Interpreting Indirect Answers Using Self-Rationalizing Models

Yun Li University of Amsterdam
Michael Neely University of Amsterdam
Frederik Nolte University of Amsterdam
{yun.li,michael.neely,frederik.nolte}@student.uva.nl

Abstract

A joint process of label prediction and free-form natural language rationale generation has been shown to offer faithful insights into a model's internal reasoning process. In this work, we formulate a text-to-text mixture task to investigate if we can successfully transfer self-rationalization capacity from existing human-provided explanations of natural language inference and question answering examples to the interpretation of indirect answers to polar questions. We show that our setup enables a model to generate faithful and coherent rationales — as measured by simulatability and human judgments of quality — with a minimal loss in predictive power when interpreting indirect answers. We identify evident shortcomings in our model's reasoning capacity through an extensive analysis of the generated rationales. Our results offer a clear direction in which the community can improve the logical prowess of current text-to-text models.

1 Introduction

Indirect answers to polar (yes/no) questions are abundant in day-to-day conversations. While such answers flout many of the Gricean maxims of the Cooperative Principle (Grice, 1967), they typically advance the state of a conversation far beyond that of a simple 'yes' or 'no'. Humans inherently perceive indirect answers to be more polite — perhaps because the original question tends to be indirect as well (Clark and Schunk, 1980) — and subconsciously translate the literal answer into an interpretation with a corresponding justification.

Correctly interpreting indirect answers relies heavily on commonsense, contextually grounded reasoning shared between the speaker and listener. For example, '*I am a vegan*' is a negative reply to the question '*Do you like red meat?*' precisely because vegans do not eat meat. Without this crucial piece of knowledge, it would not be possible to interpret the response. For this reason, indirect answer interpretation may be an excellent task to concurrently test and improve the reasoning and contextual modeling capacities of neural language models. Louis et al. (2020) introduced the *Circa* dataset of labeled (question, indirect answer) pairs precisely for this reason.

Like most apparent tests of natural language understanding (NLU), it is relatively simple to train a model to accurately classify instances of the Circa dataset. To prove this, we finetuned a pretrained T5-large model (Raffel et al., 2020) on the Circa dataset in a text-to-text multiple-choice question answering (MCQA) setting. Not only does the model reach 92% accuracy on a held-out test set in the RELAXED label setting, it also reaches 82% accuracy when only considering the answer (see Appendix A for more details). Thus, the Circa dataset — at least in its raw form — may be a flawed comprehension test because it does not require logical reasoning or contextual awareness to solve (see e.g., Sugawara et al., 2020, for a thorough investigation of the benchmarking problem in machine reading comprehension). More reliable tests of the NLU capability of language models must extend the evaluation paradigm beyond mere predictive power. One such possibility is to require the model to explain its predictions.

Explanations must be both *faithful* to the reasoning process of the model and *plausible* to humans (Jacovi and Goldberg, 2020). The standard for *faithfulness* is not only higher than that for *plausibility*, it is also more valuable to key stakeholders in critical domains, such as regulators. While the field of Explainable AI (XAI) offers many techniques with which to attribute decisions to various input features to the model (e.g, Sundararajan et al., 2017; Shrikumar et al., 2017), these are both hard to interpret in some contexts and less faithful because the attributions are calculated post-hoc. *Self-*

rationalizing models, which jointly predict the task output (i.e., the label for an indirect answer) and generate free-form explanatory text, set a higher standard for faithfulness (Wiegreffe et al., 2021).

Inspired by Narang et al. (2020), who cast various language tasks into text-to-text problems (Raffel et al., 2020) with a self-rationalizing model, we study the ability of a finetuned transformer-based language model (LM) to rationalize interpretations of indirect answers faithfully. We start from a pre-trained transformer-based LM since such models are general-purpose language learners (Radford et al., 2019) capable of storing vast amounts of relational knowledge (Petroni et al., 2019; Jiang et al., 2020). We hypothesize it should be possible to leverage this knowledge to reason over the highly logical nature of the Circa dataset. Specifically, we ask the following research questions:

RQ1: *Can transformer-based LMs faithfully rationalize and interpret indirect responses without sacrificing accuracy?* ***RQ2:*** *Can we leverage these rationales to identify and, subsequently, address shortcomings in reasoning capacity?*

Following a similar approach to Narang et al. (2020), we implement a transfer learning setup by casting the Circa dataset to a natural language inference (NLI) problem and finetuning a pre-trained, self-rationalizing T5 model on a mixture task leveraging datasets with existing human-generated free text rationales. We then evaluate the faithfulness of the generated rationales on a held-out test set of Circa instances with a previously unseen context.

We find that our model is both highly accurate and highly faithful, as measured by leakage-adjusted simulatibility (LAS) (Hase et al., 2020) and human annotations of rationale quality. Through an extensive analysis of the patterns of generated rationales, we show that our model extracts, adapts, and applies logical templates from the training data. It also generates novel rationales by combining fragments of the input sentence with self-generated text. We demonstrate that flawed rationales can usually be fixed by flipping the polarity of a single word, suggesting that T5 struggles with the concept of negation. We release our code, as well as the rationales generated by our best model, for public analysis and reproducibility[1].

[1] https://github.com/frederiknolte/indirect-response

2 Related Work

There is a rich history of work documenting the tendency of neural networks to solve tasks through simple heuristics rather than true generalization. Evidence of superficial pattern matching is prevalent on textual entailment (Gururangan et al., 2018; McCoy et al., 2019), single- and multi-hop question answering (Sen and Saffari, 2020; Trivedi et al., 2020), dialogue modelling (Sankar et al., 2019), and machine reading comprehension (Kaushik and Lipton, 2018) tasks. Efforts to identify, test, and correct these heuristics roughly fall into one of three complementary paths.

The first approach is to make the task harder for the model, either through adding adversarial examples (e.g., Jia and Liang, 2017; Jiang and Bansal, 2019; Nie et al., 2020) or distractors (Kong et al., 2020). Another possibility is to identify and remove the existing annotation artifacts entirely (Gardner et al., 2021). Since annotation artifacts are exceptionally prevalent in crowdsourced datasets (Linzen, 2020), this direction might be worth exploring for the Circa dataset. The approach of Gardner et al. (2021) presupposes that the correlation between individual features and the class label should never be higher than uniform. Whether this assumption should hold for indirect answers is unclear and would require further analysis. Our study focuses on challenging the model by adding an explainability component, rather than making indirect answer interpretation harder.

Another common research direction is to probe neural models for NLU capability. Diagnostic classifiers (Hupkes et al., 2018) directly probe the structured representations contained within a model's hidden states and can reveal embedded syntactic (Hewitt and Manning, 2019) and contextual (Tenney et al., 2019) knowledge. However, it is difficult to conclude meaningful trends from diagnostic classifiers unless overparameterization and selectivity are regulated with a control task (Hewitt and Liang, 2019). Alternatively, language models can be 'prompted' to demonstrate their knowledge with fill-in-the-blank style tasks. Several recent papers employ this approach to show that transformer-based LMs understand simple inference but struggle with compositionality, commonsense knowledge, multi-hop reasoning, and quantifiers such as 'always' and 'never' (Talmor et al., 2020; Liang and Surdeanu, 2020). Performance on probing tasks is highly dependent on the phrasing of the probe

(Gao et al., 2020; Jiang et al., 2020), suggesting that it is difficult to extract the knowledge contained within an LM manually. We believe that self-rationalization is a more natural and faithful approach to test an LM's reasoning capacity since we can generate rationales without depending on sensitive diagnostic classifiers or contrived data.

The final research direction — and the one most in line with our own — involves adding an explainability component to the model. By analyzing explanations, it becomes possible to identify the areas in which a particular model struggles. We previously mentioned feature-additive methods, but other XAI approaches like rationale extraction (Bastings et al., 2019), and minimal sufficient subsets (Chen et al., 2018; Yoon et al., 2019) are popular as well. All of these methods target different types of ground truth explanations (Camburu et al., 2020), and depend on the learned behavior of the model (Camburu et al., 2019). Another approach is to construct a pipeline model consisting of two modules, one for explanation and one for classification (e.g., Rajani et al., 2019). While this approach can improve downstream predictive performance (Latcinnik and Berant, 2020), Wiegreffe et al. (2021) suggest that self-rationalization (joint prediction and explanation) is a more faithful approach; at least as measured by two metrics they introduce. For this reason, we adopt a self-rationalizing approach in this work.

