Unsupervised Natural Language Inference Using PHL Triplet Generation

Neeraj Varshney, Pratyay Banerjee, Tejas Gokhale, Chitta Baral
Arizona State University
{nvarshn2, pbanerj6, tgokhale, cbaral}@asu.edu

Abstract

Transformer-based models achieve impressive performance on numerous Natural Language Inference (NLI) benchmarks when trained on respective training datasets. However, in certain cases, training samples may not be available or collecting them could be time-consuming and resource-intensive. In this work, we address the above challenge and present an explorative study on unsupervised NLI, a paradigm in which no human-annotated training samples are available. We investigate it under three settings: *PH*, *P*, and *NPH* that differ in the extent of unlabeled data available for learning. As a solution, we propose a procedural data generation approach that leverages a set of sentence transformations to collect PHL (Premise, Hypothesis, Label) triplets for training NLI models, bypassing the need for human-annotated training data. Comprehensive experiments with several NLI datasets show that the proposed approach results in accuracies of up to 66.75%, 65.9%, 65.39% in PH, P, and NPH settings respectively, outperforming all existing unsupervised baselines. Furthermore, fine-tuning our model with as little as ~0.1% of the human-annotated training dataset (500 instances) leads to 12.2% higher accuracy than the model trained from scratch on the same 500 instances. Supported by this superior performance, we conclude with a recommendation for collecting high-quality task-specific data.

1 Introduction

Natural Language Inference (NLI) is the task of determining whether a "hypothesis" is true (Entailment), false (Contradiction), or undetermined (Neutral) given a "premise". State-of-the-art models have matched human performance on several NLI benchmarks, such as SNLI (Bowman et al., 2015), Multi-NLI (Williams et al., 2018), and Dialogue NLI (Welleck et al., 2019). This high performance can be partially attributed to the availability of large training datasets; SNLI (570k), Multi-NLI (392k), and Dialogue-NLI (310k). For new domains, collecting such training data is time-consuming and can require significant resources. What if no training data was available at all?

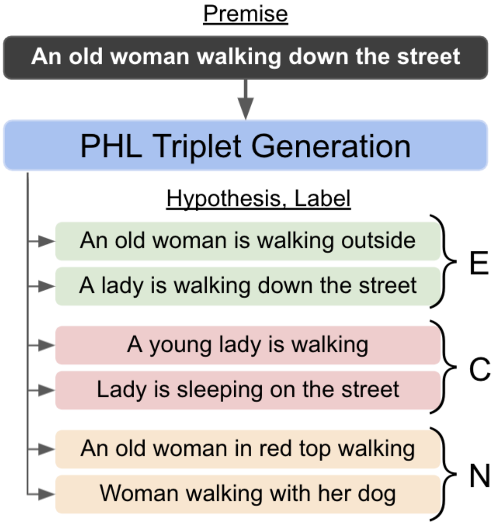
Premise
An old woman walking down the street
PHL Triplet Generation
Hypothesis, Label
An old woman is walking outside
A lady is walking down the street
E
A young lady is walking
Lady is sleeping on the street
C
An old woman in red top walking
Woman walking with her dog
N

Figure 1: Illustrating our procedural data generation approach for unsupervised NLI. A sentence is treated as premise, and multiple hypotheses conditioned on each label (Entailment- E, Contradiction- C, and Neutral- N) are generated using a set of sentence transformations.

In this work, we address the above question and explore *Unsupervised NLI*, a paradigm in which no human-annotated training data is provided for learning the task. We study three different unsupervised settings: *PH*, *P*, and *NPH* that differ in the extent of unlabeled data available for learning. In PH-setting, unlabeled premise-hypothesis pairs are available i.e. data without ground-truth labels. In P-setting, only a set of premises are available i.e. unlabeled partial inputs. The third setting NPH does not provide access to any training dataset, and thus it is the hardest among the three unsupervised settings considered in this work.

We propose to solve these unsupervised settings using a procedural data generation approach. Given a sentence, our approach treats it as a premise (P)

Findings of the Association for Computational Linguistics: ACL 2022, pages 2003 - 2016
May 22-27, 2022 ©2022 Association for Computational Linguistics

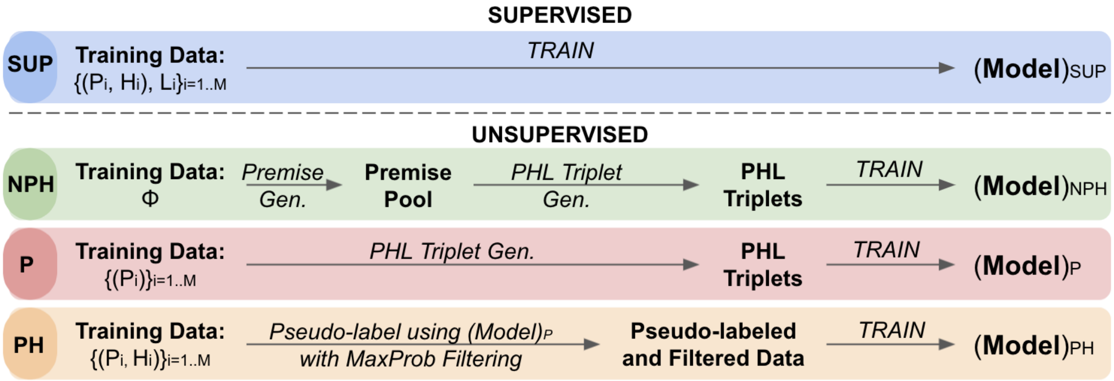

Figure 2: Comparing supervised NLI with our three unsupervised settings. For unsupervised settings, we procedurally generate PHL triplets to train the NLI model. In **NPH setting**, a premise pool is collected from raw text corpora such as Wikipedia and then used for generating PHL triplets. In **P setting**, we directly apply these transformations on the available premises. In **PH setting**, we leverage the P-setting model to pseudo-label and filter the provided unlabeled PH pairs and then train the NLI model using this pseudo-labeled dataset.

and generates multiple hypotheses (H) corresponding to each label (L = Entailment, Contradiction, and Neutral) using a set of sentence transformations (refer to Figure 1). This results in creation of Premise-Hypothesis-Label (PHL) triplets that can be used for training the NLI model. In the P and PH settings, we directly apply our sentence transformations over the available premises to generate PHL triplets. However, in the NPH setting, premises are not available. We tackle this challenge by incorporating a premise generation step that extracts sentences from various raw text corpora such as Wikipedia and short stories. We use these extracted sentences as premises to generate PHL triplets. In Figure 2, we compare the four settings (one supervised and three unsupervised) and show our approach to develop an NLI model for each setting.

