Adversarial Examples in NLP

Elisabetta Cavallo, Seyoon Ragavan

COS598C - Deep Learning for Natural Language Processing
April 16th, 2020
What are Adversarial Examples?

Generating Adversarial Examples

- **Paper I**: Adversarial Example Generation with Syntactically Controlled Paraphrase Networks (Iyyer at al., 2018)

Training with Adversarial Examples

- **Paper II**: Adversarial Examples for Evaluating Reading Comprehension Systems (Jia et Liang, 2017)

Quiz Time & Bonus Papers!
What are Adversarial Examples?

Introduction

• Applicable to any NLP task and model.
• An adversary or attack slightly perturbs the input to fool the model.

---

Article: Super Bowl 50
Paragraph: “Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver’s Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”
Question: “What is the name of the quarterback who was 38 in Super Bowl XXXIII?”
Original Prediction: John Elway
Prediction under adversary: Jeff Dean

(Jia and Liang, 2017)
What are Adversarial Examples?

**Terminology**

- **Adversarial example**: a perturbed input-output pair

- **Adversary/attack**: a method for generating examples

- **Robustness**: how well a model performs against an adversary.
  - Model evaluated with the same metric as in the standard/non-adversarial
What are Adversarial Examples?

Why do Adversarial Examples matter?

- **Security** is important for some applications
  - Spam detection
  - Healthcare

- **Evaluation** of models and datasets
  - Does the model/dataset really exhibit/test sophisticated understanding?

- **Interpretation** of models
  - What does the model care about, and what does it ignore?
  - Are these bugs that need to be addressed?

- **Robust training** of models
  - Augment training data with adversarial examples

(Singh, 2019)
What are Adversarial Examples?

Adversarial Examples in Computer Vision

• Image classification task
• Gradient-based attacks to increase loss

(Goodfellow et al., 2015)
## What are Adversarial Examples?

### Computer Vision vs. Natural Language Processing

<table>
<thead>
<tr>
<th></th>
<th>Images</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input type</strong></td>
<td>Continuous</td>
<td>Discrete</td>
</tr>
<tr>
<td><strong>Original Input</strong></td>
<td><img src="image.png" alt="Image" /> “Quarterback John Elway was 38 in Super Bowl XXXIII.”</td>
<td>“I’d have to say the star and director are the big problems here.”</td>
</tr>
<tr>
<td><strong>Adversarial Input</strong></td>
<td><img src="image.png" alt="Image" /> “Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.”</td>
<td>“By the way, you know, the star and director are the big problems.”</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td>Same</td>
<td>Different</td>
</tr>
<tr>
<td><strong>Model’s mistake</strong></td>
<td>Treats the two as different</td>
<td>Treats the two as the same</td>
</tr>
<tr>
<td><strong>Exploited weakness</strong></td>
<td>Oversensitivity</td>
<td>Overstability</td>
</tr>
</tbody>
</table>


Generating Adversarial Examples

What can we modify in the original sentence to create an adversarial example?

**Character-level:** flip / insert / delete a character.

Typoglycemic text:

“Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn’t mttae in waht ordehr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae.”

**Word-level:** change a word in a sequence.

- Random word?
  \[ x' = [\text{‘I’} \quad \text{‘lamp’} \quad \text{‘this’} \quad \text{‘movie’} \quad \text{‘.’} ] \]

- Word Embedding?
  \[ x' = [\text{‘I’} \quad \text{‘really’} \quad \text{‘this’} \quad \text{‘movie’} \quad \text{‘.’} ] \]

- Part of Speech?
  \[ x' = [\text{‘I’} \quad \text{‘eat’} \quad \text{‘this’} \quad \text{‘movie’} \quad \text{‘.’} ] \]

**Sentence-level:** replace the entire sentence.

“Susan told me she is pregnant” → “I was told by Susan she is expecting a baby”
Generating Adversarial Examples

How do we choose the adversarial attack to perform on our sentence?

- **Black-box**: close to random, relying on heuristic methods, disjoint from the “application” model → not “best” AE!

  E.g.: random shuffling of letters (character-level), word replacement based on POS tag (word-level), paraphrase with back-translation (sentence-level).
Generating Adversarial Examples

How do we choose the adversarial attack to perform on our sentence?

