Annotation Artifacts in NLP Tasks

Hao Lu and Hao Gong

April 14, 2020

Roadmap

- What is annotation artifacts?
- Why do annotation artifacts exist?
- What is the effect of annotation artifacts?
- How to deal with annotation artifacts?
- Example: Natural Language Inference

Natural Language Inference

- What is the ultimate goal of NLP?
  Machine can understand language.

- What is understanding?
- How to prove that machine can understand language?

- Let's make things simple. We have two sentences *a* and *b.* Just tell me the relationship between *a* and *b.*

  If *a*, then *b*.

  If *a*, then not *b*.

  If *a*, then could either *b* or not *b*.

Natural Language Inference

- Natural language inference is the task of determining whether a “hypothesis” is true (entailment), false (contradiction), or undetermined (neutral) given a “premise”.

Premise	Label	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction	The man is sleeping.
An older and younger man smiling.	neutral	Two men are smiling and laughing at the cats playing on the floor.
A soccer game with multiple males playing.	entailment	Some men are playing a sport.

Natural Language Inference

- Given a pair of sentences, a premise ***p*** and a hypothesis ***h***

Entailment	h is definitely true given p
Neutral	h might be true given p
Contradiction	h is definitely **not** true given p

- Now we have a well-defined task. So what is next?

  Large amounts of labeled inference data.

Outline

- Datasets for NLI
- Annotation Artifacts (Gururangan et al., 2018)
- Mitigate Artifacts (Belinkov et al., 2019)

Datasets

- SNLI Bowman et al. (2015)
- MultiNLI Williams et al. (2018)

- Crowd workers are presented with a premise ***p*** drawn from some corpus (e.g., image captions), and are required to generate three new sentences (hypotheses) based on ***p***.

Entailment	h is definitely true given p
Neutral	h might be true given p
Contradiction	h is definitely **not** true given p

SNLI

- Website: https://nlp.stanford.edu/projects/snli/

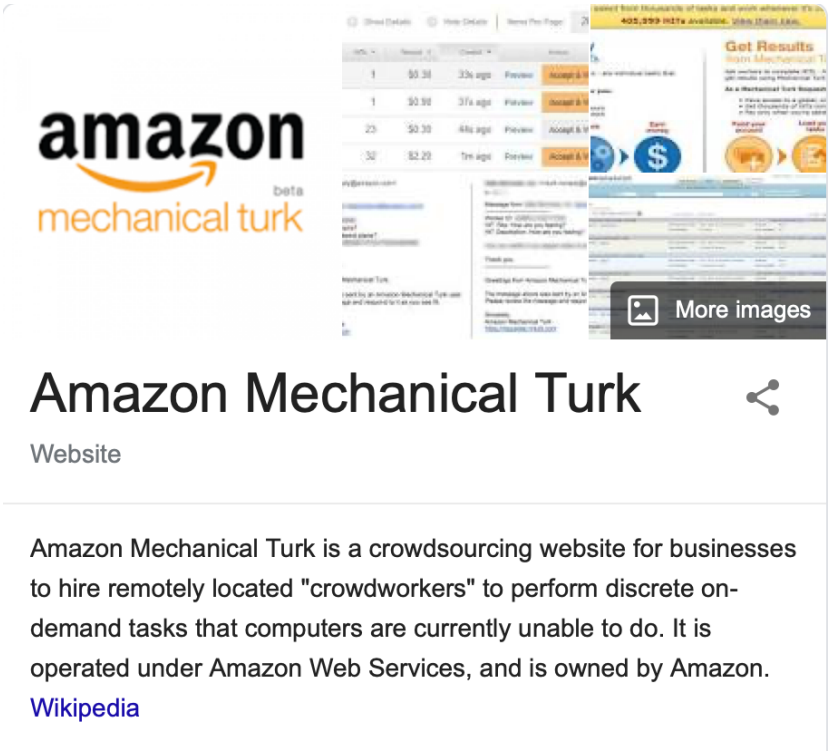
- The SNLI corpus (version 1.0) is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels *entailment*, *contradiction*, and *neutral*, supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE).

SNLI

Text	Judgments	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction C C C C C	The man is sleeping
An older and younger man smiling.	neutral N N E N N	Two men are smiling and laughing at the cats playing on the floor.
A black race car starts up in front of a crowd of people.	contradiction C C C C C	A man is driving down a lonely road.
A soccer game with multiple males playing.	entailment E E E E E	Some men are playing a sport.
A smiling costumed woman is holding an umbrella.	neutral N N E C N	A happy woman in a fairy costume holds an umbrella.

Here are a few example pairs taken from the development portion of the corpus. Each has the judgments of five mechanical turk workers and a consensus judgment.

Data Collection



They used Amazon Mechanical Turk for data collection. Sentences in SNLI are derived from only image captions.

We will show you the caption for a photo. We will not show you the photo. Using only the caption and what you know about the world:

- Write one alternate caption that is **definitely** a **true** description of the photo. *Example: For the caption "Two dogs are running through a field." you could write "There are animals outdoors."*
- Write one alternate caption that **might be** a **true** description of the photo. *Example: For the caption "Two dogs are running through a field." you could write "Some puppies are running to catch a stick."*
- Write one alternate caption that is **definitely** a **false** description of the photo. *Example: For the caption "Two dogs are running through a field." you could write "The pets are sitting on a couch." This is different from the* maybe correct *category because it's impossible for the dogs to be both running and sitting.*

Figure 1: The instructions used on Mechanical Turk for data collection.

MultiNLI

- Website: https://www.nyu.edu/projects/bowman/multinli/

- The Multi-Genre Natural Language Inference (MultiNLI) corpus is a crowd-sourced collection of 433k sentence pairs annotated with textual entailment information.

- The corpus is modeled on the SNLI corpus, but differs in that covers a range of genres of spoken and written text, and supports a distinctive cross-genre generalization evaluation.

MultiNLI

Met my first girlfriend that way.	FACE-TO-FACE **contradiction** C C N C	I didn't meet my first girlfriend until later.
8 million in relief in the form of emergency housing.	GOVERNMENT **neutral** N N N N	The 8 million dollars for emergency housing was still not enough to solve the problem.
Now, as children tend their gardens, they have a new appreciation of their relationship to the land, their cultural heritage, and their community.	LETTERS **neutral** N N N N	All of the children love working in their gardens.
At 8:34, the Boston Center controller received a third transmission from American 11	9/11 **entailment** E E E E	The Boston Center controller got a third transmission from American 11.
I am a lacto-vegetarian.	SLATE **neutral** N N E N	I enjoy eating cheese too much to abstain from dairy.
someone else noticed it and i said well i guess that's true and it was somewhat melodious in other words it wasn't just you know it was really funny	TELEPHONE **contradiction** C C C C	No one noticed and it wasn't funny at all.