3 Method

3.1 Transfer Learning from a Mixture Task

Narang et al. (2020) demonstrate the possibility of self-rationalized transfer learning, where a model learns to mimic human-provided rationales in a supervised manner on one or more datasets and then transfers the rationalizing capacity to another dataset. Inspired by their work, we propose the following method:

Given a target dataset $\mathcal{T}$ with labels but no reference rationales, and n datasets $\mathcal{D}_1, ..., \mathcal{D}_n$ drawn from similar domains which contain both labels and reference rationales, it is possible to generate a mixture $\mathcal{M}$ of $2n + 1$ tasks to successfully transfer rationalization to $\mathcal{T}$, provided the input sequences drawn from the $\mathcal{D}_1, ..., \mathcal{D}_n$ are sufficiently similar to those drawn from $\mathcal{T}$. $\mathcal{M}$ is constructed by adding two tasks per dataset D_i, one where the model is only required to predict the correct label, and another where the model is required to self-rationalize (both predict and explain). The final task added to $\mathcal{M}$ is prediction on $\mathcal{T}$ without rationalization. The mixing rate of each task in $\mathcal{M}$ during training should be proportional to the size of its corresponding dataset. Additionally, the best model should be selected based on its performance when self-rationalizing on a held-out validation set of $\mathcal{T}$. Once selected, the best model is then evaluated based on its self-rationalizing performance on a held-out test set of $\mathcal{T}$.

3.2 Evaluation

3.2.1 LAS

We use leakage-adjusted simulatability (LAS) (Hase et al., 2020) to assess the quality and faithfulness of the generated rationales on the target dataset $\mathcal{T}$. In this method, a second 'simulator' model is trained to predict the same labels as the model under evaluation in two settings with access to (1) only the original input sequences and (2) both the inputs and the model-generated rationales. Explanation quality is then estimated based on the increase in probability assigned by the simulator model to the correct label when provided with the rationale instead of just the input sequence.

As some explanations contain direct hints towards the predicted label, Hase et al. (2020) consider two groups of LAS scores: instances where label information is *leaked* by the explanation and instances where no information is leaked. Instances are marked as *leaking* if the simulator can predict the same label as the model solely using the generated rationale as input.

Using the indicator function to denote whether a prediction is correct, the final LAS score is computed as the unweighted mean of the LAS scores for the leaking (LAS_1) and non-leaking (LAS_0) groups:

$$\text{LAS}_0 = \frac{1}{n_0} \sum_{i;k_i=0} \left(\mathbb{1}\left[\hat{y}_i \mid x_i, \hat{e}_i\right] - \mathbb{1}\left[\hat{y}_i \mid x_i\right]\right)$$

$$\text{LAS}_1 = \frac{1}{n_1} \sum_{i;k_i=1} \left(\mathbb{1}\left[\hat{y}_i \mid x_i, \hat{e}_i\right] - \mathbb{1}\left[\hat{y}_i \mid x_i\right]\right)$$

$$\text{LAS} = \frac{1}{2}\left(\text{LAS}_0 + \text{LAS}_1\right)$$

where x_i is the original input sequence, $\hat{y}_i$ and $\hat{e}_i$ are the prediction and rationale of the model under evaluation, $k_i = \mathbb{1}\left[\hat{y}_i | \hat{e}_i\right]$ indicates label leakage and n_0 and n_1 are the numbers of non-leaked/leaked samples, respectively.

3.2.2 Human Evaluations

We gather human ratings of rationale quality using Amazon Mechanical Turk. We ensure that the survey covers all cases regarding LAS (leaked/nonleaked) and prediction accuracy (correct/incorrect). For each rationale, we ask three annotators to rate its quality on a 5-point Likert scale, with 1 being the worst and 5 the best. All reported ratings are normalized to account for systematic differences between workers if not indicated otherwise. We provide details about the collection and the normalization processes in Appendix D.

We present the annotators with either the gold standard or the predicted label for instances where our model makes the wrong prediction. The correctness of the label is not disclosed to the annotator — to encourage them to focus on evaluating the quality of the rationale and whether it supports the given label. This setup also allows us to measure faithfulness directly since a reasonable rationale for a wrong prediction is still desirable because it may allow us to diagnose why the model made a mistake.

4 Experiments

4.1 Datasets and Mixture

The Circa dataset (Louis et al., 2020) is a crowdsourced dataset containing (question, answer) pairs. The 'questioners' were instructed to ask polar questions in one of ten prescribed social contexts. Then, the 'answerers' were tasked with providing indirect responses to those questions. Each answer is then associated with one of six labels in the RELAXED setting and one of eight labels in the STRICT setting. Table 1 gives an example of two Circa instances. Additionally, there are two context settings. The ten social contexts are randomly distributed across all splits in the MATCHED setting, while a unique subset is held out in the validation and test splits in the UNMATCHED setting. Therefore, the UNMATCHED setting is a better test for the model's ability to generalize to unseen social contexts.

Context	Talking to a friend about food preferences.
Q	Do you like pizza?
A	I like it when the toppings are meat, not vegetable.
Gold standard	Yes, subject to some conditions
Context	Meeting a new neighbour.
Q	Would you like to grab a coffee?
A	I thought you'd never ask.
Gold standard	Yes

Table 1: Example conversations and gold standard labels from the Circa dataset.

Since the original dataset[2] does not specify splits, we created our own for both the MATCHED and UNMATCHED settings. For the MATCHED setting, all ten contexts appear during training, and we randomly divide all examples into 60%/20%/20% for train, validation, and test splits, respectively. For the UNMATCHED setting, we randomly divide the ten contexts into 6/2/2 to study the effect of unseen scenarios. We repeat both sampling procedures using three random seeds for a total of six unique versions of the Circa dataset. We make our splits publicly available[3] for reproducibility. Appendix B shows the aggregated statistics of our unique versions in both the MATCHED and UNMATCHED settings. We train, validate, and test one version of our models on each unique version of the dataset, using the same seed to control the random seeds of the Python libraries used to tune our model.

The Circa dataset is not supplied with rationales for why the gold standard labels are chosen, so we need to select datasets with human-provided rationales in similar domains in order to use the transfer learning mixture approach we described in Section 3. We therefore create a training mixture containing the e-SNLI (Camburu et al., 2018) and CoS-E (Rajani et al., 2019) datasets, drawing inspiration from the MNLI (Williams et al., 2018) baseline models of Louis et al. (2020). In this experiment, Louis et al. (2020) cast the RELAXED setting of Circa to an NLI task by mapping the labels 'no' → 'contradiction', 'yes' → 'entailment', and 'in the middle' → 'neutral'. They set the premise to the declarative form of the question and the hypothesis to the indirect answer. As the model can only predict three of the six RELAXED labels, there is a ceiling on performance. Still, their BERT model (Devlin et al., 2019) is highly accurate in this setting.

We modify the NLI setup of Louis et al. (2020) in the RELAXED setting by reducing the 'unanswerable' labels (those without NLI counterparts) into a special 'none' label, which we take to mean 'neither a definite yes, nor a definite no, nor a neutral response'. This process removes the performance ceiling and improves the robustness of our models, allowing them to deal with any ambiguous cases, e.g., when employing our model in a dialogue gen-

[2] https://github.com/google-research-datasets/circa/

[3] https://github.com/frederiknolte/indirect-response/tree/main/circa/circa_splits

eration setting, the system could be augmented to ask a clarification question whenever it classifies an answer as 'none'. The 'none' label covers 11.2% of the data points. Unless stated otherwise, all of our experiments operate in this setting.

By mimicking the NLI setup of Circa, we can leverage the e-SNLI dataset. The CoS-E dataset, on the other hand, is in a multiple-choice format. We set the question as premise and the various answers as individual hypotheses. We thus obtain our mixture $\mathcal{M}$ of five tasks: predicting e-SNLI instances, predicting and rationalizing e-SNLI instances, predicting modified CoS-E instances, predicting and rationalizing modified CoS-E instances, and predicting modified Circa instances.