To evaluate the efficacy of the proposed approach, we conduct comprehensive experiments with several NLI datasets. We show that our approach results in accuracies of 66.75%, 65.9%, and 65.39% on SNLI dataset in PH, P, and NPH settings respectively, outperforming all existing unsupervised methods by $\sim13\%$. We also conduct experiments in low-data regimes where a few human-annotated labeled instances are provided and show that further fine-tuning our models with these instances consistently achieves higher performance than the models fine-tuned from scratch. For example, with just 500 labeled instances, our models achieve 8.4% and 10.4% higher accuracy on SNLI and MNLI datasets respectively. Finally, we show that fine-tuning with 'adversarial' instances instead of randomly selected human-annotated instances further improves the performance of our models; it leads to 12.2% and 10.41% higher accuracy on SNLI and MNLI respectively.

In summary, our contributions are as follows:

1. We explore three unsupervised settings for NLI and propose a procedural data generation approach that outperforms the existing approaches by $\sim13\%$ and raises the state-of-the-art unsupervised performance on SNLI to 66.75%.
2. We also conduct experiments in low-data regimes and demonstrate that further fine-tuning our models with the provided instances achieves 8.4% and 10.4% higher accuracy on SNLI and MNLI datasets respectively.
3. Finally, we show that using 'adversarial' instances for fine-tuning instead of randomly selected instances further improves the accuracy. It leads to 12.2% and 10.41% higher accuracy on SNLI and MNLI respectively. Supported by this superior performance, we conclude with a recommendation for collecting high-quality task-specific data.

We release the implementation[1] of our procedural data generation approach and hope that our work will encourage research in developing techniques that reduce reliance on expensive human-annotated data for training task-specific models.

[1] https://github.com/nrjvarshney/unsupervised_NLI

2 Related Work

Unsupervised Question-Answering: The *unsupervised* paradigm where no human-annotated training data is provided for learning has mostly been explored for the Question Answering (QA) task in NLP. The prominent approach involves synthesizing QA pairs and training a model on the synthetically generated data. Lewis et al. (2019); Dhingra et al. (2018); Fabbri et al. (2020) propose a template-based approach, while Puri et al. (2020) leverage generative models such as GPT-2 (Radford et al., 2019) to synthesize QA pairs. Banerjee and Baral (2020) create synthetic graphs for commonsense knowledge and propose knowledge triplet learning. Wang et al. (2021) leverage few-shot inference capability of GPT-3 (Brown et al., 2020) to synthesize training data for SuperGLUE (Wang et al., 2019) tasks. For visual question answering, Gokhale et al. (2020) use template-based data augmentation methods for negation, conjunction, and Banerjee et al. (2021) utilize image captions to generate training data. Gokhale et al. (2021) use linguistic transformations in a distributed robust optimization setting for vision-and-language inference models.

Unsupervised NLI: In NLI, Cui et al. (2020) propose a multimodal aligned contrastive decoupled learning method (MACD) and train a BERT-based text encoder. They assign a label (E, C, N) based on the cosine similarity between representations of premise and hypothesis learned by their text encoder. Our approach differs from MACD as we leverage a procedural data generation step based on a set of sentence transformations and do not leverage data from other modalities. We use MACD as one of the baselines in our experiments.

3 Unsupervised NLI

In NLI, a premise-hypothesis pair (P, H) is provided as input and the system needs to determine the relationship $L \in \{\textit{Entailment}, \textit{Contradiction}, \textit{Neutral}\}$ between P and H. In the **supervised setting**, a labeled dataset $D_{train} = \{(P_i, H_i), L_i\}_{i=1}^{M}$ consisting of M instances which are usually human-annotated is available for training. However in the unsupervised setting, labels L_i are not available, thus posing a significant challenge for training NLI systems. Along with this standard unsupervised setting (referred to as PH), we consider two novel unsupervised settings (P and NPH) that differ in the extent of unlabeled data available for learning:

PH-setting: It corresponds to the standard unsupervised setting where an unlabeled dataset of PH pairs ($\{(P_i, H_i)\}_{i=1}^{M}$) is provided.

P-setting: In this setting, only premises from D_{train} i.e ($\{(P_i)\}_{i=1}^{M}$) are provided. It is an interesting setting as the large-scale NLI datasets such as SNLI (Bowman et al., 2015) and MultiNLI (Williams et al., 2018) have been collected by presenting only the premises to crowd-workers and asking them to write a hypothesis corresponding to each label. Furthermore, this setting presents a harder challenge for training NLI systems than the PH-setting as only partial inputs are provided.

NPH-setting: Here, no datasets (even with partial inputs) are provided. Thus, it corresponds to the hardest unsupervised NLI setting considered in this work. This setting is of interest in scenarios where we need to make inferences on a test dataset but its corresponding training dataset is not available in any form.

From the above formulation, it can be inferred that the hardness of the task increases with each successive setting (PH→P→NPH) as lesser and lesser information is made available. In order to address the challenges of each setting, we propose a two-step approach that includes a pipeline for procedurally generating PHL triplets from the limited information provided in each setting (Section 4), followed by training an NLI model using this procedurally generated data (Section 5). Figure 2 highlights the differences between four NLI settings (one supervised and three unsupervised) and summarizes our approach to develop an NLI model for each setting.

