Few Shot Rationale Generation using Self-Training with Dual Teachers

Aditya Srikanth Veerubhotla[1*] **Lahari Poddar** [2]

Jun Yin[2] **György Szarvas** [2] **Sharanya Eswaran** [2]

[1]Language Technologies Institute, Carnegie Mellon University
adityasv@cs.cmu.edu
[2]Amazon
{poddarl, jnyin, szarvasg, sharanye}@amazon.com

Abstract

Self-rationalizing models that also generate a free-text explanation for their predicted labels are an important tool to build trustworthy AI applications. Since generating explanations for annotated labels is a laborious and costly process, recent models rely on large pretrained language models (PLMs) as their backbone and few-shot learning. In this work we explore a self-training approach leveraging both labeled and unlabeled data to further improve few-shot models, under the assumption that neither human written rationales nor annotated task labels are available at scale. We introduce a novel dual-teacher learning framework, which learns two specialized teacher models for task prediction and rationalization using self-training and distills their knowledge into a multi-tasking student model that can jointly generate the task label and rationale. Furthermore, we formulate a new loss function, Masked Label Regularization (MLR) which promotes explanations to be strongly conditioned on predicted labels. Evaluation on three public datasets demonstrate that the proposed methods are effective in modeling task labels and generating faithful rationales.

1 Introduction

Interpretable NLP has emerged to learn models which explain their predictions through either extractive (DeYoung et al., 2020) or natural language explanations (Camburu et al., 2018; Narang et al., 2020; Wiegreffe et al., 2020). Due to higher expressivity of free text, generative self-rationalizing models have gained much research interest. However, the early works assume a fully supervised setup and require a large annotated dataset (Narang et al., 2020). Collecting large scale, manual annotations for task labels and corresponding explanations is challenging and expensive. On the other hand, a much larger unlabeled corpora is often available, making semi-supervised approaches like few-shot learning (Brown et al., 2020) and self-training (He et al., 2019) attractive solutions. In the context of self-rationalizing models, (Marasovic et al., 2022) explore few-shot learning, while (Zelikman et al., 2022) seek to improve a supervised labeler by augmenting it with rationale generation. In this work we start from a few-shot setup, assuming only a handful of examples available with their labels and hand-written rationale. We leverage a large unlabeled dataset and self-training techniques to improve over the simple few-shot model.

We hypothesize that using only a few examples, learning to generate meaningful explanations *jointly* with predicting the labels themselves, is a particularly challenging objective and self-training can suffer from a weak initial model. To address this, we propose a novel Dual Teacher learning approach to learn a self-rationalizing model from the two teacher models in a cascading manner. At first, a Predictor model is learned for predicting task labels, and then a Rationalizer model is learned to generate an explanation conditioned on an input and the task labels predicted by the Predictor model. We iteratively improve both models via self-training. In contrast to learning the Joint model directly, the Rationalizer model allows for much richer representation learning by moving the label information from decoder to the encoder part, and utilizing the encoder's self-attention mechanism to extract input-label correlations. A stronger few-shot model for rationale generation provides higher quality pseudo labels, consequently making self-training more effective.

Although the two conditional models (Predictor and Rationalizer) might be better performing, a single self-rationalizing model is still desirable for practical applications, due to its ease-of-maintenance and parameter efficiency for faster inference. We apply principles from knowledge distillation (Hinton et al., 2015; Kim and Rush, 2016) on the two conditional models to learn a

[*] Work done during an internship at Amazon

Findings of the Association for Computational Linguistics: ACL 2023, pages 4825–4838
July 9-14, 2023 ©2023 Association for Computational Linguistics

joint model that generates task label and explanation as a single sequence. The teacher models are used for generating pseudo labels on the entire unlabeled dataset. The initial few-shot labeled data and the pseudo labeled dataset are finally combined to train the joint model.

Faithfulness of explanations is an imperative property for practical applications of interpretability analysis. A model generated explanation is considered faithful if it accurately explains the decision making of the model (Alvarez Melis and Jaakkola, 2018; Wiegreffe et al., 2020). Similar to prior study (Jacovi and Goldberg, 2020), we also observe that a free text explanation generated by models might sound *plausible*, without satisfying the *faithfulness* criteria of explaining the predicted task label. This motivates us to design a masking based regularization function, Masked Label Regularizer (MLR), to encourage the model to condition on the task label while generating an explanation. MLR is an entropy based constraint that forces the Rationalizer model to be maximally uncertain in generating an explanation in absence of label tokens and is used to ensure that the Rationalizer model preserves faithfulness through the self-training iterations. To summarize, our contributions are:

- Proposing to utilize self-training for learning self-rationalizing models with free-text explanations, demonstrating that it provides significant performance boost compared to few-shot learning.
- Proposing a novel Dual Teacher framework, where two teacher models are trained with self-training in a cascading manner for learning two tasks, and a multi-task joint student model is learned through distillation from the teachers.
- Extensively studying the faithfulness property of free-text explanations, and designing an entropy based regularization to encourage label-explanation conditioning.
- Experiments on three public benchmark datasets and demonstrating the effectiveness of our proposed model in improving both task accuracy and explanation quality.

2 Related Work

Prior works on generating free text rationales have explored joint models (Narang et al., 2020; Marasović et al., 2022) as well as several variants of pipeline models (Wiegreffe et al., 2020; Jang and Lukasiewicz, 2021). We also use sequence to sequence models (Raffel et al., 2019) as our backbone models. While most of the self-rationalizing literature assumes fully supervised setups, STaR (Zelikman et al., 2022) explores an alternate bootstrapping setup where limited rationales are available, but the task labels are present for the whole dataset. We consider the generic and more restrictive setting where only limited annotations are available for both task label and rationale.

