Unsupervised Natural Language Inference Using PHL Triplet Generation

Neeraj Varshney, Pratyay Banerjee, Tejas Gokhale, Chitta Baral
Arizona State University, USA

Natural Language Inference (NLI)

Given a Premise-Hypothesis pair (P, H), identify whether the hypothesis is true *(Entailment)*, false *(Contradiction)*, or undetermined *(Neutral)* given the premise.

Premise: A man and a lady are talking in the park.

Hypothesis 1: A man and a lady are talking outdoors. Entailment
Hypothesis 2: People are sitting quietly in the park. Contradiction
Hypothesis 3: A father is talking to his daughter in park. Neutral

NLI Training Paradigms

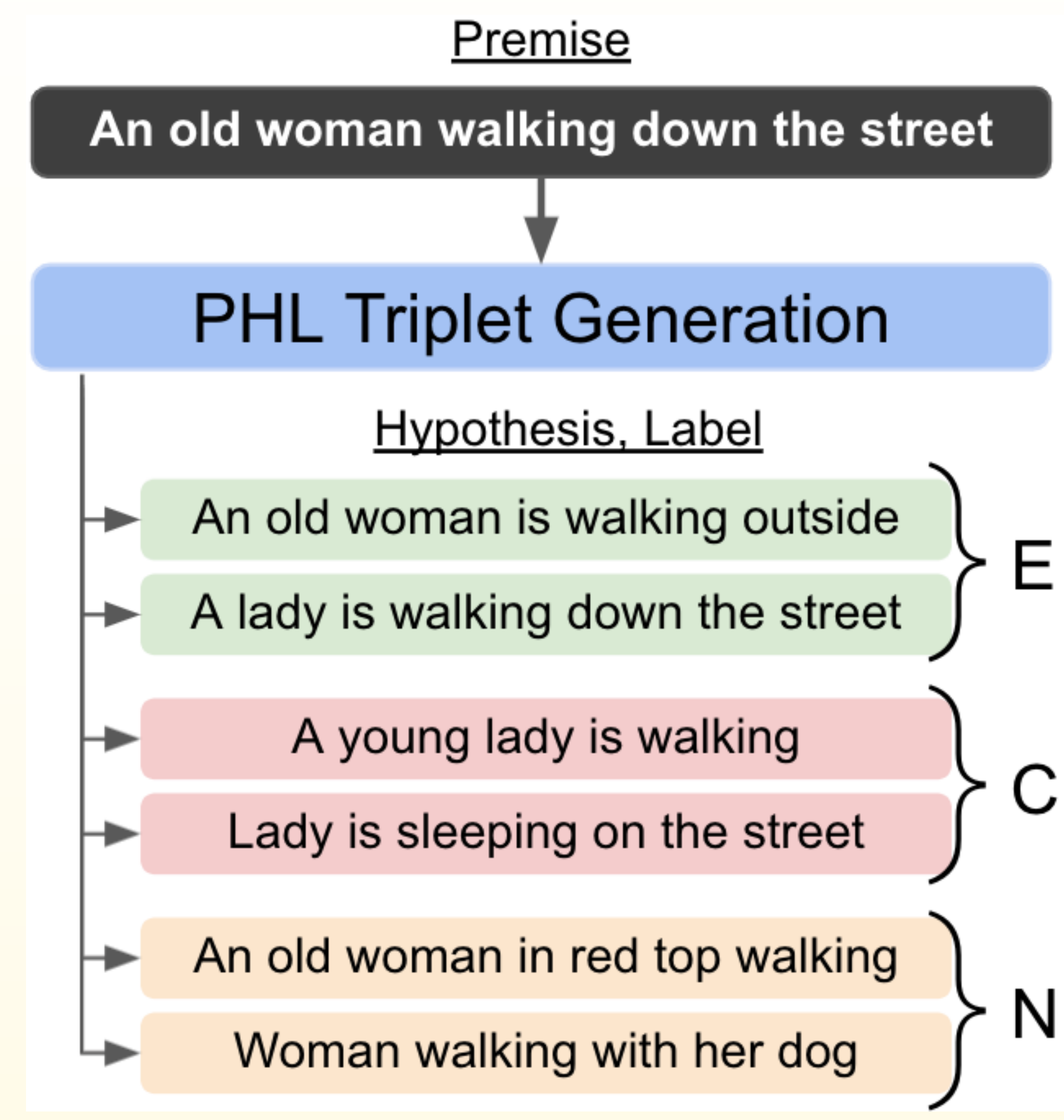
Supervised: Labeled data instances are provided for training.

- Instances are typically collected via crowdsourcing.
- Crowd-workers are given a set of premises and asked to create a hypothesis corresponding to each NLI label.
- Data Collection is **resource intensive** and **time consuming**.

Unsupervised: Labeled instances are not available for training.