4 PHL Triplet Generation

To compensate for the absence of labeled training data, we leverage a set of sentence transformations and procedurally generate PHL triplets that can be used for training the NLI model. In P and PH settings, we apply these transformations on the provided premise sentences. In the NPH setting where premises are not provided, we extract sentences from various raw text corpora and apply these transformations on them to generate PHL triplets.

4.1 $\mathcal{P}$: Premise Generation

We extract sentences from raw text sources, namely, COCO captions (Lin et al., 2014), ROC stories (Mostafazadeh et al., 2016), and Wikipedia to compile a set of premises for the NPH setting. We use these text sources as they are easily available and contain a large number of diverse sentences from multiple domains.

ROC Stories is a collection of short stories consisting of five sentences each. We include all these sentences in our premise pool. **MS-COCO** is a dataset consisting of images with five captions each. We add all captions to our premise pool. From **Wikipedia**, we segment the paragraphs into individual sentences and add them to our premise pool.

We do not perform any sentence filtration during the premise collection process. However, each transformation (described in subsection 4.2) has its pre-conditions such as presence of verbs/adjectives/nouns that automatically filter out sentences from the premise pool that can not be used for PHL triplet generation.

4.2 $\mathcal{T}$: Transformations

Now, we present our sentence transformations for each NLI label. Table 1 illustrates examples of PHL triplets generated from these transformations.

4.2.1 Entailment:

In NLI, the label is entailment when the hypothesis must be true if the premise is true.

Paraphrasing (PA): Paraphrasing corresponds to expressing the meaning of a text (restatement) using other words and hence results in entailment premise-hypothesis pairs. We use the Pegasus (Zhang et al., 2019) tool to generate up to 10 paraphrases of a sentence and use them as hypothesis with the original sentence as the premise [2].

Extracting Snippets (ES): We use dependency parse tree to extract meaningful snippets from a sentence and use them as hypothesis with the original sentence as the premise. Specifically, we extract sub-trees that form a complete phrase or a sentence. For example, from the sentence "*A person with red shirt is running near the garden*", we create entailing hypotheses "*A person is running near the garden*", "*A person is running*", "*A person is near the garden*", etc. We implement 10 such techniques using spacy (Honnibal et al., 2020)[2].

Hypernym Substitution (HS): A hypernym of a word is its supertype, for example, "animal" is a hypernym of "dog". We use WordNet (Miller, 1995) to collect hypernyms and replace noun(s) in a sentence with their corresponding hypernyms to create entailment hypothesis. For example, from the premise "*A black dog is sleeping*", we create "*A black animal is sleeping*". Note that swapping the premise and hypothesis in this case gives us another PH pair that has a 'Neutral' relationship.

Pronoun Substitution (PS): Here, we leverage Part-of-Speech (POS) tagging of spacy to heuristically substitute a noun with its mapped pronoun. For example, substituting "boy" with "he" in the sentence "*boy is dancing in arena*" results in an entailing hypothesis "*he is dancing in arena*"[2].

Counting (CT): Here, we count nouns with common hypernyms and use several templates such as "*There are {count} {hypernym}s present"* to generate entailing hypotheses. For instance, from the sentence "*A motorbike and a car are parked*", we create hypothesis "*Two automobiles are parked*". We also create contradiction hypotheses using the same templates by simply changing the *count* value such as "*There are five automobiles present*"[2].

4.2.2 Contradiction:

The label is contradiction when the hypothesis can never be true if the premise is true.

Contradictory Words (CW): We replace noun(s) and/or adjective(s) (identified using spacy POS tagging) with their corresponding contradictory words. For example, replacing the word 'big' with 'small' in "*He lives in a big house*" results in a contradictory hypothesis "*He lives in a small house*". For contradictory adjectives, we collect antonyms from wordnet and for nouns, we use the function '*most_similar*' from gensim (Rehurek and Sojka, 2011) [2].

Contradictory Verb (CV): We collect contradictory verbs from gensim and create hypothesis in the following two ways: (i) substituting verb with its contradictory verb: for example, from "*A girl is walking*", we create hypothesis "*A girl is driving*" and (ii) selecting other sentences from the premise pool that have the same subject as the original sentence but have contradictory verbs: for example, sentences like "*A young girl is driving fast on the street*" and "*There is a girl skiing with*

[2] Further details are in Appendix Section A

Transformation	Original Sentence (Premise)	Hypothesis	Label
PA	Fruit and cheese sitting on a black plate	There is fruit and cheese on a black plate	E
PA + ES + HS	A large elephant is very close to the camera	Elephant is close to the photographic equipment	E
CW-noun	Two horses that are pulling a carriage in the street	Two dogs that are pulling a carriage in the street	C
CV	A young man sitting in front of a TV	A man in green jersey jumping on baseball field	C
PA + CW	A woman holding a baby while a man takes a picture of them	A kid is taking a picture of a male and a baby	C
FCon	A food plate on a glass table	A food plate made of plastic on a glass table	N
PA + AM	Two dogs running through the snow	The big dogs are outside	N

Table 1: Illustrative examples of PHL triplets generated from our proposed transformations. E,C, and N correspond to the NLI labels Entailment, Contradiction, and Neutral respectively.

her mother". The second approach adds diversity to our synthetically generated PHL triplets[2].

Subject Object Swap (SOS): We swap the subject and object of a sentence to create a contradictory hypothesis. For example, from the sentence "*A clock is standing on top of a concrete pillar*", we create a contradictory hypothesis "*a pillar is standing on top of a concrete clock*".

Negation Introduction (NI): We introduce negation into a sentence to create a contradictory hypothesis. For example, from the sentence "*Empty fog covered streets in the night*", we create hypothesis "*Empty fog did not cover streets in the night*".

