Evaluating Explanations in Natural Language Inference: A Model Comparison

Lorenzo Rota **Filippos Andreadis** **Leonidas Zotos** **Amber Chen**

{l.d.f.rota, f.andreadis, l.zotos, a.chen.1}@student.rug.nl
University of Groningen
Groningen, The Netherlands

Abstract

Language models are able to perform impressively on the task of Natural Language Inference (NLI). However, these systems are still treated as black boxes, while there is still little to no interpretability regarding their internal state and how they produce their outputs. In this study, we leverage human-provided explanations for NLI predictions in order to train a language model on the task of predicting artificial explanations in free textual form. We gather insights about the quality of these explanations in an effort to observe where the model fails and whether there is something to be done to improve its performance. We measure the quality of outputs in a quantitative way using a lexical and a neural metric, as well as in a qualitative way in the form of manual reviewing. Our findings suggest that adding the prediction of the inference label as an auxiliary task has a negative effect on the performance. Additionally, we remain inconclusive about whether filtering out examples that lead to generic predictions can have a beneficial outcome.

1 Introduction

Natural Language Inference is a Natural Language Processing (NLP) task concerned with ascertaining whether a hypothesis (e.g. “The barking dog woke up the cat") given a premise is:

(a) “The feline woke up" –> **true** (entailment)

(b) “The feline did not wake up" –> **false** (contradiction)

(c) “The feline is brown" –> **neutral** (undetermined)

Language models have been successfully used in the classification of the labels corresponding to the hypotheses and premises (Tay et al., 2017; Liu et al., 2018), often using the Stanford Natural Language Inference (SNLI) dataset (Bowman et al., 2015) as a benchmark.

Camburu et al. (2018); Gururangan et al. (2018) argue that good model performance in NLI labeling tasks may be caused by annotation artifacts, thus generating a model without any deeper understanding of the data. Camburu et al. therefore created the e-SNLI dataset by extending the SNLI with human-annotated explanations for the label given the hypothesis and premise.

The experiments by Camburu et al. included a bidirectional-LSTM model that predicts a label, then generates an explanation based on the label, and one that generates an explanation based on the premise and hypothesis, and then predicts a label.

Our approach includes using the flan-T5 model (Raffel et al., 2019) fine-tuned on the e-SNLI dataset, creating a model that (1) generates an explanation given hypothesis, premise, and label, and (2) generates a label and explanation simultaneously given hypothesis and premise.

As Camburu et al. mentioned, roughly 11% of all collected explanations could be matched using a set of templates. The formats of these explanations made it such that they did not actually convey any (new) information, e.g. “Sentence 1 states <premise>. Sentence 2 is stating <hypothesis>". These explanations were re-annotated to improve the quality of the data.

To check for any remaining uninformative explanations, we reproduce the method by Camburu et al. and filter examples by edit distance from the given templates. Using this method, we find about 3% of the data containing uninformative explanations and remove these from the data, producing a custom dataset. Both models (explanation and label-explanation) are trained on the custom data as well as the original e-SNLI dataset.

We quantitatively compare the results of these four models, using ROUGE (Lin, 2004) and BARTScore (Yuan et al., 2021), evaluating (1) the ability of the models to simultaneously generate a label and an explanation, and (2) whether filter-

ing data (thus increasing data quality) using template matching is useful for improving model performance.

Aside from evaluation based on ROUGE and BARTScore, we perform a qualitative data analysis using the Likert scale on the helpfulness of the explanation (to understand the label given the premise and hypothesis) and a binary score for the correctness of the explanation (i.e. whether the explanation matches the label, and whether the label logically follows given the explanation). The qualitative analysis helps clarify the correlation between a high BART/ROUGE score and explanations that match the label, make sense structurally/grammatically, and are helpful in understanding the label given the hypothesis and premise.

The results show that the models that only generate an explanation outperform the label-explanation models, whereas removing data using edit distance does not have an effect on performance. This was confirmed by the qualitative evaluation, where the explanation model trained on the original data received a significantly higher score compared to both label-explanation models, but not compared to the explanation model trained on custom data. Finally, a soft BARTScore threshold was found for the correctness of explanations, with answers with a BARTScore exceeding 0.15 being very unlikely to be incorrect.

All code can be found on the GitHub repository. The models are also accessible through Huggingface.

1.1 Related Work

Liu et al. (2018) illustrated a succesfull attempt at creating a classification-explanation architecture using an Encoder-Predictor model. The model generated explanations based on the predicted labels and seemed to outperform other neural-based baselines (given their own metric called the "Explanation Factor"). Since we are using the flan-T5 model in our own experiments, a text-to-text transfer transformer model, we can compare the performance of this type of model to the Encoder-Predictor model (albeit indirectly, since we are using different metrics).

Camburu et al. (2018) performed two NLI classification-explanation experiments using a biLSTM model: "PredictAndExplain" and "ExplainThenPredict".