Table 1: Randomly chosen examples from the development set of our new corpus, shown with their genre labels, their selected gold labels, and the validation labels (abbreviated E, N, C) assigned by individual annotators.

MultiNLI

- The corpus is derived from ten different genres of written and spoken English, which are collectively meant to approximate the full diversity of ways in which modern standard American English is used.

- *matched* test examples, which are derived from the same sources as those in the training set.

- *mismatched* examples, which do not closely resemble any of those seen at training time.

Annotation Artifacts

- Gururangan et al. (2018)

- They observe that hypotheses generated by this crowdsourcing process contain **artifacts** that can help a classifier detect the correct class ***without*** ever observing the premise.

- Crowd workers adopt heuristics in order to generate hypothesis quickly and efficiently.

- The annotation task produces certain patterns in the data. We call these patterns ***annotation artifacts***. These artifacts are a product of specific annotation strategies and heuristics that crowd workers adopt.

Examples

Premise	A woman selling bamboo sticks talking to two men on a loading dock.
Entailment	There are **at least** three **people** on a loading dock.
Neutral	A woman is selling bamboo sticks **to help provide for her family.**
Contradiction	A woman is **not** taking money for any of her sticks.

Table 1: An instance from SNLI that illustrates the artifacts that arise from the annotation protocol. A common strategy for generating entailed hypotheses is to remove gender or number information. Neutral hypotheses are often constructed by adding a purpose clause. Negations are often introduced to generate contradictions.

fastText

- To prove such artifacts exist, train a model to predict the label of a given hypothesis ***without seeing the premise***.

- fastText https://fasttext.cc (Joulin et al., 2017)
- fastText is an open-source, free, lightweight library that allows users to learn text representations and text classifiers.

- fastText models text as a bag of words and bigrams.

fastText

- For a set of N documents, the fastText is to minimize the negative log-likelihood over the classes:

$$-\frac{1}{N}\sum_{n=1}^{N} y_n \log(f(BAx_n))$$

- Here f is the softmax function to compute the probability distribution over the predefined classes, x_n is the normalized bag of features of the n-th document, y_n the label, *A* and *B* the weight matrices.

Performance of fastText

Model	SNLI	MultiNLI	
		Matched	Mismatched
majority class	34.3	35.4	35.2
`fastText`	**67.0**	**53.9**	**52.3**

Table 2: Performance of a premise-oblivious text classifier on NLI. The MultiNLI benchmark contains two test sets: matched (in-domain examples) and mismatched (out-of-domain examples). A majority baseline is presented for reference.

Characteristics of Annotation Artifacts

Lexical Choice

- Pointwise mutual information (PMI) between each word and class:

$$\text{PMI}(word, class) = \log \frac{p(word, class)}{p(word, \cdot)p(\cdot, class)}$$

- A PMI indicates the use of certain words in certain class.

Top 5 words by PMI

	Entailment		Neutral		Contradiction	
SNLI	outdoors	2.8%	tall	0.7%	nobody	0.1%
	least	0.2%	first	0.6%	sleeping	3.2%
	instrument	0.5%	competition	0.7%	no	1.2%
	outside	8.0%	sad	0.5%	tv	0.4%
	animal	0.7%	favorite	0.4%	cat	1.3%
MNLI	some	1.6%	also	1.4%	never	5.0%
	yes	0.1%	because	4.1%	no	7.6%
	something	0.9%	popular	0.7%	nothing	1.4%
	sometimes	0.2%	many	2.2%	any	4.1%
	various	0.1%	most	1.8%	none	0.1%

Table 4: Top 5 words by PMI($word, class$), along with the proportion of *class* training samples containing *word*. MultiNLI is abbreviated to MNLI.

Entailment

- *animal*, *instrument*, and *outdoors*, which were probably chosen to generalize over more specific premise words such as *dog*, *guitar*, and *beach*.

- Replace exact numbers with approximates (*some*, *at least*, *various*), and to remove explicit gender (*human* and *person* appear lower down the list).

- Some artifacts are specific to the domain, such as *outdoors* and *outside*, which are typical of the personal photo descriptions on which SNLI was built.

Entailment	
outdoors	2.8%
least	0.2%
instrument	0.5%
outside	8.0%
animal	0.7%
some	1.6%
yes	0.1%
something	0.9%
sometimes	0.2%
various	0.1%

Neutral

- Modifiers (*tall*, *sad*, *popular*) and superlatives (*first*, *favorite*, *most*) are affiliated with the neutral class. These modifiers are perhaps a product of a simple strategy for introducing information that is not obviously entailed by the premise, yet plausible.

- Another formulation of neutral hypotheses seems to be through cause and purpose clauses, which increase the prevalence of discourse markers such as *because*.

Neutral	
tall	0.7%
first	0.6%
competition	0.7%
sad	0.5%
favorite	0.4%
also	1.4%
because	4.1%
popular	0.7%
many	2.2%
most	1.8%

Contradiction

- Negation words such as *nobody*, *no*, *never* and *nothing* are strong indicators of contradiction.

- Other (non-negative) words appear to be part of heuristics for contradicting whatever information is displayed in the premise; *sleeping* contradicts any activity, and *naked* (further down the list) contradicts any description of clothing.

Contradiction	
nobody	0.1%
sleeping	3.2%
no	1.2%
tv	0.4%
cat	1.3%
never	5.0%
no	7.6%
nothing	1.4%
any	4.1%
none	0.1%

Examples

Premise	Two dogs are running through a field.
Entailment	There are **animals outdoors**.
Neutral	Some puppies are running **to catch a stick**.
Contradiction	The pets are **sitting on a couch**.

Table 3: The example provided in the annotation guidelines for SNLI. Some of the observed artifacts (bold) can be potentially traced back to phenomena in this specific example.

Sentence Length

- The number of tokens in generated hypotheses is not distributed equally among the different inference classes.

- In SNLI, neutral hypotheses tend to be long, while entailed ones are generally shorter.

