Analysis methods in NLP: Adversarial training (and testing)

Christopher Potts

Stanford Linguistics

CS224u: Natural language understanding
Overview

Behavioral evaluations

- Adversarial testing
- **Adversarial training and testing**
SWAG: Situations With Adversarial Generations

**SWAG: A Large-Scale Adversarial Dataset for Grounded Commonsense Inference**

Rowan Zellers♦️  Yonatan Bisk♦️  Roy Schwartz♦️ merciless Yejin Choi♦️
♦️Paul G. Allen School of Computer Science & Engineering, University of Washington
◊Allen Institute for Artificial Intelligence

{rowanz,ybisk,roysch,yejin}@cs.washington.edu
https://rowanzellers.com/swag

**HellaSwag: Can a Machine Really Finish Your Sentence?**

Rowan Zellers♦️  Ari Holtzman♦️  Yonatan Bisk♦️  Ali Farhadi♦️ merciless Yejin Choi♦️
♦️Paul G. Allen School of Computer Science & Engineering, University of Washington
◊Allen Institute for Artificial Intelligence

https://rowanzellers.com/hellaswag
SWAG examples

Example

- **Context (given)**: He is throwing darts at a target.
- **Sentence start (given)**: Another man
- **Continuation (predicted)**: throws a dart at the target board.
- **Distractors**:
  1. comes running in and shoots an arrow at a target.
  2. is shown on the side of men.
  3. throws darts at a disk.

Sources

- ActivityNet: 51,439 exs; 203 activity types
- Large Scale Movie Description Challenge: 62,118 exs

Zellers et al. 2018; https://rowanzellers.com/swag/
Adversarial filtering for SWAG

For each example $i$:

1. The mixture creams the butter. Sugar
   a. is added.
   b. is sprinkled on top. [Model incorrect; keep this sample]
   c. is in many foods.

Repeat for some number of iterations.

Zellers et al. 2018
Figure 2: Test accuracy by AF iteration, under the negatives given by A. The accuracy drops from around 60% to close to random chance. For efficiency, the first 100 iterations only use the MLP.
SWAG in the original BERT paper

<table>
<thead>
<tr>
<th>System</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESIM+GloVe</td>
<td>51.9</td>
<td>52.7</td>
</tr>
<tr>
<td>ESIM+ELMo</td>
<td>59.1</td>
<td>59.2</td>
</tr>
<tr>
<td>BERT_BASE</td>
<td>81.6</td>
<td>-</td>
</tr>
<tr>
<td>BERT_LARGE</td>
<td>86.6</td>
<td>86.3</td>
</tr>
<tr>
<td>Human (expert)†</td>
<td>-</td>
<td>85.0</td>
</tr>
<tr>
<td>Human (5 annotations)†</td>
<td>-</td>
<td>88.0</td>
</tr>
</tbody>
</table>

Table 4: SWAG Dev and Test accuracies. Test results were scored against the hidden labels by the SWAG authors. †Human performance is measured with 100 samples, as reported in the SWAG paper.
HellaSWAG

1. ActivityNet retained
2. Large Scale Movie Description Challenge dropped
3. WikiHow data added
4. Adversarial filtering as before, now with more powerful generators and discriminators
5. Human agreement at 94%

In this section, we investigate why SWAG was solved. We focus on BERT, since it is the best underlying workhorse to construct an NLI dataset, as the test set. This contrasts with past work on in-domain data, and then selected using an ensemble is trained to classify endings as adversarial. Last, humans validate the data to remove adversarial endings that seem realistic.

Importantly, AF creates a final dataset that continues until the accuracy of these adversaries converges. This process compares BERT’s performance when trained and evaluated under several versions of SWAG, with the new dataset HellaSwag as comparison. We compare:

- **Ending Only** No context is provided; just the endings.
- **Shuffled** Endings that are individually tokenized, shuffled, and then detokenized.
- **Shuffled+ Ending Only** No context is provided and each ending is shuffled.

To distinguish word usage from absense of context

<table>
<thead>
<tr>
<th>Version</th>
<th>SWAG</th>
<th>HellaSwag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>86.7%</td>
<td>60.4%</td>
</tr>
<tr>
<td>Ending Only</td>
<td>74.8%</td>
<td>41.4%</td>
</tr>
<tr>
<td>Shuffled</td>
<td>77.0%</td>
<td>36.2%</td>
</tr>
<tr>
<td>Shuffled+ Ending Only</td>
<td>60.4%</td>
<td>31.6%</td>
</tr>
</tbody>
</table>

Figure 3: Validation accuracy on SWAG for BERT-Large versus training set size. The baseline (25% accuracy) is random chance. BERT does well given as few as 16 training examples, but requires tens of thousands of examples to approach human performance.

