotation

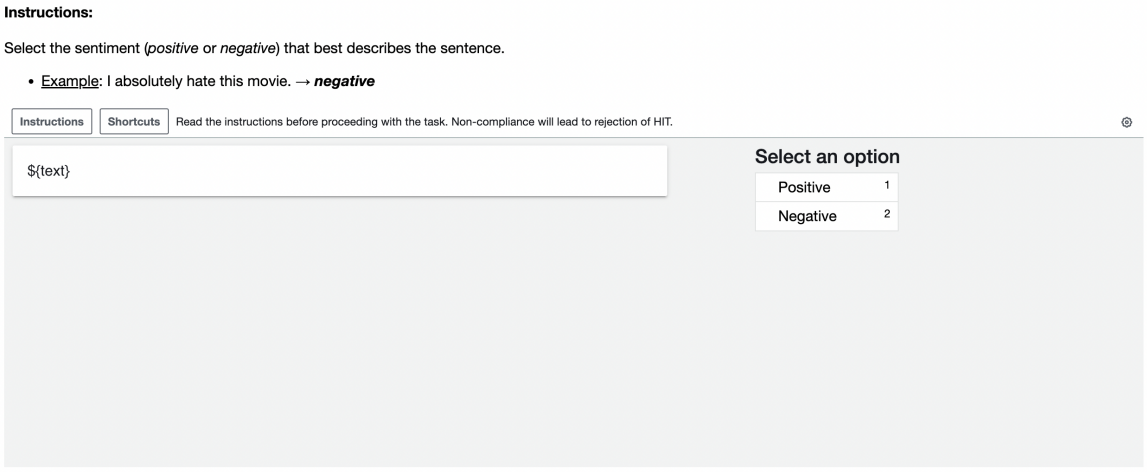

Figure 16: **Only Label:** Instructions and setup for Only Label annotation