We experimented with an MCQA mixture, where we used `question:` and `answer:` keywords to denote the Circa questions and indirect answers — as well as the premise and hypothesis for e-SNLI instances — and presented each label with a `choice:` keyword. The model was accurate in this setting but could not generate rationales. Since the bulk of the rationales available for transfer learning are from the e-SNLI dataset, we hypothesize that deviation from the most prevalent setting in the mixture $\mathcal{M}$ prevents the model from correctly rationalizing the target task $\mathcal{T}$. This limitation is unfortunate because there are not many datasets available with human-provided rationales (see Wiegreffe and Marasovic, 2021, for an overview of Explainable NLP datasets). With the currently available datasets, our approach does not work outside of the NLI setting.

4.2 Model

We select T5 (Raffel et al., 2020) as our model, since Narang et al. (2020) demonstrate success when self-rationalizing in a text-to-text format.

We use special tokens to format the instances from our three datasets. Sequences begin with an optional `explain` keyword when we want the model to generate a rationale and are followed by the dataset identifier keyword `cos_e` when predicting instances from the CoS-E dataset. This is necessary for the model to distinguish the unique format of these instances from the NLI ones. We use the `nli` keyword to indicate that we want the model to generate one of the four labels explained previously. Special `hypothesis:` and `premise:` keywords denote the hypothesis and premise, respectively. In the UNMATCHED setting of Circa we add a `context:` keyword before the context. For CoS-E instances, we prepend each multiple-choice option with a `choice:` keyword. As an example, the second Circa example in Table 1 would be fed into our model in the UNMATCHED setting as `nli context: Meeting a new neighbour. hypothesis: I would like to grab coffee. premise: I thought you'd never ask.`.

4.3 Training Details and Hyperparameters

We start from a pretrained LARGE T5 model with around 770 million parameters, preserving the hyperparameters of Narang et al. (2020). Specifically, we finetune for 20,000 steps with 65,536 tokens per batch, a maximum input sequence length of 512 tokens, and a maximum output sequence length of 256 tokens. We use the Adafactor optimizer (Shazeer and Stern, 2018) with a constant learning rate of .001. We also apply dropout with a rate of 0.1. Readers may refer to our codebase for more details. We apply greedy decoding to obtain the output sequences and repeat each experiment three times, using three random seeds. The seeds are selected based on the version of the Circa dataset used. We then select the finetuned checkpoint with the highest accuracy on the Circa validation set, which is used to predict and rationalize all instances from the held out test set. We train one model per seed in both the MATCHED and UNMATCHED settings. All experiments are performed on v3-8 TPUs offered by Google Cloud.

4.4 Models Variants

Apart from the core model trained on the mixture $\mathcal{M}$ described in section 4.1, we also perform ablation studies by varying the training data. The four baseline models are used to benchmark our core model accuracies. They are (1) **Circa only**, where the model is only finetuned on the Circa dataset without generating rationales; (2) **Hypothesis only**, where the model is finetuned on the Circa dataset without hypotheses; (3) **Premise only**, where the model is finetuned on the Circa dataset without premises; and (4) **Zero-shot transfer from e-SNLI and CoS-E**, where the model is first finetuned on a mixture containing e-SNLI and CoS-E, and then used to perform zero-shot evaluation (prediction with rationales) on the Circa dataset.

4.5 Evaluation

As a simulator for LAS, we use a pretrained BASE-CASED DistilBERT model which we fine-tune with the predictions and explanations our T5 model produced on the appropriate Circa train dataset split. Similar to the original work of Hase et al. (2020), the simulator is trained using both original input and T5 rationales as input. We mask the input and rationales with probabilities 0.2 and 0.4, respectively, in order to drive the simulator towards using both sources of information.

To build our survey, we randomly select 72 explanations from our held-out test set generated by our best-performing model in the UNMATCHED setting. We aggregate opinions from 11 annotators, including experts as well as the general public.

5 Results

5.1 Prediction Accuracy

Table 2 shows the average accuracies for all model variants on held-out test splits. Our results can be (roughly) compared to the BERT-MNLI-YN model reported by Louis et al. (2020), whose accuracy is presented in a three-class setting (entailment/contradiction/neutral). In contrast, we operate in a four-class setting with an extra 'none' label. The premise-only model also reaches a relatively high accuracy of 79%, suggesting the non-rationalizing model may rely on shortcuts in the classification process, rather than taking into account the entire conversation. Our core model performs significantly better than its zero-shot counterpart while suffering from only a 1.5% drop in accuracy compared to the Circa-only variant. This demonstrates our model's ability to learn task-specific skills while also applying the reasoning process obtained from other datasets.

Model	MATCHED	UNMATCHED
Circa only	91.46 (0.24)	90.37 (1.32)
Premise only	79.13 (0.27)	79.16 (4.06)
Hypothesis only	52.97 (0.88)	52.24 (2.86)
Zero-shot	44.09 (1.26)	43.11 (3.92)
Core model	89.90 (0.29)	88.80 (1.32)

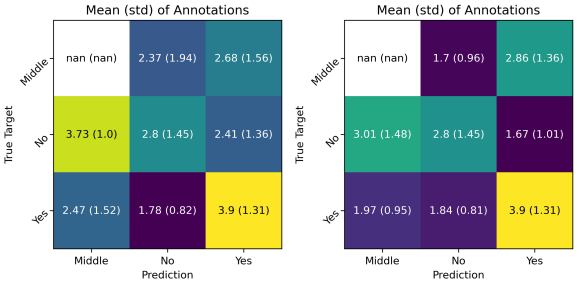
Table 2: Average classification accuracy on the Circa test splits (standard deviation estimated from 3 seeds).

5.2 LAS

The average LAS scores over all three seeds are stated in Table 3. It is apparent that most of the generated rationales contain information towards the model prediction in both MATCHED and UNMATCHED settings. This is not bad per se, as Hase et al. (2020) report that 85% of the CoS-E dataset and 97% the e-SNLI dataset leak. However, LAS scores are not directly comparable to those reported by Hase et al. (2020) as the datasets, and thus the difficulties of the tasks differ.

	MATCHED	UNMATCHED
# leaked	5940 (107)	5906 (152)
# nonleaked	253 (11)	385 (188)
LAS score	9.54 (3.64)	13.21 (5.54)

Table 3: Mean (std) of LAS scores on test splits.



(a) Prediction shown. (b) Target shown.

Figure 1: Mean (std) of annotations per true target and prediction split by whether true target or prediction is shown to the annotator.

5.3 Human Evaluations

Rating:	1	2	3	4	5
# annotations:	111	62	29	46	52

Table 4: Annotations per rating (not normalized).

For the following human evaluations, we select rationales of the UNMATCHED model seed with the highest LAS score in the non-leaking case. The number of annotations per possible rating, as shown in Table 4, show that reviewers tend towards rating rationales either very low or high. We suspect that this is a consequence of rephrasing the Circa inputs as an NLI task, in which statements are either correct or incorrect. The overall, normalized annotation mean is 2.69, which indicates annotators are neutral in their judgments on average, further supports this trend. Finally, the unnormalized annotations show an ordinal Krippendorff's alpha of 0.46, which suggests that annotator agreement is systematic.

Figure 1 depicts the mean annotations per target and prediction. In case the model predicts 'in the middle', the perceived quality of rationales is

higher for showing a wrong model prediction than for showing the true target. This suggests that if the model is unable to make a polar decision, the rationale remains faithful to the prediction.

We further observe that the model is better at generating rationales that support positive answer interpretations than those which support negative answers. This is supported by the following: (1) If the true target is 'Yes' and it is correctly identified, the rating is higher than if the true target is 'No' and it is correctly identified. (2) If the true target is 'Yes' and it is wrongly predicted as 'No', the rating is higher if 'Yes' is shown. At the same time, if the true target is 'No' and the prediction is 'Yes', the rating is again higher if 'Yes' is shown.

Leaked	Correct prediction	True target shown	Annotation mean (std)
No	False	False	2.74 (1.58)
		True	1.96 (1.08)
	True	True	3.13 (1.44)
Yes	False	False	2.12 (1.20)
		True	2.08 (1.22)
	True	True	3.64 (1.46)

Table 5: Mean annotations by label leakage, correctness of prediction and whether the true target was shown to the annotators.