In the "PredictAndExplain" experiment, the architecture generates both the label and explanation concurrently. It was found that the accuracy is equivalent to the accuracy of their architecture that only generated the label. Therefore, Camburu et al. show that generating labels and explanations simultaneously does not have a detrimental influence on the label generation accuracy. The influence of label generation on the quality of the explanations however remains unclear. This will therefore be the focus of our evaluations.

In the "ExplainThenPredict" experiment Camburu et al. created an architecture that first generates an explanation and then produces the label based on the explanation. In this case, the label accuracy decreased by 2% (compared to only generating the label), but there is "better trust that when [the model] predicts a correct label, it does so for the right reasons". While we are not reproducing this experiment, it might be an interesting future direction.

Geiger et al. (2020) evaluated neural models' ability to represent lexical relations and the ability to accurately model downward monotonicity: the notion that negation reverses entailment relations (e.g. dance entails move, but not move entails not dance). They fine-tuned four different models on the MoNLI dataset (Geiger et al., 2020), including BERT (Devlin et al., 2018). The results were initially lacking, but when pre-trained on the SNLI dataset, they found it was able to represent the lexical relations well. We are using the flan-T5 model, which was pre-trained on the e-SNLI dataset as well as further fine-tuned using the original e-SNLI and our custom cleaned version of e-SNLI. Observing the results by Geiger et al. (2020), we expect that our model will also be able to not only reproduce the model vocabulary but also accurately represent the lexical relations occurring in the data.

2 Data

The data used in this paper is the e-SNLI dataset (Camburu et al., 2018). Table 1 shows the size along with the class distribution of each one of the training, validation and test splits.

Each example consists of a premise, a hypothesis, a class label, and three different explanation strings with slight variations.

Table 1: Size and class distribution of each split of the data

Dataset	Count	Neutral	Contra-diction	Entail-ment
training	549,367	182,764	183,187	183,416
validation	9,842	3,235	3,329	3,278
test	9,824	3,219	3,368	3,237

3 Method

In this section, we describe the approaches that will be used to create NLI models, which can either generate explanations to inference triplets (premise, hypothesis and label), or jointly predict the label and generate a corresponding explanation. This is followed by an overview of the experimental design that is used to test whether removing uninformative explanations from the training data yields better results.

3.1 Overview of NLI Models

The NLI models described earlier can be separated into different tasks – out of completeness, we consider an additional task and define the three of them as follows:

0. Predicting the inference label from the premise and hypothesis.
1. Generating an explanation from the premise, hypothesis and inference label.
2. Jointly predicting the inference label and generating an explanation from the premise and hypothesis.

For each of the three tasks, a unique model will be trained to solve the specified problem, however we are only evaluating the model performance of task 1 and 2. Since they can be considered as text-to-text generation problems, a sensible choice is to fine-tune the flan-T5-base model according to the following input prompts:

0. `premise: [PREMISE]. hypothesis: [HYPOTHESIS].`
1. `premise: [PREMISE]. hypothesis: [HYPOTHESIS]. label: [LABEL]`
2. `premise: [PREMISE]. hypothesis: [HYPOTHESIS].`

where a newline is replaced by a whitespace character, and the inputs correspond to the following output prompts:

0. `[LABEL]`
1. `[EXPLANATION]`
2. `[LABEL]: [EXPLANATION]`

The models will be trained on both the full e-SNLI dataset, and a custom dataset that is prepared by filtering out uninformative explanations from the e-SNLI dataset, as described by Camburu et al. (2018). In both cases, the input and output strings are created by substituting the premises, hypotheses, labels and explanations into the corresponding prompts.

In the rest of the report, we interchangeably refer to tasks 0, 1 and 2 by the models named label, explanation and label-explanation respectively.

3.2 Preparing the Dataset

As described earlier, the relevant features from the e-SNLI dataset training split are extracted and composed according to the specific input and output prompts before fine-tuning. This is automatically taken care of as long as the dataset consists of the fields: `premise`, `hypothesis`, `label` and `explanation_1`, where the label must be represented numerically (0: entailment, 1: neutral, or 2: contradiction).

3.3 Creating the Custom Dataset

Following the method by Camburu et al., we create a custom dataset using edit distance template matching. Both models (explanation and label-explanation) are fine-tuned on this custom data as an additional experiment.

We append the suffix '-custom' to refer to the models that were trained on the custom dataset.

We follow the templates listed in Tables 8, 9, 10, 11 in the appendix, where we substitute in the hypothesis and premise for each example, after which it is compared to the golden explanation. The comparison is done by taking the edit distance, also known as the Levenshtein distance, where an example is considered a match if the distance is below a certain cutoff value. The matched examples are then removed from the dataset.

In Figure 1 we can see that explanations are matched at a linear rate with respect to the cut-off value. By qualitative inspection, we found

that the examples matched at cutoff value 13 were mostly inline with their respective templates, but any higher would lead to many false positives.