- The bias in sentence length may suggest that crowd workers created many entailed hypotheses by simply removing words from the premise.

Sentence Length

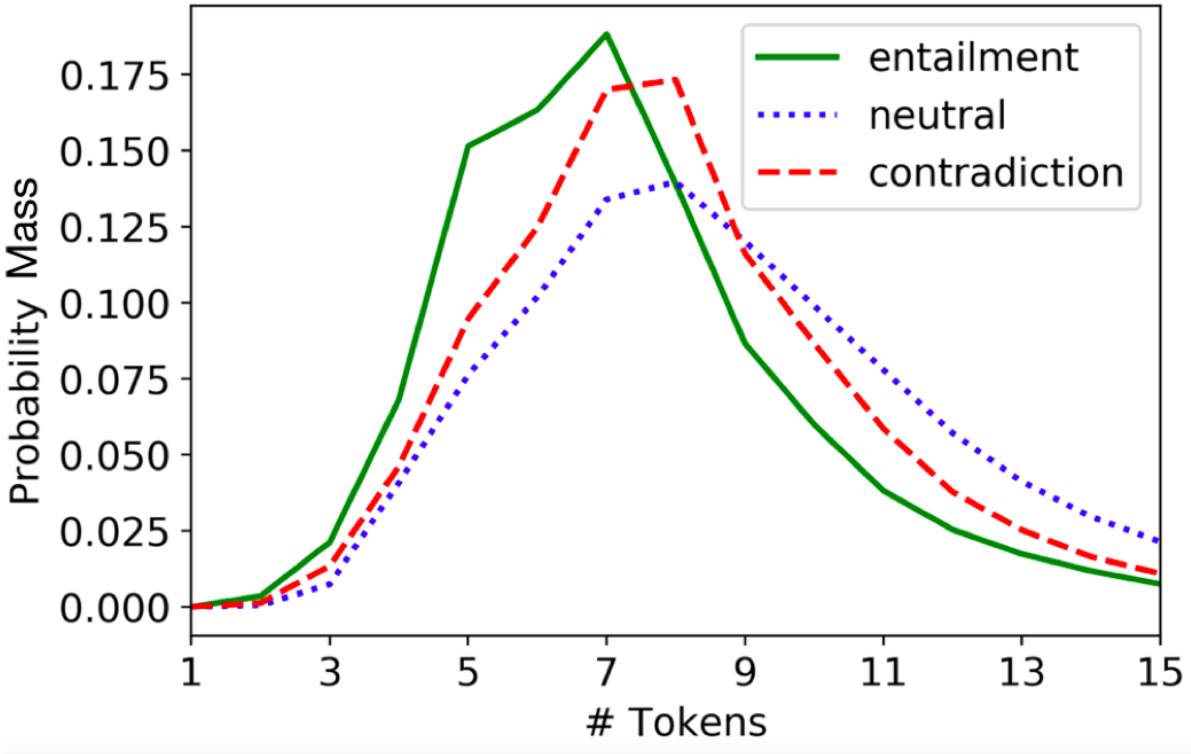


Figure 1: The probability mass function of the hypothesis length in SNLI, by class.

Re-evaluating NLI Models

- Question: To what extent are they “gaming” the task by learning to detect annotation artifacts?

- Partition each NLI test set into two subsets: examples that the premise-oblivious model classified accurately are labeled *Easy*, and those it could not are *Hard*

- train an NLI model on the original training sets (SNLI, MultiNLI), and evaluate on the full test set, the *Hard* test set, and the *Easy* test set.

Results

Model	SNLI			MultiNLI Matched			MultiNLI Mismatched		
	Full	*Hard*	*Easy*	*Full*	*Hard*	*Easy*	*Full*	*Hard*	*Easy*
DAM	84.7	69.4	92.4	72.0	55.8	85.3	72.1	56.2	85.7
ESIM	85.8	71.3	92.6	74.1	59.3	86.2	73.1	58.9	85.2
DIIN	86.5	72.7	93.4	77.0	64.1	87.6	76.5	64.4	86.8

Table 5: Performance of high-performing NLI models on the full, *Hard*, and *Easy* NLI test sets.

This result implies that the ability of NLI models to recognize textual entailment is lower than previously perceived, and that such models rely heavily on annotation artifacts in the hypothesis to make their predictions.

Improvement

- Question: Is that possible to select a set of NLI training and test samples which do not contain easy-to-exploit artifacts?

- For example, filter *Easy* examples from the training set, retaining only *Hard* examples.

- However, after removing the *Easy* examples, *Hard* examples might not necessarily be artifact-free; also *Easy* examples contain important inference, and removing these examples may hinder the model from learning such phenomena (the word “animal” is indeed a hypernym of “dog”).

Improvement

- Importantly, artifacts do not render any particular example *incorrect*; they are a problem with the sample distribution, which is skewed toward certain kinds of class.

- Therefore, a better solution might not eliminate the artifacts altogether, but rather balance them across labels.

Discussion

- Many datasets contain annotation artifacts.
  - SICK dataset (Marelli et al., 2014)
    - negation, word overlap, and hypernym relations
    - highly predictive of entailment classes
  - CNN/DailyMail dataset (Chen et al., 2016)
    - Applyed automatic tools for annotation
  - relation inference benchmark (Zeichner et al., 2012)
  - ROC stories cloze task (Schwartz et al., 2017 and Cai et al., 2017)
    - trained on the endings alone, and not the story prefix, to yield state-of-the-art results

Discussion

- Supervised models leverage annotation artifacts.
  - state-of-the-art visual question answering (Agrawal et al., 2016; Jabri et al., 2016; Goyal et al., 2017) systems leverage annotation biases in the dataset.

- supervised models will exploit shortcuts in the data for gaming the benchmark, if such exist.

- Annotation artifacts inflate model performance.
  - SQuAD (Rajpurkar et al., 2016) drops drastically by introducing simple adversarial sentences

Don’t Take the Premise for Granted: Mitigating Artifacts in Natural Language Inference (Yonatan Belinkov, ACL 2019)

- Two robust methods to deal with biases in NLI datasets.
- An empirical evaluation of the methods on synthetic & real datasets.
- An extensive analysis of the effects of the methods on handling bias.

