

NILE : Natural Language Inference with Faithful Natural Language Explanations

Sawan Kumar

Indian Institute of Science, Bangalore
sawankumar@iisc.ac.in

Partha Talukdar

Indian Institute of Science, Bangalore
ppt@iisc.ac.in

Abstract

The recent growth in the popularity and success of deep learning models on NLP classification tasks has accompanied the need for generating some form of natural language explanation of the predicted labels. Such generated natural language (NL) explanations are expected to be *faithful*, i.e., they should correlate well with the model’s internal decision making. In this work, we focus on the task of natural language inference (NLI) and address the following question: *can we build NLI systems which produce labels with high accuracy, while also generating faithful explanations of its decisions?* We propose Natural-language Inference over Label-specific Explanations (NILE), a novel NLI method which utilizes auto-generated label-specific NL explanations to produce labels along with its faithful explanation. We demonstrate NILE’s effectiveness over previously reported methods through automated and human evaluation of the produced labels and explanations. Our evaluation of NILE also supports the claim that accurate systems capable of providing testable explanations of their decisions can be designed. We discuss the faithfulness of NILE’s explanations in terms of sensitivity of the decisions to the corresponding explanations. We argue that explicit evaluation of faithfulness, in addition to label and explanation accuracy, is an important step in evaluating model’s explanations. Further, we demonstrate that task-specific probes are necessary to establish such sensitivity.

1 Introduction

Deep learning methods have been employed to improve performance on several benchmark classification tasks in NLP (Wang et al., 2018, 2019). Typically, these models aim at improving label accuracy, while it is often desirable to also produce explanations for these decisions (Lipton, 2016;

Chakraborty et al., 2017). In this work, we focus on producing natural language explanations for Natural Language Inference (NLI), without sacrificing much on label accuracy.

There has been growing interest in producing natural language explanations for deep learning systems (Huk Park et al., 2018; Kim et al., 2018; Ling et al., 2017), including NLI (Camburu et al., 2018). In general, the explanations from these methods can typically be categorized as post-hoc explanations (Lipton, 2016). Camburu et al. (2018) propose an NLI system which first produces an explanation and then processes the explanation to produce the final label. We argue that these explanations also resemble post-hoc explanations (Section 4.2). Further, existing methods don’t provide a natural way to test the *faithfulness* of the generated explanations, i.e., how well do the provided explanations correlate with the model’s decision making.

We therefore propose Natural-language Inference over Label-specific Explanations (NILE)¹, which we train and evaluate on English language examples. Through NILE, we aim to answer the following question:

Can we build NLI systems which produce faithful natural language explanations of predicted labels, while maintaining high accuracy?

Briefly, in NILE, we first generate natural language explanations for each possible decision, and subsequently process these explanations to produce the final decision. We argue that such a system provides a natural way of explaining its decisions. The key advantage is the testability of these explanations, in themselves, as well as in terms of the sensitivity of the system’s prediction

¹NILE source code available at <https://github.com/SawanKumar28/nile>

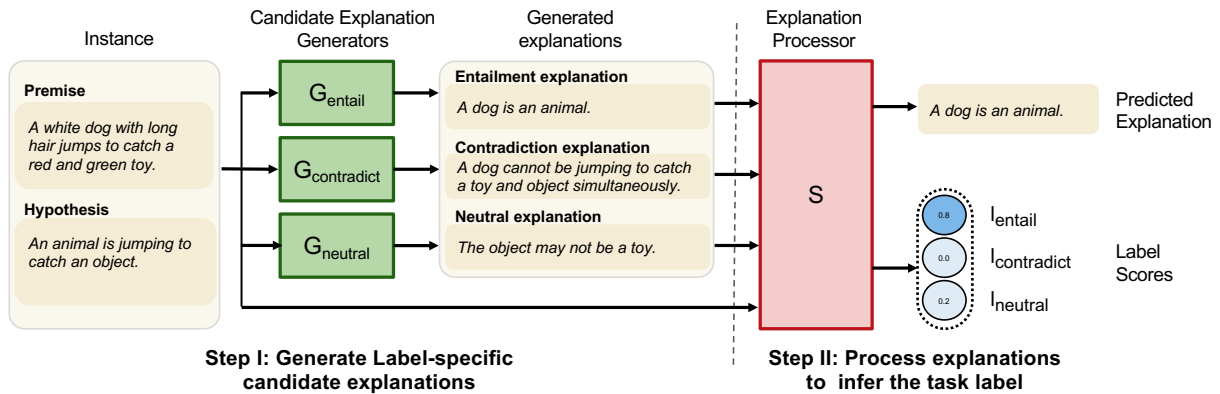


Figure 1: *Overview of NILE*: A Premise and Hypothesis pair is input to label-specific Candidate Explanation Generators G which generate natural language explanations supporting the corresponding label. The generated explanations are then fed to the Explanation Processor S , which generates label scores using the evidence present in these explanations (see Figure 3 for the architectures used in this work). In addition to the explanations, NILE also utilizes the premise and hypothesis pair (See Section 4.4.2 for a discussion on the challenges in building such a system). Please see Section 4 for details.

to these explanations.

We choose NLI due to its importance as an NLP task, and the availability of e-SNLI, a large dataset annotated both with entailment relation labels and natural language human explanations of those labels (Camburu et al., 2018; Bowman et al., 2015).

In summary, we make the following contributions in this work.

1. We propose NILE, an NLI system which generates and processes label-specific explanations to infer the task label, naturally providing explanations for its decisions.
2. We demonstrate the effectiveness of NILE compared to existing systems, in terms of label and explanation accuracy.
3. Through NILE, we provide a framework for generating falsifiable explanations. We propose ways to evaluate and improve the faithfulness of the system’s predictions to the generated explanations. We claim that task-specific probes of sensitivity are crucial for such evaluation.