For limited labeled data scenario, many NLP applications have started reporting success with self-training (Mehta et al., 2022; Yu et al., 2022; He et al., 2019; Bhat et al., 2021). Inspired from these works, we employ self-training to the self-rationalization problem. We introduce a new training framework with two conditional models and using them as teachers in a further distillation step to train the joint model. Besides the popular use for model compression, Knowledge Distillation has also shown superior performance when using the same model architecture and size for both the student and teacher models (Furlanello et al., 2018), and distilling from multiple teachers (Yuan et al., 2021; Liu et al., 2020). Recently, a work (Ghiasi et al., 2021) in computer vision domain has explored using pseudo-labels from multiple teachers to train a joint student model. However, they have multiple specialized teachers trained independently through full supervision, in contrast to the cascading nature of our dual teacher self-training setup.

Evaluating the quality of free-text rationales is significantly challenging and several works have proposed metrics to evaluate the explanations around fluency and their faithfulness properties (Hase and Bansal, 2020; Hase et al., 2020; Marasović et al., 2022). A recent work (Wang et al., 2022) also tries to imbue faithfulness through a regularizing coefficient. However, they apply the regularizer to perturb the rationale while generating task label. In contrast we use a label masking regularizer to enforce the Rationalizer model to generate an explanation which is faithful to the label.

3 Background

We first provide some necessary background on Self-Rationalizing models and a theoretical outline of Self Training based learning.

Self-Rationalization: A Self-Rationalization model tries to learn the joint distribution of output(O) and explanation(E), given an input(I),

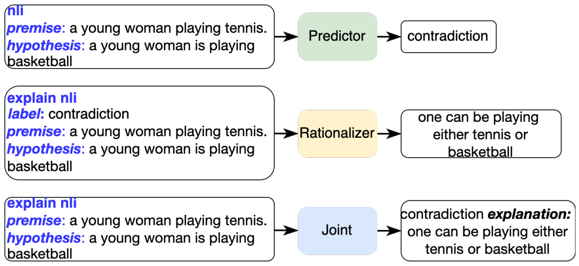

nli
premise: a young woman playing tennis.
hypothesis: a young woman is playing basketball
Predictor
contradiction
explain nli
label: contradiction
premise: a young woman playing tennis.
hypothesis: a young woman is playing basketball
Rationalizer
one can be playing either tennis or basketball
explain nli
premise: a young woman playing tennis.
hypothesis: a young woman is playing basketball
Joint
contradiction explanation: one can be playing either tennis or basketball

Figure 1: Input and output formats for Predictor, Rationalizer and Joint models.

i.e. $P(O, E|I)$. A common approach is modeling it as a sequence-to-sequence problem and generating the task prediction and the rationale jointly (Narang et al., 2020). Input-output format for a self rationalizing joint model is illustrated in Figure 1. The input consists of a task prompt, (e.g. `explain nli`), and in output sequence the task label is generated first (e.g. `contradiction`), followed by a separator token (`explanation:`), and then the free text explanation. During inference, greedy decoding is used to generate the sequence until an EOS token is produced.

Self-Training is a type of Semi-Supervised Learning based method, which assumes access to a small labeled dataset (D_l) and a large, unlabeled in-domain dataset (D_u). The algorithm progresses iteratively in four steps. First, a teacher model is trained on the labeled dataset (D_l), to obtain θ^T. The trained teacher is then used to infer *pseudo-labels* on D_u, generating the *pseudo-labeled* dataset D_{pl}. A student model is then trained on D_{pl} to obtain the θ^S. In the next iteration the teacher model is updated with the learned parameters from the student and the process repeats until a convergence criterion is met.

4 Dual Teacher for Self-Rationalization

We combine the strengths of self-training and knowledge distillation to train a self-rationalizing joint model from dual teachers. Following sections describe the components, their losses and the learning procedures in more detail. Input-output formats of the models are shown in Figure 1, and the overall framework is illustrated in Figure 2.

4.1 Problem Setup

We tackle the self-rationalization problem with few-shot labels. We consider access to a small labeled set, $D_l = \{(i_j, o_j, e_j)\}_{j=1}^{N}$, where i_j is the input, o_j is the task output, and e_j is the natural language explanation. We also leverage a much larger unlabeled dataset denoted by $D_u = \{i_j\}_{j=1}^{M}$, where $M \gg N$. In the unlabeled dataset only the input text is available and no annotation is provided for either task label or rationale.

To keep all models identical, we model all distributions in a sequence to sequence manner using T5 (Raffel et al., 2019). The teacher model in self-training is trained on few shot ground truth output sequences and the trained teacher is then used for generating output sequences for the unlabeled dataset. These sequences are considered as pseudo labels to train the student model. We re-weight the loss of each example with confidence of the teacher model. This limits error propagation through self-training iterations due to the noisy nature of pseudo labels. We use likelihood of the generated sequence as confidence estimates. Following (Bhat et al., 2021) we normalize the weights in a batch.