Overview

- **Literature Review**
- Motivation
- Proposed Methods
- Experiments and Results
- Further Analysis
- Conclusion

Literature Review

- (Sharma et al., 2018) constructed new datasets. Costly, other artifacts
- (Gururangan et al., 2018) filtering “easy” examples. New artifacts.
- (Glockner et al., 2018) Created new datasets. Limited by scale and diversity

Overview

- Literature Review
- **Motivation**
- Proposed Methods
- Experiments and Results
- Further Analysis
- Conclusion

Motivation

- Probabilistic NLI model

$$p_\theta(y|P,H)$$

H- hypothesis sentence P – premise statement y – inference label

- Problem: H may contains information about y and hinder generalizing to different datasets

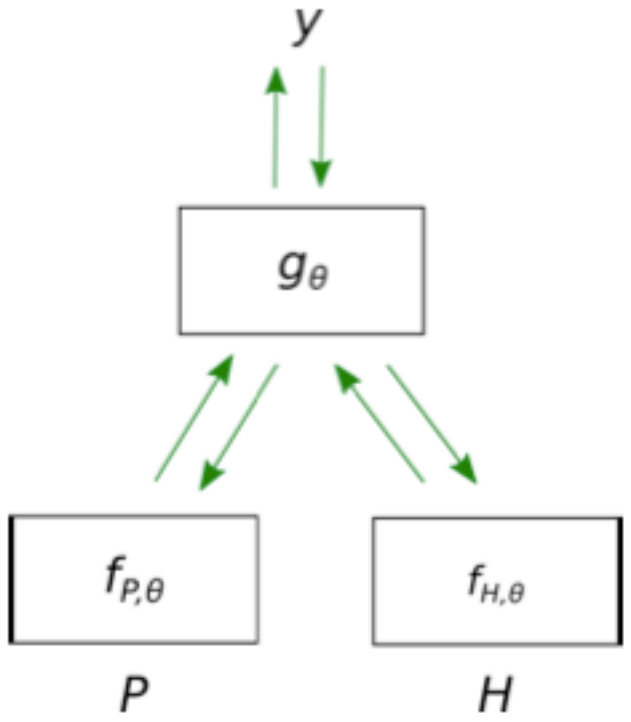
Baseline Model

- InferSent (Conneau et al. 2017)

- Two encoders $f_{P,\theta}$ and $f_{H,\theta}$ to encode premise
- and hypothesis respectively.

- Label classifier for inference g_θ

$$p_\theta(\cdot|P,H) = g_\theta(f_{P,\theta}(P), f_{H,\theta}(H))$$



Synthetic Dataset

- Dataset A

$$(a, a) \rightarrow \textsc{True} \qquad (a, b) \rightarrow \textsc{False}$$
$$(b, b) \rightarrow \textsc{True} \qquad (b, a) \rightarrow \textsc{False}$$

- Dataset B (with artifact)

$$(a, ac) \rightarrow \textsc{True} \qquad (a, b) \rightarrow \textsc{False}$$
$$(b, bc) \rightarrow \textsc{True} \qquad (b, a) \rightarrow \textsc{False}$$

Motivation

- A model that maximizes $p_\theta(y|P,H)$ can easily detect the presence/absence of c in H, ignoring more general pattern

- How about maximizing $p_\theta(P|H,y)$?

- This objective cannot be fooled by the hypothesis-only features, and it requires taking the premise into account

Overview

- Literature Review
- Motivation
- **Proposed Methods**
- Experiments and Results
- Further Analysis
- Conclusion

Training Methods

- By Bayes' Rule,

$$\log p(P \mid y, H) = \log \frac{p_\theta(y \mid P, H) p(P \mid H)}{p(y \mid H)}$$

- Assume $p(P|H)$ is a fixed constant (lacking y, P and H are independent and drawn at random)

Improve Inference

Discourage use of artifacts

- Only need to maximize $\log p_\theta(y|P, H) - \log p(y|H)$

Method1: Hypothesis-only Classifier

- Two encoders $f_{P,\theta}$ and $f_{H,\theta}$ and the inference label classifier g_θ have the same structure as the baseline model

$$p_\theta(\cdot|P,H) = g_\theta(f_{P,\theta}(P), f_{H,\theta}(H))$$

- Additional parameters ϕ for hypothesis-only classifier

$$p_{\phi,\theta}(\cdot|H) = g_\phi(f_{H,\theta}(H))$$

$$p_\theta(\cdot|P,H) = g_\theta(f_{P,\theta}(P), f_{H,\theta}(H))$$ Inference Classifier

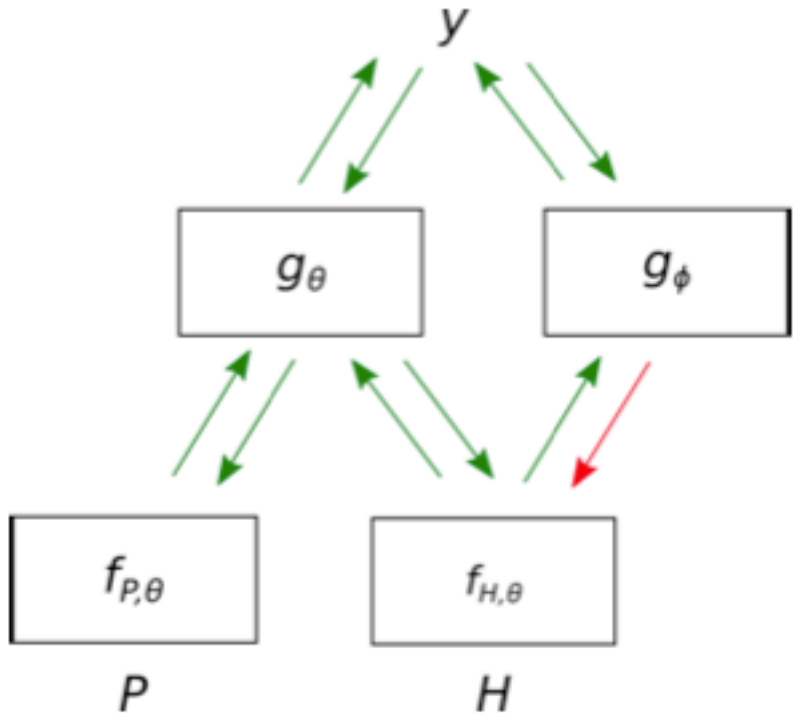
$$p_{\phi,\theta}(\cdot|H) = g_\phi(f_{H,\theta}(H))$$ Hypothesis-only Classifier