We have released the source code of NILE to aid reproducibility of the results.

2 Related Work

Explainability of a model’s predictions has been studied from different perspectives, including feature importance based explanations (Ribeiro et al., 2016; Lundberg and Lee, 2017; Chen et al., 2018), or post-hoc natural language explanations (Huk Park et al., 2018; Kim et al., 2018; Ling et al.,

2017). Hendricks et al. (2018) produce counterfactual natural language explanations for image classification given an image and a counter-class label. Camburu et al. (2018) propose a model for NLI to first generate a free-form natural language explanation and then infer the label from the explanation. However, as noted by Oana-Maria et al. (2019a), the system tends to generate inconsistent explanations. We reason that requiring a model to generate an explanation of the correct output requires it to first infer the output, and the system thus resembles post-hoc explanation generation methods.

Given the diversity of desiderata and techniques for interpretability, the need for understanding interpretation methods and evaluating them has grown. Difficulty in building interpretation models and the lack of robustness of the same are some of the major issues in existing deep neural networks systems (Feng et al., 2018; Ghorbani et al., 2019; Oana-Maria et al., 2019b). Given these observations, measuring faithfulness, i.e., how well do the provided explanations correlate with the model’s decision making, is crucial. DeYoung et al. (2019) propose metrics to evaluate such faithfulness of rationales (supporting evidence) for NLP tasks.

Through NILE, we propose a framework for generating faithful natural language explanations by requiring the model to condition on generated natural language explanations. The idea of using natural language strings as a latent space has been explored to capture compositional task structure (Andreas et al., 2018). Wu et al. (2019) explore improving

visual question answering by learning to generate question-relevant captions. [Rajani et al. \(2019\)](#) aim to improve commonsense question answering by first generating commonsense explanations for multiple-choice questions, where the question and the choices are provided as the prompt. Similar to [Camburu et al., 2018](#)), they learn by trying to generate human-provided explanations and subsequently conditioning on the generated explanation. In NILE, we instead aim to produce an explanation for each possible label and subsequently condition on the generated label-specific explanations to produce the final decision.

3 Background

In this section, we discuss the datasets (Section 3.1) and pre-trained models (Section 3.2) used to build NILE.

3.1 Data

SNLI: The Stanford NLI dataset ([Bowman et al., 2015](#)) contains samples of premise and hypothesis pairs with human annotations, using Amazon Mechanical Turk. The premises were obtained from pre-existing crowdsourced corpus of image captions. The hypotheses were obtained by presenting workers with a premise and asking for a hypothesis for each label (entailment, neutral and contradiction), resulting in a balanced set of $\sim 570\text{K}$ pairs.

e-SNLI: [Camburu et al. \(2018\)](#) extend the SNLI dataset with natural language explanations of the ground truth labels. The explanations were crowdsourced using Amazon Mechanical Turk. Annotators were first asked to highlight words in the premise and hypothesis pairs which could explain the labels. Next, they were asked to write a natural language explanation using the highlighted words.

Similar to [Camburu et al. \(2018\)](#), for all our experiments, we filter out non-informative examples where the explanations contain the entire text of the premise or hypothesis. In particular, we drop any training example where the uncased premise or hypothesis text appears entirely in the uncased explanation. This leads to a training data size of $\sim 532\text{K}$ examples.

3.2 Pretrained Language Models

Transformer architectures ([Vaswani et al., 2017](#)) pre-trained on large corpora with self-supervision have shown significant improvements on various NLP benchmarks ([Devlin et al., 2019](#); [Radford](#)

[et al., 2019](#); [Yang et al., 2019](#); [Liu et al., 2019](#); [Lan et al., 2019](#)). Improvements have been demonstrated for text classification as well as text generation tasks ([Lewis et al., 2019](#); [Raffel et al., 2019](#)). In this work, we leverage the implementation of transformer architectures and pre-trained models provided by [Wolf et al. \(2019\)](#).

GPT-2: We use the GPT-2 architecture ([Radford et al., 2019](#)), which is trained using a causal language modeling loss (CLM), and includes a left-to-right decoder suitable for text generation. In particular, we use the gpt2-medium model. This model has 24 layers, 16 attention heads and a hidden size of 1024 ($\sim 345\text{M}$ parameters). For text generation, the model can be finetuned using CLM on desired text sequences.

RoBERTa: For classification modules, we leverage RoBERTa ([Liu et al., 2019](#)), which is trained using a masked language modeling loss (MLM). In particular, we use the roberta-base model. This model has 12 layers, 12 attention heads and a hidden size of 768 ($\sim 125\text{M}$ parameters). For downstream classifications tasks, a classification layer is added over the hidden-state of the first token in the last layer.

4 Natural-language Inference over Label-specific Explanations (NILE)

The overall architecture employed in NILE is shown in Figure 1. We introduce the notation used in this paper in Section 4.1. We then discuss the motivation for the major design choices in Section 4.2.

NILE performs the following steps to produce labels and explanations:

1. **Candidate Explanation Generators:** Label-specific Candidate Explanation Generators first generate explanations supporting the respective labels (Section 4.3).
2. **Explanation Processor:** The Explanation Processor takes the explanations and also the premise and hypothesis pairs as input to produce the task label (Section 4.4). We also build NILE-PH, where the Explanation Processor has access only to the generated explanations (Section 4.4.1).

We note that NILE-PH more naturally fits the desiderata described in Section 1, while we design and evaluate NILE for the more general case

where the Explanation Processor also accesses the premise and hypothesis pair.

In Section 4.5, we describe comparable baseline architectures.