• **White-box**: approximates the worst-case attack for a particular model and input, within some allowed set of perturbations.

  - E.g. Gradient based (discrete optimization): compute the gradient of the loss function relative to the input representation $x$, step in that direction and set the adversarial example $x'$ equal to the nearest neighbour.
Generating Adversarial Examples

Discrete Optimization problem

\[ \min_{x'} ||x - x'|| \]

\[ \text{s.t. } f(x') \neq f(x) \]

Model information

Loss function and gradient w.r.t input

Possible solution:
Approximate gradient method
Adversarial Example Generation with Syntactically Controlled Paraphrase Networks

Iyyer et Al. (2018)
Adversarial Example Generation with SCPN

Paraphrase generation: previous work

- **Black Box**
  - **Template-based**: hand-crafted rules and grammars, thesaurus-based substitution, etc.
  - **Translation-based**: lattice-based SMT, statistical techniques, etc.

- **White Box**
  - **Gradient based**: atomic flip operation (HotFlip by Ebrahimi et al, 2018), etc.

→ Lexical adversaries

“Exactly the kind of **unexpected delight** one hopes for every time the lights go down”

Positive

“Exactly the kind of **thrill** one hopes for every time the lights go down”

Negative
Adversarial Example Generation with SCPN

Paraphrase generation: SCPN

This paper: first learning approach for generate a syntactically controlled paraphrase of a given sentence.

→ Syntactical adversaries

```
"American drama doesn't get any more meaty and muscular than this"

Positive

+ Target syntactic form (e.g., a constituency parse)

SCPN

"Doesn't get any more meaty and muscular than this American drama"

Negative
```

Black box with output feedback
Adversarial Example Generation with SCPN

Overview

1. Training Data
   Backtranslation from Wieting et al. (2017)

2. Sentence parsing
   Get $\langle p_1, p_2 \rangle$ for each $\langle s_1, s_2 \rangle$
   Template relaxation
   Use template $t_2$ for $p_2$

3. Model
   1) Parse generator
      Produce complete parses from template
      Parses (from $t_2$ to $p_2$).
   2) SCPN
      Generate $s_2$ from $s_1, p_2$.

4. Evaluation
   Intrinsic evaluation
   Adversarial evaluation
Adversarial Example Generation with SCPN

1. Training data

No large-scale dataset of sentential paraphrases exists publicly.

→ use the pre-trained PARA-NMT-50M corpus from Wieting and Gimpel (2017): 50 million paraphrases obtained by backtranslating the Czech side of the CzEng.

Original sentence: EN

“Despite being scared of flying, I went to visit my sister in Japan.”

Translated sentence: CZ

"Přestože jsem se bál létání, šel jsem navštívit svou sestru v Japonsku.”

Paraphrased sentence: EN

"Although I was afraid of flying, I went to visit my sister in Japan.”
2. Sentence Parsing

Parse the backtranslated paraphrases using the Stanford parser.

→ Get the pair of constituency parses \( \langle p_1, p_2 \rangle \) for each \( \langle s_1, s_2 \rangle \).

“She drove home.”

\( (S(NP(PRP)) \ (VP(VBD) \ (NP(NN)))) \ (.) ) \)

Relax the target syntactic form to a parse template (top two levels of the linearized parse tree):

“She drove home.”

\( S \rightarrow NP \ VP \)

Consider 20 most frequent templates in PÂRA-NMT-50M

To overcome learned biases, also include the reversed pairs \( \langle s_2, s_1 \rangle \) are included during training
### Adversarial Example Generation with SCPN

#### 2. Sentence Parsing - Template filtering

<table>
<thead>
<tr>
<th>template</th>
<th>paraphrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>with the help of captain picard, the borg will be prepared for everything.</td>
</tr>
<tr>
<td>(SBARQ (ADVP) (,) (S) (,) (SQ))</td>
<td>now, the borg will be prepared by picard, will it?</td>
</tr>
<tr>
<td>(S (NP) (ADVP) (VP))</td>
<td>the borg here will be prepared for everything.</td>
</tr>
<tr>
<td>(S (S) (,) (CC) (S) (:) (FRAG))</td>
<td>with the help of captain picard, the borg will be prepared, and the borg will be prepared for everything ... for everything.</td>
</tr>
<tr>
<td>(FRAG (INTJ) (,) (S) (,) (NP))</td>
<td>oh, come on captain picard, the borg line for everything.</td>
</tr>
</tbody>
</table>

