Faithfulness Tests for Natural Language Explanations

Atanasova, Pepa; Camburu, Oana Maria; Lioma, Christina; Lukasiewicz, Thomas; Simonsen, Jakob Grue; Augenstein, Isabelle

Published in:
Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)

DOI:
10.18653/v1/2023.acl-short.25

Publication date:
2023

Document version
Publisher's PDF, also known as Version of record

Document license:
CC BY

Citation for published version (APA):
Faithfulness Tests for Natural Language Explanations

Pepa Atanasova¹, Oana-Maria Camburu², Christina Lioma¹, Thomas Lukasiewicz³, Jakob Grue Simonsen¹, Isabelle Augenstein¹
¹Department of Computer Science, University of Copenhagen, Denmark
²University College London, UK ³TU Wien, Austria ⁴University of Oxford, UK
pepa@di.ku.dk

Abstract

Explanations of neural models aim to reveal a model’s decision-making process for its predictions. However, recent work shows that current methods giving explanations such as saliency maps or counterfactuals can be misleading, as they are prone to present reasons that are unfaithful to the model’s inner workings. This work explores the challenging question of evaluating the faithfulness of natural language explanations (NLEs). To this end, we present two tests. First, we propose a counterfactual input editor for inserting reasons that lead to counterfactual predictions but are not reflected by the NLEs. Second, we reconstruct inputs from the reasons stated in the generated NLEs and check how often they lead to the same predictions. Our tests can evaluate emerging NLE models, proving a fundamental tool in the development of faithful NLEs.

1 Introduction

Explanations of neural models aim to uncover the reasons behind model predictions in order to provide evidence on whether the model is trustworthy. To this end, explanations have to be faithful, i.e., reflect the decision-making process of the model, otherwise, they could be harmful (Hancox-Li, 2020). However, recent studies show that explanations can often be unfaithful, covering flaws and biases of the model. Adebayo et al. (2018) show that certain widely deployed explainability approaches that provide saliency maps (with importance scores for each part of the input, e.g., words or super-pixels) can even be independent of the training data or of the model parameters. Others also question the effectiveness and reliability of counterfactuals (Slack et al., 2021), concept activations, and training point ranking explanations (Adebayo et al., 2022).

In this work, we investigate the degree of faithfulness of natural language explanations (NLEs), which explain model predictions with free text. NLEs are not constrained to contain only input segments, thus they provide more expressive (Camburu et al., 2021) and usually more human-readable explanations than, e.g., saliency maps (Wiegrefe and Marasovic, 2021). Evaluating the faithfulness of explanations is very challenging in general, as the ground-truth reasons used by a model for a prediction are usually unknown. Evaluating the faithfulness of NLEs is further complicated, as they often include words not present in the input. Thus, existing tests evaluating other types of explanations, e.g., saliency maps, cannot be directly applied to NLEs. As a stepping stone towards evaluating how faithful NLEs are, we design two tests. Our first test investigates whether NLE models are faithful to reasons for counterfactual predictions. We introduce a counterfactual input editor that makes counterfactual interventions resulting in new instances on which the model prediction changes but the NLE does not reflect the intervention leading to the change. Our second test reconstructs an input from the reasons stated in a generated NLE, and checks whether the new input leads to a different prediction. We apply our tests to four NLE models over three datasets. We aim for our tests to be an important tool to assess the faithfulness of existing and upcoming NLE models.¹

2 The Faithfulness Tests

Given a dataset \( X = (x_i, e_i, y_i) \), with an input \( x_i \), a gold NLE \( e_i \), and a gold label \( y_i \in L \), where \( L \) is the set of all labels for \( X \), a model \( f \) is trained to produce an NLE and a task prediction for each input: \( f(x_i) = (\hat{e}_i, \hat{y}_i) \). We also refer to \( \hat{e}_i \) as \( f(x_i)_{ex} \) and to \( \hat{y}_i \) as \( f(x_i)_{pr} \).

2.1 The Counterfactual Test: Are NLE models faithful to reasons for counterfactual predictions? Studies in cognitive science show that humans usually seek counterfactuals by looking for

¹The code is available at https://github.com/copenlu/nle_faithfulness
Table 1: Examples of unfaithful explanations detected with our tests for the task of NLI (see §2). We apply the tests on an original instance (second column), which results in a new instance (third column). The parts of the input changed by the test are marked with $\rightarrow$, and the intervention made by the test is in blue. $\times$ marks an NLE or a prediction that does not match the expectation, thus pointing to the underlined NLE as being unfaithful.