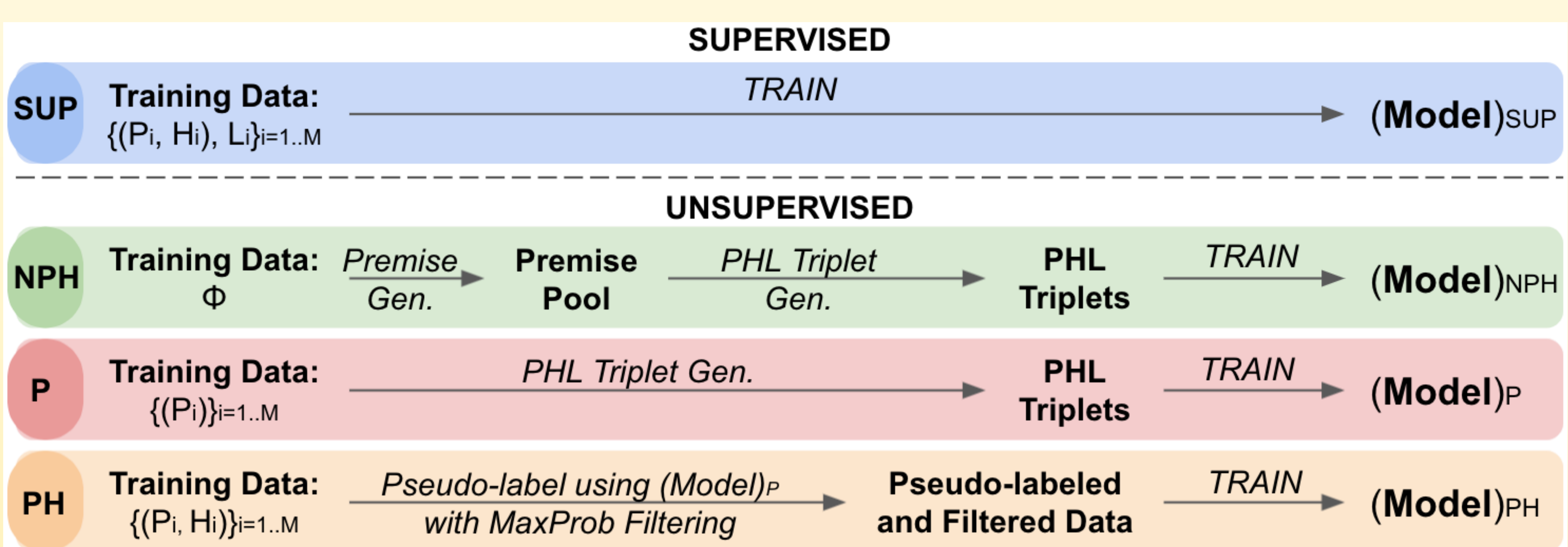
- **PH-Setting** : Unlabeled premise-hypothesis pairs are available.
- **P-Setting** : Only premises (partial inputs) are available.
- **NPH-Setting**: Neither premises nor hypotheses are available.

PHL Triplet Generation for Unsupervised NLI



A sentence is treated as a **premise** and multiple hypotheses conditioned on each label (Entailment- E, Contradiction- C, and Neutral- N) are generated using a set of **sentence transformations**.

Training NLI model for each Setting



For **supervised** setting, the provided labeled dataset is used for training.
For **unsupervised** settings, we procedurally generate PHL triplets to train NLI model.

- In **NPH**-setting, a premise pool is collected from **raw text corpora** such as Wikipedia and then PHL triplets are generated using the premises.
- In **P**-setting, we directly apply **PHL method** on the available premises.
- In **PH** setting, we leverage the P-setting model to **pseudo-label and filter** the provided unlabeled PH pairs for training.

Performance in P-Setting

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

- **MACD** (a baseline method) performs multi-modal pretraining using COCO and Flickr30K captions for the unsupervised NLI task.
- Our proposed approach **outperforms** the previous best method by ~13% on SNLI.
- Adding PHL triplets generated from COCO and ROC to the training dataset **further improves the accuracy** to 65.90%.
- In all three unsupervised settings, our approach achieves **SOTA performance.**

Performance in Low-Data Regimes

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

- In low-data regimes, **a few labeled** instances are provided for training the NLI model.
- Further fine-tuning our model with the provided labeled instances **achieves superior performance** over the model that is trained from scratch on these provided instances.
- Performance improves further on using **adversarial** instances for finetuning. With just 500 adversarial instances (~0.1% of SNLI), our method achieves 70.85% accuracy.
- This suggests possibility of an **alternative data collection strategy** that results in high-quality instances and is resource efficient.
  Specifically, a dataset designer can develop a set of simple transformations to procedurally generate data, train a model using this generated data and instruct humans to create adversarial instances over this model.

Code and Resources

https://github.com/nrjvarshney/unsupervised_NLI