 Failure

 templates may be not be appropriate for particular input sentences (semantic divergence/ungrammatical)

→ **Feedback mechanism from output:** generated paraphrases are filtered using n-gram overlap and paraphrastic similarity (Wieting and Gimpel, 2017).
Adversarial Example Generation with SCPN

3. Model - Parse Generator + SCPN

Given a paraphrase pair \( \langle s_1, s_2 \rangle \) and corresponding target syntax trees \( \langle p_1, p_2 \rangle \), the model is such that:

**Inputs:** \( s_1 \) and \( p_2 \)  \quad \rightarrow \quad **Output:** trained to produce \( s_2 \)

*trained separately*
Adversarial Example Generation with SCPN

3. Paraphrase Generator Architecture - Encoder

1. Input sentence encoding

The man is standing in the water...
Adversarial Example Generation with SCPN

3. Paraphrase Generator Architecture - Encoder

1. Input sentence encoding

```
The man is standing in the water ...
```

2. Linearized parse token sequence encoding
Adversarial Example Generation with SCPN

3. Paraphrase Generator Architecture - Decoder

- Copy mechanism on input encoder
  - Input sentence $s_1$
  - Target sentence $s_2$

- Attention on parse encoder
  - Target parse $p_2$

- LSTM decoder to produce $s_2$
  - $h_t = LSTM([w_{t-1}; a_t; z_t])$

Previous word in $s_2$
Attention-weighted average of LSTM parse hidden encoding
Copy-mechanism over encoded input
Adversarial Example Generation with SCPN

3. Parse Generator Architecture

Generate complete target parses from parse templates → similar architecture to the paraphrase generator.
Adversarial Example Generation with SCPN

Evaluation

1) **Intrinsic evaluations**: paraphrase quality, do the generated paraphrases follow the target distribution?

2) **Adversarial evaluations**: validity of adversarial examples, improvement in robustness of downstream models.

**Baseline**: NMT-BT → uncontrolled neural back-translation.

→ compare the ten most probable beams from NMT-BT to controlled paraphrases generated by SCPN

<table>
<thead>
<tr>
<th>template</th>
<th>original</th>
<th>paraphrase</th>
</tr>
</thead>
<tbody>
<tr>
<td>((S\ ADVP) (NP) (VP)) )</td>
<td>moody, heartbreaking, and filmed in a natural, unforced style that makes its characters seem entirely convincing even when its script is not. there is no pleasure in watching a child suffer.</td>
<td>so he’s filmed in a natural, unforced style that makes his characters seem convincing when his script is not. in watching the child suffer, there is no pleasure.</td>
</tr>
<tr>
<td>((S\ PP) ,(,) (NP) (VP)) )</td>
<td>every nanosecond of the the new guy reminds you that you could do something else far more pleasurable. harris commands the screen, using his frailty to suggest the ravages of a life of corruption and ruthlessness.</td>
<td>each nanosecond from the new guy reminds you that you could do something else much more enjoyable. harris commands the screen, using his weakness to suggest the ravages of life of corruption and recklessness.</td>
</tr>
</tbody>
</table>

→ **SCPN**: Syntactic adversaries

→ **NMT-BT**: Lexical adversaries
Adversarial Example Generation with SCPN

Intrinsic Evaluation

1) Paraphrase quality: score of a paraphrase pair \langle source, generated \rangle by crowdworkers

→ SCPN vs. NMT-BT outputs: comparable in quality and grammatical correctness (but not in terms of syntactic difference from original).

→ Templates-fed vs. Full parses-fed SCPN quality: close to same.

2) Do the paraphrases follow the target specification?