$$\max_\theta L_1(\theta) = \log p_\theta(y \mid P,H) - \alpha \log p_{\phi,\theta}(y \mid H)$$

$$\max_\phi L_2(\phi) = \beta \log p_{\phi,\theta}(y \mid H)$$

$$\max_\theta L_1(\theta) = \log p_\theta(y|P,H) + \alpha(-\log p_{\phi,\theta}(y|H))$$

$$\min_\phi L_2'(\phi) = \beta(-\log_{\phi,\theta}(y|H))$$

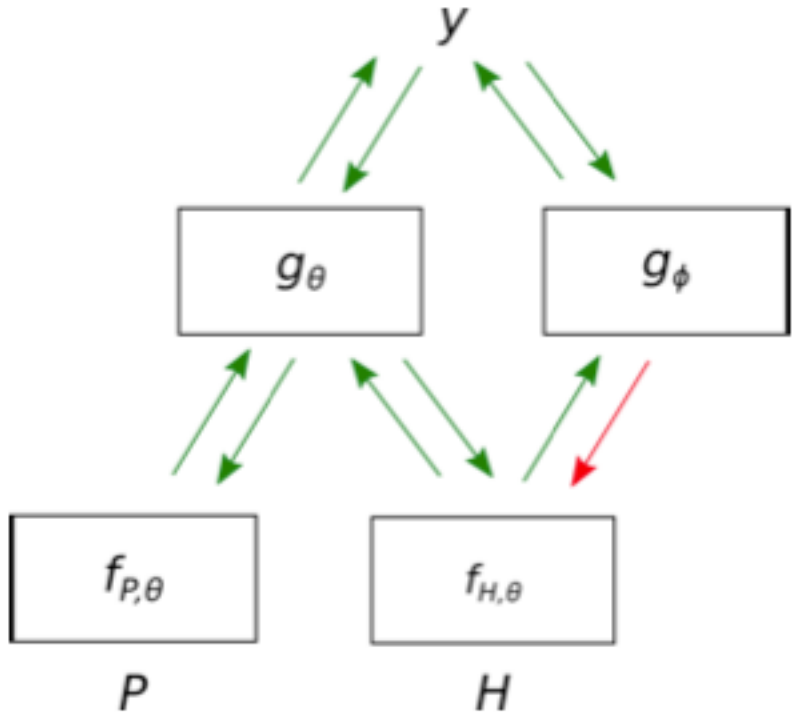


Implementation

- Gradient reversal layer (Ganin et al., 2015) to do

$$\max_{\phi} \min_{\theta} \beta \log p_{\phi,\theta}(y|H)$$

- During backpropagation, first pass the gradients through the hypothesis-only classifier g_{Φ} then reverse the gradient to the hypothesis encoder $g_{H,\theta}$



Implementation

- Add pseudo-function R_λ such that

$$R_\lambda(\mathbf{x}) = \mathbf{x}$$
$$\frac{dR_\lambda}{d\mathbf{x}} = -\lambda\mathbf{I}$$

```
class GradReverse(Function):

    def __init__(self, lambd=1.0):
        self.lambd = lambd

    def forward(self, x):
        return x.view_as(x)

    def backward(self, grad_output):
        return (grad_output * -self.lambd)

def grad_reverse(x, lambd=1.0):
    return GradReverse(lambd)(x)
```

Question on Method 1?

Method2: Negative Sampling

- Recall we wanted to maximize

$$\log p_\theta(y|P,H) - \log p(y|H)$$

$$\begin{aligned} -\log p(y \mid H) &= -\log \sum_{P'} p(P' \mid H) p(y \mid P', H) \\ &= -\log \mathbb{E}_{P'} p(y \mid P', H) \\ &\geq -\mathbb{E}_{P'} \log p(y \mid P', H), \end{aligned}$$

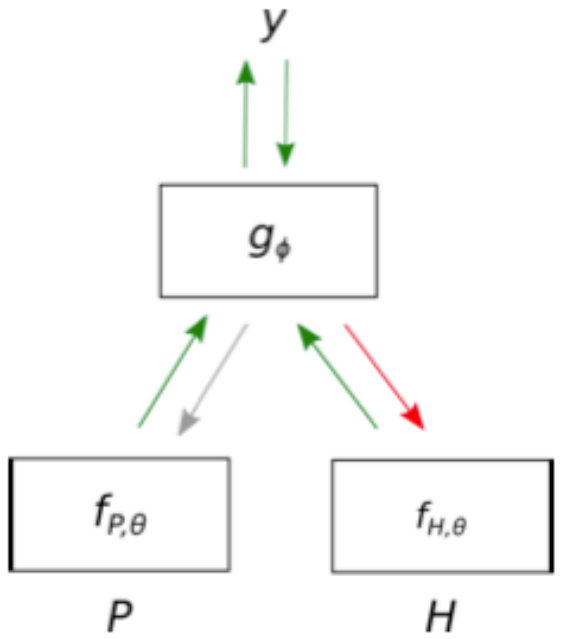
Jensen's Inequality

$$\mathbb{E}_{P'} \log p(y|P', H)$$

- P' is sampled uniformly from other training examples
- Its frequency is controlled by α, similar to Method 1
- $\log(y|P', H)$ is parameterized the same as $\log(y|P, H)$

$$\max_{\theta} L_1 = (1-\alpha) \log p_{\theta,\phi}(y|P, H) - \alpha \log p_{\theta,\phi}(y|P', H)$$

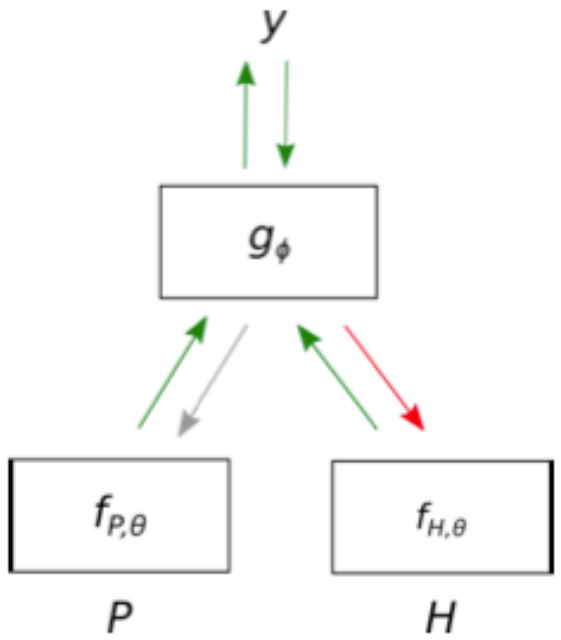
$$\max_{\phi} L_2(\phi) = \beta \log p_{\theta,\phi}(y|P', H)$$



Tricks

- Block the gradients to premise encoder when training with random premise P', because attempting to unlearn only hypothesis biases

- Gradient reversal layer again.