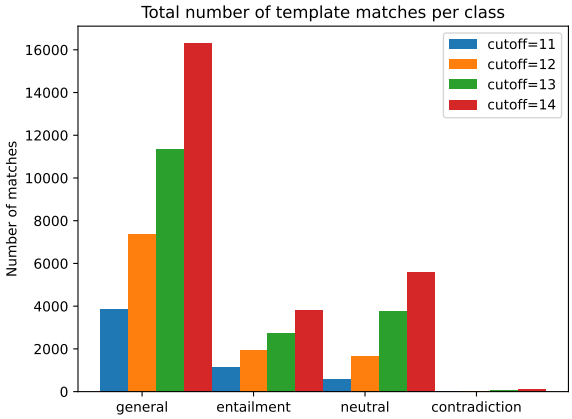

Figure 1: The total number of template matches over the training split, with varying cutoff values. The larger the value, the less precise the comparisons are.

After removing the uninformative examples from the dataset, which accounts for 3% of the original dataset, the models are fine-tuned and evaluated using the same methods as described in Section 3.1.

For training the models, we prepare the labeled data by composing the input and output strings according to the prompts for the respective task. The base model is then downloaded from the Huggingface repository and fine-tuned using the Huggingface Trainer class, which loads the latest checkpoint and updates the model parameters according to training arguments, labeled data and specified loss function. Since flan-T5-base is a language model, it uses the standard cross-entropy loss to adjust the model parameters. We additionally specify that the validation loss is computed alongside the three ROUGE metrics to visualize how well the fine-tuned models perform with respect to unseen data during training. These metrics can be viewed under the training metrics section of our Huggingface model repositories. In all cases, we observe that the ROUGE scores increase across the training epochs and reach a stable plateau.

3.4 Quantitative Assessment Method

The performance of each model is evaluated using the text-based metric ROUGE (Lin, 2004) and the neural-based BARTScore (Yuan et al., 2021).

We use three different scores from the available ROUGE metrics: ROUGE_1, ROUGE_2, and ROUGE_L. ROUGE_1 and ROUGE_2 measure the overlap between the generated text and the reference text based on unigrams and bigrams respectively. ROUGE_L measures the similarity between the generated and reference text based on the longest common subsequence.

The added benefit to this last metric is that longer sequences inherently contain more structural information of a text. A longer common subsequence therefore also points to a greater degree of common sentence structure. Since a useful generated explanation should not just be a reproduction of the model vocabulary, but also convey correct structure and grammar, we focus our evaluation on the ROUGE_L metric.

Since each example contains three (different) reference explanations, three scores are computed for each metric. Of those three, the highest is used as the score for that example.

In contrast to ROUGE, BARTScore considers text evaluation as a generation task. Specifically, given a source sentence A and a reference sentence B, BARTScore attempts to paraphrase sentence A and produces the log-likelihood (measure of probability) of the paraphrasing being sentence B. Especially helpful for the task at hand is BARTScore's ability to generate a score given multiple reference sentences, creating one score for each example in the test set.

3.5 Qualitative Assessment Method

As was described in the previous subsection, ROUGE and BARTScore were used to quantitatively evaluate the models on the full test set. However, with these scores it is difficult to actually understand whether the generated explanations are correct (e.g. a BARTScore of 0.1 does not have a clear correlation in terms of explanation correctness).

To fill this gap, two sets of 40 randomly[1] sampled examples each were manually annotated using a Likert scale. For each generated explanation, the annotators gave a score from one to five (ranging from "Not at all helpful and incorrect" to "Entirely helpful and correct"), without being aware of which explanation was generated by which model. Additionally, they marked whether the explanation was intuitively correct (i.e. whether the explanation directly explains why the hypothesis is an entailment,

[1]To ensure we have a variety of bad/good explanations, samples were uniformly sampled from each quartile based on the neural score. In this way, we decrease the chance of only sampling bad/good explanations

a contradiction or a neutral statement). Each set was annotated by two members of the team. In case the annotators gave different scores, the average of the two was taken. If this average was not in the Likert scale (e.g. 2.5), the other two annotators decided whether it should be rounded up or down. We chose to create two sets in order to have a large number of samples (80 samples in total), but also to have at least two evaluations per sample.

4 Results

In this section we describe our newly made dataset and provide a quantitative evaluation of the four models trained, in the form of the ROUGE and BARTScore. Additionally, we make a qualitative assessment of the quality of the generated explanations by manually reviewing and grading a subset of the outputs ourselves. Our focus is on comparing the models among each other and inspecting whether the explanations given are in line with the corresponding inference labels, in order to gather insights about our proposed methodologies and pick a candidate model.

In Section C of the Appendix, we provide full tables for the performance of each model that include the average and standard deviation of the two metrics, per target-prediction pair, for both the standard and the cleaned dataset.

Finally, Section A of the Appendix includes confusion matrices of the entailment classification of the explanation-label model for both datasets.

4.1 Quantitative Assessment

Table 2 shows the mean ROUGE and BARTScore for each one of the models. Preliminarily examining the data, we observe that the model that outputs only explanations clearly outperforms the jointly predicting model with a 10-point difference in ROUGE_L score and a 2-point difference in BARTScore.

In Section C of the Appendix, we provide full tables for the performance of both models that include the average and standard deviation of the metrics.

