Adversarial Examples for Evaluating Reading Comprehension Systems

R. Jia, P. Liang

Stanford University

arXiv: 1707.07328
Reviewed by: Bill Zhang
University of Virginia
https://qdata.github.io/deep2Read/
Outline

Introduction

SQuAD Task and Models

Adversarial Evaluation

Experiments

Discussion and Conclusion

References
Introduction
Basic Premise and Motivation

- Qualifying a computer’s ability to exhibit intelligent behavior is a long-standing problem
- Recognizing patterns that happen to be predictive on most samples can yield great success
- Propose adversarial evaluation for NLP, specifically SQuAD which answers questions about paragraphs in Wikipedia
- Want a method which does not contradict correct answer or confuse humans
SQuAD Task and Models

- 107,785 human-generated reading comprehension questions about Wikipedia articles
- Each question refers to one paragraph in article, answer is guaranteed to be in paragraph
- Focused on BiDAF and Match-LSTM which predict probability distributions over correct answer; each has single and ensemble version
- Validate results on 12 other public models; did not run during development
- Accuracy Evaluation where $v$ is the F1 score, $D_{test}$ is the test set, and $(p, q, a)$ is a paragraph, question, answer tuple

$$\text{Acc}(f) = \frac{1}{|D_{test}|} \sum_{(p, q, a) \in D_{test}} v((p, q, a), f)$$
Adversarial Evaluation

Main Idea

- A model which relies on superficial cues without understanding language can perform well
- Define adversary $A$ as a function which takes in $(p, q, a)$ (and optionally $f$) and outputs new examples $(p', q', a')$
- Adversarial accuracy is therefore

$$\text{Adv}(f) = \frac{1}{|D_{test}|} \sum_{(p, q, a) \in D_{test}} v(A(p, q, a, f), f)$$

- For meaningful results, $(p', q', a')$ should be valid (human would answer $a'$ given $p'$ and $q'$); also, should be close to original $(p, q, a)$
Adversarial Evaluation

General Method

- In image classification, usually add small perturbation while preserving semantics of image; analogy in NLP is paraphrasing, which is hard to do in high-precision
- Thus, rely on concatenative adversaries: generate adversaries of the form \((p + s, q, a)\) which adds a new sentence to end of paragraph without changing question and answer
- Valid \(s\) do not contradict correct answer
- Overstability vs oversensitivity of model
- Could append \(s\) at beginning, but would violate first sentence being topic sentence; appending in middle could break links between sentences
Adversarial Evaluation

AddSent

1. Take question and make semantics-altering perturbations: replace n. and adj. with antonyms from WordNet, entities and numbers to nearest word in GloVe space with same part of speech
   - What ABC division handles domestic TV distribution? → What NBC division handles foreign TV distribution?

2. Create fake answer with same "type" as original answer: manually associated fake answer for each type

3. Combine 1 and 2 in declarative form
   - What ABC division handles domestic television distribution? → The NBC division of Central Park handles foreign television distribution.

4. Fix grammar errors via crowdsourcing, pick best sentence from black-box tests

Minimal interaction with model, AddOneSent variant without black-box tests
Adversarial Evaluation

AddAny

- Choose any sequence of $d$ words, regardless of grammar
- Initialize $d$ words randomly from common English words
- Run 6 epochs of local search, each of which iterates through indices 1 to $d$ in random order
- For each index, generate candidate words from 20 randomly sampled common words and all words in $q$
- Replace word at index with each candidate word, greedily choose word which minimizes expected F1 score Requires significantly more model queries, requires model output distribution, not just single choice
- Variant AddCommon which only uses common words
Experiments