Question on Method 2?

Overview

- Literature Review
- Motivation
- Proposed Methods
- **Experiments and Results**
- Further Analysis
- Conclusion

Experiments

- Synthetic dataset
- Common NLI datasets

Synthetic Dataset

- Test set

$$(a, a) \rightarrow \textsc{True} \qquad (a, b) \rightarrow \textsc{False}$$
$$(b, b) \rightarrow \textsc{True} \qquad (b, a) \rightarrow \textsc{False}$$

- Training set

$$(a, ac) \rightarrow \textsc{True} \qquad (a, b) \rightarrow \textsc{False}$$
$$(b, bc) \rightarrow \textsc{True} \qquad (b, a) \rightarrow \textsc{False}$$

β	α 0.1	0.25	0.5	1	2.5	5
0.1	50	50	50	50	50	50
0.5	50	50	50	50	50	50
1	50	50	50	50	50	50
1.5	50	50	50	50	50	100
2	50	50	50	50	100	100
2.5	50	50	100	75	100	100
3	50	100	100	100	100	100
3.5	100	100	100	100	100	100
4	100	100	100	100	100	100
5	100	100	100	100	100	100
10	100	100	100	100	100	100
20	100	100	100	100	100	100

(a) Method 1

β	α 0.1	0.25	0.5	0.75	1
0.1	50	50	50	50	50
0.5	50	50	50	50	50
1	50	50	50	50	50
1.5	50	50	50	50	50
2	50	50	50	50	50
2.5	50	50	50	50	50
3	50	50	100	50	50
3.5	50	50	100	50	50
4	50	100	100	50	50
5	50	50	100	100	50*
10	75	100	100	100	50*
20	100	100	100	50*	50*

(b) Method 2

- α – weight of second goal / negative sample frequency
- β – adversary learning rate

Common NLI datasets

- Training set
  - SNLI (Bowman et al., 2015)
- Dev set / Test set (target datasets)
  - SCITAIL (Khot et al., 2018)
  - ADD-ONE-RTE (Pavlick et al., 2016)
  - JOCI (Zhang et al., 2017)
  - MPE (Lai et al., 2017)
  - DPR (Rahman et al., 2012)
  - MNLI matched (William et al., 2018)
  - FN+ (Pavlick et al., 2016)
  - MNLI mismatched (William et al., 2018)
  - SICK (Marelli et al., 2014)
  - GLUE (Wang et al., 2018)
  - SPR (Reisinger et al., 2015)
  - SNLI-hard (Bowman et al., 2015)

Model Structure

- Baseline model: InferSent (Conneau et al., 2017)
  - GloVe word embeddings (Pennington et al., 2014)
  - Separate BiLSTMs as premise and hypothesis encoders $f_{P,\theta}$ and $f_{H,\theta}$
  - Vector representations are concatenated, subtracted and multiplied element-wise
  - Followed by an MLP with one hidden layer

- Proposed methods used same structure to represent and combine sentences

Performances

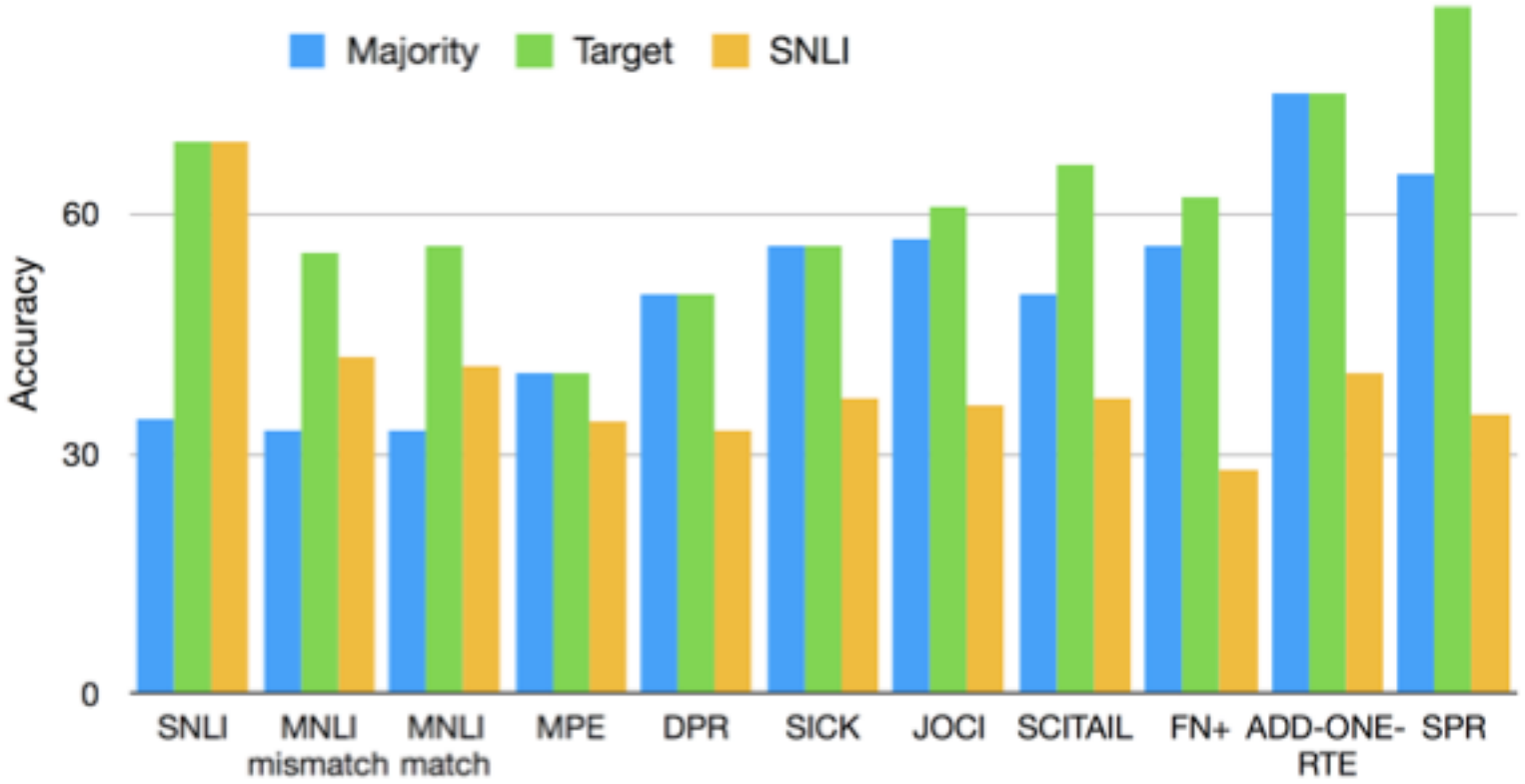
	Test On Target Dataset			Test On SNLI	
Target Test Dataset	Baseline	Δ Method 1	Δ Method 2	Δ Method 1	Δ Method 2
SCITAIL	58.14	-0.47	-7.06	-0.18	-9.06
ADD-ONE-RTE	66.15	0.00	17.31	-2.29	-49.63
JOCI	41.50	0.24	-1.87	-0.44	-5.92
MPE	57.65	0.45	-5.30	-0.57	-0.54
DPR	49.86	1.10	-0.45	-0.73	-7.81
MNLI matched	45.86	1.38	-2.10	-1.25	-8.93
FN+	50.87	1.61	6.16	-1.94	-0.44
MNLI mismatched	47.57	1.67	-3.91	-1.25	-8.93
SICK	25.64	1.80	31.11	-0.57	-8.93
GLUE	38.50	1.99	4.71	-1.25	-8.93
SPR	52.48	6.51	12.94	-1.76	-14.01
SNLI-hard	68.02	-1.75	-12.42		