Table 5 shows an apparent difference in faithfulness of the rationales between leaking and non-leaking samples. In the leaking case, showing the true target for a wrong prediction instead of the prediction itself only leads to a marginal decrease in perceived explanation quality. As leaking explanations contain information towards the model prediction, an erroneous prediction could negatively affect the explanation quality for a given sample. However, as annotators were asked to judge the suitability of explanations towards the shown target, regardless of the correctness of this target, this result is peculiar. Opposed to that, for non-leaking samples, the perceived rationale quality drastically increases when showing the wrong model prediction instead of the true target. These observations could indicate that it is significantly harder for the annotators to judge the suitability of a rationale for a wrong prediction when the rationale leaks information about this wrong prediction. This could be due to wrong predictions being easy to identify by humans and thus rationales that contain information towards wrong predictions being negatively perceived. In contrast, the suitability of a rationale for a wrong prediction could be easier to judge if the rationale does not contain much information about the label and is therefore not as easily perceived to be of low quality. This lets us observe that non-leaking rationales are highly faithful towards the model predictions.

6 Discussion

6.1 Quantifying Indirectness

We note that some indirect answers from the Circa dataset are, in fact, rather direct and do not require much 'reasoning' for the model to arrive at the right conclusions. These are often generic polar responses (e.g. 'Not really', 'Sounds good'), or rephrasings of the question (e.g. X: 'Would you have to work weekends?' Y: 'I never work weekends.'). The directness is especially prominent in conversations labelled with 'Yes' or 'No', and could affect the quality of the rationales and the benchmarking capacity of the Circa dataset.

To investigate the extent of this effect, we collect some generic responses from the Circa dataset, and calculate the BLEU score — using the SacreBLEU package (Post, 2018) — between the answers and a reference set containing the generic responses and the declarative form of the questions. Table 6 shows the BLEU scores grouped by the RELAXED labels. All values are significantly different from each other, apart from the 'other' category (see Appendix C for details). Of the three NLI labels we consider in this report, responses that are 'in the middle' are less likely to be generic or to be restatements of the question.

Label	BLEU
No / Contradiction	3.19
Yes / Entailment	2.67
In the middle / Neutral	2.04
Yes, subject to conditions	1.19
Other	2.35

Table 6: BLEU score between responses and questions (supplemented by a list of generic answers, see Appendix C) in the Circa dataset, grouped by target label.

We also compare this with the LAS score per target category of the generated rationales in the UNMATCHED setting, shown in Table 7. As expected, the 'in the middle / neutral' category has the highest rationale quality based on this metric. Interestingly, our model struggles to predict the correct label for neutral instances (average accuracy $= 0.214$). This could be due to label imbalance since neutral examples are less prevalent. Regard-

less, we have identified a clear shortcoming in its reasoning capacity.

	Train	Dev	Test
Contradiction	18.04	20.55	23.96
Entailment	0.77	16.72	9.21
Neutral	**31.97**	**33.00**	**30.08**

Table 7: LAS scores of the rationales for the UNMATCHED setting, grouped by target label.

6.2 Patterns in Rationales

We also identify several 'templates' that our model uses to generate rationales for Circa, most of which follow the pattern of a phrase, followed by a logical connector, and concluded by another phrase. We find three categories of templates with decreasing complexity: (1) complex logical templates with verbose connectors like 'then it is logical to conclude that', (2) naive logical templates with simple connectors like 'is a' or 'not', and finally (3) *parroting* templates, where the model generates some combination of the premise, hypothesis, and /or context. Together, templates of these forms account for 92% of all generated rationales (41% complex, 50% naive, 1% parrot). Crucially, the complex templates are quite frequent, which is essential because these templates have longer natural language phrases. We find that, when generating templated rationales, the model adds new text or directly modifies existing text from the input. For this reason, and because the parroting templates are so infrequent, it is clear that our model understands how to format novel logical statements correctly — even if the truth value of that statement clashes with the interpretation of the indirect answer.

For the sake of brevity, readers may refer to Appendix E for a description of the templates, as well breakdowns of the templates by predicted label, micro F1 and LAS scores, and average annotator quality, as well as analysis of rationale novelty. The model's decision to apply a particular template largely depends on the label prediction. For example, negation templates are almost exclusively applied when the label is contradiction. This tendency further reinforces our opinion that the survey annotators were evaluating *plausibility* to the ground-truth label, rather than *faithfulness* to the predicted label. If the model applies a negation template when it predicts a contradiction, this should inherently be faithful.

The most common (yet, still largely infrequent) mistake our model makes is inappropriately applying 'negation' templates (e.g., 'is not a') to predictions of entailment, and 'assignment' templates (e.g., 'is a') to contradictions. Thus, simply removing or adding a single 'not' token could drastically improve the quality of the rationale. Interestingly, one *modus tollens* style logical template (e.g., 'just because *some phrase*, does not imply *some phrase*') accounts for almost all of the neutral cases. Based on our frequent observation of this template — and the model's low accuracy in the neutral setting — we hypothesize that the reasoning capacity of T5 can be improved by teaching it to understand modus tollens logic (see Betz, 2020, for an example of training a transformer-based LM to understand logical schemes).

7 Conclusion

We can use our results from Sections 5.1 and 5.2 to answer **RQ1** affirmatively: it is possible to learn to faithfully rationalize without sacrificing accuracy. Our results from Section 5.3 are less clear and we treat them as a useful pilot study to indicate deficiencies in our survey setup we could fix in future work. We can also partially answer **RQ2**. We have identified shortcomings in the model's modus tollens reasoning, particularly for neutral responses. We observe our model occasionally using some templates in the wrong cases. This is an unfortunate consequence of the transfer learning setup: the model does not receive any supervision on which templates are appropriate for which labels on the Circa dataset.

In future work, we will formulate strategies to improve logical reasoning, particularly around negation and modus tollens schemes. Transformer-based LMs struggle with the contextual impacts of negation and quantifiers (Rogers et al., 2020; Ettinger, 2020), both of which are essential reasoning skills to interpret indirect answers. Perhaps this is because Transformers may model language purely from a distributional perspective and exploit co-occurrence statistics to solve language tasks (Sinha et al., 2021). Such behavior does not align with the way humans process language, and may be corrected by augmenting the training scheme to promote sensitivity to desirable phenomena (Glass et al., 2020; Yang et al., 2019). Whether we should expect neural networks to solve tasks in the same manner as humans is another question entirely.

Acknowledgments

This work was supported by the Google TPU Research Cloud Program. We would also like to thank Dr. Raquel Fernández, Mario Giulianelli and Ece Takmaz for their assistance in conducting this research project.

References

Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2963–2977, Florence, Italy. Association for Computational Linguistics.

Gregor Betz. 2020. Critical thinking for language models. *CoRR*, abs/2009.07185.

Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. 2019. Can I trust the explainer? verifying post-hoc explanatory methods. *CoRR*, abs/1910.02065.

Oana-Maria Camburu, Eleonora Giunchiglia, Jakob Foerster, Thomas Lukasiewicz, and Phil Blunsom. 2020. The struggles of feature-based explanations: Shapley values vs. minimal sufficient subsets. *CoRR*, abs/2009.11023.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural language inference with natural language explanations.

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

Herbert H. Clark and Dale H. Schunk. 1980. Polite responses to polite requests. *Cognition*, 8(2):111–143.

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

Allyson Ettinger. 2020. What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models. *Transactions of the Association for Computational Linguistics*, 8:34–48.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *CoRR*, abs/2012.15723.

Matt Gardner, William Merrill, Jesse Dodge, Matthew E. Peters, Alexis Ross, Sameer Singh, and Noah Smith. 2021. Competency problems: On finding and removing artifacts in language data.

Michael Glass, Alfio Gliozzo, Rishav Chakravarti, Anthony Ferritto, Lin Pan, G P Shrivatsa Bhargav, Dinesh Garg, and Avi Sil. 2020. Span selection pre-training for question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2773–2782, Online. Association for Computational Linguistics.