<table>
<thead>
<tr>
<th>Model</th>
<th>Parse Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCPN w/ gold parse</td>
<td>64.5</td>
</tr>
<tr>
<td>SCPN w/ generated parse</td>
<td>51.6</td>
</tr>
<tr>
<td>Parse generator</td>
<td>99.9</td>
</tr>
</tbody>
</table>

Accuracy is measured by exact template match (i.e., how often do the top two levels of the parses match).
Adversarial Example Generation with SCPN

Adversarial Evaluation

1) **Sentiment Analysis** - Stanford Sentiment Tree-bank (SST) (Socher et al., 2013)
   → contains complicated sentences with high syntactic variance.

2) **Entailment Detection** - SICK (Marelli et al., 2014)
   → almost exclusively consists of short, simple sentences.

<table>
<thead>
<tr>
<th>Model</th>
<th>Task</th>
<th>Validity</th>
<th>No augmentation</th>
<th>With augmentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCPN</td>
<td>SST</td>
<td>77.1</td>
<td>83.1</td>
<td>41.8</td>
</tr>
<tr>
<td>NMT-BT</td>
<td>SST</td>
<td>68.1</td>
<td>83.1</td>
<td>20.2</td>
</tr>
<tr>
<td>SCPN</td>
<td>SICK</td>
<td>77.7</td>
<td>82.1</td>
<td>33.8</td>
</tr>
<tr>
<td>NMT-BT</td>
<td>SICK</td>
<td>81.0</td>
<td>82.1</td>
<td>20.4</td>
</tr>
</tbody>
</table>

- SCPN generates more legitimate adversarial examples than NMT-BT
- Augmenting data improves robustness of downstream models
Adversarial Example Generation with SCPN

Conclusions

SCPN:

- avoids lexical substitution in favor of making syntactic changes
- paraphrases follow their target specifications without decreasing paraphrase quality of unrestricted baselines.
- no quality drop when trained with templates vs. full parses.
- generates valid adversarial examples.

Possible future research:

- Provide down-stream signals to SCPN when training to allow for further lexical and syntax substitution.
- Dynamically integrates templates based on factors such as the length of the input sentence.
Adversarial Examples for Evaluating Reading Comprehension Systems

Jia and Liang (2017)
Adversarial Examples for Reading Comprehension

Contributions

- Show that simple adversarial attacks are effective against models trained on SQuAD.
- Analyse adversarial examples → evidence that many models trained on SQuAD rely on shallow heuristics, e.g. keyword matching.

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

Refresher: Reading Comprehension and SQuAD

- Input: (paragraph, question)
- Output: span of the paragraph
- Evaluation: F1 score

**Computational complexity theory** is a branch of the **theory** of computation in theoretical computer science that focuses on classifying **computational** problems according to their **inherent difficulty**, and relating those classes to each other. A **computational** problem is understood to be a task that is in principle amenable to being solved by a computer, which is equivalent to stating that the problem may be solved by mechanical application of mathematical steps, such as an algorithm.

By what main attribute are computational problems classified utilizing computational complexity theory?

**Ground Truth Answers:** inherent difficulty, their inherent difficulty

**Prediction:** inherent difficulty

(Rajpurkar et al., 2016)
Adversarial Examples for Reading Comprehension

Refresher: Limitations of SQuAD

- Questions were constructed looking at passages → lexical and syntactic overlap.
- Should be doable with type and keyword-matching.
- **Goal**: create an adversary that exploits this.

---

By what main attribute are computational problems classified utilizing computational complexity theory?

*Ground Truth Answers*: inherent difficulty, their inherent difficulty, inherent difficulty

*Prediction*: inherent difficulty

(Ref: Rajpurkar et al., 2016)
Adversarial Examples for Reading Comprehension

Adversaries: AddSent and AddOneSent

- **Concatenative**: append a distracting sentence to the input paragraph
- **Word-level** changes to the question/answer
  - High lexical overlap with the question but does not actually answer it
- **Semantics-altering**
- **No dependence on the input paragraph**

What is the name of the quarterback who was 38 in Super Bowl XXXIII? (A: John Elway)

Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.

What ABC division handles domestic television distribution? (A: Disney-ABC)

The NBC division of Central Park handles foreign television distribution.

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

AddSent and AddOneSent: Overview

What ABC division handles *domestic* television distribution?