Finally, Section A of the Appendix includes confusion matrices of the entailment classification of the explanation-label model.

4.2 Qualitative Assessment

As was mentioned in the Methods section, a qualitative analysis was setup to evaluate the models' performance. In this section, these findings are presented, and will be later discussed in the Discussion section of this report. The four models that are compared are: explanation-original, label-explanation-original, explanation-custom and label-explanation-custom ("custom" indicates that the model was trained on the dataset where the templates are filtered out).

Table 2: Average quantitative metrics for each model

Model	Mean ROUGE_L	Mean BARTScore
explanation	**0.59**	**0.11**
label-explanation	0.49	0.09
explanation-custom	**0.59**	**0.11**
label-explanation-custom	0.49	0.09

A set of Wilcoxon T-tests was conducted to identify which pairs of models had significantly different qualitative assessment. The "explanation-original" model (M = 3.89, SD = 1.16) received significantly higher qualitative assessment compared to the "label-explanation-original" model (M = 3.63, SD = 1.43) (t = 95.5, p = 0.0384) and the "label-explanation-custom" model (M = 3.59, SD = 1.47) (t = 96.5, p = 0.0129). The difference between the qualitative assessment of all other models was not significantly different.

4.3 Further Insights

The following analyses were performed to gain a deeper understanding of the data and the metrics used in our experiments.

4.3.1 Template Matching

Since the removal of uninformative examples did not seem to have an effect on the model's performance to explain and/or classify examples, we examine the data more closely.

To gain insight into how beneficial the edit distance method is for matching uninformative explanations based on the templates shown in Appendix D, we generate a distribution of matches for both the original and custom models. The goal is to understand whether the two types of models follow the same distribution of template matches over the training data, and whether filtering out such explanations lowers the number of matches in the custom models. Across all four types of distributions, we note that the model predictions consistently follow a similar distribution, albeit different

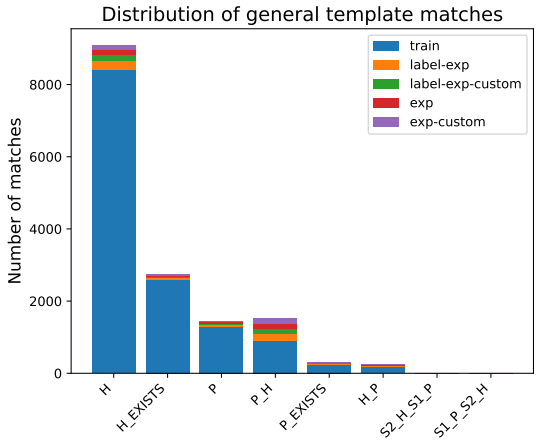

Figure 2: Distribution of general template matches.

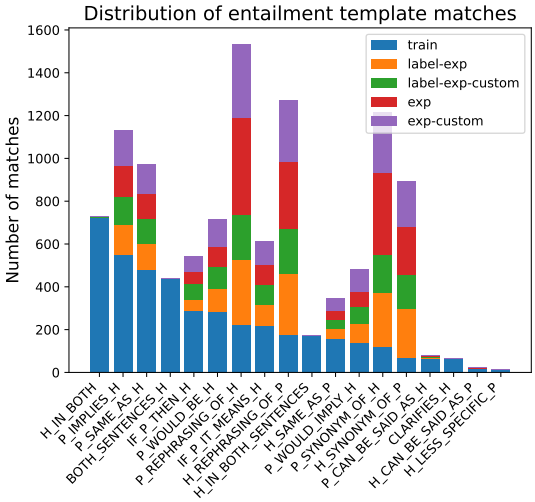

Figure 3: Distribution of entailment template matches.

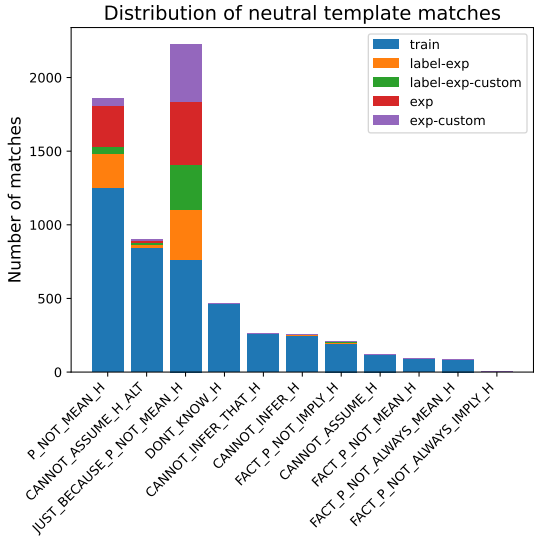

Figure 4: Distribution of neutral template matches.