Herbert Paul Grice. 1967. Logic and conversation. In Paul Grice, editor, *Studies in the Way of Words*, pages 41–58. Harvard University Press.

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

Peter Hase, Shiyue Zhang, Harry Xie, and Mohit Bansal. 2020. Leakage-adjusted simulatability: Can models generate non-trivial explanations of their behavior in natural language? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4351–4367.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2733–2743. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4129–4138. Association for Computational Linguistics.

Dieuwke Hupkes, Sara Veldhoen, and Willem H. Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *J. Artif. Intell. Res.*, 61:907–926.

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, Online. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2021–2031. Association for Computational Linguistics.

Yichen Jiang and Mohit Bansal. 2019. Avoiding reasoning shortcuts: Adversarial evaluation, training, and model development for multi-hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2726–2736, Florence, Italy. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Divyansh Kaushik and Zachary C. Lipton. 2018. How much reading does reading comprehension require? a critical investigation of popular benchmarks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5010–5015, Brussels, Belgium. Association for Computational Linguistics.

Xiang Kong, Varun Gangal, and Eduard Hovy. 2020. SCDE: Sentence cloze dataset with high quality distractors from examinations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5668–5683, Online. Association for Computational Linguistics.

Veronica Latcinnik and Jonathan Berant. 2020. Explaining question answering models through text generation.

Zhengzhong Liang and Mihai Surdeanu. 2020. Do transformers dream of inference, or can pretrained generative models learn implicit inferential rules? In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 76–81, Online. Association for Computational Linguistics.

Tal Linzen. 2020. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.

Annie Louis, Dan Roth, and Filip Radlinski. 2020. "I'd rather just go to bed": Understanding indirect answers.

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.

Sharan Narang, Colin Raffel, Katherine Lee, Adam Roberts, Noah Fiedel, and Karishma Malkan. 2020. WT5?! Training text-to-text models to explain their predictions.

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

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases?

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

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

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

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! Leveraging language models for commonsense reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Chinnadhurai Sankar, Sandeep Subramanian, Christopher Pal, Sarath Chandar, and Yoshua Bengio. 2019. Do neural dialog systems use the conversation history effectively? an empirical study.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online. Association for Computational Linguistics.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4603–4611. PMLR.

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.

Koustuv Sinha, Robin Jia, Dieuwke Hupkes, Joelle Pineau, Adina Williams, and Douwe Kiela. 2021. Masked language modeling and the distributional hypothesis: Order word matters pre-training for little. *CoRR*, abs/2104.06644.

Saku Sugawara, Pontus Stenetorp, Kentaro Inui, and Akiko Aizawa. 2020. Assessing the benchmarking capacity of machine reading comprehension datasets. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8918–8927. AAAI Press.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org.

Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2020. oLMpics-on what language model pre-training captures. *Transactions of the Association for Computational Linguistics*, 8:743–758.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2020. Is multihop QA in DiRe condition? measuring and reducing disconnected reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8846–8863, Online. Association for Computational Linguistics.

Sarah Wiegreffe and Ana Marasovic. 2021. Teach me to explain: A review of datasets for explainable NLP. *CoRR*, abs/2102.12060.

Sarah Wiegreffe, Ana Marasović, and Noah A. Smith. 2021. Measuring association between labels and free-text rationales.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019. Assessing the ability of self-attention networks to learn word order. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3635–3644, Florence, Italy. Association for Computational Linguistics.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. 2019. INVASE: instance-wise variable selection using neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

A Multiple Choice Question Answering Baseline

To estimate the relative difficulty of the Circa dataset, we finetune a pretrained T5 LARGE model in the RELAXED, MATCHED setting using a text-to-text multiple-choice question answering format. We add a special `circa` dataset identifier and preface the question and answers with `question:` and `answer:`, respectively. Like Narang et al. (2020), we append each label with a `choice:` keyword. For example, a full input would like: `'circa question: Are you into books? answer: I prefer movies. choice: Yes choice: Yes, subject to some conditions choice: No choice: In the middle, neither yes nor no choice: NA choice: Other'`. Unlike our NLI setting, the MCQA format has no ceiling on performance, since the model can predict all of the labels. As an ablation test, we also measure the performance of our T5 model when we remove the Circa question from the input. The tuning process and hyperparameters for these experiments are identical to those discussed in Section 4.3. We can conclude that the Circa task is relatively easy because the model is highly accurate in both settings.

B Statistics on Circa splits

Table 8 shows relevant statistics for our custom Circa splits, aggregated over three random seeds.

C Significance Test for BLEU Scores

While the 'directness' of a response is hard to quantify, we can approximate it by measuring word overlap between the responses and a list of generic responses, or the overlap between the responses and the questions rephrased in declarative form. The former can capture generic polar responses that do not contain 'yes' or 'no' explicitly, and the latter can capture responses that are restatements of the question.

Below are some examples of generic responses found in the Circa dataset. This is not a comprehensive list, but rather clues to help us study how the directness of answers affect rationale quality.

Generic positive answers:

```
sounds good / sounds great
good idea / great idea
let's do it
I think so
I'd love to
```

Generic negative answers:

```
not anymore
not really
not yet
I'm not a fan
I have not
```

We choose BLEU as a metric for measuring word overlap, but in principle any similar metric could be used.

Since we do not know the distributions of the BLEU scores for different target labels, we use a non-parametric test to calculate statistical significance between different labels. We first group all conversations by RELAXED labels, and for every label we create 100 samples each containing 100 conversations (sampled with replacement). A BLEU score is calculated for each sample, and finally we calculate the two-sided Wilcoxon signed-rank test between labels using pairs of 100 BLEU scores. The p-values are shown in Table 9.

D Data Collection for Human Evaluation

We use two surveys on Amazon Mechanical Turk sandbox to evaluate the quality of rationales generated by our model. The first survey asks the annotators to classify three indirect responses into one of 5 categories as in the UNMATCHED setting, similar to Step 4 of the process used by Louis et al. (2020). This is to ensure we only select annotators that have a good grasp of indirect responses.

In the second survey we present explanations along with the original conversation to the annotator, and ask them to rate the quality of the explanations on a scale of 1 to 5, with 1 being the worst and 5 the best. The user interface and an example annotation task are shown in Figure 2.

We generate two batches of questions. For the first batch we uniformly sampled 15 examples from each of the four categories (LAS leaked/nonleaked) $\times$ (correct/incorrect prediction). One of these questions has a formatting error and is later discarded. For the second batch we randomly sampled 41 examples from the remaining part of the test set. The conversations used as questions in the first part of the survey or as examples for survey instructions are excluded from the second survey. Due to time constraints, not all questions are rated by 3 annota-

Setting	Split	Count per label none	Entailment	Contradiction	Neutral	Avg seq length
matched	train	2341 ± 45	9980 ± 33	7669 ± 40	570 ± 18	57.1 ± 0.1
matched	val	748 ± 12	3337 ± 17	2582 ± 42	188 ± 17	57.1 ± 0.3
matched	test	769 ± 37	3311 ± 50	2582 ± 27	191 ± 7	57.0 ± 0.2
unmatched	train	2277 ± 185	10081 ± 307	7664 ± 236	600 ± 3	120.1 ± 0.4
unmatched	val	795 ± 70	3215 ± 178	2556 ± 266	163 ± 31	117.0 ± 5.9
unmatched	test	785 ± 183	3331 ± 138	2614 ± 31	186 ± 34	117.1 ± 6.8

Table 8: Aggregated statistics for our Circa splits (average $\pm$ standard deviation).

	Yes	In the middle	Yes, subject to conditions	Other
No	0.014	7.21×10^{-10}	4.47×10^{-17}	5.13×10^{-6}
Yes	-	0.002	5.78×10^{-16}	0.014
In the middle	-	-	5.24×10^{-8}	0.083
Yes, subject to conditions	-	-	-	2.67×10^{-11}

Table 9: p-values for Wilcoxon signed-rank test on BLEU scores between target labels.

tors. Questions with less than 3 annotations were discarded before evaluating the results.