4.2 Splitting the Joint into Conditionals

In order to make the learning task easier, we break down the joint probability of modeling task and rationale, into its conditionals.

$$\underbrace{P(O, E|I)}_{\text{Joint}} = \underbrace{P(O|I)}_{\text{Predictor}} \times \underbrace{P(E|I, O)}_{\text{Rationalizer}} \quad (1)$$

This allows us to build two separate models in a cascading manner: (1) Predictor Model for predicting task label, i.e. $P(O|I)$, and (2) Rationalizer Model for rationalizing the task label for an input, i.e. $P(E|I, O)$. Prior works (Jang and Lukasiewicz, 2021) have shown that factorization of this distribution to predicting the output first (Prediction) and generating an explanation for the prediction (Rationalization) has obtained better performance than alternate factorizations.

We hypothesize that with limited labeled examples, learning a joint distribution for <task label+rationale> sequence would be much harder than focusing on learning to predict only the task label. More importantly, for rationale generation we move the task label from output sequence (in the joint model) to input sequence (in Rationalizer model). This allows the encoder to capture much richer interactions between task label and the input through its self-attention network, compared to only the decoder in joint model. The stronger initial few-shot models for predictor and rationalizer would be further boosted through self-training in generating higher quality pseudo labels.

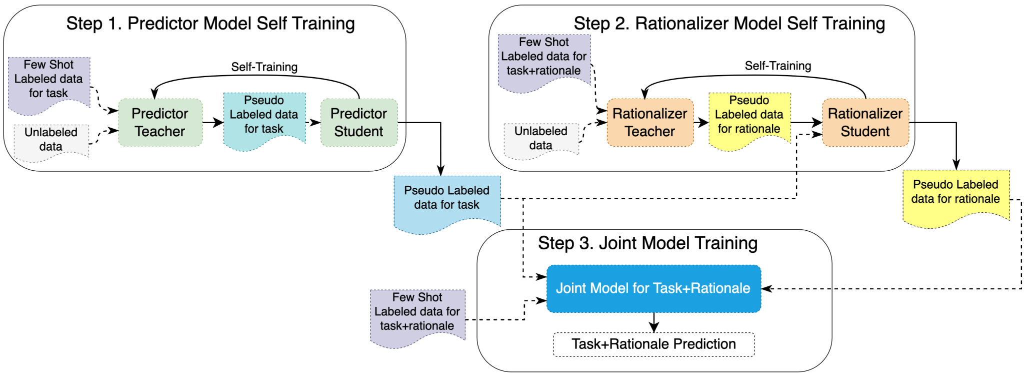

Step 1. Predictor Model Self Training
Few Shot Labeled data for task
Unlabeled data
Self-Training
Predictor Teacher
Pseudo Labeled data for task
Predictor Student
Step 2. Rationalizer Model Self Training
Few Shot Labeled data for task+rationale
Unlabeled data
Self-Training
Rationalizer Teacher
Pseudo Labeled data for rationale
Rationalizer Student
Pseudo Labeled data for task
Pseudo Labeled data for rationale
Step 3. Joint Model Training
Few Shot Labeled data for task+rationale
Joint Model for Task+Rationale
Task+Rationale Prediction

Figure 2: Dual Teacher Training Framework. Predictor and Rationale models are trained in their own Self-training loop. Pseudo labels generated from the trained predictor and rationale model are used for training the Joint model.

4.3 Predictor Teacher

In the first step of our framework, we train a Predictor model with self-training. The Predictor is trained to model the probability of the task output given the input, i.e. $P(O|I)$. The task output is decomposed into subwords, and the model is trained to minimize the negative log likelihood of the output token sequence:

$$\mathcal{L}_{pred}(\theta) = \mathbb{E}_{(i,o)\sim\mathcal{D}}\left[-\log P_\theta(o|i)\right] \qquad (2)$$

The predictor model is trained within its own self training loop, utilizing the few shot ground truth task labels and unlabeled inputs. After self-training has converged, we store the predictor and use it for generating pseudo task labels on all unlabeled data.

$$D_{pl} = \{(i, p_{\theta_{pred}}(o|i))\}_{i\in D_u} \qquad (3)$$

These pseudo labels are then used for training the Rationalizer model and the Joint model.

4.4 Rationalizer Teacher

In the second stage we train a Rationalizer model that can generate natural language explanations given an input and the predicted task output, modeling the conditional distribution $P(E|I,O)$.

$$\mathcal{L}_{rat\_gen}(\theta) = \mathbb{E}_{(i,o,e)\sim\mathcal{D}}\left[-\log P_\theta(e|i,o)\right] \qquad (4)$$

For training the teacher model we use the few-shot ground truth labeled dataset for task label and rationale. For generating rationale pseudo-labels on the unlabeled set, we use the task pseudo labels generated by the predictor model as input. The generated rationale pseudo labels are then used to train a student rationalizer model in self-training loop until convergence.

Faithfulness of Explanations

For a Rationalizer model to generate a *faithful* explanation, we want the explanation to be strongly conditioned on the label. The rationalizer should not be able to generate an explanation solely based on the input, but must take into consideration the label for which it is rationalizing. We introduce a regularizing constraint in our rationalizer model to explicitly encode this property.

Masked Label Regularization

We design an entropy based regularization which tells the model to be maximally uncertain in generating the explanation in absence of a task label. We achieve this by replacing the task output with mask tokens and maximizing the per-token entropy of the explanation sequence.

$$\mathcal{L}_{MLR}(\theta) = \mathbb{E}_{(i,e)\sim\mathcal{D}}\left[-H_\theta[e|i]\right] \qquad (5)$$

where $H_\theta[e|i]$ refers to the entropy of producing an explanation from input directly.