4.1 Notation

We denote each data point by (p, h) , where p is the premise and h the hypothesis sentence. G denotes a model trained to generate natural language explanations. Specifically, G_x denotes a model which generates natural language explanations t_x of type x , where $x \in \{\text{entail, contradict, neutral}\}$. We denote the human-provided gold explanation for the correct predictions as t_g . S denotes a module which predicts label scores. The true label for an example is denoted by y , while a model prediction is denoted by y' , and label scores by l_x .

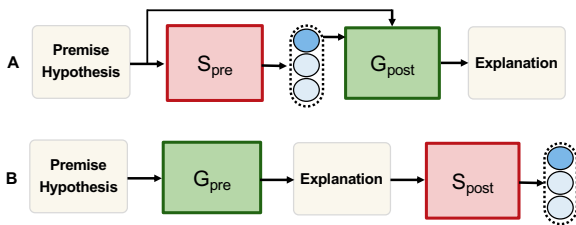


Figure 2: *Existing alternative architectures.* **A.** *Post-hoc generation*: Given an input instance, first the label is predicted and then an explanation generated conditioned on the label and the input text. **B.** *ExplainThen-Predict* (Camburu et al., 2018): Given the input instance, first the desired explanation is generated, and then the label is predicted using only the generated explanation. We argue that neither architecture provides a natural way to test the sensitivity of the model’s predictions to the generated explanation. Please see Section 4.2 for details.

4.2 Why do it this way?

In this section, we describe the motivation for adopting a two-step pipelined approach.

Label-specific explanations: Consider two alternative existing architectures in Figure 2. In Figure 2A, a model S_{pre} is trained directly on the example sentences (p & h) to produce a label (y'), which together with the example sentences are used to produce an explanation t'_g using G_{post} . It can be argued that while the target explanations may regularize the system, there is no reason for t'_g to be aligned with the reason why the model chose a particular label.

Figure 2B corresponds to a model which has also been trained on e-SNLI (Camburu et al., 2018).

G_{pre} is first trained to produce natural language explanations t'_g using human-provided explanations (t_g) as targets, using only the example sentences as inputs. A model S_{post} then chooses the label corresponding to the generated explanation t'_g . While at first, it appears that this system may provide faithful explanations of its decisions, i.e., the generated explanations are the reason for the label prediction, we argue that it may not be so.

In Figure 2B, G_{pre} is required to generate the explanation of the correct label for an example. It must first infer that label and then produce the corresponding explanation. Further analysis of the free-form human-provided explanations has revealed clear differences in the form of explanations, through alignment to label-specific templates (Camburu et al., 2018; Oana-Maria et al., 2019a). The Explanation Processor S_{post} then only needs to infer the form of t'_g . G_{pre} then resembles post-hoc generation methods, with the label (as the form of t'_g) and explanation t'_g being produced jointly. The claim is supported by inconsistencies found in the generated explanations (Oana-Maria et al., 2019a).

Neither architecture allows a natural way to test the sensitivity of the model’s predictions to its explanations. In NILE, we first allow explanations for each label, and then require the Explanation Processor to select the correct explanation. This allows us to naturally test whether the model’s predictions are indeed due to the selected explanation. This can be done, for example, by perturbing the input to the Explanation Processor.

A pipelined approach: We use a pipelined approach in NILE (Figure 1). The Candidate Explanation Generators are first trained using human-provided explanations. The Explanation Processor takes as input the generated label-specific explanations. This prevents the system from producing degenerate explanations to aid task performance. It also allows perturbing the generated explanations to probe the system in a more natural way compared to an unintelligible intermediate state of a learnt model. We believe that systems can be designed to work in this setting without compromising task performance.

4.3 Candidate Explanation Generators

We train label-specific explanation generators, G_x , $x \in \{\text{entail, contradict, neutral}\}$, using human-provided explanations of examples with the corresponding label. For example, to train G_{entail} , we

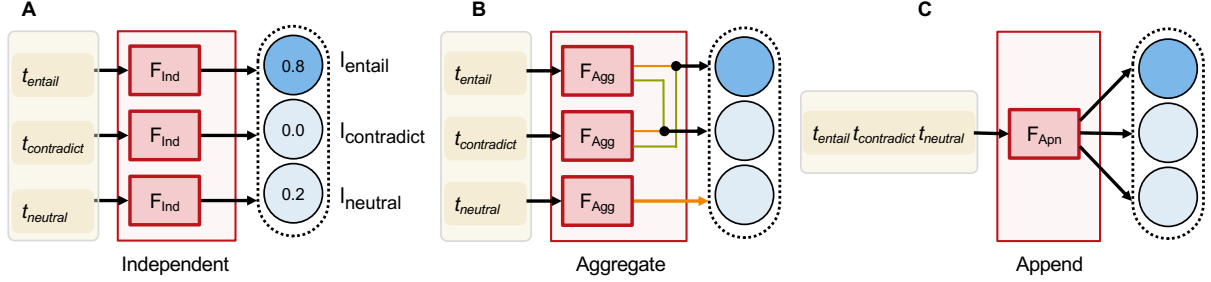


Figure 3: *Explanation Processor architectures*. **A.** Independent (Ind) collects evidence for a label symmetrically from the corresponding explanation. **B.** Aggregate (Agg) allows handling missing explanations by looking for contradictory evidence. **C.** Append (Apn) allows arbitrary evidence collection for each label. Please see Section 4.4.1 for details. Premise and hypothesis sentences are processed by additionally providing them to each block F_z where $z \in \{\text{Ind, Agg, Apn}\}$. Please see Section 4.4.2 for details.

collect all triplets (p, h, t_g) annotated as entailment. We create text sequences of the form: “*Premise: p Hypothesis: h [EXP] t_g [EOS]*” to fine-tune a pre-trained language model, where $[EXP]$ and $[EOS]$ are special tokens added to the vocabulary. During fine-tuning, the language modeling loss function is used only over the explanation tokens.