These biases are similar to those in NLI datasets, as found by Gururangan et al. (2018); Poliak et al. (2018).

Figure 4: BERT validation accuracy when trained and evaluated under several versions of SWAG, with the new dataset HellaSwag as comparison. We compare:

- **Ending Only** No context is provided; just the endings.
- **Shuffled** Endings that are individually tokenized, shuffled, and then detokenized.
- **Shuffled+ Ending Only** No context is provided and each ending is shuffled.

### Table 1: Performance of models, evaluated with accuracy (%).
We report results on the full validation and test sets (Overall), as well as results on informative subsets of the data: evaluated on in-domain, versus zero-shot situations, along with performance on the underlying data sources (ActivityNet versus WikiHow). All models substantially underperform humans: the gap is over 45% on in-domain categories, and 50% on zero-shot categories.

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall Val</th>
<th>Overall Test</th>
<th>In-Domain Val</th>
<th>In-Domain Test</th>
<th>Zero-Shot Val</th>
<th>Zero-Shot Test</th>
<th>ActivityNet Val</th>
<th>ActivityNet Test</th>
<th>WikiHow Val</th>
<th>WikiHow Test</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10K</td>
<td>10K</td>
<td>5K</td>
<td>5K</td>
<td>5K</td>
<td>5K</td>
<td>3.2K</td>
<td>3.5K</td>
<td>6.8K</td>
<td>6.5K</td>
</tr>
<tr>
<td>Chance</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fastText</td>
<td>30.9</td>
<td>31.6</td>
<td>33.8</td>
<td>32.9</td>
<td>28.0</td>
<td>30.2</td>
<td>27.7</td>
<td>28.4</td>
<td>32.4</td>
<td>33.3</td>
</tr>
<tr>
<td>LSTM+GloVe</td>
<td>31.9</td>
<td>31.7</td>
<td>34.3</td>
<td>32.9</td>
<td>29.5</td>
<td>30.4</td>
<td>34.3</td>
<td>33.8</td>
<td>30.7</td>
<td>30.5</td>
</tr>
<tr>
<td>LSTM+ELMo</td>
<td>31.7</td>
<td>31.4</td>
<td>33.2</td>
<td>32.8</td>
<td>30.4</td>
<td>30.0</td>
<td>33.8</td>
<td>33.3</td>
<td>30.8</td>
<td>30.4</td>
</tr>
<tr>
<td>LSTM+BERT-Base</td>
<td>35.9</td>
<td>36.2</td>
<td>38.7</td>
<td>38.2</td>
<td>33.2</td>
<td>34.1</td>
<td>40.5</td>
<td>40.5</td>
<td>33.7</td>
<td>33.8</td>
</tr>
<tr>
<td>ESIM+ELMo</td>
<td>33.6</td>
<td>33.3</td>
<td>35.7</td>
<td>34.2</td>
<td>31.5</td>
<td>32.3</td>
<td>37.7</td>
<td>36.6</td>
<td>31.6</td>
<td>31.5</td>
</tr>
<tr>
<td>OpenAI GPT</td>
<td>41.9</td>
<td>41.7</td>
<td>45.3</td>
<td>44.0</td>
<td>38.6</td>
<td>39.3</td>
<td>46.4</td>
<td>43.8</td>
<td>39.8</td>
<td>40.5</td>
</tr>
<tr>
<td>BERT-Base</td>
<td>39.5</td>
<td>40.5</td>
<td>42.9</td>
<td>42.8</td>
<td>36.1</td>
<td>38.3</td>
<td>48.9</td>
<td>45.7</td>
<td>34.9</td>
<td>37.7</td>
</tr>
<tr>
<td>BERT-Large</td>
<td><strong>46.7</strong></td>
<td><strong>47.3</strong></td>
<td><strong>50.2</strong></td>
<td><strong>49.7</strong></td>
<td><strong>43.3</strong></td>
<td><strong>45.0</strong></td>
<td><strong>54.7</strong></td>
<td><strong>51.7</strong></td>
<td><strong>42.9</strong></td>
<td><strong>45.0</strong></td>
</tr>
<tr>
<td>Human</td>
<td>95.7</td>
<td>95.6</td>
<td>95.6</td>
<td>95.6</td>
<td>95.8</td>
<td>95.7</td>
<td>94.0</td>
<td>94.0</td>
<td>96.5</td>
<td>96.5</td>
</tr>
</tbody>
</table>