<table>
<thead>
<tr>
<th>Test</th>
<th>Original Instance</th>
<th>Instance After Test Intervention</th>
</tr>
</thead>
</table>
| Counterfactual test (§2)    | **Premise:** Man in a black suit, white shirt and black bowtie playing an instrument with the rest of his symphony surrounding him.  
**Prediction:** neutral  
**NLE:** Not all men are tall.                                                                 | **Premise:** Man in a black suit, white shirt and black bowtie playing an instrument with the rest of his symphony surrounding him.  
$\rightarrow$**Hypothesis:** A tall person in a suit.  
$\times$**Prediction:** contradiction  
$\times$**NLE:** A man is not a tall person.  
Unfaithfulness cause: inserted word “blue” $\notin$ NLE but changed the prediction.                                                                 |
| Input reconstruction test (§2) | **Premise:** Many people standing outside of a place talking to each other in front of a building that has a sign that says “HE-POINTE”  
**Hypothesis:** The people are having a chat before going into the work building.  
**Prediction:** neutral  
**NLE:** Just because people are talking does not mean they are having a chat. | $\rightarrow$**Premise:** People are talking.  
$\rightarrow$**Hypothesis:** They are having a chat.  
$\times$**Prediction:** entailment  
**NLE:** People are talking is a rephrasing of they are having a chat.  
Unfaithfulness cause: The reasons in the NLE for the original instance lead to a different prediction. |

Factors that explain why event $A$ occurred instead of $B$ (Miller, 2019). Counterfactual explanations were proposed for ML models by making interventions either on the input (Wu et al., 2021; Ross et al., 2021) or on the representation space (Jacovi et al., 2021). An intervention $h(x_i, y_i^C) = x_i'$ is produced over an input instance $x_i$ w.r.t. a target counterfactual label $y_i^C$, $y_i^C \neq \hat{y}_i$, such that the model predicts the target label: $f(x_i') = y_i^C$. For our test, we search for interventions that insert tokens into the input such that the model gives a different prediction, and we check whether the NLE reflects these tokens. Thus, we define an intervention $h(x_i, y_i^C) = x_i'$ that, for a given counterfactual label $y_i^C$, generates a set of words $W = \{w_j\}$ that, inserted into $x_i$, produces a new instance $x_i' = \{x_i,1, \ldots, x_i,k, W, x_i,k+1, \ldots, x_i,|x_i|\}$ such that $f(x_i') = y_i^C$. While one can insert each word in $W$ at a different position in $x_i$, here we define $W$ to be a contiguous set of words, which is computationally less expensive. As $W$ is the counterfactual for the change in prediction, then at least one word from $W$ should be present in the NLE for the counterfactual prediction:

$$h(x_i, y_i^C) = x_i'$$

$$x_i' = \{x_i,1, \ldots, x_i,k, W, x_i,k+1, \ldots, x_i,|x_i|\}$$

$$f(h(x_i, y_i^C)) = f(x_i') = y_i^C \neq \hat{y}_i = f(x_i)$$

If $W \cap x_i' = \emptyset$, then $x_i'$ is unfaithful,

where the $s$ superscript indicates that the operator is used at the semantic level. Sample counterfactual interventions satisfying Eq. 1 are in Table 1. More examples are in Tables 4 and 5 in the Appendix.

To generate the input edits $W$, we propose an editor $h$ as a neural model and follow Ross et al. (2021). The authors generate input edits that change the model prediction to target predictions and refer to these edits as explanations. We note that besides the input edits, confounding factors could cause the change in prediction, e.g., the edits could make the model change its focus towards other parts of the input and not base its decision on the edit itself. In this work, we presume that it is still important for the NLEs to point to the edits, as the model changed its prediction when the edit was inserted. This aligns with the literature on counterfactual explanations, where such edits are seen as explanations (Guidotti, 2022). We also hypothesize that confounding factors are rare, especially when insertions rather than deletions are performed. We leave such investigation for future work.

During the training of $h$, we mask $n_1\%$ tokens in $x_i$, provide as an input to $h$ the label predicted by the model, i.e., $y_i^C = \hat{y}_i$, and use the masked tokens to supervise the generation of the masked text (corresponding to $W$). During inference, we provide as target labels $y_i^C \in Y, y_i^C \neq \hat{y}_i$, and we search over $n_2$ different positions to insert $n_3$ candidate tokens at each position at a time. The training objective is the cross-entropy loss for generating the inserts.

We use as a metric of unfaithfulness the percentage of the instances in the test set for which $h$ finds counterfactual interventions that satisfy Eq. 1. To compute this automatically, we use $\cap s$ at the syntactical level. As paraphrases of $W$ might appear in the NLEs, we manually verify a subset of NLEs. We leave the introduction of an automated evaluation for the semantic level for future work.