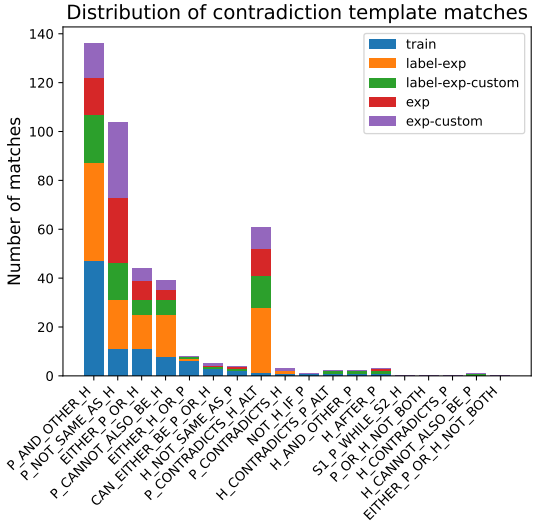

Figure 5: Distribution of contradiction template matches.

from that of the training explanations. We also note that the custom models consistently have slightly fewer matches than their original counterparts, but that this difference is very small. This indicates that the filtered dataset ultimately still contains too many uninformative explanations, and that the current edit distance method is unable to capture them well. It also means that the method may mismatch a lot of useful explanations.

4.3.2 BARTScore Correctness Threshold

Since we now have a qualitative evaluation of which produced explanations are correct, as well as the BARTScore of these evaluations, it is interesting to attempt and find the BARTScore threshold above which an explanation is likely to be correct. To this end, a boxplot is created comparing the BARTScore of correct ($M = 0.127$, $SD = 0.0856$) and incorrect ($M = 0.0720$, $SD = 0.0497$) explanations, as evaluated by the annotators (Figure 6).

It is clear from Figure 6 that the BARTScore for incorrect solutions has a significantly smaller spread. Additionally, it is also clear that there is no clear threshold below which explanations are likely to be incorrect, since correct explanation can have lower scores compared to incorrect explanations. However, as can be seen from the figure, it is very unlikely for an answer to be incorrect if its BARTScore exceeds 0.15.

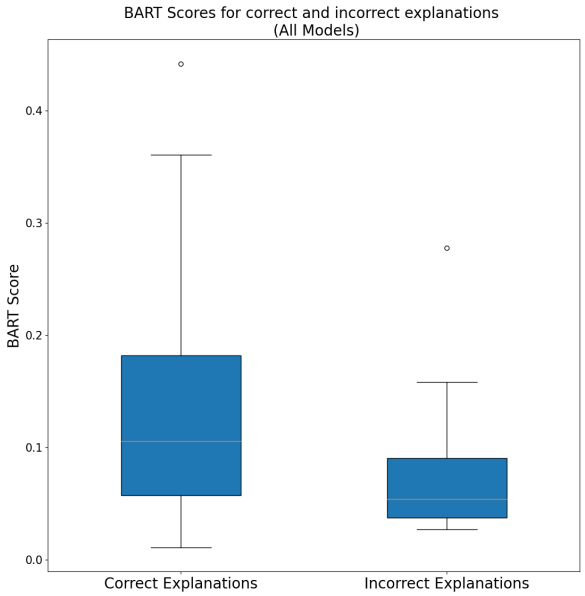

Figure 6: BARTScore of correct and incorrect explanations, based on the qualitative evaluation

5 Discussion

In this study we fine-tuned a pre-trained large language model on the e-SNLI dataset in order to evaluate its performance on generating free-form explanations for the given natural language inference prediction. We additionally assess the model's performance in outputting an inference label along with the explanation. Lastly, we used template matching to detect uninformative explanations and filter out the corresponding instances from the dataset in an effort to make the model generate answers of higher quality.

Based on both the lexical and the neural metrics we can safely conclude that when adding the task of predicting the inference label to the model, there is a noticeable drop in the quality of the generated explanation. Typically, providing models with auxiliary tasks during training (i.e., multi-task learning) has proven to be beneficial. However, the case of moving the inference label from the input to the target output means that we are also denying the model useful information needed for creating a better explanation.

It is also quite easy to observe that cleaning the dataset has a marginal effect on the scores, meaning that it does not make any significant impact on the quality of the explanations. Moreover, the confusion matrices of the entailment classification also show that the change of the results between the two datasets is minor. This could be due to the fact that only 3% of the examples are filtered out the data. Hence, a more effective template detection method is clearly an important piece of future work.

While qualitatively assessing the generated explanations, we encountered the following example that nicely demonstrates each model's functioning.

- **Premise**: Man jumping over a rusty fence on a blue bicycle.
- **Hypothesis**: Man doing a trick riding his bicycle.
- **Correct Label**: Entailment
- **Explanation-original prediction**: A man jumping over a fence on a blue bicycle is doing a trick riding his bicycle.
- **Label-Explanation-original prediction**: neutral: Jumping over a fence does not imply doing a trick.
- **Explanation-custom prediction**: Jumping over a fence is a trick.
- **Label-Explanation-custom prediction**: neutral: Jumping over a fence does not imply doing a trick.