Next, we create prompts of the form “*Premise: p Hypothesis: h [EXP]*” and require each trained language model to independently complete the sequence. In this way we obtain label specific explanations $t_x, t_x = G_x(p, h)$, for $x \in \{\text{entail, contradict, neutral}\}$.

4.4 Explanation Processor

The Explanation Processor in NILE takes as input the generated label-specific explanations, as well as the premise and hypothesis pair to generate label scores $l_x, x \in \{\text{entail, contradict, neutral}\}$. During training, these scores are passed through a softmax layer and a cross-entropy loss is used to generate the training signal. During testing, the label with the maximum score is selected.

We leverage a pre-trained roberta-base model for all our experiments, and fine-tune it as specified in the following subsections. In each case, any intermediate scores are generated through transformations of the first token ($[CLS]$) embedding from the last layer. We define:

$$F_{\text{model}}(\text{inp}) = \tanh(W \cdot \text{CLS}_{\text{embed}}(\text{inp}))$$

where inp is a pair of sequences in NILE, a single sequence in NILE-PH, and W are the learnable parameters for the model.

For simplicity, and to elucidate the desired behavior, we first describe how explanations are processed in NILE-PH (Section 4.4.1). We then dis-

cuss the construction of NILE, a potential issue, and a fix for the same (Section 4.4.2).

4.4.1 Processing Explanations

In this section, we describe how explanations are processed in NILE-PH, which is generalized in NILE (Section 4.4.2). We experiment with three architectures, described below (also see Figure 3).

A. Independent: In the Independent model, explanations are fed to F_{Ind} , which generates a score for each explanations independently:

$$l_x = W_{\text{Ind}} F_{\text{Ind}}(t_x) \quad (1)$$

where $x \in \{\text{entail, contradict}\}$. We expect this score to represent the truthfulness of the input explanation.

B. Aggregate: The Independent model would need all three explanations to be available to reliably produce label scores. We believe a system should be able to handle one or more missing or ambiguous explanations. For example, the entailment explanation: “ t_{entail} : *A dog is a cat*” would provide evidence for contradiction. To capture this notion, we require the Explanation Processor to produce two intermediate scores V_1 and V_2 , where we expect V_1 to collect evidence supporting an input claim and V_2 to collect evidence against an input claim:

$$V_i(x) = W_{\text{Agg},i} F_{\text{Agg}}(t_x), \text{ where } i \in \{1, 2\} \quad (2)$$

The intermediate score are then aggregated into the final label scores:

$$\begin{aligned} l_{\text{entail}} &= \text{Cmb}(V_1(t_{\text{entail}}), V_2(t_{\text{contradict}})) \\ l_{\text{contradict}} &= \text{Cmb}(V_1(t_{\text{contradict}}), V_2(t_{\text{entail}})) \\ l_{\text{neutral}} &= V_1(t_{\text{neutral}}) \end{aligned} \quad (3)$$

where `Cmb` is the `LogSumExp` function. The reason for this choice of aggregation is that while evidence against entailment might point to contradiction and vice versa, evidence against neutral doesn’t necessarily provide any information about entailment or contradiction relations.

C. Append: Finally, to allow the model to reason arbitrarily between the three generated explanations, we created a single sequence, `concatecn`: “*entailment: t_{entail} contradiction: t_{contradict} neutral: t_{neutral}*”, and generate the scores as follows:

$$l_x = W_{\text{Apn},x} F_{\text{Apn}}(\text{concat}_{ecn}) \quad (4)$$

where $x \in \{\text{entail, contradict, neutral}\}$.

4.4.2 Processing Premise and Hypothesis

In NILE, to process premise p and hypothesis h , we first concatenate p and h into `concatph`: “*Premise: p Hypothesis: h*”. The label scores are then obtained as in Section 4.4.1, by modifying Equation 1, 2 and 4 as follows: replace $F_z(x)$ by $F_z(\text{concat}_{ph}, x)$, where $z \in \{\text{Ind, Agg, Apn}\}$. We note that appending the example sentences to the generated explanations (as in Append) would result in having no control over whether the explanations are used for the final prediction. The case for Independent and Aggregate is not immediately clear. We now discuss a potential issue with these architectures when processing premise and hypothesis text, and suggest a fix for the same.

The issue: We expect NILE to answer the question: Is $(\text{concat}_{ph}, t_x)$, where $x \in \{\text{entail, contradict, neutral}\}$, a valid instance-explanation pair? The Independent and Aggregate architectures for NILE have been designed such that the model can’t ignore the label-specific explanations. For example, the Independent model will produce identical scores for each output label, if it chooses to completely ignore the input explanations. However, the model is still free to learn a different kind of bias which is an outcome of the fact that natural language explanations convey ideas through both content and form. If the form for explanations of different labels is discriminative, an unconstrained learning algorithm could learn to infer first the type of explanation and use it to infer the task. For example, given the input $(\text{concat}_{ph}, t_x)$, where $x \in \{\text{entail, contradict, neutral}\}$, if a model could learn whether t_x is an entailment explanation, it then only has to output whether `concatph` corresponds

to an entailment relation. Essentially, high label accuracy can be achieved by inferring first what task to do using only the form of t_x .

The fix: To prevent NILE from exploiting the form of an explanation as described above, we create additional training examples, where we require NILE to score valid instance-explanation pairs higher. In particular, we sample negative explanations for an instance, of the same form as the correct label. For example, an instance labeled as entailment would have an additional training signal: Score $(\text{concat}_{ph}, t_{\text{entail}})$ higher than $(\text{concat}_{ph}, t'_{\text{entail}})$ and $(\text{concat}_{ph}, t''_{\text{entail}})$, where t'_{entail} and t''_{entail} are randomly sampled entailment form explanations.