Overview

- Literature Review
- Motivation
- Proposed Methods
- Experiments and Results
- **Further Analysis**
- Conclusion

Interplay with known biases

- Training set SNLI is known to have biases
- Conjecture: their methods provide the most benefit when a target dataset has no hypothesis-only biases or different biases
- Need a measurement of the bias difference
- For each target set, compare the three models
  - Hypothesis-only classifier trained on SNLI and tested on the target dataset
  - Majority baseline of most frequent class in the target set
  - Hypothesis-only classifier trained and tested on the target dataset



- With different biases: SPR -> Huge improvement
- Similar biases: MNLI -> mild improvement
- No/little biases: ADD-ONE-RTE, SICK, MPE -> Medium improvement

Stronger hyper-parameters

- Synthetic dataset experiment showed larger α and β
  - Hurt performance on the original dataset
  - Generalize better on the target dataset

- Different search region
  - {0.05, 0.1, 0.2, 0.4, 0.8, 1.0} -> {1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0}

Dataset	Base	Method 1	Δ
JOCI	41.50	39.29	-2.21
SNLI	84.22	82.40	-1.82
DPR	49.86	49.41	-0.45
MNLI matched	45.86	46.12	0.26
MNLI mismatched	47.57	48.19	0.62
MPE	57.65	58.60	0.95
SCITAIL	58.14	60.82	2.68
ADD-ONE-RTE	66.15	68.99	2.84
GLUE	38.50	41.58	3.08
FN+	50.87	56.31	5.44
SPR	52.48	58.68	6.20
SICK	25.64	36.59	10.95
SNLI-hard	68.02	63.81	-4.21

Fine-tuning on target datasets

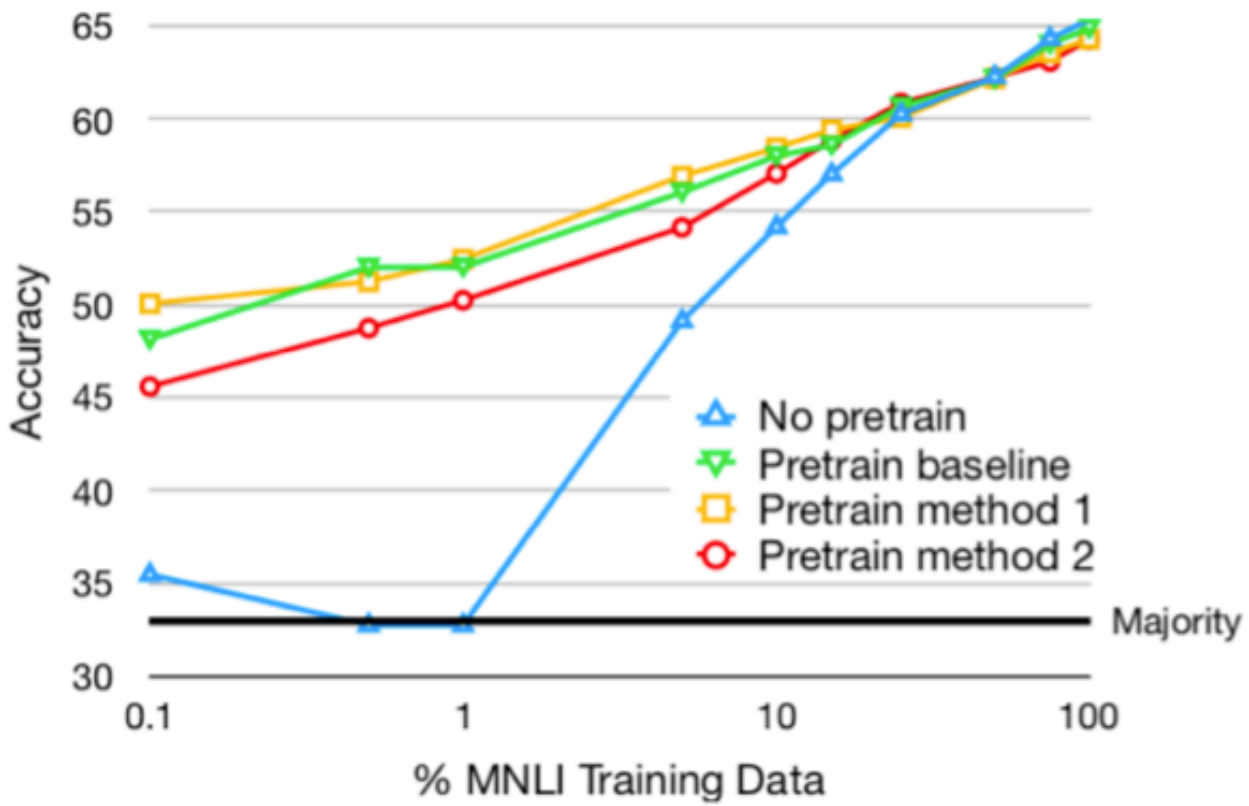
- Previous experiment only used target datasets to tune hyper-parameters