This example demonstrates that the models that also generate a label force their explanation to be on the same line as the label. For example, the "Label-Explanation-original" prediction contains the right content, but since it generated a "neutral" label, it used "does not imply" as this format is commonly used in neutral explanations. Additionally, comparing the two "Explanation" models, we can see that the one trained on the custom model did not copy the entire premise and hypothesis, in contrast to the model that was trained on the original dataset. This indicates that the filtering can be useful, although it has to be mentioned that in some cases the model trained on the filtered dataset still produced explanations with an exact repetition of the premise & hypothesis.

Lastly, it was also found that, based on the qualitative assessment, it is possible to determine a neural score threshold above which explanations are unlikely to be incorrect. However, we also found out that it is challenging to determine a threshold below which answers are likely to be incorrect. This demonstrates the difficulty of using numerical metrics to evaluate the correctness of generated free-form explanations.

In conclusion, from the presented results and discussion, we can see that for the generation of the best explanations, the model that was trained on the full dataset performs best. When it comes to also generating labels, the use of filtering did not have a considerable impact and therefore the two models (models trained on the full or filtered dataset) performed equally well.

5.1 Potential Problems

The work presented in this report might have been affected by a number of potential problems. First and foremost, as mentioned previously, only 3% of the data was removed due to the explanations following uninformative templates. This is in contrast to the 11% of data removed in the original publication (Camburu et al., 2018), even though we attempted to follow the same methodology. It is not clear how our criteria for data removal were stricter. This difference might have affected the potential benefits from removing these samples.

Another potential problem of the current research is the lack of clear annotation guidelines. We annotated the findings using a Likert Scale ranging from one, “Not at all helpful and incorrect", to five “Entirely helpful and correct". However, it was up to the annotator’s discretion to decide how to interpret “helpful". Additionally, this scale did not consider grammatical correctness and fluency, and therefore annotators penalised differently for the lack of either (e.g. a grammatically incorrect explanation might still be a correct explanation). However, since the statistical analysis was done considering paired samples, the bias of the individual annotators is reduced. A potential workaround for similar work in the future is to create a more fine-grained qualitative analysis in terms of the questions asked (e.g. asking the annotator to evaluate the grammatical correctness separately from the correctnss of the explanation), as well as creating clearer and stricter annotation guidelines.

5.2 Future Work

For future work, it is worthwhile evaluating the models using a feature attribution tool to better understand how each model comes to the generated predictions. With such a comparison, the difference between using the full or filtered dataset might become apparent: The model trained on the filtered dataset might pay more attention to the content words of the input, instead of the template words.

Another future work is the inclusion of visual input in the training set, visualising the premise (e.g. image of “Man jumping over a rusty fence on a blue bicycle"). With such a model, it will be interesting to see if it is able to generate explanations that are more plausible in the visualised world, or perhaps using elements that are present in the picture but not the premise/hypothesis.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. The snli corpus.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Atticus Geiger, Kyle Richardson, and Christopher Potts. 2020. Neural natural language inference models partially embed theories of lexical entailment and negation. *arXiv preprint arXiv:2004.14623*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xiaodong Liu, Kevin Duh, and Jianfeng Gao. 2018. Stochastic answer networks for natural language inference. *arXiv preprint arXiv:1804.07888*.

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

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. Compare, compress and propagate: Enhancing neural architectures with alignment factorization for natural language inference. *arXiv preprint arXiv:1801.00102*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 27263–27277. Curran Associates, Inc.

A Label classification results (label-explanation model)

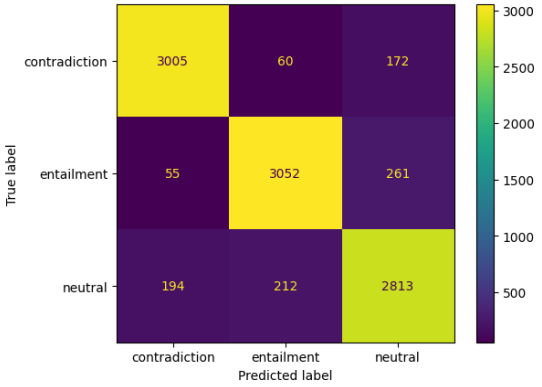

Figure 7: Confusion matrix of the label classification by the label-explanation model trained on the original dataset

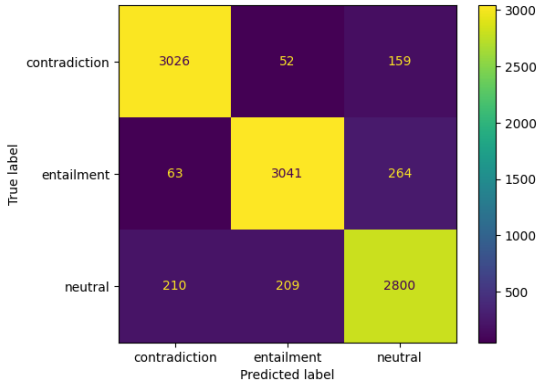

Figure 8: Confusion matrix of the label classification by the label-explanation model trained on the cleaned dataset

Table 3: Accuracy and F1-score of the label classification by the label-explanation model, for the original and the custom dataset