Number Substitution (NS): Here, we change numbers (tokens with dependency tag '*nummod*' in the parse tree) in a sentence. For example, changing 'four' to 'seven' in the sentence "*Car has four red lights*" results in a contradictory hypothesis.

Irrelevant Hypothesis (IrH): We sample sentences that have different subjects and objects than the premise sentence. For example, for the premise "*Sign for an ancient monument on the roadside*", we sample "*A man goes to strike a tennis ball*" as a contradictory hypothesis.

4.2.3 Neutral:

The label is neutral when the premise does not provide enough information to classify a PH pair as either entailment or contradiction.

Adding Modifiers (AM): We introduce a *relevant* modifier for noun(s) in premise to generate a neutral hypothesis. For instance, in the sentence "*A car parked near the fence*", we insert modifier 'silver' for the noun 'car' and create hypothesis "*A silver car parked near the fence*". We collect relevant modifiers for nouns by parsing sentences in the premise pool and selecting tokens with dependency tag '*amod*' and POS tag '*ADJ*'[2].

ConceptNet (Con): We add relevant information from ConceptNet (Speer et al., 2017) relations ('AtLocation', 'DefinedAs', etc.) to the premise and create a neutral hypothesis. For instance, from the sentence "*Bunch of bananas are on a table*", we create hypothesis"*Bunch of bananas are on a table at kitchen*" using the 'AtLocation' relation.

Same Subject but Non-Contradictory Verb (SSNCV) : For a premise, we select sentences from the premise pool that have the same subject as the premise, contain additional noun(s) but no contradictory verbs as neutral hypotheses. For instance, for premise "*A small child is sleeping in a bed with a bed cover*", we sample "*A child laying in bed sleeping with a chair near by*" as a hypothesis.

We create more examples by swapping premise and hypothesis of the collected PHL triplets and accordingly change the label. For instance, swapping P and H in **HS, ES, etc.** results in neutral examples, swapping P and H in **AM, Con** results in entailment examples. Furthermore, we note that transformations **ES, HS, PS, SOS, NI** result in PH pairs with high word overlap between premise and hypothesis sentences, whereas, transformation **PA, CV, IrH, SSNCV, etc.** result in PH pairs with low word overlap. In order to add more diversity to the examples, we use composite transformations on the same sentence such as **PA + ES** ($L = E$), **PA + CW** ($L = C$) as shown in Table 1.

4.3 Data Validation

In order to measure the correctness of our procedurally generated PHL triplets, we validate randomly sampled 50 instances for each transformation. We find that nearly all the instances get correct label assignments in case of **PA, HS, PS, NI, NS, IrH, AM** transformations. While transformations **CW, Con, SSNCV** result in a few mislabeled instances. Specifically, **SSNCV** transformation results in the maximum errors (5). Appendix Section B provides examples of such instances. While it is beneficial to have noise-free training examples, doing so would require more human effort and increase the data collection cost. Thus, in this work, we study how well we can do solely using the procedurally generated data without investing human effort in either creating instances or eliminating noise.

5 Training NLI Model

In this section, we describe our approach to develop NLI models for each unsupervised setting. Table 13 (in Appendix) shows sizes of the generated PHL datasets for each setting.

5.1 NPH-Setting

We use the Premise Generation function ($\mathcal{P}$) over raw-text sources, namely, COCO captions, ROC stories, and Wikipedia i.e., $\mathcal{P}$(COCO), $\mathcal{P}$(ROC), and $\mathcal{P}$(Wiki) to compile a set of premises and apply the transformations ($\mathcal{T}$) over them to generate PHL triplets. We then train a transformer-based 3-class classification model (Section 6.1) using the generated PHL triplets for the NLI task.

5.2 P-Setting

In this slightly relaxed unsupervised setting, premises of the training dataset are provided. We directly apply the transformation functions ($\mathcal{T}$) on the given premises and generate PHL triplets. Similar to the NPH setting, a 3-class classification model is trained using the generated PHL triplets.

5.3 PH-Setting

In this setting, unlabeled training data is provided. We present a 2-step approach to develop a model for this setting. In the first step, we create PHL triplets from the premises and train a model using the generated PHL triplets (same as the P-setting). In the second step, we **pseudo-label** the unlabeled PH pairs using the model trained in Step 1.

Here, a naive approach to develop NLI model would be to train using this pseudo-labeled dataset. This approach is limited by confirmation bias i.e overfitting to incorrect pseudo-labels predicted by the model (Arazo et al., 2020). We address this by filtering instances from the pseudo-labeled dataset based on the model's prediction confidence. We use the maximum softmax probability (maxProb) as the confidence measure and select only the instances that have high prediction confidence for training the final NLI model. This approach is based on prior work (Hendrycks and Gimpel, 2017) showing that correctly classified examples tend to have greater maximum softmax probabilities than erroneously classified examples. Furthermore, we investigate two ways of training the final NLI model:

Augmenting with $\mathcal{T}(P)$: Train using the selected pseudo-labeled dataset and the PHL triplets generated in Step 1.

Further Fine-tune P-Model: Further fine-tune the model obtained in Step 1 with the selected pseudo-labeled dataset instead of fine-tuning one from scratch.

6 Experiments

6.1 Experimental Setup

Datasets: We conduct comprehensive experiments with a diverse set of NLI datasets: SNLI (Bowman et al., 2015) (sentence derived from only a single text genre), Multi-NLI (Williams et al., 2018) (sentence derived from multiple text genres), Dialogue NLI (Welleck et al., 2019) (sentences from context of dialogues), and Breaking NLI (Glockner et al., 2018) (adversarial instances).

Model: We use BERT-BASE model (Devlin et al., 2019) with a linear layer on top of [CLS] token representation for training the 3-class classification model. We trained models for 5 epochs with a batch sizes of 32 and a learning rate ranging in $\{1{-}5\}e{-}5$. All experiments are done with Nvidia V100 16GB GPUs.