D.1 Normalization of Ratings

To account for systematic differences between annotators, the annotations were normalized such that each annotator has the same mean annotation and the same standard deviation across annotations. The ratings were altered to have the average intra-annotator mean m as per-annotator mean and the average intra-worker standard deviation std as per-annotator standard deviation:

$$m = \text{mean}\Big(\quad \text{mean}(r_{1,1}, \ldots, r_{1,m}), \quad \ldots, \quad \text{mean}(r_{n,1}, \ldots, r_{n,m}) \Big)$$

$$std = \text{mean}\Big(\quad \text{std}(r_{1,1}, \ldots, r_{1,m}), \quad \ldots, \quad \text{std}(r_{n,1}, \ldots, r_{n,m}) \Big)$$

where $\text{mean}(\cdot)$ computes the mean over all arguments, $\text{std}(\cdot)$ computes the standard deviation over all arguments and $r_{i,j}$ is the rating of annotator $i \in \{1, \ldots, n\}$ for explanation $j \in \{1, \ldots, m\}$. If an annotator has not rated an explanation, the corresponding item is left out of the calculation.

D.2 Comparison of Human Scores to LAS

Table 10 shows the average human rating per prediction type and simulator correctness as well as the coefficients for a linear regression predicting the human annotations from the accuracy of the simulator. Similar to the observations by Hase et al. (2020), we note that human judgements and simulator correctness seem to be correlated. However, in our results, the null-hypothesis of no correlation would not be rejected under any reasonable significance level. This might be due to the very small number of human annotations on which we can base our analysis. Nevertheless, we note that in contrast to the results of Hase et al. (2020), the simulator correctness using only rationales as input shows the least correlation with human ratings whereas the simulator correctness based on both inputs and rationales shows the strongest correlation. While the validity of this observation should be confirmed in future work with more extensive human annotations, this result could indicate that the LAS framework is a reasonable choice to estimate the quality of rationales in our case.

E Patterns in Circa Rationales

E.1 Descriptions of Templates

We name and describe the complex and naive logical templates we observed in the Circa rationales generated by our T5 model. Naive templates are prefaced with 'naive_'. For the exact regular expressions used to capture the templates, we refer

You will be shown short dialogues between two friends/colleagues X and Y. X and Y are in a certain context. In all the dialogues, X asks a simple 'Yes/No' question, and Y answers the question indirectly with a short sentence or phrase. For example:

Context: X wants to know about Y's food preferences.

Question (X): "Do you eat red meat?"

Answer (Y): "I am a vegetarian."

An interpretation to Y's answer is given, and it is from one of these categories:

1. Yes
2. Yes, subject to some conditions
3. No
4. In the middle, neither yes nor no
5. Other

For this task we provide explanations for the interpretations, and we need your help to rate the quality of the explanations on a scale of 1 - 5. Below is an example of a good explanation that supports the interpretation:

Interpretation: No

Explanation: Vegetarians don't eat meat.

Here are some important criteria to keep in mind:

1. 1 is the worst, which means the explanation either contradicts the answer choice or is meaningless. 5 is the best, which means the explanation explains the answer choice very well with meaningful content.
2. Explanations in following cases should be rated low:
    1. Contradict the answer choice, or support a different answer choice;
    2. Meaningless or irrelevant, e.g., "this is the only/best choice";
    3. Only repeat the question;
    4. Only repeat the answer choice without any other content;
    5. Internally contradictory, e.g., "choice A is right because choice B is right".

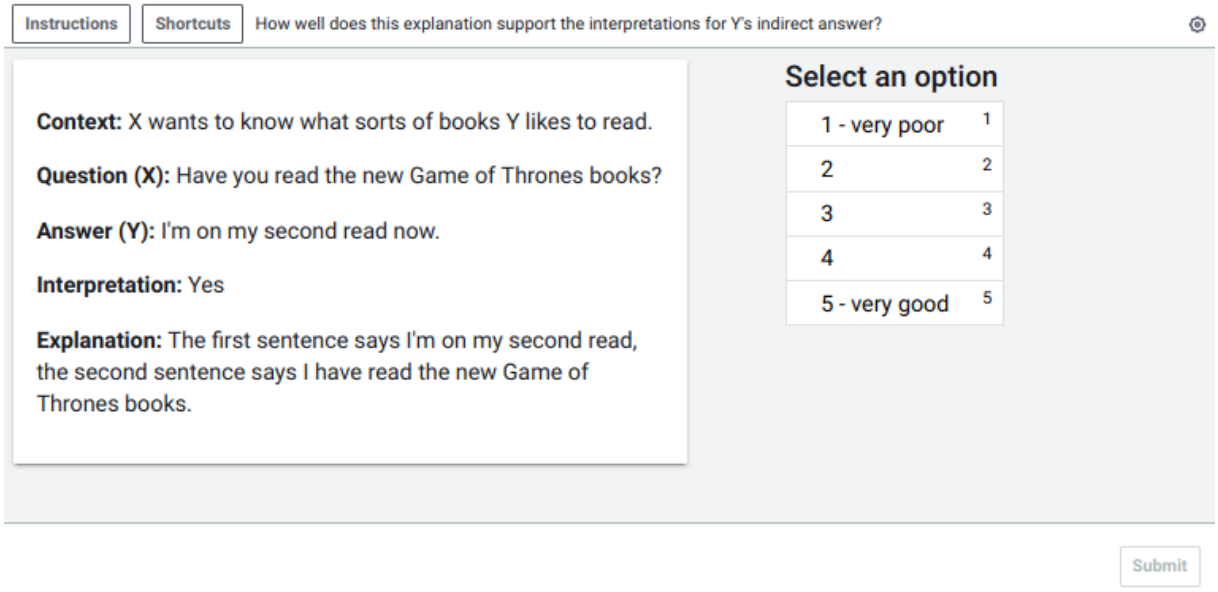


Figure 2: User Interface of Mechanical Turk for the second part of our survey.

Prediction Type	Simulator Correctness		Regression Coef.	
	0	1	β	p
$\hat{y}\|x, \hat{e}$	2.41 (1.40)	2.77 (1.51)	.35	.10
$\hat{y}\|x$	2.58 (1.48)	2.82 (1.50)	.24	.16
$\hat{y}\|\hat{e}$	2.61 (1.45)	2.76 (1.52)	.15	.38

Table 10: Mean (std) human annotation by simulator correctness for different inputs. Regression coefficients for linear regression predicting annotations from simulator correctness.

readers to our codebase.

- Entailment Templates
  - rephrasing/synonymy: `<blank> is a rephrasing of <blank>.`
  - equality: `<blank> is the same as <blank>.`
  - only option: `<blank> is the only <blank>.`
  - implication: `<blank> implies <blank>.`
  - naive assignment: `<blank> is a <blank>.`
  - naive conditional: `if <blank> then <blank>.`
- Contradiction Templates
  - comparison: `<blank> is greater than <blank>.`
  - contradictory: `<blank> contradicts <blank>.`
  - opposite: `<blank> is the opposite of <blank>.`
  - either or: `<blank> is either <blank> or <blank>.`
  - naive negation: `<blank> is not <blank>.`
- Neutral Templates
  - denying the consequent: `Just because <blank>, does not mean <blank>.`

We name templates which 'parrot' (i.e., restate) different components of the input sequences with a 'parrot_' preference. For example, if our T5 model simply restates the premise and the hypothesis of the input sequence, we label that rationale as following the 'parrot_premise_hypothesis' template. We denote any rationale which does not match either a complex, naive, or parroting template as 'unique'.

E.2 Distributions of Templates

Figures 3 and 4 show the distribution of templates across the predicted labels for the entire UNMATCHED dataset, as well as the held-out test set, for our best performing model (as measured by accuracy). The distributions for the full dataset and held-out test set are roughly identical. We see that 'naive assignment', 'naive negation', and 'rephrasing' are among the most frequent patterns. This coincides with the observation in Section 6.1 that many Circa answers are in fact rather direct, so explanations following these patterns may naturally arise. The model has clearly learned which patterns are (generally) acceptable for each predicted label. For example, 'equality', 'implication', 'naive assignment', and 'rephrasing' are common when predicting entailment, and 'either or', 'naive negation', and 'opposite' are common when predicting contradictions. As noted in Section 6.2, nearly all neutral prediction use the 'denying the consequent' template.