Our metric is not a complete measure of the overall faithfulness of the NLEs, as (1) we only check whether the NLEs are faithful to the reasons for counterfactual predictions, and (2) it depends on the performance of $h$. But if $h$ does not succeed in finding a significant number of counterfactual rea-
Table 2: Results for the counterfactual test. For each setup (Eq. 3), we include the results of the random baseline (Rand), the counterfactual editor (Edit), and their union (Rand+Edit). The “% Counter” column indicates the editor’s success in finding inserts that change the model’s prediction. “% Counter Unfaith” presents the percentage of instances where the inserted text was not found in the associated NLE among the instances where the prediction was changed. “% Total Unfaith” presents the percentage of instances where the prediction was changed and the inserted text was not found in the model’s prediction. 

<table>
<thead>
<tr>
<th>Model</th>
<th>% Counter</th>
<th>% Counter Unfaith</th>
<th>% Total Unfaith</th>
</tr>
</thead>
<tbody>
<tr>
<td>e-SNLI</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Re-Rand</td>
<td>38.85</td>
<td>60.39</td>
<td>23.46</td>
</tr>
<tr>
<td>MT-Re-Edit</td>
<td>56.70</td>
<td>46.12</td>
<td>26.15</td>
</tr>
<tr>
<td>MT-Re-Rand+Edit</td>
<td>64.98</td>
<td>51.29</td>
<td>34.63</td>
</tr>
<tr>
<td>ST-Re-Rand</td>
<td>37.14</td>
<td>54.26</td>
<td>20.15</td>
</tr>
<tr>
<td>ST-Re-Edit</td>
<td>49.64</td>
<td>52.74</td>
<td>26.18</td>
</tr>
<tr>
<td>ST-Re-Rand+Edit</td>
<td>61.15</td>
<td>58.27</td>
<td>35.63</td>
</tr>
<tr>
<td>MT-Ra-Rand</td>
<td>37.17</td>
<td>54.93</td>
<td>20.42</td>
</tr>
<tr>
<td>MT-Ra-Edit</td>
<td>55.04</td>
<td>41.34</td>
<td>22.75</td>
</tr>
<tr>
<td>MT-Ra-Rand+Edit</td>
<td>63.94</td>
<td>46.63</td>
<td>31.05</td>
</tr>
<tr>
<td>ST-Ra-Rand</td>
<td>35.21</td>
<td>57.82</td>
<td>20.36</td>
</tr>
<tr>
<td>ST-Ra-Edit</td>
<td>60.00</td>
<td>45.66</td>
<td>27.39</td>
</tr>
<tr>
<td>ST-Ra-Rand+Edit</td>
<td>67.31</td>
<td>55.03</td>
<td>37.04</td>
</tr>
<tr>
<td>CoS-E</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Re-Rand</td>
<td>44.89</td>
<td>83.18</td>
<td>37.34</td>
</tr>
<tr>
<td>MT-Re-Edit</td>
<td>50.00</td>
<td>77.23</td>
<td>38.62</td>
</tr>
<tr>
<td>MT-Re-Rand+Edit</td>
<td>59.89</td>
<td>85.26</td>
<td>51.06</td>
</tr>
<tr>
<td>ST-Re-Rand</td>
<td>52.34</td>
<td>79.47</td>
<td>41.60</td>
</tr>
<tr>
<td>ST-Re-Edit</td>
<td>53.83</td>
<td>86.17</td>
<td>46.38</td>
</tr>
<tr>
<td>ST-Re-Rand+Edit</td>
<td>67.45</td>
<td>87.54</td>
<td>59.04</td>
</tr>
<tr>
<td>MT-Ra-Rand</td>
<td>39.26</td>
<td>84.01</td>
<td>32.98</td>
</tr>
<tr>
<td>MT-Ra-Edit</td>
<td>50.00</td>
<td>78.72</td>
<td>39.36</td>
</tr>
<tr>
<td>MT-Ra-Rand+Edit</td>
<td>56.81</td>
<td>85.58</td>
<td>48.62</td>
</tr>
<tr>
<td>ST-Ra-Rand</td>
<td>46.70</td>
<td>75.85</td>
<td>35.43</td>
</tr>
<tr>
<td>ST-Ra-Edit</td>
<td>52.02</td>
<td>75.05</td>
<td>39.04</td>
</tr>
<tr>
<td>ST-Ra-Rand+Edit</td>
<td>63.62</td>
<td>81.77</td>
<td>52.02</td>
</tr>
<tr>
<td>ComVE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Re-Rand</td>
<td>35.60</td>
<td>83.43</td>
<td>29.70</td>
</tr>
<tr>
<td>MT-Re-Edit</td>
<td>50.90</td>
<td>70.53</td>
<td>35.90</td>
</tr>
<tr>
<td>MT-Re-Rand+Edit</td>
<td>61.10</td>
<td>78.89</td>
<td>48.20</td>
</tr>
<tr>
<td>ST-Re-Rand</td>
<td>41.90</td>
<td>74.22</td>
<td>31.10</td>
</tr>
<tr>
<td>ST-Re-Edit</td>
<td>48.40</td>
<td>76.45</td>
<td>37.00</td>
</tr>
<tr>
<td>ST-Re-Rand+Edit</td>
<td>62.90</td>
<td>77.42</td>
<td>48.70</td>
</tr>
<tr>
<td>MT-Ra-Rand</td>
<td>33.70</td>
<td>75.67</td>
<td>25.50</td>
</tr>
<tr>
<td>MT-Ra-Edit</td>
<td>47.20</td>
<td>66.53</td>
<td>31.40</td>
</tr>
<tr>
<td>MT-Ra-Rand+Edit</td>
<td>58.10</td>
<td>73.84</td>
<td>42.90</td>
</tr>
<tr>
<td>ST-Ra-Rand</td>
<td>36.30</td>
<td>80.17</td>
<td>29.10</td>
</tr>
<tr>
<td>ST-Ra-Edit</td>
<td>49.50</td>
<td>79.80</td>
<td>39.50</td>
</tr>
<tr>
<td>ST-Ra-Rand+Edit</td>
<td>61.90</td>
<td>85.98</td>
<td>51.90</td>
</tr>
</tbody>
</table>