We note that the fix leaves room for other kinds of biases to be learnt. However, the key advantage with NILE is that it is easy to design probes to test for such biases and subsequently fix them (see Section 5.3).

4.5 Baselines

We now describe baselines which use the same underlying blocks as NILE, for generating explanations and classification.

NILE:post-hoc: To understand the drop in performance which could be associated with constraining models as we have done, we train a model with full access to input examples (See Figure 2A).

$$l_x = W_x F_{\text{pre}}(p, h)$$

where $x \in \{\text{entail, contradict, neutral}\}$.

Further, we provide a strong baseline for post-hoc generators using this model, where using the model’s predictions, we simply pick the corresponding label-specific generated explanation.

$$t'_g = G_{\text{post}}(l_x) = t_x$$

We note that the model’s predictions have no sensitivity to the generated explanations in NILE: post-hoc.

ExplainThenPredictAttention (ETPA): Following (Camburu et al., 2018), (see Figure 2B), we train a pipelined system, where we first learn to generate the gold explanation t'_g , followed by a classification of t'_g to predict the label:

$$\begin{aligned} t'_g &= G_{\text{pre}}(\text{concat}_{ecn}) \\ l_x &= W_x F_{\text{post}}(t'_g) \end{aligned}$$

where $x \in \{\text{entail, contradict, neutral}\}$.

Model	SNLI Dev	SNLI Test	Explanation evaluation on first 100 SNLI Test Samples					
			Label Accuracy	Label Accuracy	A: Correct Labels	Averaged over annotators		Annotators in-agreement
	B: Correct Expl.	B/A				C: Correct Expl.	C/A	
SemBERT [#] (Zhang et al., 2019)	92.2	91.9	-	-	-	-	-	-
ETPA (Camburu et al., 2018)	Reported	81.71	-	-	64.27	-	-	-
	Reproduced	86.98	86.22	77	71.2	92.47	59	76.62
NILE:post-hoc		91.86	91.49	90	81.4	90.44	68	75.56
NILE-PH	Independent	84.69	84.13	78	72.0	92.31	61	78.21
	Aggregate	85.71	85.29	80	73.4	91.75	62	77.50
	Append	88.49	88.11	85	78.0	91.76	66	77.65
NILE-NS	Independent	91.56	90.91	88	80.8	91.82	69	78.41
	Aggregate	91.55	91.08	89	80.6	90.56	68	76.40
	Append	91.74	91.12	89	80.4	90.34	67	75.28
NILE	Independent	91.29	90.73	91	82.4	90.55	69	75.82
	Aggregate	91.19	90.91	90	81.4	90.44	68	75.56

Table 1: Comparison of label and explanation accuracy on the *in-domain* SNLI evaluation sets. Models are selected using the Dev set label accuracy over 5 runs with different seeds of random initialization. Mean (and standard deviation) over the 5 runs are reported in the Appendix. [#] indicates the best reported result at <https://nlp.stanford.edu/projects/snli/> at the time of writing. Note that SemBERT does not provide natural language explanations and is reported here only for reference. Bold numbers indicate highest among methods that produce explanations. Explanations are evaluated on the first 100 SNLI Test examples. We present reported numbers of ETPA (Camburu et al., 2018) as well as the results with our reproduction of ETPA. ETPA (reproduced) is directly comparable with NILE (Section 4.5). NILE-PH competes with or outperforms ETPA baselines on label accuracy, while NILE-NS and NILE provide significant gains in label accuracy. NILE and NILE-NS are competitive with the best reported results in terms of label accuracies. We report the number of correct explanations, averaged across annotators (B) as well as when all annotators agree on correctness (C). All NILE variants are able to provide more correct explanations than the ETPA baseline. We also report the percentage of correct explanations in the subset of correct label predictions (B/A, C/A). On this metric, NILE variants are comparable with the ETPA baseline. However, the real value of NILE lies in being able to probe the faithfulness of its decisions (Section 5.3). Further, NILE explanations generalize significantly better on out-of-domain examples (See Table 2). Please see Section 5.1 for details.

5 Experiments

In this section, we aim to answer the following questions:

- Q1** How does NILE compare with the baselines and other existing approaches in terms of final task performance, and explanation accuracy, on in-domain evaluation sets (train and test on SNLI)? (Section 5.1)
- Q2** How well does NILE transfer to out-of-domain examples (train on SNLI, and test on MNLI)? (Section 5.2)
- Q3** How faithful are the model’s predictions to the generated explanations? (Section 5.3)

We provide training details in Appendix A, and examples of generated label-specific explanations in Appendix B.

5.1 In-domain Results

We report the label accuracies of the baselines and proposed architectures on the SNLI Dev and Test set in Table 1. We also report explanation accuracies, obtained through human evaluation of the generated explanations in the first 100 test examples. Binary scores on correctness were sought from five annotators (non-experts in NLP) on the generated explanations. For both label and explanation accuracies, we report using a model selected using the SNLI Dev set label accuracy across 5 runs with 5 different seeds of random initialization. Please see the Appendix for more details on the the 5 runs. First, through NILE:post-hoc, we provide a strong baseline for obtaining high label and explanation accuracy. Our aim in this work is to learn explanations that serve as the reason for the model’s