	Original		**Custom**	
Label	**Accuracy**	**F1-score**	**Accuracy**	**F1-score**
Contradiction	0.93	0.32	0.93	0.32
Entailment	0.91	0.32	0.90	0.32
Neutral	0.87	0.31	0.87	0.31

B Results per label (explanation model)

Table 4: Text-based (ROUGE_1, ROUGE_2, ROUGE_L) and neural-based (BARTScore) mean with standard deviation per label for the explanation model, trained on the standard dataset

Label	**Mean ROUGE_1**	**Std ROUGE_1**	**Mean ROUGE_2**	**Std ROUGE_2**	**Mean ROUGE_L**	**Std ROUGE_L**	**Mean BARTScore**	**Std BARTScore**
Contradiction	0.66	0.17	0.43	0.23	0.58	0.18	0.14	0.11
Entailment	0.67	0.17	0.49	0.22	0.61	0.18	0.12	0.1
Neutral	0.63	0.18	0.45	0.24	0.59	0.2	0.16	0.11

Table 5: Text-based (ROUGE_1, ROUGE_2, ROUGE_L) and neural-based (BARTScore) mean with standard deviation per label for the explanation model, trained on the custom dataset. Improvements compared to the model trained on the original dataset are displayed in boldface.

Label	**Mean ROUGE_1**	**Std ROUGE_1**	**Mean ROUGE_2**	**Std ROUGE_2**	**Mean ROUGE_L**	**Std ROUGE_L**	**Mean BARTScore**	**Std BARTScore**
Contradiction	0.66	0.17	**0.44**	**0.22**	0.58	0.18	0.14	0.11
Entailment	0.67	**0.16**	0.49	0.22	0.61	0.18	0.12	0.1
Neutral	0.63	0.18	0.44	**0.23**	0.59	0.2	0.15	0.11

C Results per target-prediction pair (label-explanation model)

Table 6: Text-based (ROUGE_1, ROUGE_2, ROUGE_L) and neural-based (BARTScore) mean with standard deviation per target-prediction pair for the label-explanation model, trained on the standard dataset

Target	Prediction	Mean ROUGE_1	Std ROUGE_1	Mean ROUGE_2	Std ROUGE_2	Mean ROUGE_L	Std ROUGE_L	Mean BARTScore	Std BARTScore
Contradiction	Contradiction	0.63	0.17	0.42	0.22	0.56	0.18	0.13	0.09
Contradiction	Entailment	0.56	0.15	0.33	0.17	0.48	0.17	0.08	0.06
Contradiction	Neutral	0.51	0.13	0.29	0.15	0.44	0.14	0.07	0.04
Entailment	Entailment	0.64	0.16	0.47	0.22	0.58	0.17	0.11	0.09
Entailment	Contradiction	0.51	0.15	0.27	0.18	0.43	0.16	0.05	0.04
Entailment	Neutral	0.53	0.15	0.34	0.17	0.47	0.15	0.06	0.05
Neutral	Neutral	0.61	0.17	0.43	0.23	0.57	0.19	0.14	0.1
Neutral	Entailment	0.49	0.15	0.3	0.18	0.44	0.16	0.08	0.06
Neutral	Contradiction	0.49	0.14	0.28	0.15	0.43	0.13	0.07	0.05

Table 7: Text-based (ROUGE_1, ROUGE_2, ROUGE_L) and neural-based (BARTScore) mean with standard deviation per target-prediction pair for the label-explanation model, trained on the custom dataset. Improvements compared to the model trained on the original dataset are displayed in boldface.

Target	Prediction	Mean ROUGE_1	Std ROUGE_1	Mean ROUGE_2	Std ROUGE_2	Mean ROUGE_L	Std ROUGE_L	Mean BARTScore	Std BARTScore
Contradiction	Contradiction	0.63	0.17	0.42	0.22	0.56	0.18	0.13	0.09
Contradiction	Entailment	0.54	0.17	0.3	0.18	0.47	0.18	0.08	0.06
Contradiction	Neutral	0.51	0.13	**0.31**	0.15	0.45	0.14	0.07	0.05
Entailment	Entailment	**0.65**	0.16	0.47	0.22	0.59	0.18	0.11	0.09
Entailment	Contradiction	**0.53**	0.15	0.27	**0.17**	**0.44**	0.16	0.05	0.04
Entailment	Neutral	0.52	0.15	0.33	0.17	0.46	0.15	0.06	**0.04**
Neutral	Neutral	0.62	0.17	0.43	0.23	**0.58**	0.19	0.14	0.1
Neutral	Entailment	0.49	0.16	0.3	0.18	0.44	0.16	0.08	**0.05**
Neutral	Contradiction	0.49	0.14	0.27	0.15	0.42	0.13	0.07	0.05

D Templates

Table 8: General templates used for matching and filtering out uninformative explanations