Table 1: Performance of models, evaluated with accuracy (%). We report results on the full validation and test sets (Overall), as well as results on informative subsets of the data: evaluated on in-domain, versus zero-shot situations, along with performance on the underlying data sources (ActivityNet versus WikiHow). All models substantially underperform humans: the gap is over 45% on in-domain categories, and 50% on zero-shot categories.
Adversarial NLI

Adversarial NLI: A New Benchmark for Natural Language Understanding

Yixin Nie*, Adina Williams†, Emily Dinan†, Mohit Bansal*, Jason Weston†, Douwe Kiela†

*UNC Chapel Hill
†Facebook AI Research
## Adversarial NLI: Dataset creation

A direct response to adversarial test failings

*NLI datasets:

1. The annotator is presented with a premise sentence and a condition (entailment, contradiction, neutral).
2. The annotator writes a hypothesis.
3. A state-of-the-art model makes a prediction about the premise–hypothesis pair.
4. If the model's prediction matches the condition, the annotator returns to step 2 to try again.
5. If the model was fooled, the premise–hypothesis pair is independently validated by other annotators.
Adversarial NLI: Dataset creation

A direct response to adversarial test failings *

1. The annotator is presented with a premise sentence and a condition (entailment, contradiction, neutral).
Adversarial NLI: Dataset creation

A direct response to adversarial test failings *NLI datasets:

1. The annotator is presented with a premise sentence and a condition (entailment, contradiction, neutral).

2. The annotator writes a hypothesis.
Adversarial NLI: Dataset creation

A direct response to adversarial test failings *NLI datasets:

1. The annotator is presented with a premise sentence and a condition (entailment, contradiction, neutral).
2. The annotator writes a hypothesis.
3. A state-of-the-art model makes a prediction about the premise–hypothesis pair.
Adversarial NLI: Dataset creation

A direct response to adversarial test failings *NLI datasets:

1. The annotator is presented with a premise sentence and a condition (entailment, contradiction, neutral).

2. The annotator writes a hypothesis.

3. A state-of-the-art model makes a prediction about the premise–hypothesis pair.

4. If the model’s prediction matches the condition, the annotator returns to step 2 to try again.
Adversarial NLI: Dataset creation

A direct response to adversarial test failings *NLI datasets:

1. The annotator is presented with a premise sentence and a condition (entailment, contradiction, neutral).
2. The annotator writes a hypothesis.
3. A state-of-the-art model makes a prediction about the premise–hypothesis pair.
4. If the model’s prediction matches the condition, the annotator returns to step 2 to try again.
5. If the model was fooled, the premise–hypothesis pair is independently validated by other annotators.
### Adversarial NLI: Example

<table>
<thead>
<tr>
<th>Premise</th>
<th>Hypothesis</th>
<th>Reason</th>
<th>Label</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>A melee weapon is any weapon used in direct hand-to-hand combat; by contrast with ranged weapons which act at a distance. The term “melee” originates in the 1640s from the French word “mêlée”, which refers to hand-to-hand combat, a close quarters battle, a brawl, a confused fight, etc. Melee weapons can be broadly divided into three categories</td>
<td>Melee weapons are good for ranged and hand-to-hand combat.</td>
<td>Melee weapons are good for hand to hand combat, but NOT ranged.</td>
<td>E</td>
<td>N</td>
</tr>
</tbody>
</table>
### Adversarial NLI results