where $R$ is the function that builds a new input $r_i$ given $x_i$ and $\hat{e}_i$. Sufficiency has been employed to evaluate saliency explanations, where the direct mapping between tokens and saliency scores allows $r_i$ to be easily constructed (by preserving only the top-N most salient tokens) (DeYoung et al., 2020; Atanasova et al., 2020a). For NLEs, which lack such direct mapping, designing an automated extraction $R$ of the reasons in $\hat{e}_i$ is challenging.

Here, we propose automated agents $Rs$ that are task-dependent. We build $Rs$ for e-SNLI (Camburu et al., 2018) and ComVE (Wang et al., 2020), due to the structure of the NLEs and the nature of these datasets. However, we could not construct an $R$ for CoS-E (Rajani et al., 2019). For e-SNLI, a large number of NLEs follow certain templates. Camburu et al. (2020) provide a list of templates covering 97.4% of the NLEs in the training set. For example, “<X> is the same as <Y>” is an NLE template for entailment. Thus, many of the generated NLEs also follow these templates. In our test, we simply use <X> and <Y> from the templates as the reconstructed pair of premise and hypothesis, respectively. We keep only those <X> and <Y> that are sentences containing at least one subject and at least one verb. If the NLE for the original input was faithful, then we expect the prediction for the reconstructed input to be the same as for the original.

Given two sentences, the ComVE task is to pick the one that contradicts common sense. If the generated NLE is faithful, replacing the correct sentence with the NLE should lead to the same prediction.

3 Experiments

Following Hase et al. (2020), we experiment with four setups for NLE models, which can be grouped
by whether the prediction and NLE generation are trained with a multi-task objective using a joint model (MT) or with single-task objectives using separate models (ST). They can also be grouped by whether they generate NLEs conditioned on the predicted label (rationalizing models (Ra)), or not conditioned on it (reasoning models (Re)). The general notation \( f(x_i) = (\hat{c}_i, \hat{y}_i) \) used in §2 includes all four setups:

\[
\begin{align*}
    \text{MT-Re:} & \quad f_{p, ex}(x_i) = (\hat{c}_i, \hat{y}_i) \\
    \text{MT-Ra:} & \quad f_{p, ex}(x_i) = (\hat{c}_i | \hat{y}_i), \\
    \text{ST-Re:} & \quad f_{ex}(x_i) = \hat{c}_i; f_p(x_i, \hat{c}_i) = \hat{y}_i, \\
    \text{ST-Ra:} & \quad f_{ex}(x_i, y_j) = \hat{c}_{i,j}; f_p(x_i, \hat{c}_i) = \hat{y}_{i,j} \\
    j = \arg \max_{j \in [1, \ldots, |L|]} & \left( f_p(x_i, \hat{c}_{i,j}) \right) 
\end{align*}
\]

where \( f_{p, ex} \) is a joint model for task prediction and NLE generation, \( f_p \) is a model only for task prediction, and \( f_{ex} \) is a model only for NLE generation. The ST-Ra setup produces one NLE \( \hat{c}_{i,j} \) for each \( y_j \in L \). Given \( \hat{c}_{i,j} \) and \( x_i \), \( f_p \) predicts the probability of the corresponding label \( y_{i,j} \) and selects as \( \hat{y}_i \) the label with the highest probability.