Abbreviation	Template
P	<PREMISE>
H	<HYPOTHESIS>
H_P	<HYPOTHESIS><PREMISE>
P_H	<PREMISE><HYPOTHESIS>
S1_P_S2_H	sentence 1 states <PREMISE>. sentence 2 is stating <HYPOTHESIS>
S2_H_S1_P	sentence 2 states <HYPOTHESIS>. sentence 1 is stating <PREMISE>
H_EXISTS	there is <HYPOTHESIS>
P_EXISTS	there is <PREMISE>

Table 9: Entailment templates used for matching and filtering out uninformative explanations

Abbreviation	Template
P_IMPLIES_H	<PREMISE>implies <HYPOTHESIS>
IF_P_THEN_H	if <PREMISE>then <HYPOTHESIS>
P_WOULD_IMPLY_H	<PREMISE>would imply <HYPOTHESIS>
H_REPHRASING_OF_P	<HYPOTHESIS>is a rephrasing of <PREMISE>
P_REPHRASING_OF_H	<PREMISE>is a rephrasing of <HYPOTHESIS>
BOTH_SENTENCES_H	in both sentences <HYPOTHESIS>
P_WOULD_BE_H	<PREMISE>would be <HYPOTHESIS>
P_CAN_BE_SAID_AS_H	<PREMISE>can also be said as <HYPOTHESIS>
H_CAN_BE_SAID_AS_P	<HYPOTHESIS>can also be said as <PREMISE>
H_LESS_SPECIFIC_P	<HYPOTHESIS>is a less specific rephrasing of <PREMISE>
CLARIFIES_H	this clarifies that <HYPOTHESIS>
IF_P_IT_MEANS_H	if <PREMISE>it means <HYPOTHESIS>
H_IN_BOTH_SENTENCES	<HYPOTHESIS>in both sentences
H_IN_BOTH	<HYPOTHESIS>in both
H_SAME_AS_P	<HYPOTHESIS>is same as <PREMISE>
P_SAME_AS_H	<PREMISE>is same as <HYPOTHESIS>
P_SYNONYM_OF_H	<PREMISE>is a synonym of <HYPOTHESIS>
H_SYNONYM_OF_P	<HYPOTHESIS>is a synonym of <PREMISE>

Table 10: Neutral templates used for matching and filtering out uninformative explanations

Abbreviation	Template
JUST_BECAUSE_P_NOT_MEAN_H	just because <PREMISE>doesn't mean <HYPOTHESIS>
CANNOT_INFER_H	cannot infer the <HYPOTHESIS>
CANNOT_ASSUME_H	one cannot assume <HYPOTHESIS>
CANNOT_INFER_THAT_H	one cannot infer that <HYPOTHESIS>
CANNOT_ASSUME_H_ALT	cannot assume <HYPOTHESIS>
P_NOT_MEAN_H	<PREMISE>does not mean <HYPOTHESIS>
DONT_KNOW_H	we don't know that <HYPOTHESIS>
FACT_P_NOT_MEAN_H	the fact that <PREMISE>doesn't mean <HYPOTHESIS>
FACT_P_NOT_IMPLY_H	the fact that <PREMISE>does not imply <HYPOTHESIS>
FACT_P_NOT_ALWAYS_MEAN_H	the fact that <PREMISE>does not always mean <HYPOTHESIS>
FACT_P_NOT_ALWAYS_IMPLY_H	the fact that <PREMISE>doesn't always imply <HYPOTHESIS>

Table 11: Contradiction templates used for matching and filtering out uninformative explanations

Abbreviation	Template
S1_P_WHILE_S2_H	in sentence 1 <PREMISE>while in sentence 2 <HYPOTHESIS>
CAN_EITHER_BE_P_OR_H	it can either be <PREMISE>or <HYPOTHESIS>
NOT_H_IF_P	it cannot be <HYPOTHESIS>if <PREMISE>
EITHER_P_OR_H	either <PREMISE>or <HYPOTHESIS>
EITHER_H_OR_P	either <HYPOTHESIS>or <PREMISE>
P_AND_OTHER_H	<PREMISE>and other <HYPOTHESIS>
H_AND_OTHER_P	<HYPOTHESIS>and other <PREMISE>
H_AFTER_P	<HYPOTHESIS>after <PREMISE>
P_NOT_SAME_AS_H	<PREMISE>is not the same as <HYPOTHESIS>
H_NOT_SAME_AS_P	<HYPOTHESIS>is not the same as <PREMISE>
P_CONTRADICTS_H	<PREMISE>is contradictory to <HYPOTHESIS>
H_CONTRADICTS_P	<HYPOTHESIS>is contradictory to <PREMISE>
P_CONTRADICTS_H_ALT	<PREMISE>contradicts <HYPOTHESIS>
H_CONTRADICTS_P_ALT	<HYPOTHESIS>contradicts <PREMISE>
P_CANNOT_ALSO_BE_H	<PREMISE>cannot also be <HYPOTHESIS>
H_CANNOT_ALSO_BE_P	<HYPOTHESIS>cannot also be <PREMISE>
EITHER_P_OR_H_NOT_BOTH	either <PREMISE>or <HYPOTHESIS>not both at the same time
P_OR_H_NOT_BOTH	<PREMISE>or <HYPOTHESIS>not both at the same time