<table>
<thead>
<tr>
<th>Model</th>
<th>Data</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>ANLI</th>
<th>ANLI-E</th>
<th>SNLI</th>
<th>MNLI-m/-mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>S,M</td>
<td>00.0</td>
<td>28.9</td>
<td>28.8</td>
<td>19.8</td>
<td>19.9</td>
<td>91.3</td>
<td>86.7 / 86.4</td>
</tr>
<tr>
<td></td>
<td>+A1</td>
<td>44.2</td>
<td>32.6</td>
<td>29.3</td>
<td>35.0</td>
<td>34.2</td>
<td>91.3</td>
<td>86.3 / 86.5</td>
</tr>
<tr>
<td></td>
<td>+A1+A2</td>
<td>57.3</td>
<td>45.2</td>
<td>33.4</td>
<td>44.6</td>
<td>43.2</td>
<td>90.9</td>
<td>86.3 / 86.3</td>
</tr>
<tr>
<td></td>
<td>+A1+A2+A3</td>
<td>57.2</td>
<td>49.0</td>
<td>46.1</td>
<td>50.5</td>
<td>46.3</td>
<td>90.9</td>
<td>85.6 / 85.4</td>
</tr>
<tr>
<td></td>
<td>S,M,F,ANLI</td>
<td>57.4</td>
<td>48.3</td>
<td>43.5</td>
<td>49.3</td>
<td>44.2</td>
<td>90.4</td>
<td>86.0 / 85.8</td>
</tr>
<tr>
<td>XLNet</td>
<td>S,M,F,ANLI</td>
<td>67.6</td>
<td>50.7</td>
<td>48.3</td>
<td>55.1</td>
<td>52.0</td>
<td>91.8</td>
<td>89.6 / 89.4</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>S,M</td>
<td>47.6</td>
<td>25.4</td>
<td>22.1</td>
<td>31.1</td>
<td>31.4</td>
<td>92.6</td>
<td>90.8 / 90.6</td>
</tr>
<tr>
<td></td>
<td>+F</td>
<td>54.0</td>
<td>24.2</td>
<td>22.4</td>
<td>32.8</td>
<td>33.7</td>
<td>92.7</td>
<td>90.6 / 90.5</td>
</tr>
<tr>
<td></td>
<td>+F+A1</td>
<td>68.7</td>
<td>19.3</td>
<td>22.0</td>
<td>35.8</td>
<td>36.8</td>
<td>92.8</td>
<td>90.9 / 90.7</td>
</tr>
<tr>
<td></td>
<td>+F+A1+A2</td>
<td>71.2</td>
<td>44.3</td>
<td>20.4</td>
<td>43.7</td>
<td>41.4</td>
<td>92.9</td>
<td>91.0 / 90.7</td>
</tr>
<tr>
<td></td>
<td>S,M,F,ANLI</td>
<td>73.8</td>
<td>48.9</td>
<td>44.4</td>
<td>53.7</td>
<td>49.7</td>
<td>92.6</td>
<td>91.0 / 90.6</td>
</tr>
</tbody>
</table>

Table 3: Model Performance. ‘Data’ refers to training dataset (‘S’ refers to SNLI, ‘M’ to MNLI dev (-m=matched, -mm=mismatched), and ‘F’ to FEVER); ‘A1–A3’ refer to the rounds respectively. ‘-E’ refers to test set examples written by annotators exclusive to the test set. Datasets marked ‘*n’ were used to train the base model for round n, and their performance on that round is underlined.
A vision for future development

Zellers et al. (2019)
“a path for NLP progress going forward: towards benchmarks that adversarially co-evolve with evolving state-of-the-art models.”

Nie et al. (2019)
“This process yields a “moving post” dynamic target for NLU systems, rather than a static benchmark that will eventually saturate.”
Rethinking AI Benchmarking

Dynabench is a research platform for dynamic data collection and benchmarking. Static benchmarks have well-known issues: they saturate quickly, are susceptible to overfitting, contain exploitable annotator artifacts and have unclear or imperfect evaluation metrics.

This platform in essence is a scientific experiment: can we make faster progress if we collect data dynamically, with humans and models in the loop, rather than in the old-fashioned static way?
## Dynabench

1. NLI  
   (see Nie et al. 2020)
2. QA  
   (see Bartolo et al. 2020)
3. Sentiment  
   (DynaSent; Potts et al. 2020)
4. Hate Speech  
   (Vidgen et al. 2020)
Can adversarial training improve systems?

1. Jia and Liang (2017:§4.6): Training on adversarial examples makes them more robust to those examples but not to simple variants.

2. Alzantot et al. (2018:§4.3): “We found that adversarial training provided no additional robustness benefit in our experiments using the test set, despite the fact that the model achieves near 100% accuracy classifying adversarial examples included in the training set.”

3. Liu et al. (2019): Fine-tuning with a few adversarial examples improves systems in some cases (as discussed under ‘inoculation’ just above).

References