Model		MNLI Dev	MNLI Dev-mm	Explanation evaluation on first 100 MNLI Dev Samples				
		Label Accuracy	Label Accuracy	A: Correct Labels	Averaged over annotators		Annotators in-agreement	
B: Correct Expl.	B/A				C: Correct Expl.	C/A		
ETPA (Camburu et al., 2018)	Reproduced	56.11	56.42	48	22.67	47.22	14	29.17
NILE:post-hoc		79.29	79.29	69	47.67	69.08	35	50.72
NILE-PH	Independent	54.95	55.35	46	34.33	74.64	28	60.87
	Aggregate	56.45	56.66	49	34.67	70.75	26	53.06
	Append	61.33	61.98	58	43.33	74.71	34	58.62
NILE-NS	Independent	74.84	75.20	68	49.67	73.04	37	54.41
	Aggregate	75.73	76.22	69	49.33	71.50	37	53.62
	Append	77.07	77.22	72	52.33	72.69	38	52.78
NILE	Independent	72.91	73.04	64	45.67	71.35	33	51.56
	Aggregate	72.94	73.01	63	45.67	72.49	34	53.97

Table 2: *Testing the generalization capability of NILE on the out-of-domain MNLI Dev sets.* Training and model selection is done on the SNLI dataset (Section 5.1), and evaluation on the out-of-domain MNLI Dev (matched) and MNLI Dev-mm (mismatched) sets. Label accuracies are reported for both MNLI Dev (matched) and MNLI Dev-mm (mismatched) sets, while explanations are evaluated on the first 100 MNLI Dev set examples. We report the number of correct explanations, averaged across annotators (B) as well as when all annotators agree on correctness (C). All NILE variants provide more correct explanations than the ETPA baseline (B, C). Further, the percentage of correct explanations in the subset of correct label predictions (B/A, C/A) is significantly better for all NILE variants. The results demonstrate that NILE provides a more generalizable framework for producing natural language explanations. Please see Section 5.2 for details.

predictions. Nevertheless, we are able to compete or outperform this baseline, in terms of explanation accuracy, while incurring a only a small drop in label accuracy. All variants of NILE, including NILE-PH and NILE-NS (which is not trained using negative samples of explanations as described in Section 4.4.2), produce more correct explanations than the ETPA baseline. NILE-PH:Append, NILE and NILE-NS provide gains over label accuracies compared to the ETPA baseline. Additionally, NILE and its variants provide natural ways to probe the sensitivity of the system’s predictions to the explanations, as demonstrated in the subsequent sections. Finally, the explanations generated by all NILE variants generalize significantly better on out-of-distribution examples when compared to the ETPA baseline (See Section 5.2).

5.2 Transfer to Out-of-domain NLI

To test the generalization capability of NILE, we do training and model selection on the SNLI dataset (Section 5.1), and evaluate on the out-of-domain MNLI (Williams et al., 2018) development sets. Transfer without fine-tuning to out-of-domain NLI has been a challenging task with transfer learning

for generating explanations in MNLI being particularly challenging (Camburu et al., 2018). We report label accuracies on the Dev (matched) and Dev-mm (mismatched) sets, and explanation evaluation on the first 100 Dev samples in Table 2. Explanation evaluation was done by three annotators (who also annotated the SNLI explanations). While the label accuracies follow a similar pattern as the in-domain SNLI Test set, all variants of NILE provide gains in the quality of generated explanations. All variants of NILE produce more correct explanations (B, C) as well as a higher percentage of correct generated explanations among correct predictions (B/A, C/A). This demonstrates that NILE, through intermediate label-specific natural language explanations, provides a more general way for building systems which can produce natural language explanations for their decisions.

5.3 Evaluating Faithfulness using Sensitivity Analysis

NILE and its variants allow a natural way to probe the sensitivity of their predictions to the generated explanations, which is by perturbing the explanations themselves. In this way, NILE resembles

Model		I+ Exp	I only	Exp only
NILE-NS	Independent	91.6	33.8	69.4
	Aggregate	91.6	33.8	74.5
	Append	91.7	91.2	72.9
NILE	Independent	91.3	33.8	46.1
	Aggregate	91.2	33.8	40.7

Table 3: *Estimating the sensitivity of the system’s predictions to input explanations through erasure.* During testing, we erase either the instance or the explanations from the input to NILE-NS and NILE. The results seem to indicate that NILE-NS’s predictions are more faithful, in the sense of having a higher sufficiency. However, as demonstrated subsequently, the sensitivity of NILE-NS’s prediction to the input explanations is not as desired. Please see Section 5.3 for details.

Model		Dev Set	Shuffled Dev Set
NILE-NS	Independent	91.6	88.1
	Aggregate	91.6	89.6
	Append	91.7	88.5
NILE	Independent	91.3	35.3
	Aggregate	91.2	31.6

Table 4: *Probing the sensitivity of the system’s predictions by shuffling instance-explanation pairs.* Each instance is attached to a randomly selected explanation of the same form as the original pair. The results demonstrate a much weaker link between NILE-NS’s predictions and associated explanations. On the other hand, NILE behaves more expectedly. Note that the baselines don’t allow a similar mechanism to test their faithfulness, and such testability is a key advantage of NILE. Please see Section 5.3 for details.

explanation systems which provide input text fragments as reasons for their decisions. DeYoung et al. (2019) propose metrics to evaluate the faithfulness of such explanations. Following their work, we first attempt to measure the explanations generated by the methods proposed in this paper for comprehensiveness (what happens when we remove the explanation from the input) and sufficiency (what happens if we keep only the explanations). In Table 3, we show these measures for NILE and NILE-NS. The results seem to indicate that explanations for both NILE and NILE-NS are comprehensive, while having higher sufficiency in the case of NILE-NS. We first note that the comprehensiveness of these systems is ensured by design, and the input is indistinguishable without an explanation. Second, we argue that sufficiency may indicate correlations which don’t necessarily exist

in the system otherwise. We study the sensitivity of the explanations through a probe motivated by an understanding of the task and the training examples (see Section 4.4.2). We perturb the instance-explanation inputs such that for each test instance, the explanation is replaced by a randomly selected explanation of the same label. The results (Table 4) indicate that NILE-NS is more robust to random perturbations of input explanations, and presumably uses the form of the explanation to infer the task (see Section 4.4.2 for a discussion). It is true that NILE behaves expectedly as we have specifically designed NILE to prevent the associated bias, and that this could potentially lead the system to learn other such biases. However, a key advantage of the proposed architecture is the ability to identify and fix for such biases. We leave it as an interesting and challenging future work to find and fix more such biases.