Main Experiments

- Measure adversarial F1 score across 1000 random samples from SQuAD

<table>
<thead>
<tr>
<th>Model</th>
<th>Original</th>
<th>ADDSENT</th>
<th>ADDONESENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ReasonNet-E</td>
<td>81.1</td>
<td>39.4</td>
<td>49.8</td>
</tr>
<tr>
<td>SEDT-E</td>
<td>80.1</td>
<td>35.0</td>
<td>46.5</td>
</tr>
<tr>
<td>BiDAF-E</td>
<td>80.0</td>
<td>34.2</td>
<td>46.9</td>
</tr>
<tr>
<td>Mnemonic-E</td>
<td>79.1</td>
<td>46.2</td>
<td>55.3</td>
</tr>
<tr>
<td>Ruminating</td>
<td>78.8</td>
<td>37.4</td>
<td>47.7</td>
</tr>
<tr>
<td>jNet</td>
<td>78.6</td>
<td>37.9</td>
<td>47.0</td>
</tr>
<tr>
<td>Mnemonic-S</td>
<td>78.5</td>
<td>46.6</td>
<td>56.0</td>
</tr>
<tr>
<td>ReasoNet-S</td>
<td>78.2</td>
<td>39.4</td>
<td>50.3</td>
</tr>
<tr>
<td>MPCM-S</td>
<td>77.0</td>
<td>40.3</td>
<td>50.0</td>
</tr>
<tr>
<td>SEDT-S</td>
<td>76.9</td>
<td>33.9</td>
<td>44.8</td>
</tr>
<tr>
<td>RaSOR</td>
<td>76.2</td>
<td>39.5</td>
<td>49.5</td>
</tr>
<tr>
<td>BiDAF-S</td>
<td>75.5</td>
<td>34.3</td>
<td>45.7</td>
</tr>
<tr>
<td>Match-E</td>
<td>75.4</td>
<td>29.4</td>
<td>41.8</td>
</tr>
<tr>
<td>Match-S</td>
<td>71.4</td>
<td>27.3</td>
<td>39.0</td>
</tr>
<tr>
<td>DCR</td>
<td>69.3</td>
<td>37.8</td>
<td>45.1</td>
</tr>
<tr>
<td>Logistic</td>
<td>50.4</td>
<td>23.2</td>
<td>30.4</td>
</tr>
</tbody>
</table>

Table 2: Adversarial evaluation on the MatchLSTM and BiDAF systems. All four systems can be fooled by adversarial examples.
Experiments
Human Evaluation

- Make sure that humans are not fooled by examples

<table>
<thead>
<tr>
<th></th>
<th>Human</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>92.6</td>
</tr>
<tr>
<td>ADDSENT</td>
<td>79.5</td>
</tr>
<tr>
<td>ADDONESENT</td>
<td>89.2</td>
</tr>
</tbody>
</table>

Table 4: Human evaluation on adversarial examples. Human accuracy drops on ADDSENT mostly due to unrelated errors; the ADDONESENT numbers show that humans are robust to adversarial sentences.
Experiments

Analysis

- Manually verify that sentences do not contradict answer and are grammatically accurate for AddSent
- In 96.6% of model failures, predicted a span within adversarial sentence for AddSent
- Humans only picked adversarial spans in 27.3% of failures, which shows that humans make many mistakes unrelated to adversarial sentences
- Models do well when there is a n-gram match in question and original paragraph
- Short questions tend to increase model success
- AddSent generalized well to other models, AddAny more limited
Experiments

Analysis

Table 5: Transferability of adversarial examples across models. Each row measures performance on adversarial examples generated to target one particular model; each column evaluates one (possibly different) model on these examples.

Figure 3: Fraction of model successes and failures on ADDSENT for which the question has an exact n-gram match with the original paragraph. For each model and each value of n, successes are more likely to have an n-gram match than failures.

Figure 4: For model successes and failures on ADDSENT, the cumulative distribution function of the number of words in the question (for each k, what fraction of questions have ≤ k words). Successes are more likely to involve short questions.
Experiments

Analysis

- Also, attempt adversarial training while performing only steps 1 to 3 of AddSent
- Results look good, but modifying method slightly to prepend sentence and change words for each category makes model perform poorly
- Suggests model has learned to reject specific fake answers and the last sentence

<table>
<thead>
<tr>
<th>Test data</th>
<th>Training data Original</th>
<th>Augmented</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>75.8</td>
<td>75.1</td>
</tr>
<tr>
<td>ADDSENT</td>
<td>34.8</td>
<td>70.4</td>
</tr>
<tr>
<td>ADDSENTMOD</td>
<td>34.3</td>
<td>39.2</td>
</tr>
</tbody>
</table>

Table 6: Effect of training the BiDAF Single model on the original training data alone (first column) versus augmenting the data with raw ADDSENT examples (second column).
Discussion and Conclusion

- Despite appearing successful by common metrics, reading comprehension systems perform poorly under adversarial evaluation; models are overly stable to perturbations.
- Adversarial evaluation method is primarily for evaluation, not training because of how slow it is.
- Concatenative adversaries are good for reading comprehension, but other methods may be better for other, more general tasks.
References