Baseline Methods: We compare our approach with Multimodal Aligned Contrastive Decoupled learning **(MACD)** (Cui et al., 2020) , Single-modal pre-training model **BERT** (Devlin et al., 2019), Multi-modal pre-training model **LXMERT** (Tan and Bansal, 2019), and **VilBert** (Lu et al., 2019).

6.2 Results

NPH-Setting: We utilize three raw text sources: COCO, ROC, and Wikipedia to compile a premise pool and then generate PHL triplets from those premises. Table 2 shows the accuracy of models in this setting. We use equal number of PHL triplets ($150k$ class-balanced) for training the NLI models. We find that **the model trained on PHL triplets generated from COCO captions as premises outperforms ROC and Wikipedia models on all datasets**. We attribute this superior performance

Model	SNLI	MNLI mat.	MNLI mis.	DNLI	BNLI
BERT*	35.09	-	-	-	-
LXMERT*	39.03	-	-	-	-
VilBert*	43.13	-	-	-	-
$\mathcal{T}(\mathcal{P}(\text{C}))$	64.8	**49.01**	**50.0**	**50.26**	74.73
$\mathcal{T}(\mathcal{P}(\text{R}))$	58.51	45.44	45.93	47.4	67.9
$\mathcal{T}(\mathcal{P}(\text{W}))$	55.06	44.15	44.25	48.48	62.58
$\mathcal{T}(\mathcal{P}(\text{C+R}))$	**65.39**	46.83	46.92	47.95	**77.37**
$\mathcal{T}(\mathcal{P}(\text{C+R+W}))$	65.09	46.63	46.83	44.74	56.11

Table 2: Comparing accuracy of models in the **NPH-setting**. C, R, and W correspond to the premise sources COCO, ROC, and Wikipedia respectively. Results marked with * have been taken from (Cui et al., 2020).

Approach	SNLI	MNLI mat.	MNLI mis.	DNLI	BNLI
BERT*	35.09	-	-	-	-
LXMERT*	39.03	-	-	-	-
VilBert*	43.13	-	-	-	-
MACD*	52.63	-	-	-	-
$\mathcal{T}(\text{SNLI})$	65.72	49.56	50.00	43.27	67.78
$+\mathcal{T}(\mathcal{P}(\text{C}))$	65.36	49.91	49.24	46.25	70.07
$+\mathcal{T}(\mathcal{P}(\text{R}))$	**65.90**	48.53	48.36	44.97	66.43

Table 3: Comparing accuracy of various approaches in the **P-Setting**. Results marked with * have been taken from (Cui et al., 2020). Note that we utilize the premises of the SNLI training dataset only but evaluate on SNLI (in-domain), and MNLI, DNLI, BNLI (out-of-domain).

to the short, simple, and diverse sentences present in COCO that resemble the premises of SNLI that were collected from Flickr30K (Plummer et al., 2015) dataset. In contrast, Wikipedia contains lengthy and compositional sentences resulting in premises that differ from those present in SNLI, MNLI, etc. Furthermore, we find that **combining the PHL triplets of COCO and ROC leads to a slight improvement in performance on SNLI** (65.39%)**, and BNLI** (77.37%) **datasets**.

P-Setting: Cui et al. (2020) presented MACD that performs multi-modal pretraining using COCO and Flick30K caption data for the unsupervised NLI task. It achieves 52.63% on the SNLI dataset. **Our approach outperforms MACD and other single-modal and multi-modal baselines by** ∼13% on SNLI as shown in Table 3. We also experiment by adding PHL triplets generated from COCO and ROC to the training dataset that further improves the accuracy to 65.90% and establish a new state-of-the-art performance in this setting.

Method	Data	SNLI	MNLI mat.	MNLI mis.
From Scratch	MaxProbFilt	66.67	**53.37**	**55.17**
From Scratch	MaxProbFilt+$\mathcal{T}(P)$	**66.75**	50.22	50.37
Finetune P-model	MaxProbFilt	65.60	52.97	53.44

Table 4: Comparing accuracy of our proposed approaches in the **PH-Setting**. Note that the models are trained using PH pairs only from the SNLI train-set but evaluated on MNLI (out-of-domain dataset) also.

PH-Setting: Here, we first pseudo-label the given unlabeled PH pairs using the P-model and then select instances based on the maximum softmax probability (Section 5.3). We refer to this set of selected instances as *MaxProbFilt* dataset. This approach results in accuracy of 66.67% on the SNLI dataset as shown in Table 4. We investigate two more approaches of training the NLI model. In the first approach, we train using *MaxProbFilt* and PHL triplets generated from premises. In the second approach, we further fine-tune the P-model with *MaxProbFilt* dataset. We find that the first approach slightly improves the accuracy to 66.75%. This also represents our best performance across all the unsupervised settings. Furthermore, we observe **improvement in the Out-of-domain datasets also** (53.37% and 55.17% on MNLI matched and mismatched datasets respectively).

6.3 Low-Data Regimes

We also conduct experiments in low-data regimes where a few labeled instances are provided. We select these instances from the training dataset of SNLI/MNLI using the following two strategies:

Random: Here, we randomly select instances from the corresponding training dataset. Further fine-tuning our NPH model with the selected instances consistently achieves higher performance than the models fine-tuned from scratch as shown in Table 5. **With just** 500 **SNLI instances i.e.** ∼ 0.1% **of training dataset, our models achieve** 8.4% **and** 8.32% **higher accuracy on SNLI (in-domain) and MNLI (out-of-domain) respectively.** Furthermore, with 500 MNLI instances, our models achieve 10.37% and 18.07% higher accuracy on MNLI (in-domain) and SNLI (out-of-domain) respectively.

