Simple but Effective Techniques to Reduce Dataset Biases

Rabeeh Karimi$^{1,2}$, James Henderson$^1$

1. Idiap Research Institute
2. École Polytechnique Fédérale de Lausanne (EPFL)

November 13th, 2019
Overview

1. Introduction
2. Our Model
3. Experimental Results
4. Takeaways
Example: Biases in Visual Question Answering

Q: What color is the grass?
A: Green

Q: What color is the banana?
A: Yellow

Q: What color is the skye?
A: Blue
A VQA system that fails to ground questions in image content would likely perform poorly in real-world settings.

Q: What color is the banana?
A: Yellow

Q: What color is the banana?
A: Yellow ✗
Example: Natural language Inference (NLI)

Premise

- SNLI (Bowman et. al, 2015) 570 K
- MultiNLI (Williams et. al., 2017) 433 K
- SNLI premises are Flickr captions.
- MultiNLI premises are collected from diverse genre.
- Hypotheses are crowdsourced-generated.
While NLI is a hard task, the community has made significant progress on large-scale NLI datasets.

Significant NLI Progress, almost human performance

- Dagan et al., 2005
- Bowman et al., 2015
- Williams et al., 2018
- Wang et al., 2018
- Conneau et al., 2017
- Lin et al., 2018
- Devlin et al, 2019
- Liu et al, 2019
- Yang et al., 2019

(among others)
Kicking out premises ...

Figure: Figure from [GSL+18]

- Over 50% of NLI examples can be correctly classified without ever observing the premise!
Biases in NLI - Patterns in the hypothesis

A group of female athletes are gathered together and excited.

Purpose clauses

They are gathered together because they are working together.

Neutral

Some men and boys are playing frisbee in a grassy area.

Generalization

People play frisbee outdoors.

Entailment

A man with a black cap is looking at the street.

Negation

Nobody wears a cap.

Contradiction
Can we avoid biases?

- This is hard to avoid biases during the creation of datasets.
- Constructing new datasets, specially in large-scale is costly and still could results in other artifacts.
- This is important to develop techniques which to prevent models from using known biases to be able to leverage existing datasets.
- Goal: train robust model to improve their generalization performance on evaluation phase, where typical biases observed in the training data do not exist.
Overview of Our Model

Figure: An illustration of our debiasing strategies on NLI. Solid arrows show the flow of input information, and dotted arrows show the back-propagation flow of error. Blue highlighted modules are removed after training. At test time, only the predictions of the base model $f_M$ are used.
Steps to make the models robust to biases ...

- Identify the biases
- Train the bias-only branch \( f_B \).
- Compute the combination of the two models \( f_C \)
  - Motivate the base model to learn different strategies than the ones used by the bias-only branch \( f_B \).
- Remove the bias-only classifier and use the predictions of the base model.
Step 1: Bias-only Model

- Fortunately often times, we know what are the domain-specific biases
- Train the bias-only model using only biased features

Hypothesis

A woman is not taking money for any of her sticks.
A boy with no shirt on throws rocks.
A man is asleep and dreaming while sitting on a bench.
A naked man is posing on a ski board with snow in the background.

Labels

f_B

Contradiction
Step 2: Training a Robust Model

- Classical learning strategy:
  \[
  \mathcal{L}(\theta_M) = -\frac{1}{N} \sum_{i=1}^{N} a_i \log(\text{softmax}(f_M(x_i))),
  \]  
  (1)

- Down-weighting the impact of the biased examples so that the model focuses on learning hard examples.
- Avoid major gradient updates from trivial predictions.
- Ensemble techniques:
  - Method 1: Product of experts [Hin02]
  - Method 2: RUBI [CDBy+19]
- Weight the loss of the base model depending on the accuracy of the bias-only model
  - Method 3: Debiased Focal Loss
Method 1: Product of Experts

- Combine multiple probabilistic models of the same data by multiplying the probabilities together and then renormalizing.
- Combine the bias-only and base model predictions:

\[ f_C(x_i, x^b_i) = f_B(x^b_i) \odot f_M(x_i), \]  

(2)

where \( x^b_i \) is the biased features, and \( x_i \) is the whole input.
- Update the model parameters based on the cross-entropy loss of the combined classifier.
Apply a sigmoid function to the bias-only model's predictions to obtain a mask containing an importance weight between 0 and 1 for each possible label.

\[
f_C(x_i, x_i^b) = f_M(x_i) \odot \sigma(f_B(x_i^b)),
\]
Method 2: RUBi [CDBy+19]

Figure: Detailed illustration of the RUBi impact on the learning [CDBy+19].