There could be alternate ways of encoding the constraint of label-explanation association. We experimented with one such variant where the ground truth explanation would be generated with a high entropy in case of a wrong label. We observed similar empirical results in our experiments for this alternative. However, it is strictly less general - since it becomes limited to only categorical problems, and also is computationally more expensive due the necessity of computing entropy for multiple wrong labels. Therefore, we use the simpler and generic form of masking the label tokens.

The overall loss of the Rationalizer is a weighted summation of the sequence generation loss and the regularization loss:

$$\mathcal{L}_{rat} = \mathcal{L}_{rat\_gen}(\theta) + \lambda_{MLR}\mathcal{L}_{MLR}(\theta) \qquad (6)$$

Algorithm 1 Dual Teacher Training Algorithm

```
Require: $D_l = \{(i_i, o_i, e_i)\}_{i=1}^{N}$
Require: $D_u = \{i_j\}_{j=1}^{M}$
Require: $D_{val} = \{(i_k, o_k, e_k)\}_{i=1}^{K}$
  Initialize $\theta_{pred}, \theta_{rat}, \theta_{joint}$ randomly

  /* Train Predictor model */
  $\theta_{pred}^{*} \leftarrow SelfTraining(\mathcal{D}_l, \mathcal{D}_u, D_{val}, \theta_{pred})$
  $\mathcal{D}_{pred} \leftarrow \{(i_j, \hat{o}_j)\}_{j=1}^{M}, \hat{o}_j \sim p_{\theta_{pred}^{*}}(.|I)$

  /* Train Rationalizer model */
  $\theta_{rat}^{*} \leftarrow SelfTraining(\mathcal{D}_l, \mathcal{D}_{pred}, D_{val}, \theta_{rat})$
  $\mathcal{D}_{pl} \leftarrow \{(i_j, \hat{o}_j, \hat{e}_j)\}_{j=1}^{M},$
  $\hat{o}_j \sim p_{\theta_{pred}^{*}}(.|I), \hat{e}_j \sim p_{\theta_{rat}^{*}}(.|I, O)$

  /* Train Joint model */
  $D_{final} \leftarrow D_{pl} \cup D_l$
  $\theta_{joint}^{*} \leftarrow Train(D_{final}, D_{val}, \theta_{joint})$
```

λ_{MLR} is empirically set to $1e^{-4}$ in our experiments for all datasets.

4.5 Learning from Multiple Teachers: Distilling a Joint from the Conditionals

Knowledge Distillation is an effective learning paradigm to train a lighter student model with rich supervision signals from better performing teacher model(s). To alleviate the limitations of limited labeled data for learning a good self-rationalization model, we leverage the unlabeled data and collect task and rationale pseudo-labels sequentially from trained Predictor and Rationalizer teacher models. The final pseudo-labeled dataset is then combined with the few-shot labeled data and a joint model is trained on this set. This allows the knowledge from both the Predictor and Rationalizer models to be distilled into the student Joint model through pseudo labels and the teachers' confidence weights.

The joint model is trained to maximize the likelihood of a concatenated sequence of task output and explanation, as illustrated in Figure 1. The detailed training algorithm is described in Algorithm 1.

Loss Re-weighting: Similar to most sequence-to-sequence models, in WT5 (Narang et al., 2020), all output tokens in the generated sequence have uniform weights in the loss. However, in the joint task setup, the number of tokens from task label is substantially smaller than those in the explanation. To balance this, we re-weight the token-level losses between the output and the explanation. For a tuple (i_j, o_j, e_j), the loss is computed as:

$$\mathcal{L} = \lambda \sum_{y_m \in o_j} -\log p_\theta(y_m | i_j, y_1, \cdots y_{m-1})$$
$$+ (1-\lambda) \sum_{y_n \in e_j} -\log p_\theta(y_n | i_j, y_1, \cdots y_{n-1})$$

where $\lambda \in [0.5, 1)$ is a weight coefficient.

	e-SNLI	ComVE	ECQA
# classes	3	2	5
total train size	549,367	10,000	7,598
few shot dataset size	300	200	500
validation size	9,842	1,000	1,090
test size	9,824	1,000	2,194
Avg. tokens in output	2.0	2.0	1.9
Avg. tokens in explanation	16.8	26.0	14.5

Table 1: Dataset Statistics. Token-level statistics were generated using the T5-base tokenizer.

5 Results and Discussion

We evaluate on public datasets for three different tasks. Table 1 shows statistics of the datasets.

e-SNLI (Camburu et al., 2018) extends the popular SNLI dataset (Bowman et al., 2015) by adding human-annotated explanations to the NLI labels. The task requires generation of a task label which describes the relationship between a premise and a hypothesis as entailment/contradiction/neutral, and a free text explanation for the prediction.

ComVE (Wang et al., 2020) aims to evaluate if a model can distinguish between sensible and nonsensical statements based on common knowledge. We combine the data from SubTask A (Validation) and SubTask C (Generation) for our experiments.

ECQA (Aggarwal et al., 2021) augments the Commonsense QA dataset (Talmor et al., 2019) with free-text explanations that support the the correct answer choice and refute the incorrect ones. We utilize the explanations for the correct output (Positive Property) as the explanation.

For few-shot settings we sample 100 examples per class for each dataset. The self-training setup leverages the few-shot labeled dataset(D_l) and the rest of the training set as unlabeled dataset(D_u).