E.3 Correctness of Templates

Figure 5 shows the micro F1 scores for the distribution of templates by predicted label on the held-out test set. We use the micro F1 score, instead of accuracy, to account for label imbalance. We observe that most of model's mistakes fall into two categories: (1) inappropriately using 'naive assignment' or 'naive_conditional' for the contradiction case; (2) inappropriately using 'naive negation' for the entailment case. The F1 scores (shown in Figure 5) are still high despite the inappropriate usage of some templates. As discussed in Section 6.2, this is probably due to the lack of Circa rationale supervision in our mixture task. Additionally, we note a poor F1 score for modus tollens logic applied to the neutral case. In Section 7, we discuss possible ways to address this problem in our T5 model.

E.4 Faithfulness of Templates

A natural question to ask is whether some templates are more *faithful* than others. We use LAS as proxy for faithfulness, as it measures how well an observer can use model explanations to predict the model's output, while controlling for explanation leakage. Figure 6 shows the percentage of leakage by template. Most templates do, in fact, leak

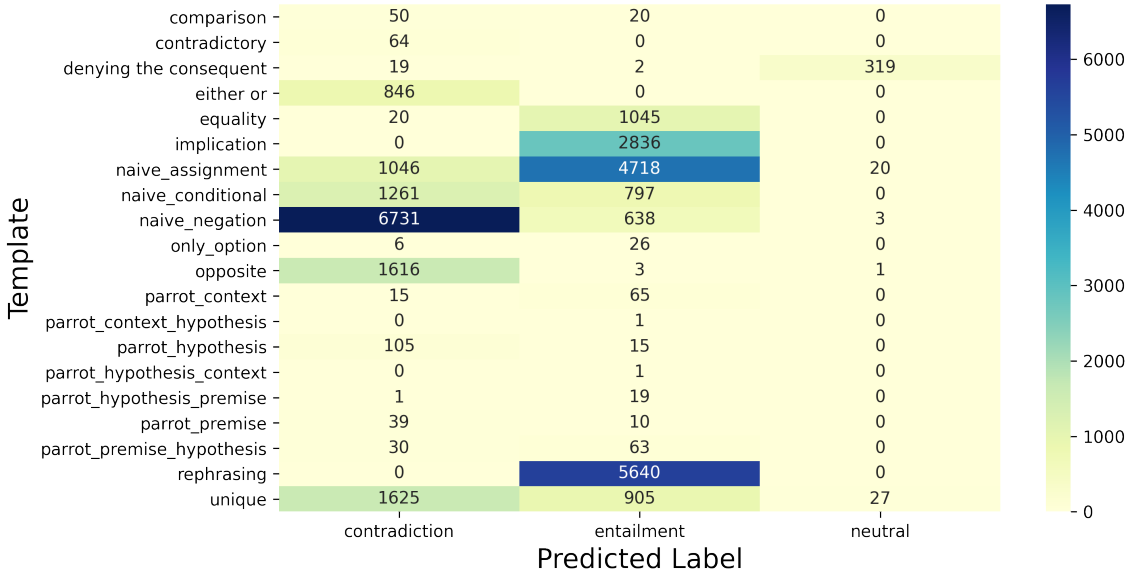


Figure 3: Template distribution by predicted label for the entire UNMATCHED Circa dataset.

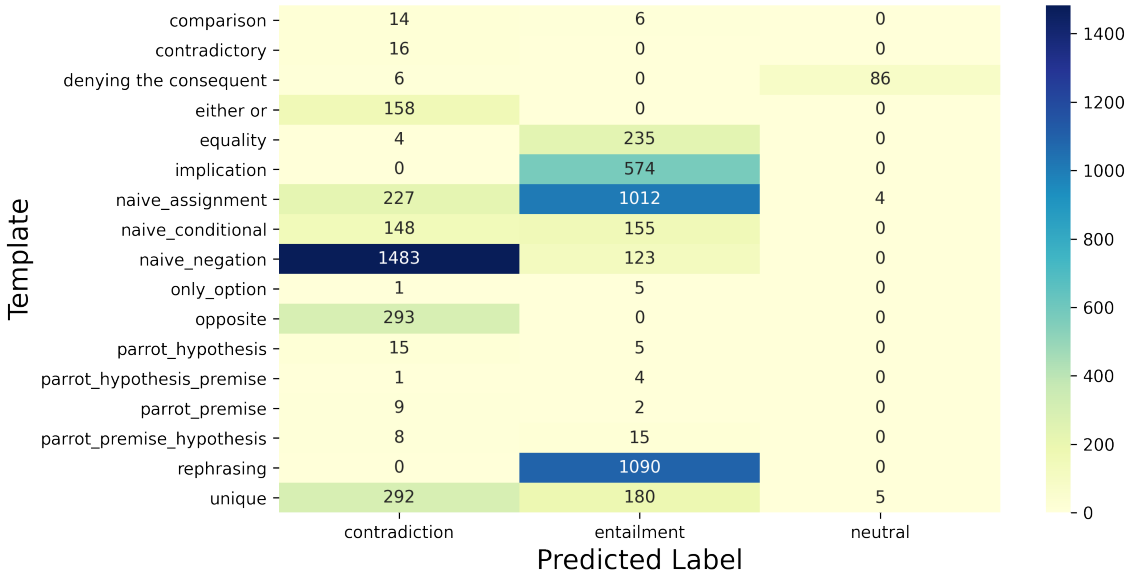


Figure 4: Template distribution by predicted label for the held-out UNMATCHED test set.

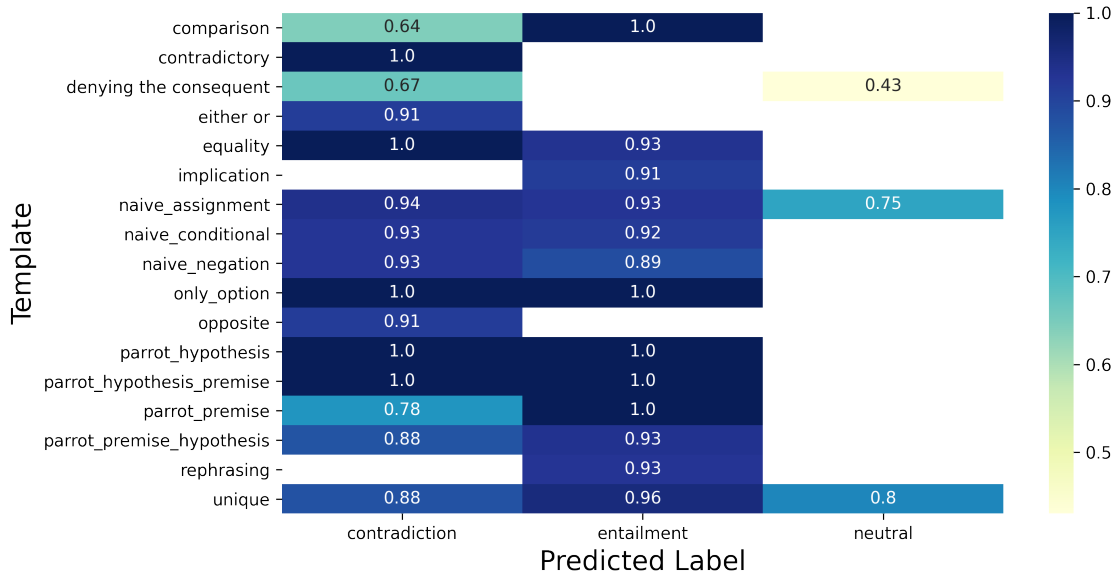


Figure 5: F1 Score by predicted label for the held-out test set.

the label. They may not directly state 'entailment', 'contradiction', or 'neutral', but they follow such a predictable pattern that our DistilBERT simulator model can easily predict the correct label when provided with T5-generated rationale. Interestingly, we note that 'parroting the premise' commonly leads to leakage. This can be explained by the fact that indirect answers alone are often sufficient to classify even without knowledge of the original question. Also, among all non-parroting templates, those that leak the least are 'comparison' and 'only option'. This is perhaps because DistilBERT, like its full BERT ancestor, struggles with comparisons and quantifiers such as 'only', 'every', etc (Talmor et al., 2020; Ettinger, 2020; Rogers et al., 2020).