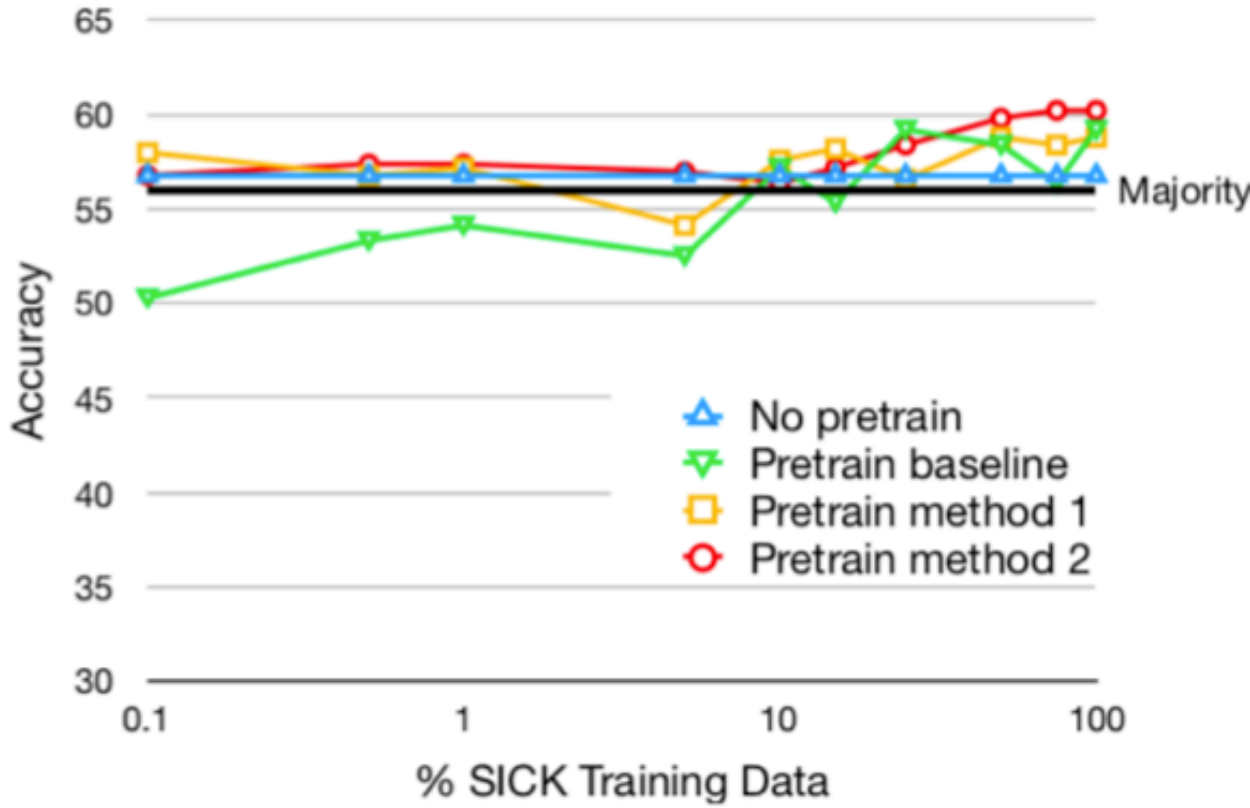
- What if also using target datasets to update parameters?

- Pick two target datasets
  - SICK – where their methods resulted in good gains
  - MNLI – which has large training set

- Train four models
  - Baseline, Method 1 and Method 2 pretrained on SNLI and fine-tuned on the target dataset
  - Baseline model trained only on the target dataset

- With varying target set sizes


65
60
55
50
45
40
35
30
Accuracy
0.1
1
10
100
% MNLI Training Data
No pretrain
Pretrain baseline
Pretrain method 1
Pretrain method 2
Majority


65
60
55
50
45
40
35
30
Accuracy
0.1
1
10
100
% SICK Training Data
No pretrain
Pretrain baseline
Pretrain method 1
Pretrain method 2
Majority

Overview

- Literature Review
- Motivation
- Proposed Methods
- Experiments and Results
- Further Analysis
- **Conclusion**

Related topic:
Biases and artifacts in NLU dataset

- ROC Story (Schwartz et al., 2017a; Cai et al., 2017)
  - Pick a coherent ending for a story without looking at the story
- Reading Comprehension (Kaushik, et al., 2018)
  - Answer questions only looking at the last sentence of the passage
- Visual Question Answering
  - Zhang et al., 2016; Kafle & Kanan, 2016, 2017; Goyal et al., 2017; Agrawal et al., 2017

Example

- (Kaushik et al., 2018) How Much *Reading* Does Reading Comprehension Require? A Critical Investigation of Popular Benchmarks
- Original task: (passage, question) -> answer

	bAbI Tasks 1-10									
Dataset	1	2	3	4	5	6	7	8	9	10
True dataset	**100%**	**100%**	39%	**100%**	**99%**	**100%**	**94%**	**97%**	**99%**	**98%**
Question only	18%	17%	22%	22%	34%	50%	48%	34%	64%	44%
Passage only	53%	86%	**60%**	59%	31%	48%	85%	79%	63%	47%
$\Delta(min)$	−47	−14	+21	−41	−65	−52	−9	−18	−35	−51

Related topics

- Transferability across NLI datasets

- Improving model robustness
  - Adversarial examples
  - Domain-adversarial neural networks (Ganin et al., 2015)
  - Removing biases from the representations

Takeaways

- Learn inference and discourage use of hypothesis-only biases simultaneously by balancing two terms in objective function
- Tradeoff the two goals with two hyperparameters
- Quantify the difference between the biases of two NLI datasets

Questions?