5.1 Implementation Details

We use the base variant of T5 (Raffel et al., 2019) as backbone model for the Predictor, Rationalizer and Joint models. Following (Narang et al., 2020), we also measure task performance using accuracy, and rationalization using SacreBLEU (Post, 2018). Label smoothing was set to 0.1 and early stopping

Model	e-SNLI		ComVE		ECQA		Average	
/Metric	Acc	BLEU	Acc	BLEU	Acc	BLEU	Acc	BLEU
Fully Supervised (Narang et al., 2020)	90.44	33.76	86.2	14.53	53.6	16.25	76.75	21.5
Few-Shot	82.57	24.21	73.77	12.74	34.29	9.77	63.54	15.57
Self-Training techniques								
Vanilla	83.35	25.18	78.83	10.44	41.8	9.7	67.99	15.11
Confidence Weighted	83.41	24.54	79.23	11.04	41.75	9.85	68.13	15.14
Dual Teacher	**83.95**	**30.17**	**79.61**	**14.83**	**44.26**	**17.12**	**69.27**	**20.71**

Table 2: Results on different baselines and other self-training techniques on three datasets, measured using Accuracy for label prediction, and BLEU for explanation

with a patience of 5 was used for model selection. The few-shot examples were sampled randomly by stratifying across classes. We trained on 4 NVIDIA v100-16GB GPUs with a batch size of 8 and 16 for D_l and D_{pl}, respectively. The token re-weighting coefficient λ is set to 0.8 for eSNLI and ComVE, and 0.9 for ECQA via grid search based on validation scores and average length of the explanations in the dataset. All results are reported after averaging 3 runs.

5.2 Main Results

In Table 2 we compare the various training paradigms, namely, fully supervised, few-shot training, and self-training on all three datasets. For self-training we explore two setups - one without pseudo label re-weighting on a Joint model, which we call Vanilla Joint. Confidence-weighted Joint performs self-training on a Joint model where the pseudo labels are weighted by the confidence of the teacher model. The Dual Teacher refers to the proposed Joint model in Section 4.5 that is trained with distillation from two teachers.

Few-shot vs Fully Supervised results: Metrics from the fully-supervised setup provide an upper bound on the scores achievable when trained on complete dataset of labels and rationales. Aggregated across datasets, the few-shot performance of the model is around 13% behind the fully supervised model, and around 5 BLEU lower in rationalization performance.

Self-Training helps boosting few-shot results. Our experiments show that self-training is a promising direction in bridging the performance gap, improving accuracy and BLEU across all the tasks over the few-shot counterparts. We observe that re-weighting the pseudo-labels with the confidence of the teacher models, provides small improvements in the overall performance and is in alignment with previous findings (Bhat et al., 2021).

Stronger Results with Dual Teacher Self-Training Framework. Finally, we observe a further improvement by our proposed method of performing self-training on the Predictor and Rationalizer models, and subsequently distilling the knowledge to a joint student model through pseudo labels. The improvement in aggregate scores shows that the accuracy is within 8% of a fully supervised model, and 5% higher than the few-shot baseline. The improvements from the proposed model are most prominent for the Rationale generation task - the BLEU scores are improved by a large margin compared to learning both tasks jointly in a self-training setup. Impressively, the dual-teacher approach achieves an aggregated result of 20.71 BLEU which is close to the aggregate performance of the *Fully Supervised* model (21.5 BLEU). We even obtained higher performance (BLEU score) than the supervised model on the two smaller datasets, ComVE and ECQA.

5.3 Discussion

Next we conduct several deeper analysis of the models and provide detailed insight to the overall results presented in Section 5.2.

RQ1: Does breaking the joint into conditionals improve performance for task label prediction and explanation quality?

We first want to analyze the effectiveness of breaking the joint model into conditionals and learning two separate models for task prediction and rationalization. From the results in Table 3, it is evident that by breaking the joint distribution into conditionals, we obtain significantly higher performance across all datasets, especially for explanation generation. This validates our hypothesis that with limited labels, it is much harder for the model to learn the joint distribution of output and explanation, compared to learning the conditionals separately. With self-training, the gap in performance

	Model	e-SNLI		ComVE		ECQA		Avg	
	/Metric	Acc	BLEU	Acc	BLEU	Acc	BLEU	Acc	BLEU
Fully Supervised	Predictor	89.7	–	**90.2**	–	**55.9**	–	**78.6**	–
	Rationalizer	–	**34.9**	–	**16.8**	–	**18.8**	–	**23.5**
	Joint	**90.4**	33.7	86.2	14.5	53.6	16.2	76.7	21.5
Few-Shot	Predictor	**82.9**	–	**75.7**	–	**39.7**	–	**66.1**	–
	Rationalizer	–	**27.1**	–	**14.8**	–	**16.3**	–	**19.4**
	Joint	82.6	24.2	73.8	12.7	34.3	9.8	63.5	15.6
Self-Training	Predictor	**83.8**	–	**78.8**	–	**44.4**	–	**69**	–
	Rationalizer	–	**31.2**	–	**17.0**	–	**19.2**	–	**22.5**
	Joint	83.4	24.5	79.2	11.0	41.8	9.8	68.1	15.1

Table 3: Performance of the Joint model compared to Predictor and Rationalizer models in Fully supervised, Few-Shot and Self-Training setup.