(a) Classical learning strategy

(b) RUBi learning strategy
Debiased Focal Loss

- Explicitly modulating the loss depending on the accuracy of the bias-only model:

\[
\mathcal{L}_C(\theta_M; \theta_B) = -\frac{1}{N} \sum_{i=1}^{N} a_i (1 - f_B(x_i^b))^{\gamma} \log(f_M(x_i)),
\]

(4)

**Observations**

- When the example is unbiased, and bias-only branch does not do well, \(f_B(x_i^b)\) is small, and the loss remains unaffected.
- As the sample is more biased and \(f_B(x_i^b)\) is closer to 1, the loss for the most biased examples is down-weighted.
Evaluation of Generalization Performance

- We train our models on two large-scale NLI datasets, namely SNLI and MNLI, and FEVER dataset.
- Evaluate performance on the challenging unbiased datasets.

Figure: Figure from [GSL+18]

Figure: Figure from [SJSJSY+19]
Obtaining 9.76 points gain on FEVER symmetric test set, improving the results of prior work by 4.65 points.

Table: Results on FEVER development (Dev) set and FEVER symmetric test set.

<table>
<thead>
<tr>
<th>Debiasing method</th>
<th>Dev</th>
<th>Symmetric test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>85.99</td>
<td>56.49</td>
</tr>
<tr>
<td>RUBI</td>
<td>86.23</td>
<td>57.60</td>
</tr>
<tr>
<td>Debiased Focal Loss</td>
<td>83.07</td>
<td>64.02</td>
</tr>
<tr>
<td>Product of experts</td>
<td>86.46</td>
<td>66.25</td>
</tr>
<tr>
<td>[SJSJSY+19]</td>
<td>84.6</td>
<td>61.6</td>
</tr>
</tbody>
</table>
### Experimental Results - MNLI

Table: Results on MNLI matched (MNLI) and mismatched (MNLI-M) sets.

<table>
<thead>
<tr>
<th>Debiasing Method</th>
<th>MNLI Test</th>
<th>MNLI Hard</th>
<th>MNLI-M Test</th>
<th>MNLI-M Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>84.11</td>
<td>75.88</td>
<td>83.51</td>
<td>75.75</td>
</tr>
<tr>
<td>Product of experts</td>
<td>84.11</td>
<td>76.81</td>
<td>83.47</td>
<td>76.83</td>
</tr>
</tbody>
</table>

Table: Results on MNLI matched and HANS datasets

<table>
<thead>
<tr>
<th>Debiasing Method</th>
<th>MNLI</th>
<th>HANS</th>
<th>Constituent</th>
<th>Lexical</th>
<th>Subsequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>83.99</td>
<td>61.10</td>
<td>61.11</td>
<td>68.97</td>
<td>53.21</td>
</tr>
<tr>
<td>RUBI</td>
<td>83.93</td>
<td>60.35</td>
<td>56.51</td>
<td>71.09</td>
<td>53.44</td>
</tr>
<tr>
<td>Debiased Focal Loss</td>
<td>84.33</td>
<td>64.99</td>
<td>62.42</td>
<td>74.45</td>
<td>58.11</td>
</tr>
<tr>
<td>Product of experts</td>
<td>84.04</td>
<td>66.55</td>
<td>64.29</td>
<td>77.61</td>
<td>57.75</td>
</tr>
</tbody>
</table>
Gain of 4.78 points on SNLI hard set.

<table>
<thead>
<tr>
<th>Debiasing method</th>
<th>BERT Test</th>
<th>BERT Hard</th>
<th>InferSent Test</th>
<th>InferSent Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>90.53</td>
<td>80.53</td>
<td>84.24</td>
<td>68.91</td>
</tr>
<tr>
<td>RUBI</td>
<td>90.69</td>
<td>80.62</td>
<td>83.93</td>
<td>69.64</td>
</tr>
<tr>
<td>Debiased Focal Loss</td>
<td>89.57</td>
<td>83.01</td>
<td>73.54</td>
<td>73.05</td>
</tr>
<tr>
<td>Product of experts</td>
<td>90.11</td>
<td>82.15</td>
<td>80.35</td>
<td>73.69</td>
</tr>
<tr>
<td>AdvCls belinkov2019adversarial</td>
<td>-</td>
<td>-</td>
<td>83.56</td>
<td>66.27</td>
</tr>
<tr>
<td>AdvDat belinkov2019adversarial</td>
<td>-</td>
<td>-</td>
<td>78.30</td>
<td>55.60</td>
</tr>
</tbody>
</table>
High performance of neural models could be due to leveraging superficial cues in the data.

This is hard to avoid biases during creation of datasets.

We need to develop methods robust to existing biases.

Let bias-only model capture the biases and we adjust cross-entropy loss to focus learning on the hard examples.

Substantial improvement in the model robustness and better generalization performance.
Thank you. Any questions?