For both \( f \) and the editor \( h \), we employ the pre-trained T5-base model (Raffel et al., 2020). The editor uses task-specific prefixes for insertion and NLE generation. We train both \( f \) and \( h \) for 20 epochs, evaluate them on the validation set at each epoch, and select the checkpoints with the highest success rate (see §2). We use a learning rate of 1e-4 with the Adam optimizer (Kingma and Ba, 2014). For the editor, during training, we mask \( n_1 \) consecutive tokens with one mask token, where \( n_1 \) is chosen at random in \([1, 3]\). During inference, we generate candidate insertions for \( n_2 = 4 \) random positions, with \( n_3 = 4 \) candidates for each position at a time. The hyper-parameters are chosen with a grid search over the validation set.\(^3\) For the manual evaluation, an author annotated the first 100 test instances for each model (800 in total). The manual evaluation has been designed in accordance with related work (Camburu et al., 2018), which also evaluated 100 instances per model. We found that no instances were using paraphrases. Hence, in our work, the automatic metric can be trusted.

**Baseline.** For the counterfactual test, we incorporate a random baseline as a comparison. Specifically, we insert a random adjective before a noun or a random adverb before a verb. We randomly select \( n_2 = 4 \) positions where we insert the said words, and, for each position at a time, we consider \( n_3 = 4 \) random candidate words. The candidates are single words randomly chosen from the complete list of adjectives or adverbs available in WordNet (Fellbaum, 2010). We identify the nouns and verbs in the text with spaCy (Honnibal et al., 2020). We also observe that the random baseline finds words that change the prediction of the model, the words are more often not found in the corresponding NLE compared to the counterfactual editor (% Counter Unfaith). We conjecture that this is because the randomly selected words are rare for the dataset compared to the words that the editor learns to insert. Second, the counterfactual editor is better at finding words that lead to a change in the model’s prediction, which in turn results in a higher percentage of unfaithful instances in general (% Total Unfaith).

We also observe that the insertions \( W \) lead to counterfactual predictions for up to 56.70% of the instances (for MT-Re-Edit on e-SNLI). For up to 46.38% of the instances (for ST-Re-Edit on CoS-E), the editor is able to find an insertion for which the counterfactual NLE is unfaithful. Table 1, row 1, presents one such example. More examples for the random baseline can be found in Table 4, and for the counterfactual editor in Table 5. Finally, the union of the counterfactual interventions discovered by the random baseline and the editor, we observe total percentages of up to 59.04% unfaithfulness to the counterfactual.

\(^{\text{1}}\)During training, the gold label is used.

\(^{\text{2}}\)When \( n_2 \) and \( n_3 \) are increased, a higher number of insertions are generated, which in turn could result in a higher percentage of unfaithful NLEs. However, increasing these parameters also leads to higher computational demands. Future research could explore strategies for efficiently searching the space of insertion candidates.
We see that for all datasets and models, the total percentages of unfaithfulness to counterfactual are high, between 37.04\% (for MT-Ra-Rand+Edit on e-SNLI) and 59.04\% (ST-Re-Rand+Edit for CoS-E). We re-emphasize that this should not be interpreted as an overall estimate of unfaithfulness, as our test is not complete (see §2).

The Input Reconstruction Test. Table 3 shows the results of the input reconstruction test. We were able to reconstruct inputs for up to 4487 out of the 10K test instances in e-SNLI, and for all test instances in ComVE. There are, again, a substantial number of unfaithful NLEs: up to 14\% for e-SNLI, and up to 40\% for ComVE. An example is in Table 1, row 2. More examples can be found in Table 6. We also notice that this test identified considerably more unfaithful NLEs for ComVE than for e-SNLI, while for our first test, the gap was not as pronounced. This shows the utility of developing diverse faithfulness tests.

Finally, all four types of models had similar faithfulness results\(^5\) on all datasets and tests, with no consistent ranking among them. This opposes the intuition that some configurations may be more faithful than others, e.g., Camburu et al. (2018) hypothesized that ST-Re may be more faithful than MT-Re, which is the case in most but not all of the cases, e.g., on CoS-E the editorial finds more unfaithfulness for ST-Re (44.04\%) than for MT-Re (42.76\%). We also observe that Re models tend to be less faithful than Ra models in most cases.