6 Conclusion

In this paper we propose NILE, a system for Natural Language Inference (NLI) capable of generating labels along with natural language explanations for the predicted labels. Through extensive experiments, we demonstrate the effectiveness of this approach, in terms of both label and explanation accuracy. NILE supports the hypothesis that accurate systems can produce testable natural language explanations of their decisions. In the paper, we also argue the importance of explicit evaluation of faithfulness of the generated explanations, i.e., how correlated are the explanations to the model’s decision making. We evaluate faithfulness of NILE’s explanations using sensitivity analysis. Finally, we demonstrate that task-specific probes are necessary to measure such sensitivity.

Acknowledgments

We thank the anonymous reviewers for their constructive comments. This work is supported by the Ministry of Human Resource Development (Government of India). We would also like to thank HuggingFace for providing a state-of-the-art Transformers library for natural language understanding. Finally, we want to thank the annotators who annotated generated explanations for correctness.

References

- Jacob Andreas, Dan Klein, and Sergey Levine. 2018. [Learning with latent language](#). 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 2166–2179, New Orleans, Louisiana. Association for Computational Linguistics.
- Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. [A large annotated corpus for learning natural language inference](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.
- Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-SNLI: Natural Language Inference with Natural Language Explanations. In *Advances in Neural Information Processing Systems*, pages 9539–9549.
- Supriyo Chakraborty, Richard Tomsett, Ramya Raghavendra, Daniel Harborne, Moustafa Alzantot, Federico Cerutti, Mani Srivastava, Alun Preece, Simon Julier, Raghuvier M Rao, et al. 2017. Interpretability of Deep Learning Models: A Survey of Results. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, pages 1–6. IEEE.
- Jianbo Chen, Le Song, Martin Wainwright, and Michael Jordan. 2018. Learning to Explain: An Information-Theoretic Perspective on Model Interpretation. In *International Conference on Machine Learning*, pages 882–891.
- 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.
- Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C Wallace. 2019. ERASER: A Benchmark to Evaluate Rationalized NLP Models. *arXiv preprint arXiv:1911.03429*.
- Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. [Pathologies of Neural Models Make Interpretations Difficult](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.
- Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of Neural Networks is Fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3681–3688.
- Lisa Anne Hendricks, Ronghang Hu, Trevor Darrell, and Zeynep Akata. 2018. Generating counterfactual explanations with natural language. In *ICML Workshop on Human Interpretability in Machine Learning*, pages 95–98.
- Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. 2018. Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8779–8788.
- Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. 2018. Textual Explanations for Self-Driving Vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *International Conference on Learning Representations*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv preprint arXiv:1910.13461*.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. [Program Induction by Rationale Generation: Learning to Solve and Explain Algebraic Word Problems](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.
- Zachary C Lipton. 2016. The Mythos of Model Interpretability. *arXiv preprint arXiv:1606.03490*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint arXiv:1907.11692*.
- Scott M Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.
- Camburu Oana-Maria, Shillingford Brendan, Minervini Pasquale, Lukasiewicz Thomas, and Blunsom Phil. 2019a. Make Up Your Mind! Adversarial Generation of Inconsistent Natural Language Explanations. *arXiv preprint arXiv:1910.03065*.

- Camburu Oana-Maria, Giunchiglia Eleonora, Foerster Jakob, Lukasiewicz Thomas, and Blunsom Phil. 2019b. Can I Trust the Explainer? Verifying Post-hoc Explanatory Methods. *arXiv preprint arXiv:1910.02065*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8).
- 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. *arXiv preprint arXiv:1910.10683*.
- 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.
- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in neural information processing systems*, pages 5998–6008.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In *Advances in Neural Information Processing Systems*, pages 3261–3275.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. [GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.
- 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, New Orleans, Louisiana. Association for Computational Linguistics.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace’s Transformers: State-of-the-art Natural Language Processing. *ArXiv*, abs/1910.03771.
- Jialin Wu, Zeyuan Hu, and Raymond Mooney. 2019. [Generating question relevant captions to aid visual question answering](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3585–3594, Florence, Italy. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in neural information processing systems*, pages 5754–5764.
- Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2019. Semantics-aware BERT for Language Understanding. *arXiv preprint arXiv:1909.02209*.

Appendix A. Experimental Setup

Model		SNLI Dev Label Accuracy	
		Mean	Stddev
ETPA	Reproduced	86.96	0.02
NILE:post-hoc		91.77	0.06
NILE-PH	Independent	84.53	0.18
	Aggregate	85.47	0.26
	Append	88.30	0.12
NILE-NS	Independent	90.17	2.76
	Aggregate	91.44	0.06
	Append	91.57	0.14
NILE	Independent	91.09	0.19
	Aggregate	90.94	0.22

Table 5: Mean and Standard Deviation for label accuracies on SNLI Dev set are reported. NILE-NS:Independent system has a high standard deviation and relatively lower mean accuracy. This is due to a bad random initialization with seed 219. When seed 219 results are excluded, the mean and standard deviation are 91.41 and 0.20 respectively.