Step 1: Mutate question

What NBC division handles *foreign* television distribution?

Step 2: Generate fake answer

**Step 3: Convert into statement**

The NBC division of **Central Park** handles foreign television distribution.

**Step 4: Fix errors and verify corrected sentences with crowdworkers**

The NBC division of **Central Park** handles foreign television distribution.

Model Predicts: **Central Park**

(Jia and Liang, 2017)
What are Adversarial Examples?

Step 1: Mutate Question

• Alter the question’s semantics → generated sentence will not contradict the paragraph

• Nouns, adjectives → antonyms from WordNet

• Named entities, numbers → nearest word in GloVe embedding space with the same POS

What ABC division handles domestic television distribution?

What NBC division handles foreign television distribution?
Adversarial Examples for Reading Comprehension

Step 2: Generate Fake Answer

- Fake answer should have the same “type” as the original answer
- Predefine 26 types
  - NER and POS tags from Stanford CoreNLP
  - Custom categories e.g. abbreviations
- Fix a fake answer for each type

(Jia and Liang, 2017)
Step 3: Combine Fake Question/Answer

- Use 50 manually defined rules over CoreNLP constituency parses
- Incomplete and error-prone

What *NBC division* handles foreign television distribution?

What/which NP1 VP1?

The NP1 of ANSWER VP1.

The *NBC division of Central Park* handles foreign television distribution.

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

**Step 4: Fix Grammatical Errors**

- Crowdsource via Amazon Mechanical Turk
- Edited independently by 5 workers → 5 sentences

- **AddSent**: try all 5 sentences on the model and choose the one where the model gives the worst (in terms of F1 score) answer
  - This is the only part where the model is used!

- **AddOneSent**: choose one of these 5 sentences randomly
  - Completely model-independent

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

**Adversaries: AddAny**

- Still concatenative
- But the appended “sentence” can be any sequence of d words → could (and will probably) be total nonsense
- Step 1: Initialise the words randomly from a list of common English words

Spring attention income getting reached

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

Adversaries: AddAny

- Step 2: Use local search to greedily change one word at a time to worsen the model’s performance
- Search space: 20 randomly sampled common words and all words in the question
- Performance measure: expected F1 score over the model’s output distribution
- Requires several queries to the model and “grey-box” access to the output distribution

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

AddAny: Example \( (d = 5) \)

Spring attention income `{getting}` reached

Spring attention income `{NBC}` reached

George attention income `{NBC}` reached

George handle handle `{NBC, other}`

Model predicts: `{George}`

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

Adversaries: AddCommon

- All the adversaries so far rely in part on “baiting” the model with keywords from the question

- Can we trick the model in a less straightforward way?
  - Identify **subtler error patterns** of the model

- AddCommon: same as AddAny but the local search is restricted to common words

(Spring attention income getting reached)

Briefcase escalator gossip cough other

(Jia and Liang, 2017)
### Adversarial Examples for Reading Comprehension

#### Adversaries: Overview

<table>
<thead>
<tr>
<th>Adversary</th>
<th>Access to model</th>
<th>Appends sensible sentences</th>
<th>Uses words from question</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddSent</td>
<td>Black-box 5 queries/example</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>AddOneSent</td>
<td>Black-box Model-independent</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>AddAny</td>
<td>“Grey-box” 1000s of queries/example</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>AddCommon</td>
<td>“Grey-box” 1000s of queries/example</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

**Experiments: Setup**

- Evaluate on 2 models during development
  - BiDAF (Seo et al, 2016)
  - Match-LSTM (Wang and Jiang, 2016)
    - Single and ensemble version for each
  - Use 10 other models for validation as well
Adversarial Examples for Reading Comprehension

Results: Main Experiments

Mnemonic Reader models long-range dependencies within the paragraph → can locate correct answer

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

Results: Human Evaluation

• **This is important!** If humans are consistently getting adversarial examples “wrong” then the examples are not valid.

• AddSent < AddOneSent only because humans naturally make mistakes

<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>

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

Results: AddSent Error Analysis

Question: The number of Huguenot colonists declined after what year?