Adversarial: Here, we select those instances from the training dataset on which the NPH model makes incorrect prediction. This is similar to the ad-

Training Dataset	Method	100		200		500		1000		2000	
		SNLI	MNLI	SNLI	MNLI	SNLI	MNLI	SNLI	MNLI	SNLI	MNLI
SNLI	BERT	44.62	37.36	48.97	34.71	58.54	44.01	65.36	37.24	72.51	45.59
	NPH (Random)	**64.82**	**49.72**	**65.06**	**50.48**	**66.97**	**52.33**	**70.61**	**56.75**	**73.7**	**59.0**
	NPH (Adv.)	**68.21**	**51.93**	**69.23**	**56.55**	**70.85**	**58.46**	**73.62**	**59.47**	**74.31**	**60.43**
MNLI	BERT	35.12	36.01	35.14	36.58	46.16	47.1	47.64	56.21	53.68	**63.3**
	NPH (Random)	**63.87**	**52.85**	**63.87**	**53.61**	**64.23**	**57.47**	**65.62**	**60.42**	**66.87**	62.89

Table 5: Comparing performance of various methods on in-domain and out-of-domain datasets in **low-data regimes** (100-2000 training instances). 'BERT' method corresponds to fine-tuning BERT over the provided instances from SNLI/MNLI, 'NPH (Random)' corresponds to further fine-tuning our NPH model with the randomly sampled instances from SNLI/MNLI, 'NPH (Adv.)' corresponds to further fine-tuning our NPH model with the adversarially selected instances from SNLI/MNLI.

Approach	**Δ Accuracy**
NPH model	64.8%
- CV	-5.88%
- CW	-3.07%
- SSNCV	-2.63%
- Neg.	-0.70%
- IrH	-0.50%
- PS	-0.00%

Table 6: **Ablation Study of transformations** in the NPH-Setting. Each row corresponds to the drop in performance on the SNLI dataset when trained without PHL triplets created using that transformation.

versarial data collection strategy (Nie et al., 2020; Kiela et al., 2021) where instances that fool the model are collected. Here, we do not simply fine-tune our NPH model with the adversarial examples as it would lead to catastrophic forgetting (Carpenter and Grossberg, 1988). We tackle this by including 20000 randomly sampled instances from the generated PHL triplets and fine-tune on the combined dataset. **It further takes the performance to 70.85%, 58.46% on SNLI and MNLI respectively with 500 instances.**

6.4 Analysis

Ablation Study: We conduct ablation study to understand the contribution of individual transformations on NLI performance. Table 6 shows the performance drop observed on removing PHL triplets created using a single transformation in the NPH-Setting. We find that **Contradictory Words (CW) and Contradictory Verbs (CV) lead to the maximum drop in performance, 5.88% and 3.07% respectively.** In contrast, Pronoun Substitution (PS) transformation doesn't impact the performance significantly. Note that this does not imply that this transformation is not effective, it means that the evaluation dataset (SNLI) does not contain instances requiring this transformation.

Setting	**Metric**	**Label**		
		C	**E**	**N**
NPH	Precision	0.65	0.71	0.6
	Recall	0.68	0.77	0.51
P	Precision	0.66	0.72	0.58
	Recall	0.67	0.78	0.52
PH	Precision	0.64	0.74	0.60
	Recall	0.73	0.77	0.50

Table 7: **Precision and Recall values** achieved by our models under each unsupervised setting.

NC	**RS**	**SNLI-RS**	**SNLI-NC**
84.22	50.07	58.59	75.39

Table 8: Performance of our NPH model on **Names-Changed (NC) and Roles-Switched (RS) adversarial test sets** (Mitra et al., 2020).

NC and RS Evaluation: We evaluate our model on NER-Changed (NC) and Roles-Switched (RS) datasets presented in (Mitra et al., 2020) that test the ability to distinguish entities and roles. **Our model achieves high performance on these datasets**. Specifically, 84.22% on NC and 75.39% on SNLI-NC as shown in Table 8.

Label-Specific Analysis: Table 7 shows the precision and recall values achieved by our models. We observe that our models perform better on Entailment and Contradiction than Neutral examples. This suggests that **neutral examples are relatively more difficult.** We provide examples of instances where our model makes incorrect predictions and conduct error analysis in Appendix.

7 Conclusion and Discussion

We explored three different settings in unsupervised NLI and proposed a procedural data generation approach that outperformed the existing unsupervised methods by $\sim$13%. Then, we showed that fine-tuning our models with a few human-authored instances leads to a considerable improvement in performance. We also experimented using adversarial instances for this fine-tuning step instead of randomly selected instances and showed that it further improves the performance. Specifically, in presence of just 500 adversarial instances, the proposed method achieved 70.85% accuracy on SNLI, 12.2% higher than the model trained from scratch on the same 500 instances.

This improvement in performance suggests possibility of an alternative data collection strategy that not only results in high-quality data instances but is also resource efficient. Using a model-in-the-loop technique has been shown to be effective for adversarial data collection (Nie et al., 2020; Kiela et al., 2021; Li et al., 2021; Sheng et al., 2021; Arunkumar et al., 2020). In these techniques, a model is first trained on a large dataset and then humans are instructed to create adversarial samples that fool the model into making incorrect predictions. Thus, requiring the crowd-sourcing effort twice. However, in our method, a dataset designer can develop a set of simple functions (or transformations) to procedurally generate training data for the model and can directly instruct humans to create adversarial samples to fool the trained model. This is resource efficient and allows dataset designers to control the quality of their dataset.

Ethical Considerations

We use existing public-domain text corpora such as Wikipedia, ROC Stories, and MS-COCO, and follow the protocol to use and adapt research data to generate our weakly-labeled dataset. We will release the code to generate our dataset. Any bias observed in NLI systems trained using our methods can be attributed to the source data and our transformation functions. However, no particular sociopolitical bias is emphasized or reduced specifically by our methods.

Acknowledgements

We thank the anonymous reviewers for their insightful feedback. This research was supported by DARPA SAIL-ON and DARPA CHESS programs. The views and opinions of the authors expressed herein do not necessarily state or reflect those of the funding agencies and employers.