E.5 Quality of Templates

Not only do we have insufficient human evaluations to get an accurate idea of the perceived faithfulness of our generated rationales, we also have evidence that our survey annotators were actually evaluating the plausibility of the rationales to the true label (see Section 5.3). Nevertheless, we can still can approximate perceived rationale quality. Figure 7 displays the average human-annotated quality per template based on our Mechanical Turk survey responses.

E.6 Quantifying Rationale Novelty

Finally, we consider whether the model can truly generate novel, free-form rationale, or rather blindly substitute in some combination of the hypothesis, premise, and/or context into the templates. We compare how often the regular expression capture groups of our templates exactly match part of the input, shown in Figure 8. The rationales are clearly novel, since the model adds or modifies text in the majority of cases.

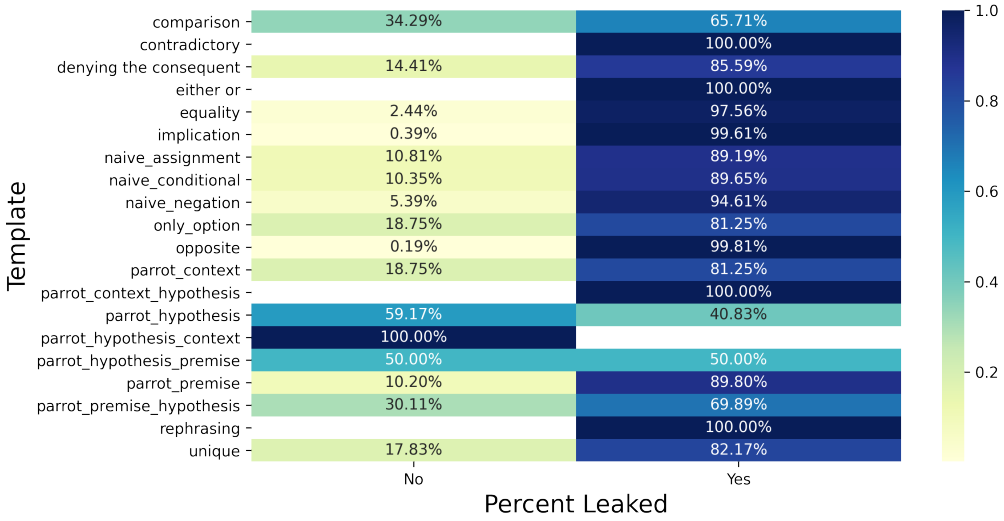


Figure 6: Percent of leaked templates across all generated rationales.

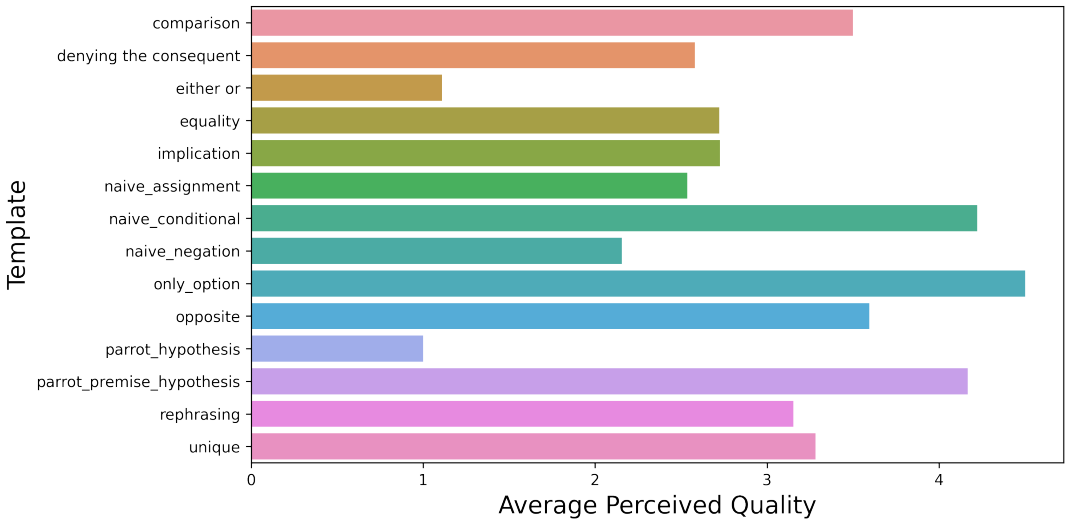


Figure 7: Average human-annotated quality per template measured via a Likert scale.

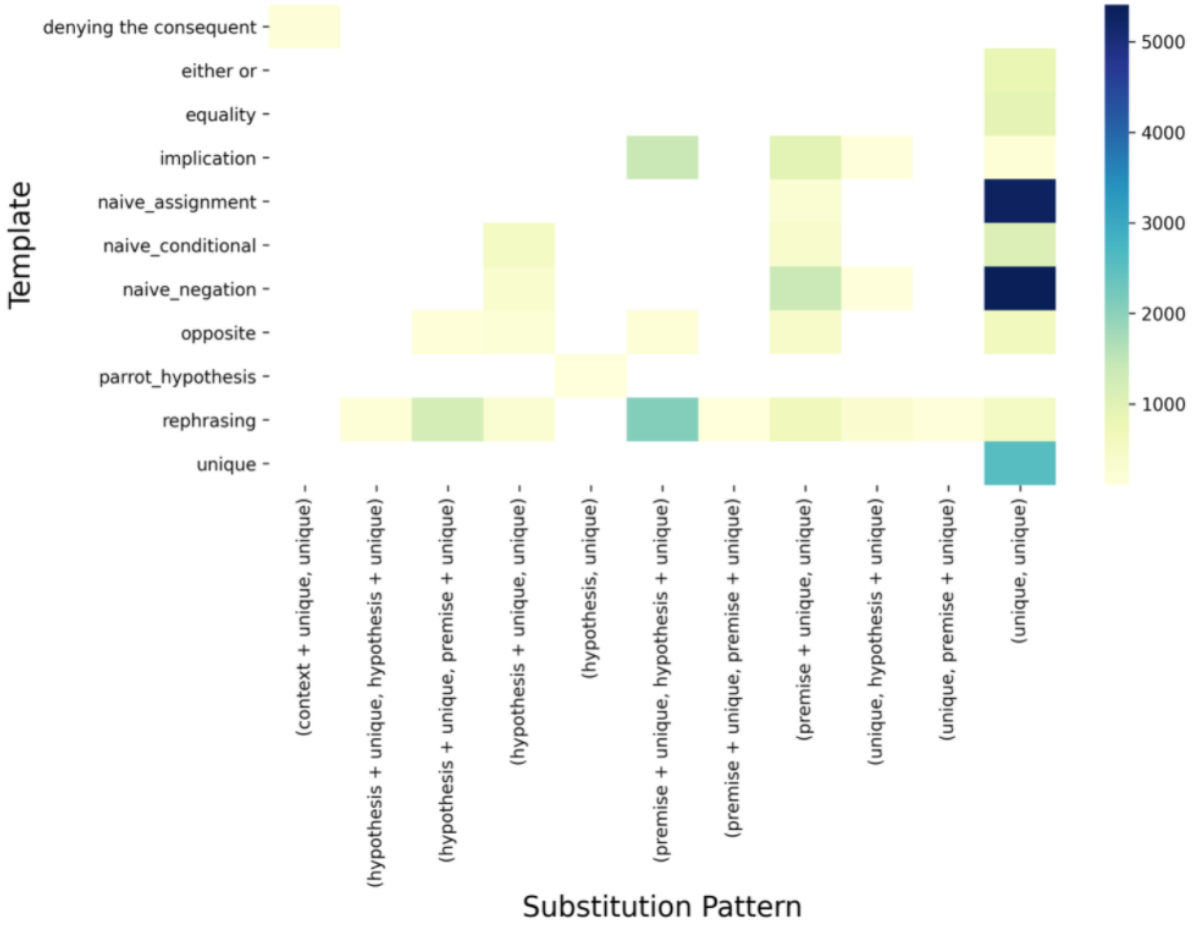


Figure 8: Distributions of different substitution schemes by templates across all generated rationales.