Paragraph: The largest portion of the Huguenots to settle in the Cape arrived between 1688 and 1689, in seven ships as part of the organised migration, but quite a few arrived as late as 1700; thereafter, the numbers declined, and only small groups arrived at a time. The number of old Acadian colonists declined after the year of 1675.

Correct answer: 1700

Model predicts: 1675

(After Jia, 2017; Jia and Liang, 2017; Rajpurkar et al., 2016)
Adversarial Examples for Reading Comprehension

Results: AddAny Error Analysis

Question: What city did Tesla move to in 1880?

Paragraph: In January 1880, two of Tesla’s uncles put together enough money to help him leave Gospić for Prague… what 30 city 1880 what move city city medical move.

Correct answer: Prague

Model predicts: medical

(Jia, 2017; Jia and Liang, 2017; Rajpurkar et al., 2016)
Adversarial Examples for Reading Comprehension

Results: AddCommon Error Analysis

Question: Where did he (Tesla) claim the blueprint was stored?

Paragraph: During the period in which the negotiations were being conducted... the **blueprint** for the teleforce weapon was all **in his mind**. **Doubt was did about** carried wasn’t year 1961 near policy.

Correct answer: **in his mind**

Model predicts: **near policy**

(Jia, 2017; Jia and Liang, 2017; Rajpurkar et al., 2016)
Adversarial Examples for Reading Comprehension

Results: Transferability Across Models

• AddSent examples transfer well, AddAny examples do not

• Suggests that the attacks exploit general limitations of SQuAD rather than model-specific limitations

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDSENT</td>
<td>ML Single</td>
<td>27.3</td>
<td>33.4</td>
<td>40.3</td>
<td>39.1</td>
</tr>
<tr>
<td></td>
<td>ML Ens.</td>
<td>31.6</td>
<td>29.4</td>
<td>40.2</td>
<td>38.7</td>
</tr>
<tr>
<td></td>
<td>BiDAF Single</td>
<td>32.7</td>
<td>34.8</td>
<td>34.3</td>
<td>37.4</td>
</tr>
<tr>
<td></td>
<td>BiDAF Ens.</td>
<td>32.7</td>
<td>34.2</td>
<td>38.3</td>
<td>34.2</td>
</tr>
<tr>
<td>ADDANY</td>
<td>ML Single</td>
<td>7.6</td>
<td>54.1</td>
<td>57.1</td>
<td>60.9</td>
</tr>
<tr>
<td></td>
<td>ML Ens.</td>
<td>44.9</td>
<td>11.7</td>
<td>50.4</td>
<td>54.8</td>
</tr>
<tr>
<td></td>
<td>BiDAF Single</td>
<td>58.4</td>
<td>60.5</td>
<td>4.8</td>
<td>46.4</td>
</tr>
<tr>
<td></td>
<td>BiDAF Ens.</td>
<td>48.8</td>
<td>51.1</td>
<td>25.0</td>
<td>2.7</td>
</tr>
</tbody>
</table>

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

Results: Adversarial Training

- **AddSentMod**:
  - Use a different set of fake answers for each type e.g. Jeff Dean → Charles Babbage
  - Prepend (rather than append) the adversarial sentence to the paragraph

- Model overfits the adversary used for training

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

(Jia and Liang, 2017)
Adversarial Examples for Reading Comprehension

**Takeaways**

- Adversarial examples can expose models that rely on shallow heuristics and provide insights into these heuristics.
- They can also expose datasets that are simpler than they seem.
- Just appending a sentence is effective as an attack.
- **For future work:** Haven't successfully used adversarial examples to train robust models yet.
Quiz time!
Quiz Time

Discussion Question #2

Q: Both papers investigated the effect of training the model on these generated adversarial examples. Do you think this would eventually fix the problem or not?

• Adversarial training can help in some applications - notable success in computer vision.
• But this is probably harder in NLP's discrete space. Can help in improving the robustness of the model at test time, and also in reducing its likelihood to "break" at train time, but far from a solution.
• Belinkov and Bisk, 2018
• Some types of adversarial examples do not improve robustness as the model is incapable of learning any patterns.
• Training on a specific type of error/adv. example does not allow to generalise on other errors.