References

Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. 2020. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Anjana Arunkumar, Swaroop Mishra, Bhavdeep Sachdeva, Chitta Baral, and Chris Bryan. 2020. Real-time visual feedback for educative benchmark creation: A human-and-metric-in-the-loop workflow.

Pratyay Banerjee and Chitta Baral. 2020. Self-supervised knowledge triplet learning for zero-shot question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 151–162, Online. Association for Computational Linguistics.

Pratyay Banerjee, Tejas Gokhale, Yezhou Yang, and Chitta Baral. 2021. WeaQA: Weak supervision via captions for visual question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3420–3435, Online. 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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Gail A. Carpenter and Stephen Grossberg. 1988. The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88.

Wanyun Cui, Guangyu Zheng, and Wei Wang. 2020. Unsupervised natural language inference via decoupled multimodal contrastive learning. In *Proceedings of the 2020 Conference on Empirical Methods* *in Natural Language Processing (EMNLP)*, pages 5511–5520, Online. Association for Computational Linguistics.

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.

Bhuwan Dhingra, Danish Danish, and Dheeraj Rajagopal. 2018. Simple and effective semi-supervised question answering. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 582–587, New Orleans, Louisiana. Association for Computational Linguistics.

Alexander Fabbri, Patrick Ng, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. Template-based question generation from retrieved sentences for improved unsupervised question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4508–4513, Online. Association for Computational Linguistics.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, Melbourne, Australia. Association for Computational Linguistics.

Tejas Gokhale, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2020. Vqa-lol: Visual question answering under the lens of logic. In *European conference on computer vision*, pages 379–396. Springer.

Tejas Gokhale, Abhishek Chaudhary, Pratyay Banerjee, Chitta Baral, and Yezhou Yang. 2021. Semantically distributed robust optimization for vision-and-language inference.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. 2021. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4110–4124, Online. Association for Computational Linguistics.

Patrick Lewis, Ludovic Denoyer, and Sebastian Riedel. 2019. Unsupervised question answering by cloze translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4896–4910, Florence, Italy. Association for Computational Linguistics.

Linjie Li, Jie Lei, Zhe Gan, and Jingjing Liu. 2021. Adversarial vqa: A new benchmark for evaluating the robustness of vqa models. In *International Conference on Computer Vision (ICCV)*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and Larry Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*. European Conference on Computer Vision.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

A. Mitra, Ishan Shrivastava, and Chitta Baral. 2020. Enhancing natural language inference using new and expanded training data sets and new learning models. In *AAAI*.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

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

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

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

Radim Rehurek and Petr Sojka. 2011. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Sasha Sheng, Amanpreet Singh, Vedanuj Goswami, Jose Alberto Lopez Magana, Wojciech Galuba, Devi Parikh, and Douwe Kiela. 2021. Human-adversarial visual question answering.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *NeurIPS*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. 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.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization.

Appendix

A Transformations

In this section, we provide details about the proposed sentence transformations.

A.1 Entailment

Table 9 shows examples of our transformations.

Paraphrasing (PA): It is an effective way of creating entailment examples as the hypothesis which is simply a paraphrased version of the premise is always entailed Furthermore, since the Pegasus tool is trained for abstractive text summarization, it often removes some information from the original sentence while paraphrasing. For instance, a paraphrase of the sentence "*A boy is playing with a red ball*" could be "*Boy is playing with a ball*". This restricts us from using the paraphrased sentence as the premise with the original sentence as the hypothesis as the formed PH pair does not represent an entailment scenario (neutral in this case). It is non-trivial to detect such instances in an automated way. Hence, in order to avoid noisy examples, we only use the original sentence as premise and paraphrased sentences as hypothesis. We also explore back-translation (Sennrich et al., 2016) but it often results in noisy outputs and provides less diversity than the Pegasus tool. Hence, we use only the Pegasus tool for generating paraphrases of sentences.

Extracting Snippets (ES): Here, we provide details of the techniques used for extracting snippets from a text. Note that we use dependency parse tree of the sentence to select/skip the tokens to create the hypothesis.

(i) We skip modifiers (tokens with dependency **amod**) that have no children in the parse tree. For example, from the sentence "*The male surfer is riding a small wave*", we create "*The surfer is riding a small wave*", "*The male surfer is riding a wave*", and "*The surfer is riding a wave*" as entailing hypotheses.

(ii) Similar to the previous technique, we skip adverb modifier (**advmod**). For example, from the sentence "*A very beautiful girl is standing outside the park*", we create an entailment hypothesis "*A beautiful girl is standing outside the park*".

(iii) We skip adjectives that do not have dependency token **conj** and also have 0 children in the parse tree. For example, from the sentence "*A middle-aged man in a beige vest is sleeping on a wooden bench.*", we create "*A middle-aged man in a vest is sleeping on a bench.*".

(iv) In another technique, we select the root token and all the tokens to the left of it. If this results in selection of at least 3 tokens and if one of them is a verb then we consider it to be a valid sentence and use it as an entailing hypothesis. For example, from the sentence "*The male surfer is riding a small wave*", we create "*surfer is riding*".

Hypernym Substitution (HS): Examples of hypernyms:

'alcohol': ['beverage', 'drink']

'apple': ['fruit']

'axe': ['edge tool']

'banana': ['fruit']

etc.

Pronoun Substitution (PS): For words in the list ['man', 'boy', 'guy', 'lord', 'husband', 'father', 'boyfriend', 'son', 'brother', 'grandfather', 'uncle'], we use ('he'/ 'someone'/ 'they', etc.) and for words in the list ['woman', 'girl', 'lady', 'wife', 'mother', 'daughter', 'sister', 'girlfriend', 'grandmother', 'aunt'], we use 'she'/ 'someone'/ 'they', etc.). In other cases, we use the pronoun 'they' or 'someone' or 'somebody'.