4 Related Work

Tests for Saliency Maps. The faithfulness and, more generally, the utility of explanations were predominantly explored for saliency maps. Comprehensiveness and sufficiency (DeYoung et al., 2020) were proposed for evaluating the faithfulness of existing saliency maps. They measure the decrease in a model’s performance when only the most or the least important tokens are removed from the input. Madsen et al. (2022) propose another faithfulness metric for saliency maps, ROAR, obtained by masking allegedly important tokens and then retraining the model. In addition, Yin et al. (2022) and Hsieh et al. (2021) evaluate saliency maps through adversarial input manipulations presuming that model predictions should be more sensitive to manipulations of the more important input regions as per the saliency map. Chan et al. (2022b) provide a comparative study of faithfulness measures for saliency maps. Further faithfulness testing for saliency maps was introduced by Camburu et al. (2019). Existing studies also pointed out that saliency maps can be manipulated to hide a classifier’s biases towards dataset properties such as gender and race (Dombrowski et al., 2019; Slack et al., 2020; Anders et al., 2020). While diagnostic methods for saliency maps rely on the one-to-one correspondence between the saliency scores and the regions of the input, this correspondence is not present for NLEs, where text not in the input can be included. Thus, diagnostic methods for saliency maps are not directly applicable to NLEs. To this end, we propose diagnostic tests that can be used to evaluate NLE model faithfulness.

Tests for NLEs. Existing work often only looks at the plausibility of the NLEs (Rajani et al., 2019; Kayser et al., 2021; Marasović et al., 2022; Narang et al., 2020; Kayser et al., 2022; Yordanov et al., 2022). In addition, Sun et al. (2022) investigated whether the additional context available in human-and model-generated NLEs can benefit model prediction as they benefit human users. Differently, Hase et al. (2020) proposed to measure the utility of NLEs in terms of how well an observer can simulate a model’s output given the generated NLE. The observer could be an agent (Chen et al., 2022a) or a human (Jolly et al., 2022; Atanasova et al., 2020b). The only work we are aware of that introduces sanity tests for the faithfulness of NLEs is that of Wiegrefe et al. (2021), who suggest that an association between labels and NLEs is necessary for faithful NLEs and propose two pass/fail tests: (1) whether the predicted label and generated NLE are similarly robust to noise, (2) whether task prediction and NLE generation share the most important input tokens for each. Majumder et al. (2022) use these tests as a sanity check for the faithfulness of their model. Our tests are complementary and offer quantitative metrics.

5 Summary and Outlook

In this work, we introduced two tests to evaluate the faithfulness of NLE models. We find that all four high-level setups of NLE models are prone to generate unfaithful NLEs, reinforcing the need for proof of faithfulness. Our tests can be used to ensure the faithfulness of emerging NLE models and inspire the community to design complementary faithfulness tests.
Limitations

While our tests are an important stepping stone for evaluating the faithfulness of NLEs, they are not comprehensive. Hence, a model that would perform perfectly on our tests may still generate unfaithful NLEs.

Our first test inspects whether NLE models are faithful to reasons for counterfactual predictions. It is important to highlight that NLEs may not comprehensively capture all the underlying reasons for a model’s prediction. Thus, an NLE that fails to accurately represent the reasons for counterfactual predictions may still offer faithful explanations by reflecting other relevant factors contributing to the predictions. Additionally, both the random baseline and the counterfactual editor can generate insertions that result in text lacking semantic coherence. To address this limitation, future research can explore methods to generate insertion candidates that are both semantically coherent and reveal unfaithful NLEs.

Our second test uses heuristics that are task-dependent and may not be applicable to any task. The reconstruction functions $R$s proposed in this work are based on hand-crafted rules for the e-SNLI and ComVE datasets. However, due to the nature of the CoS-E NLEs, rule-based input reconstructions were not possible for this dataset. To address this limitation, future research could investigate automated reconstruction functions that utilize machine learning models. These models would be trained to generate reconstructed inputs based on the generated NLEs, where a small number of annotations would be provided as training instances. For example, for CoS-E, one such training annotation could be: **Original Question:** After getting drunk people couldn’t understand him, it was because of his what? **Choices:** lower standards, slurred speech, or falling down. **Answer:** slurred speech. **NLE:** People who are drunk have difficulty speaking. → **Reconstructed Question:** What do drunk people have difficulty with? **Reconstructed Choices:** lower standards, speaking, or falling down. This approach would enable the development of machine learning models capable of generating reconstructed inputs for various datasets.

Acknowledgements

The research documented in this paper has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199. Isabelle Augenstein’s research is further partially funded by a DFF Sapere Aude research leader grant under grant agreement No 0171-00034B, as well as by the Pioneer Centre for AI, DNRF grant number P1. Thomas Lukasiewicz was supported by the Alan Turing Institute under the UK EPSRC grant EP/N510129/1, the AXA Research Fund, and the EU TAILOR grant 952215. Oana-Maria Camburu was supported by a UK Leverhulme Early Career Fellowship. Christina Lioma’s research is partially funded by the Vilum and Velux Foundations Algorithms, Data and Democracy (ADD) grant.