For fine-tuning gpt2-medium language models for explanation generation as well as roberta-base models, we leverage code and pre-trained models from the “transformers” library available at <https://github.com/huggingface>. In each case we train on the train split for three epochs. Apart from batch size, sequence length and seed for random initialization, we keep the other hyperparameters fixed throughout the experiments. We don’t do any fine-tuning on seeds of random initialization.

For roberta-base models, we report results through model selection on models trained using 5 seeds of random initialization - 42, 219, 291, 67 and 741. Model selection is done using label accuracies on SNLI Dev set. In Table 5, we report the mean and standard deviation for the label accuracies across 5 runs.

We ran our experiments on GeForce GTX 1080 Ti GPUs. We adjust the batch size to be the largest multiple of 16 to fit on the GPU memory (~ 12 GB). We now list all the hyper-parameters used.

GPT2 : The hyper-parameters used for fine-tuning gpt2-medium include a maximum sequence length of 128, batch size of 2, learning rate of $5e-5$, Adam epsilon of $1e-8$, max gradient norm of 1.0 and a seed of 42. For generating text, we used greedy decoding.

Model		Batch size	Max seq length
ETPA	Reproduced	32	100
NILE:post-hoc		32	100
NILE-PH	Independent	32	50
	Aggregate	32	50
	Append	32	100
NILE-NS	Independent	16	100
	Aggregate	16	100
	Append	16	200
NILE	Independent	16	100
	Aggregate	16	100

Table 6: Hyper-parameters (batch size and maximum sequence length) used for fine-tuning roberta-base

RoBERTa : The fixed parameters for fine-tuning roberta-base included a learning rate of $2e-5$, Adam epsilon of $1e-8$ and max gradient norm 1.0. Other parameters are captured in Table 6.

Appendix B. Generated Explanations

In this section, we list the label-specific explanations generated for five Dev set examples, along with the premise, hypothesis and gold labels, for both SNLI (Section B.1) and MNLI (Section B.2) datasets.

B.1 SNLI

- **Example 1**

PairID: 4705552913.jpg#2r1n

gold label: neutral

Premise: Two women are embracing while holding to go packages.

Hypothesis: The sisters are hugging goodbye while holding to go packages after just eating lunch.

entailment: Sisters are women.

contradiction: The women cannot be embracing if they are hugging goodbye.

neutral: Two women embracing does not imply they are sisters hugging goodbye.

- **Example 4**

PairID: 2407214681.jpg#0r1e

gold label: entailment

Premise: Two young children in blue jerseys, one with the number 9 and one with the number 2 are standing on wooden steps in a bathroom and washing their hands in a sink.

Hypothesis: Two kids in numbered jerseys wash their hands.

entailment: Two young children are two kids, and blue jerseys implies numbered jerseys.

contradiction: The kids are either standing on wooden steps or wash their hands.

neutral: The kids could be washing their hands in a sink for any reason, not just because they wash their hands.

- **Example 18**

PairID: 3667788497.jpg#0r1c

gold label: contradiction

Premise: Under a blue sky with white clouds, a child reaches up to touch the propeller of a plane standing parked on a field of grass.

Hypothesis: A child is playing with a ball.

entailment: A propeller is a type of ball.

contradiction: A child cannot touch the propeller of a plane while playing with a ball.

neutral: Just because the child reaches up to touch the propeller of a plane does not mean the child is playing with a ball.

- **Example 24**

PairID: 6170429974.jpg#3r1c

gold label: contradiction

Premise: Two men on bicycles competing in a race.

Hypothesis: A few people are catching fish.

entailment: Two men are a few people, and

catching fish is a form of competition.

contradiction: competing and catching are different actions

neutral: Competing in a race does not imply catching fish.

- **Example 30**

PairID: 173499538.jpg#2r1c

gold label: contradiction

Premise: Families waiting in line at an amusement park for their turn to ride.

Hypothesis: People are waiting to see a movie.

entailment: People are waiting to see a movie at an amusement park.

contradiction: People cannot be waiting to ride an amusement park and a movie at the same time.

neutral: Not all rides at amusement parks are a movie.

B.2 MNL1

- **Example 1**

PairID: 63735n

gold label: neutral

Premise: The new rights are nice enough

Hypothesis: Everyone really likes the newest benefits

entailment: Nice enough is another way of saying everyone really likes.

contradiction: If everyone really likes the benefits, then the rights wouldn't be nice enough.

neutral: New benefits are not necessarily the newest benefits.

- **Example 14**

PairID: 73734e

gold label: entailment

Premise: You and your friends are not welcome here, said Severn.

Hypothesis: Severn said the people were not welcome there.

entailment: Friends are people.

contradiction: People who are not welcome here are not welcome here.

neutral: You can not infer they were not welcome.

- **Example 17**

PairID: 70726e

gold label: entailment

Premise: so i have to find a way to supplement that

Hypothesis: I need a way to add something extra.

entailment: Add something extra is a rephrasing of Supplement that.

contradiction: A way to supplement something extra is not the same as a way to add something extra.

neutral: Just because someone is adding something doesn't mean they are doing it extra.

- **Example 26**

PairID: 67610c

gold label: contradiction

Premise: Sorry but that's how it is.

Hypothesis: This is how things are and there are no apologies about it.

entailment: oops that's how it is is same as there are no apologies about it

contradiction: A person can't be sorry and have no apologies.

neutral: Just because someone is sorry does not mean they are saying no apologies.

- **Example 45**

PairID: 98811c

gold label: contradiction

Premise: yeah i mean just when uh the they military paid for her education

Hypothesis: The military didn't pay for her education.

entailment: The military paid for her education, doesn't matter if it was for college or not.

contradiction: The military either paid for her education or they didn't.

neutral: Just because the military paid for her education doesn't mean she didn't get paid for it.