Model/Dataset	e-SNLI	ComVE	ECQA	Avg
Joint	76.8	52.0	89.0	72.6
Dual Teacher − MLR	86.5	54.8	93.9	78.4
Dual Teacher	95.7	74.7	95.8	88.7
Rationalizer − MLR	97.8	69.9	95.3	87.7
Rationalizer	**99.4**	**84.8**	**96.8**	**93.7**

Table 4: Label-Explanation association measured as % of inputs with distinct explanations for each task label.

between the joint and the conditionals decreases, but the individual models still outperform the joint model.

These results align with the improvement observed from the Dual Teacher framework over Joint model in Table 2. Training the Predictor and Rationalizer models in their own self-training loops creates two strong teacher models and provides better pseudo labels. This allows us to train a strong self-rationalizing model through distillation than training a joint model directly through self-training.

RQ2: Does the Masked Label Regularization help to generate more faithful explanations?

While our method achieves better BLEU scores compared to different baselines, it is also important to evaluate whether the generated explanations are *faithful* to the predictions, i.e. provide reasoning that support the predicted label. During creation of the datasets, the annotators were instructed to assign a label and then explain the assignments with a natural language explanation. Therefore, it is desirable for the models to preserve the faithfulness properties in generated explanations.

We perform two tests to analyze whether (1) the explanations are dependent on the output and (2) if they reflect the intended label. Through these experiments we also conduct an ablation study to estimate the effect of the proposed Masked Label Regularization (MLR) constraint in improving the faithfulness of explanations.

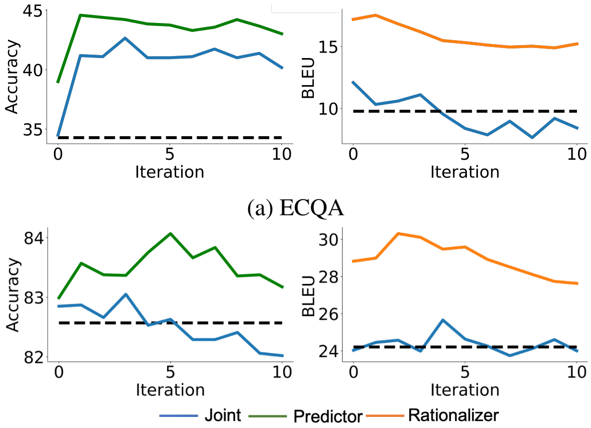
Label-Explanation Association. We first conduct a simple analysis to check if the explanations are dependent on the model predictions. As a necessary condition for generating faithful explanations, different predicted labels have to produce different explanations. We measure this association as the number of test instances for which the model generates a distinct explanation for all labels.

We vary the task label and ask the model to generate an explanation. For joint models, we replace the generated label with other possible labels and ask the decoder to continue generating an explanation. For Rationalizer model, we simply generate predictions with providing different labels in the input. We study the effect of MLR by removing the entropy regularization loss while training the Rationalizer. We denote this variant as Rationalizer − MLR. Dual teacher − MLR refers to the Joint model trained using Rationalizer − MLR.

Results in Table 4 show that for the Joint model, only 72% of the examples have unique explanations per output on an average across datasets. This implies that the label-explanation association is not inherently captured in the decoder and for 28% of instances the generated explanation is constant and has no association with the labels. Adding the MLR loss encourages the model to condition on labels, and thereby provides a substantial improvement of over 10% for the Dual Teacher model. This indicates a strong association between the generated label and explanation, where the explanations are unique to the label in over 88% of cases. As can be seen from the Table, the Rationalizer teacher achieves significantly better label-explanation association compared to the Joint counterparts. The MLR constraint further improves the results, especially in the ComVE dataset where explanations

Model/Dataset	e-SNLI	ECQA	ComVE	Avg
Fully-Supervised	8.64	30.45	2.4	13.83
Few-Shot	2.61	16.91	0.2	6.57
Joint	1.87	14.22	0.7	5.6
Dual Teacher - MLR	**6.01**	17.96	0.73	7.62
Dual Teacher	4.54	**18.19**	**0.9**	**7.88**

Table 5: Simulatability score of the explanations from different methods. The higher the score the more aligned the explanation is with the predicted label



(a) ECQA

(b) e-SNLI

Figure 3: Performance across self-training iterations on ECQA and e-SNLI datasets of the Confidence Weighted Joint, Predictor and Rationalizer models. Dashed lines show the performance of the Few-Shot Joint model.

are much longer on average.

Simulatability of Explanations. We utilize the Simulatability metric as defined in prior work (Chan et al., 2022; Hase and Bansal, 2020) to evaluate how well an external system, human or AI, is able to simulate the prediction made by a black-box, self-rationalizing model using the explanation it generates. As simulators, two models are trained to predict the task label - (1) a control model $P(O|I)$, which predicts the output given input and (2) a treatment model, $P(O|I, E)$ predicting output given input and an explanation. The simulators are used to measure how much the explanations generated by the self-rationalizing model help in 'guessing' its predicted label. The simulatability score is defined as

$$\Phi = \mathbb{1}(y^T = \hat{y}) - \mathbb{1}(y^C = \hat{y}) \qquad (7)$$

where $\hat{y}$ refers to predicted label from the self-rationalizing model, y^C and y^T refers to predictions from the control and treatment simulators, respectively. The higher the faithfulness of a model, the better aligned its explanations are with its predicted labels, relative to the control simulator which does not consider explanations.

Table 5 shows the simulatability scores of the various self-rationalizing models under consideration. We observe a similar trend as in Table 4 while comparing the different models, with the exception of e-SNLI. For e-SNLI the control simulator was notably stronger compared to treatment, potentially due to the overlap with pre-training tasks of T5. We note that overall there is significant gap in the simulatability between our models and the Fully-Supervised model, indicating a large room for improvement in the faithfulness of explanation for weakly supervised models.