References


Andreas Madsen, Nicholas Meade, Vaibhav Adlakha, and Siva Reddy. 2022. Evaluating the Faithfulness of Importance Measures in NLP by Recursively


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


## A More Examples of Unfaithful NLEs

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Original Instance</th>
<th>Instance After Test Intervention</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoS-E</td>
<td><strong>Question:</strong> What happens when spending money without paying someone back?</td>
<td>➤Question: What happens when increasingly spending money without paying someone back?</td>
</tr>
<tr>
<td></td>
<td><strong>Choice:</strong> poverty, debt, and bankruptcy</td>
<td><strong>Prediction:</strong> bankruptcy</td>
</tr>
<tr>
<td></td>
<td><strong>Prediction:</strong> debt</td>
<td>➤ <strong>NLE:</strong> bankruptcy is the only option that is a result of spending money.</td>
</tr>
<tr>
<td></td>
<td><strong>NLE:</strong> debt is the only option that is not something that can be paid back.</td>
<td>Unfaithfulness cause: inserted word ‘increasingly’ ‡ NLE but changed the prediction.</td>
</tr>
<tr>
<td></td>
<td><strong>Sent:</strong></td>
<td>➤ <strong>Sent:</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Premise:</strong> A man wearing glasses and a ragged costume is playing a Jaguar electric guitar and singing with the accompaniment of a drummer.</td>
<td><strong>Hypothesis:</strong> A man wearing glasses and a ragged costume is playing a Jaguar electric guitar and singing along with a semi-formal drummer.</td>
</tr>
<tr>
<td></td>
<td><strong>Hypothesis:</strong> A man with glasses and a disheveled outfit is playing a guitar and singing along with a drummer.</td>
<td><strong>Prediction:</strong> neutral</td>
</tr>
<tr>
<td></td>
<td><strong>Unfaithfulness cause:</strong> inserted word ‘ardently’ ‡ NLE but changed the prediction.</td>
<td><strong>NLE:</strong> Not all ragged costumes are disheveled.</td>
</tr>
<tr>
<td></td>
<td><strong>NLE:</strong> A ragged costume is a disheveled outfit.</td>
<td>Unfaithfulness cause: inserted word ‘semi-formal’ ‡ NLE but changed the prediction.</td>
</tr>
</tbody>
</table>

Table 4: Examples of unfaithful explanations detected with random insertion baseline. (see §2). The examples are selected for the MT-RA models for all three datasets. We apply the tests on an original instance (second column), which results in a new instance (third column). The parts of the input changed by the test are marked with ➤, and the intervention made by the test is in blue. ✗ marks an NLE or a prediction that does not match the expectation, thus pointing to the underlined NLE being unfaithful.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Original Instance</th>
<th>Instance After Test Intervention</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoS-E</td>
<td><strong>Question:</strong> Where can books be read?</td>
<td>➤Question: Where outside can books be read?</td>
</tr>
<tr>
<td></td>
<td><strong>Choice:</strong> shelf, table, and backpack</td>
<td><strong>Prediction:</strong> backpack</td>
</tr>
<tr>
<td></td>
<td><strong>NLE:</strong> books are usually stored on a table.</td>
<td><strong>NLE:</strong> books are usually stored in backpacks.</td>
</tr>
<tr>
<td></td>
<td><strong>Sent:</strong></td>
<td>➤ <strong>Sent:</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Premise:</strong> When people are hungry they drink water and do not eat food.</td>
<td><strong>Hypothesis:</strong> When people are hungry they drink water and do not eat food.</td>
</tr>
<tr>
<td></td>
<td><strong>Sent:</strong> People eat food when they are hungry.</td>
<td><strong>Prediction:</strong> first sentence</td>
</tr>
<tr>
<td></td>
<td><strong>Prediction:</strong> first sentence</td>
<td><strong>NLE:</strong> Eating food is not a good way to get rid of hunger.</td>
</tr>
<tr>
<td></td>
<td><strong>NLE:</strong> Water is not a food and cannot satisfy people’s hunger.</td>
<td>Unfaithfulness cause: inserted words ‘so many times’ ‡ NLE but changed the prediction.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Premise:</strong> Two women having drinks at the bar.</td>
</tr>
<tr>
<td></td>
<td><strong>Hypothesis:</strong> Three women are at a bar.</td>
<td><strong>Prediction:</strong> entailment</td>
</tr>
<tr>
<td></td>
<td><strong>Prediction:</strong> contradiction</td>
<td>✗ <strong>NLE:</strong> Two women are three women.</td>
</tr>
<tr>
<td></td>
<td><strong>NLE:</strong> Two women are not three women.</td>
<td>Unfaithfulness cause: inserted word ‘together’ ‡ NLE but changed the prediction.</td>
</tr>
</tbody>
</table>

Table 5: Examples of unfaithful explanations detected with counterfactual editor. (see §2). The examples are selected for the MT-RA models for all three datasets. We apply the tests on an original instance (second column), which results in a new instance (third column). The parts of the input changed by the test are marked with ➤, and the intervention made by the test is in blue. ✗ marks an NLE or a prediction that does not match the expectation, thus pointing to the underlined NLE being unfaithful.