→
Quiz Time

Discussion Question #2

Q: Both papers investigated the effect of training the model on these generated adversarial examples. Do you think this would eventually fix the problem or not?

Answer:

• Adversarial training can help in some applications - notable success in computer vision.
• But this is harder in NLP’s discrete space. Can help in improving the robustness of the model at test time, and also in reducing its likelihood to “break” at train time, but far from a solution.
• Belinkov and Bisk, 2018
  • some types of adversarial examples do not improve robustness as the model is incapable of learning any patterns.
  • training on a specific type of error/adv. example does not allow to generalise on other errors.
Bonus Papers

Natural vs. Synthetic Noise: Success in Improving Robustness? - Belinkov and Bisk, 2018

Natural errors: collected from real examples at word level (e.g. Wikipedia edit histories, manually annotated essays written by non-native speakers, etc.), across 3 languages - German, French and Czech.

Synthetic errors: four types of noise

- Swap: e.g. noise → nosie
- Middle Random: e.g. noise → nisoe
- Fully Random: e.g. noise → nisoe
- Keyboard typo: e.g. noise → noide
### Bonus Papers

**Natural vs. Synthetic Noise: Success in Improving Robustness?** - Belinkov and Bisk, 2018

#### Results (BLEU Scores)

<table>
<thead>
<tr>
<th>Language</th>
<th>Model</th>
<th>Vanilla</th>
<th>Swap</th>
<th>Mid</th>
<th>Rand</th>
<th>Key</th>
<th>Nat</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>French</strong></td>
<td>charCNN</td>
<td>42.54</td>
<td>10.52</td>
<td>9.71</td>
<td>1.71</td>
<td>8.26</td>
<td>17.42</td>
</tr>
<tr>
<td></td>
<td>charCNN</td>
<td>34.79</td>
<td>9.25</td>
<td>8.37</td>
<td>1.02</td>
<td>6.40</td>
<td>14.02</td>
</tr>
<tr>
<td></td>
<td>char2char</td>
<td>29.97</td>
<td>5.68</td>
<td>5.46</td>
<td>0.28</td>
<td>2.96</td>
<td>12.68</td>
</tr>
<tr>
<td></td>
<td>Nematus</td>
<td>34.22</td>
<td>3.39</td>
<td>5.16</td>
<td>0.29</td>
<td>0.61</td>
<td>10.68</td>
</tr>
<tr>
<td><strong>German</strong></td>
<td>charCNN</td>
<td>25.99</td>
<td>6.56</td>
<td>6.67</td>
<td>1.50</td>
<td>7.13</td>
<td>10.20</td>
</tr>
<tr>
<td></td>
<td>char2char</td>
<td>25.71</td>
<td>3.90</td>
<td>4.24</td>
<td>0.25</td>
<td>2.88</td>
<td>11.42</td>
</tr>
<tr>
<td></td>
<td>Nematus</td>
<td>29.65</td>
<td>2.94</td>
<td>4.09</td>
<td>0.66</td>
<td>1.41</td>
<td>11.88</td>
</tr>
</tbody>
</table>

- Significant drop in BLEU when evaluated on noisy texts → the more the noise the worse.
- Worst results on languages with complex structures (Czech).
- Other results: training on a **specific type of noise** makes the model more robust to that type of noise, but not to others (except random which never improves robustness).
Certified Robustness to Word-Level Attacks!

- Interval Bound Propagation → upper bound on the model’s loss for any combination of these substitutions
- Optimise this upper bound directly!