RQ3: How does the performance change as self-training progresses?

Figure 3 shows the performance of different models over self-training iterations. We observe that the two teacher models consistently outperform the joint model over iterations in both datasets. In ECQA dataset there is a large jump in accuracy in the first iteration and the algorithm converges soon. A similar trend is observed for BLEU scores, with a slight improvement in the Rationalizer in first iteration and the score plateauing or even declining in case of the Joint model. For e-SNLI dataset, accuracy continues to improve till five iterations for the Predictor, and three for the Joint model. The rationalization performance also converges after nearly five iterations for both the models. Convergence of the algorithm could be explained by the poor separability of the class labels in the datasets, causing more erroneous pseudolabels and plateauing of performance as time progresses.

RQ4: How does the performance change with increase in labelled dataset size?

We study the performance of our model by conducting experiments with different dataset sizes. We only vary the labeled dataset size and keep the remaining training set as unlabeled data. For example, for ECQA the total size (D) is $7.5K$, and we conduct experiments with labeled data (D_l) in the range $\{50, 2.5K\}$ and the remaining data size ($D - D_l$) as unlabeled data. Table 6 reports the accuracy and BLEU score of our proposed model for dataset sizes ranging from 50 to 2500 samples. We see that there is a improvement in the test accuracy and BLEU score as the labeled data size increases. With as few as 500 examples per label, the model is able to achieve accuracy within 6% of

	ECQA		ComVE		eSNLI	
Size of D_l	Accuracy	BLEU	Accuracy	BLEU	Accuracy	BLEU
10	30.63	16.36	49.6	15.55	81.97	24.62
100	44.26	17.12	79.61	14.83	83.95	30.17
200	47.81	17.01	80.1	14.45	83.61	29.59
500	51.28	19.03	82.4	15.88	84.43	29.3
5000	-	-	-	-	85.85	29.37
Fully supervised	53.6	16.25	86.2	14.53	90.44	33.76

Table 6: Effect of size of the labeled set on the final performance.

the fully supervised model across all datasets. Interestingly, we note that with limited supervision the self-training setup is able to outperform the Fully supervised model in the BLEU scores, demonstrating the data efficiency of the Rationalizer teacher in achieving good performance.

6 Conclusion

We study the self-rationalization problem with few-shot labels and demonstrate that self-training is an effective learning paradigm and can significantly reduce the gap between few shot and fully supervised model performance. We present a novel dual teacher learning framework that learns two models for task label prediction and rationale generation through self-training and efficiently distill the knowledge in a single self-rationalizing joint model. With a masking based loss formulation we enforce label-explanation association in the rationalizer, leading to generation of more faithful explanations. We conduct experiments on three public benchmark datasets for free text explanations, and show that the proposed methods are effective in improving task performance while generating accurate and faithful explanations.

7 Limitations

Despite strong performance compared to few-shot our self-training methods still contain significant room for improvement compared to the fully supervised benchmarks. It would be interesting to try larger language models to see if it is possible to close this gap with more knowledge embedded into the pre-trained models. Our evaluation of free text rationales are limited by the automatic metrics, which are necessary but not sufficient to analyze quality of an explanation for decision making of the model. From example explanations (a few of which are shown in Appendix), it is evident that we still lack understanding on multiple dimensions such as, when an explanation is factually wrong, is it due to the model believing in the wrong knowledge or is unable to retrieve the correct one. Works that probe a language model with various prompts could be useful for investigating in these directions.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for CommonsenseQA: New Dataset and Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*.

David Alvarez Melis and Tommi Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31.

Meghana Moorthy Bhat, Alessandro Sordoni, and Subhabrata Mukherjee. 2021. Self-training with few-shot rationalization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Aaron Chan, Shaoliang Nie, Liang Tan, Xiaochang Peng, Hamed Firooz, Maziar Sanjabi, and Xiang Ren. 2022. Frame: Evaluating rationale-label consistency metrics for free-text rationales. *arXiv preprint arXiv:2207.00779*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2020. Eraser: A benchmark to evaluate rationalized nlp models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR.

Golnaz Ghiasi, Barret Zoph, Ekin D Cubuk, Quoc V Le, and Tsung-Yi Lin. 2021. Multi-task self-training for learning general representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8856–8865.

Peter Hase and Mohit Bansal. 2020. Evaluating explainable ai: Which algorithmic explanations help users predict model behavior? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

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

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205.

Myeongjun Jang and Thomas Lukasiewicz. 2021. Are training resources insufficient? predict first then explain! *CoRR*, abs/2110.02056.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Yuang Liu, Wei Zhang, and Jun Wang. 2020. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113.

Ana Marasovic, Iz Beltagy, Doug Downey, and Matthew Peters. 2022. Few-shot self-rationalization with natural language prompts. In *Findings of the Association for Computational Linguistics: NAACL 2022*.

Sanket Vaibhav Mehta, Jinfeng Rao, Yi Tay, Mihir Kale, Ankur Parikh, and Emma Strubell. 2022. Improving compositional generalization with self-training for data-to-text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. Wt5?! training text-to-text models to explain their predictions. *ArXiv*, abs/2004.14546.

Matt Post. 2018. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Cunxiang Wang, Shuailong Liang, Yili Jin, Yilong Wang, Xiaodan Zhu, and Yue Zhang. 2020. SemEval-2020 task 4: Commonsense validation and explanation. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 307–321, Barcelona (online). International Committee for Computational Linguistics.