Counting (CT): We provide examples of templates we use to create counting hypotheses:

"*There are {count} {hypernym} present*",

"*{count} {hypernym} are present*",

"*Several {hypernym} present*",

"*There are multiple {hypernym} present*",

"*There are more than {count'} {hypernym} present*",

"*There are at least {count'} {hypernym} present*",

etc.

We also substitute the hypernym in the original sentence directly to create hypotheses as shown in Table 9.

A.2 Contradiction

Table 10 shows examples of our transformations.

Contradictory Words (CW): For contradictory adjectives, we collect antonyms from wordnet and for contradictory nouns, we use the function '$most\_similar$' from gensim (Rehurek and Sojka, 2011) library. that returns words close (but distinct) to a given word[2]. For instance, it returns words like 'piano', 'flute', 'saxophone' when given the word 'violin' In order to filter out the inflected forms of the same word or its synonyms from the list returned by $most\_similar$ function, we remove words that have high STS with the given word. This step removes noisy contradictory word pairs to a large extent. Here, we provide examples of contradictory words:

'stove': ['heater']

'cucumber': ['onion', 'carrot', 'melon', 'turnip', 'eggplant', 'watermelon', 'radish']

'motorcycle': ['truck', 'scooter', 'car']

'kitchen': ['bedroom', 'bathroom', 'toilet']

etc.

Contradictory Verb (CV): We provide examples of contradictory verbs:

'stand': ['sprint', 'cycle', 'drive', 'jump', 'sit', etc.]

'play':['sleep', 'cry', 'fight', 'drink', 'hunt', etc.]

'smile': ['cry', 'anger', 'frown', etc.]

etc.

A.3 Neutral

Table 11 shows examples of our transformations.

Adding Modifiers (AM): We provide examples of modifiers collected using our approach:

'metal': ['large', 'circular', 'galvanized','heavy', 'dark', etc.]

'vegetable': ['steamed', 'cruciferous', 'green', 'uncooked', 'raw', etc.]

'park': ['quiet', 'neglected', 'vast', 'square', 'crowded', etc.]

etc.

ConceptNet: We use ConceptNet relations *AtLocation*, *DefinedAs*, etc. and insert the node connected by these relations to the sentence resulting in a neutral hypothesis.

Category	Original Sentence (Premise)	Hypothesis
PA	Fruit and cheese sitting on a black plate.	There is fruit and cheese on a black plate.
ES	person relaxes at home while holding something.	person relaxes while holding something.
HS.	A girl is sitting next to a blood hound.	A girl is sitting next to an animal.
PS	People are walking down a busy city street.	they are walking down a busy city street
CT	A man and woman setup a camera.	Two people setup a camera
Composite	A large elephant is very close to the camera.	elephant is close to the photographic equipment.

Table 9: Illustrative examples of entailment transformations.

Category	Original Sentence (Premise)	Hypothesis
CW-noun	A small bathroom with a sink under a cabinet.	a small kitchen with a sink under a cabinet.
CW-adj	A young man is doing a trick on a surfboard.	A old man is doing a trick on a surfboard.
CV	A couple pose for a picture while standing next to a couch.	A couple sit in a chair on laptops
SOS	A man is flying a kite on the beach.	a beach is flying a kite on the man
NS	Two green traffics lights in a European city.	nine green traffics lights in a European city
IrH.	A flock of sheep grazing in a field.	A man having fun as he glides across the water.
NI.	A boy with gloves on a field throwing a ball.	a boy with gloves on a field not throwing a ball
Composite	A woman holding a baby while a man takes a picture of them	a kid is taking a picture of a male and a baby.

Table 10: Illustrative examples of contradiction transformations.

Category	Original Sentence (Premise)	Hypothesis
AM	two cats are eating next to each other out of the bowl	two cats are eating next to each other out of the same bowl
SSNCV	A man holds an electronic device over his head.	man is taking photo with a small device
FCon	a food plate on a table with a glass.	a food plate on a table with a glass which is made of plastic.
Composite	two dogs running through the snow.	The big dogs are outside.

Table 11: Illustrative examples of neutral transformations.

Trans.	Premise	Hypothesis	Assigned Label	True Label
PS	Two dogs on leashes sniffing each other as people walk in a outdoor market	Two dogs on leashes sniffing each other as they walk in a market	E	N
CT	Adult woman eating slice of pizza while standing next to building	There are 2 humans present	E	C
CW	Meal with meat and vegetables served on table	There is a meal with cheese and vegetables	C	N
SSNCV	A person riding skis down a snowy slope	A person riding skis in a body of water	N	C
SSNCV	A person on a skateboard jumping up into the air	A person jumping up in the air on a snowboard	N	C
CV	A male surfer riding a wave on the ocean	A surfer is surfing in the ocean near some swimmers	C	N

Table 12: Examples of mis-labeled PHL triplets generated by our transformations.

Transformation $\mathcal{T}$	NPH-Setting			P-Setting
	$\mathcal{T}(\mathcal{P}(\text{C}))$	$\mathcal{T}(\mathcal{P}(\text{R}))$	$\mathcal{T}(\mathcal{P}(\text{W}))$	$\mathcal{T}(\text{SNLI})$
Raw Sentences	591	490	600	548
PA	5083	3072	273	475
ES	2365	196	87	516
PS	37	41	137	38
CT	25	8	2	43
Neg.	1175	1175	2053	990
CW	978	119	116	265
CV	1149	63	5	505
NS	73	16	224	91
SOS	428	180	229	76
AM	1048	125	535	327
SSNCV	1363	2	7	405

Table 13: Sizes of PHL triplet datasets generated by our transformations for the unsupervised settings. All numbers are in thousands. C, R, W denote COCO, ROC Stories, and Wikipedia respectively. For P-Setting, we show stats for SNLI dataset. We do not include PH-Setting in this table because we leverage the PHL triplets generated using the P-Setting to solve it as described in Section 5.3.

B Data Validation

Table 12 shows examples of mis-labeled instances generated by our transformations.

C Training NLI Model

Table 13 shows sizes of the generated PHL datasets for each setting.