### A.1 Model Performance
Dataset, Model | Original Instance | Instance After Test Intervention
--- | --- | ---
ComVE, ST-RE | Sent 1: Crack addicts are addicted to chocolate milk. | ➔ Sent 1: Crack addicts are addicted to chocolate milk. 
Sent 2: He visited a doctor to cure his sickness | 

Explanation: Chocolate milk contains a lot of addictive chemicals.

Predict: first sentence

X Prediction: second sentence

Explanation: Lawyers do not treat people.

Sent 1: Lawyers treat people

Sent 2: He went to a lawyer to cure his sickness

X Prediction: first sentence

Explanation: Lawyers do not treat people.

Sent 1: Monkeys have short necks.

Sent 2: My knee was scrambled and I put ointment on it.

X Prediction: first sentence

Explanation: Ointment is not used to scrape a knee.

Ointment is used to scrape a knee.

X Prediction: second sentence

Explanation: Ointment is used to scrape a knee.

Premise: People riding bicycles are not necessarily friends.

Hypothesis: They are friends.

X Prediction: neutral

Explanation: People are talking.

People are talking.

X Prediction: entailment

Explanation: They are having a chat.

People are talking is a rephrasing of they are having a chat.

Table 6: Examples of unfaithful explanations detected with the Input Reconstruction Test. (see §2). We apply the test on an original instance (second column), which results in a new instance (third column). The parts of the input changed by the test are marked with ➔, and the intervention made by the test is in blue. X marks an NLE or a prediction that does not match the expectation, thus pointing to the underlined NLE being unfaithful. The unfaithfulness cause for the instances is that the reasons in the NLE for the original instance lead to a different prediction.

<table>
<thead>
<tr>
<th>Model</th>
<th>Acc↑</th>
<th>BLEU↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNLI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Re</td>
<td>88.24</td>
<td>20.01</td>
</tr>
<tr>
<td>ST-Re</td>
<td>87.68</td>
<td>19.67</td>
</tr>
<tr>
<td>MT-Ra</td>
<td>88.10</td>
<td>20.67</td>
</tr>
<tr>
<td>ST-Ra</td>
<td>87.63</td>
<td>20.59</td>
</tr>
<tr>
<td>CoS-E</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Re</td>
<td>65.79</td>
<td>5.75</td>
</tr>
<tr>
<td>ST-Re</td>
<td>66.11</td>
<td>6.66</td>
</tr>
<tr>
<td>MT-Ra</td>
<td>66.95</td>
<td>5.55</td>
</tr>
<tr>
<td>ST-Ra</td>
<td>67.79</td>
<td>7.85</td>
</tr>
<tr>
<td>ComVE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Re</td>
<td>85.70</td>
<td>7.53</td>
</tr>
<tr>
<td>ST-Re</td>
<td>84.40</td>
<td>6.68</td>
</tr>
<tr>
<td>MT-Ra</td>
<td>86.40</td>
<td>7.03</td>
</tr>
<tr>
<td>ST-Ra</td>
<td>86.40</td>
<td>7.21</td>
</tr>
</tbody>
</table>

Table 7: Performance of the models described in Eq 3. Acc denotes the prediction performance of the model on the corresponding task. BLEU denotes the BLEU score of the generated explanation compared to the gold human ones.
ACL 2023 Responsible NLP Checklist

A For every submission:

A1. Did you describe the limitations of your work?
   Limitations Section

A2. Did you discuss any potential risks of your work?
   Limitations Section (the risk of our tests being seen as comprehensive has been addressed)

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

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

B □ Did you use or create scientific artifacts?
   Not applicable. We only used existing datasets that are not specifically created for artifacts

B1. Did you cite the creators of artifacts you used?
   No response.

B2. Did you discuss the license or terms for use and/or distribution of any artifacts?
   No response.

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)?
   No response.

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?
   No response.

B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
   No response.

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.
   No response.

C □ Did you run computational experiments?

3. Experiments

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

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?

3. Experiments

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?

just a single run, sections 2 and 3

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.)?

No response.

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

Sections 2 and 3

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.?

Sections 2 and 3

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)?

The annotation was done by an author

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. The annotation was done by an author

D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?

Not applicable. We did not collect data, the annotations were done for evaluation of our methods only

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.