PeiFeng Wang, Aaron Chan, Filip Ilievski, Muhao Chen, and Xiang Ren. 2022. Pinto: Faithful language reasoning using prompt-generated rationales. In *Workshop on Trustworthy and Socially Responsible Machine Learning, NeurIPS 2022*.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2020. Measuring association between labels and free-text rationales. In *Conference on Empirical Methods in Natural Language Processing*.

Yue Yu, Lingkai Kong, Jieyu Zhang, Rongzhi Zhang, and Chao Zhang. 2022. AcTune: Uncertainty-based active self-training for active fine-tuning of pretrained language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. 2021. Reinforced multi-teacher selection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14284–14291.

Eric Zelikman, Yuhuai Wu, and Noah D Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*.

8 Appendix

We include some qualitative analysis of the different design choices in our method.

8.1 Impact of moving the task label to the input from the output sequence

We observed substantial improvement in rationale performance with the Rationalizer teacher model compared to the Joint model. This is can be attributed to the the prediction being passed as an input to encoder of the Seq2Seq model, generating better representation of the predictions and yielding better quality rationales. Table 7 shows a sample of explanations generated from the Joint and Rationalizer models, for cases when the label was predicted correctly. We see that the Rationalizer generally produces higher quality explanations, and in contrast, while the Joint model often generates nonsensical explanations with frequently repeated words. The better quality rationales obtained from the Rationalizer teacher helps generate better pseudo labels and the final model is able to capture those with distillation.

8.2 Effect of the Masked Label Regularization on faithfulness

Table 8 shows a few examples of rationales generated by the Dual Teacher model with and without the MLR loss. From the first example on e-SNLI dataset, we see that without MLR constraint the model generates same explanation for neutral and contradiction labels, and the explanation for the neutral label indicates a contradiction. In contrast when trained with MLR, it outputs an explanation which is in alignment with the assigned label. In the second example from ComVE, the model without MLR outputs the same explanation showing that it ignores the label assigned. With MLR constraint the model is able to generate explanations sensitive to the assigned label. Although the reasoning for the incorrect label is wrong, this behavior is still desired for an interpretable system elucidating why a prediction was made.

8.3 Error Analysis

Table 9 shows a snapshot of the qualitative analysis of the errors from our model. From the explanations generated for the predictions, we see that the model is unaware of situations which require additional background information, such as the existence of hair on eyes, or subtle differences between words, such as paws and feet. We believe a better pretrained Language Model can help alleviate some of these issues.

Input: *premise*: a woman plays guitar for tips. *hypothesis*: lessons are there to learn **Label**: neutral **Gold**: a woman playing a guitar does not imply she is giving lessons. **Joint**: a woman can be a teacher and not be a teacher. **Rationalizer**: just because a woman plays guitar does not mean they are there to learn.
Input: *choice1*: she turned in her homework before it was due and got full credit *choice2*: she turned in her homework five years after it was due and got full credit **Label**: choice2 **Gold**: homework must typically be turned in before the due date in order to receive full credit **Joint**: homework is not done when it is due. **Rationalizer**: you cannot turn in your homework five years after it was due
Input: what will friends in elementary school likely do? *choice*: go somewhere *choice*: tell story *choice*: going out at night *choice*: jump rope *choice*: teach **Label**: jump rope **Gold**: kids in elementary school generally play games like hide and seek, tag, musical chairs and so on. **Joint**: jump rope is a rope used to teach others to jump rope. friends in elementary school likely will do something like jumping rope. **Rationalizer**: friends in elementary school will jump rope.

Table 7: Sample explanations generated from a few-shot Joint model as compared to a few shot Rationalizer model.

premise: two little boys wearing athletic jerseys are washing their hands in a public restroom. *hypothesis*: they were playing soccer. **With MLR** **entailment** explanation: boys are washing their hands while playing soccer. **neutral** explanation: just because boys are wearing athletic jerseys, it does not mean they are playing soccer. **contradiction** explanation: boys cannot be wearing athletic jerseys and playing soccer at the same time.
Without MLR **entailment** explanation: boys are boys. **neutral** explanation: boys cannot be washing their hands while playing soccer. **contradiction** explanation: boys cannot be washing their hands while playing soccer.
choice1: bats can fly perfectly. *choice2*: bats can ride bicycles. **With MLR** **choice1** explanation: bats cannot fly. **choice2** explanation: bats cannot ride bicycles.
Without MLR **choice1** explanation: bats cannot ride bicycles. **choice2** explanation: bats cannot ride bicycles.

Table 8: Case studies of generated explanations for varied task labels with and without the MLR loss constraint

Input: choice1: she shaved her eyes. choice2: she shaved her legs. **Label**: choice1 **Gold Explanation**: there is no hairs to shave on the eyes. **Predicted Label** choice2 **Generated explanation:** legs are not razors.
Input: cats have how many appendages? choice: tail choice: whiskers choice: two eyes choice: four paws choice: four legs **Label** four legs **Gold Explanation**: appendage refers to something that is attached four legs are attached to cats four legs are used to walk **Predicted Label** four paws **Generated explanation**: four paws are appendages. cats have two eyes.

Table 9: Qualitative analysis of the prediction errors of our model.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Left blank.

□ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

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

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

B ☑ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

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

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

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

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

C ☑ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
section 5.1

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
section 5.1

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
section 5.1

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
section 5.1

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

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.