(Jia et al., 2019; Gowal et al., 2018)
## Comparison with Data Augmentation

<table>
<thead>
<tr>
<th>System</th>
<th>Genetic attack (Upper bound)</th>
<th>IBP-certified (Lower bound)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard training</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoW</td>
<td>9.6</td>
<td>0.8</td>
</tr>
<tr>
<td>CNN</td>
<td>7.9</td>
<td>0.1</td>
</tr>
<tr>
<td>LSTM</td>
<td>6.9</td>
<td>0.0</td>
</tr>
<tr>
<td><strong>Robust training</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoW</td>
<td><strong>70.5</strong></td>
<td><strong>68.9</strong></td>
</tr>
<tr>
<td>CNN</td>
<td><strong>75.0</strong></td>
<td><strong>74.2</strong></td>
</tr>
<tr>
<td>LSTM</td>
<td>64.7</td>
<td>63.0</td>
</tr>
<tr>
<td><strong>Data augmentation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoW</td>
<td>34.6</td>
<td>3.5</td>
</tr>
<tr>
<td>CNN</td>
<td>35.2</td>
<td>0.3</td>
</tr>
<tr>
<td>LSTM</td>
<td>33.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Genetic attack (Upper bound)</th>
<th>IBP-certified (Lower bound)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Normal training</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoW</td>
<td>40.5</td>
<td>2.3</td>
</tr>
<tr>
<td>DECOMPATTN</td>
<td>40.3</td>
<td>1.4</td>
</tr>
<tr>
<td><strong>Robust training</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoW</td>
<td><strong>75.0</strong></td>
<td><strong>72.7</strong></td>
</tr>
<tr>
<td>DECOMPATTN</td>
<td>73.7</td>
<td>72.4</td>
</tr>
<tr>
<td><strong>Data augmentation</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BoW</td>
<td>68.5</td>
<td>7.7</td>
</tr>
<tr>
<td>DECOMPATTN</td>
<td>70.8</td>
<td>1.4</td>
</tr>
</tbody>
</table>

Sentiment Analysis on IMDB

Textual Entailment on SNLI

(Jia et al., 2019)
Genetic Algorithms to Generate Examples

- Semantics-preserving
- Word-level perturbations
- Grey-box (access to output probabilities)

(Alzantot et al., 2018; Mallawaarachchi, 2017)
Generating Natural Adversarial Examples

- Search in **continuous space** via **sentence embeddings**
- **Black-box, sentence-level** perturbations
- Applied to computer vision as well

\[
\begin{align*}
\min_{x'} \| z - z' \| \\
\text{s.t. } f(x') \neq f(x)
\end{align*}
\]

(Zhao et al., 2018; Singh, 2019)
Semantically Equivalent Adversarial Rules

- Extract general patterns from backtranslation attacks

<table>
<thead>
<tr>
<th>SEAR</th>
<th>Questions / SEAs</th>
<th>f(x)</th>
<th>Flips</th>
</tr>
</thead>
<tbody>
<tr>
<td>What VBZ →</td>
<td>What is What’s the NASUWT?</td>
<td>Trade union</td>
<td>2%</td>
</tr>
<tr>
<td>What’s</td>
<td></td>
<td>Teachers in Wales</td>
<td></td>
</tr>
<tr>
<td>What NOUN →</td>
<td>What resource Which resource was mined in the Newcastle area?</td>
<td>Coal wool</td>
<td>1%</td>
</tr>
<tr>
<td>Which NOUN</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>What VERB →</td>
<td>What was So what was Ghandi’s work called?</td>
<td>Satyagraha</td>
<td>2%</td>
</tr>
<tr>
<td>So what VERB</td>
<td></td>
<td>Civil Disobedience</td>
<td></td>
</tr>
<tr>
<td>What VBD →</td>
<td>What was And what was Kenneth Swezey’s job?</td>
<td>Journalist sleep</td>
<td>2%</td>
</tr>
<tr>
<td>And what VBD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(Ribeiro et al., 2018; Singh, 2019)
Bonus Slide and References
Results: Reasons for Model Success

Models did better with an exact n-gram match between question and paragraph. Models also did better with short questions.

(Jia and Liang, 2017)
References

- Alzantot et al., 2018: https://arxiv.org/abs/1804.07998
- Goodfellow et al., 2015: https://arxiv.org/abs/1412.6572
- Gowal et al., 2018: https://arxiv.org/abs/1810.12715
- Jia et al., 2019: https://arxiv.org/abs/1909.00986
- Mallawaarachchi, 2017: https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3
- Rajpurkar et al., 2016: https://arxiv.org/abs/1606.05250
- Ribeiro et al., 2018: https://www.aclweb.org/anthology/P18-1079.pdf
- Zhao et al., 2018: https://arxiv.org/abs/1710.11342
Thank you

Any questions?