Deep Reinforcement Learning for Mention-Ranking Coreference Models

Kevin Clark and Christopher D. Manning
Stanford University

Presented by Zubin Pahuja
Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.
Coreference Resolution

- Identify all mentions that refer to the same real world entity

_Barak Obama_ nominated Hillary Rodham Clinton as _his_ secretary of state on Monday. _He_ chose her because she had foreign affairs experience as a former First Lady.
Coreference Resolution

• Identify all mentions that refer to the same real world entity
• A document-level structured prediction task

Barack Obama nominated Hillary Rodham Clinton as his secretary of state on Monday. He chose her because she had foreign affairs experience as a former First Lady.
Applications

• Full text understanding
  Information extraction, question answering, summarization

“He was born in 1961”
Applications

• Dialog

“Book tickets to see James Bond”

“Spectre is playing near you at 2:00 and 3:00 today. How many tickets would you like?”

“Two tickets for the showing at three”
Coreference Resolution is Hard!

- “She poured water from the pitcher into the cup until it was full”
- “She poured water from the pitcher into the cup until it was empty”

- The trophy would not fit in the suitcase because it was too big.
- The trophy would not fit in the suitcase because it was too small.
Coreference Resolution is Hard!

• “She poured water from the pitcher into the cup until it was full”
• “She poured water from the pitcher into the cup until it was empty”

• The trophy would not fit in the suitcase because it was too big.
• The trophy would not fit in the suitcase because it was too small.

• These are called Winograd Schema
Three Kinds of Coreference Models

• Mention Pair

• Mention Ranking

• Clustering
“I voted for Nader because he was most aligned with my values,” she said.
Mention Ranking

• Assign each mention its highest scoring candidate antecedent
• Dummy mention **NA** allows model to decline assigning antecedent to current mention
Mention Ranking

• Assign each mention its highest scoring candidate antecedent
• Dummy mention **NA** allows model to decline assigning antecedent to current mention

$$p(NA, she) = 0.1$$
$$p(I, she) = 0.5$$
$$p(Nader, she) = 0.1$$
$$p(he, she) = 0.1$$
$$p(my, she) = 0.2$$
Mention Ranking

• Assign each mention its highest scoring candidate antecedent
• Dummy **NA** mention allows model to decline linking the current mention to anything

\[
\begin{align*}
p(\text{NA, she}) &= 0.1 \\
p(\text{I, she}) &= 0.5 \\
p(\text{Nader, she}) &= 0.1 \\
p(\text{he, she}) &= 0.1 \\
p(\text{my, she}) &= 0.2 \\
\end{align*}
\]
Mention Ranking

- Infer global structure by making a sequence of local decisions
Mention Ranking

• Infer global structure by making a sequence of local decisions
Mention Ranking

• Infer global structure by making a sequence of local decisions
Mention Ranking

- Infer global structure by making a sequence of local decisions
Mention Ranking

• Infer global structure by making a sequence of local decisions
Challenge

How to train a model to make local decisions such that it produces a global structure?
Some Local Decisions Matter More than Others

"it was raining, but the car stayed dry because it was under cover"

Bill Clinton, he, Clinton, Clinton, Hillary, her

Severe error

it, the car, it

Minor error
Prior Work

Heuristically defines which error types are more important than others
Prior Work: Coreference Error Types

1. False New
   - NEW
   - I
   - Nader
   - he
   - my
   - she

2. False Anaphoric
   - NEW
   - I
   - Nader
   - he
   - my
   - she

3. Wrong Link
   - NEW
   - I
   - Nader
   - he
   - my
   - she

Correct Decision
---
Model Decision
Learning Algorithms

Heuristic Loss Function
Prior Work: Heuristic Loss Function

Heuristically costs for mistakes

\[ \Delta_h(c, m_i) = \begin{cases} 
0 & \text{if } c \text{ and } m_i \text{ are coreferent} \\
\alpha_{FN} & \text{if false new error} \\
\alpha_{FA} & \text{if false anaphoric error} \\
\alpha_{WL} & \text{if wrong link error}
\end{cases} \]
Prior Work: Heuristic Loss Function

Max-Margin Loss (Wiseman et al)

$$\max_{c \in \mathcal{C}(m_i)} \Delta_h(c, m_i)(1 + s(c, m_i) - s(t_i, m_i))$$

- max over candidate coreference decisions
- cost for this coref decision
- loss for scoring this decision too highly
Prior Work: Heuristic Loss Function

Disadvantages

• Requires careful tuning of hyperparameters using slow grid search

• Does not generalize across datasets, languages, metrics

• Does not optimize for evaluation metric
  • At best loss is correlated with metric
Reinforcement Learning to the Rescue!

• Does not require hyperparameter training

• Small boost in accuracy
Coref Resolution with Reinforcement Learning

- Model takes a sequence of actions $a_{1:T} = a_1, a_2, ..., a_T$
  - action $a_i = (c, m_i)$ adds a coreference link between the $i^{th}$ mention and candidate antecedent $c$
Coref Resolution with Reinforcement Learning

- After completing a sequence of actions, model receives a reward ($B^3$ metric)

$$R(a_{1:5}) = 100$$
Learning Algorithms

REINFORCE algorithm (Williams, 1992)
REINFORCE Algorithm

- Define probability distribution over actions:
  \[ p_\theta((c, m)) \propto e^{s(c, m)} \text{ for any action } a = (c, m) \]

- Maximize expected reward
  \[ J(\theta) = \mathbb{E}_{a_{1:T} \sim p_\theta} R(a_{1:T}) \]
REINFORCE Algorithm

• Competitive with heuristic loss

• Disadvantage Vs. Max-Margin Loss
  • REINFORCE maximizes performance in expectation
    • We only need the highest scoring action(s) to be correct, not low scoring actions
Combine best of both worlds!

Improve cost-function in Max-Margin Loss
Learning Algorithms

Reward-Rescaling
Reward-Rescaling

• Since actions are independent, we can change an action $a_i$ to a different one $a'_i$ and see what reward we would have gotten instead.
Reward-Rescaling

• Since actions are independent, we can change an action $a_i$ to a different one $a'_i$ and see what reward we would have gotten instead.

Reward = 85
Regret = 15
Reward-Rescaling

• Since actions are independent, we can change an action $a_i$ to a different one $a'_i$ and see what reward we would have gotten instead.
Reward-Rescaling

- Use this idea to do reward-based slack-rescaling

\[ \Delta_r(c, m_i) = \max_{a'_i \in A_i} R(a_1, ..., a'_i, ..., a_T) \quad \text{reward for best action} \]

\[ - R(a_1, ..., (c, m_i), ..., a_T) \quad \text{reward for current action} \]

- Cost is the regret of taking the action
  - Replaces the heuristic cost, otherwise use the same max-margin loss function
Experimental Setup

• English and Chinese CoNLL 2012 Shared Task dataset
• Mentions predicted using Stanford rule-based system (Lee et al, 2011)
• Scores are CoNLL F-1 scores
  • Average of MUC, $B^3$ and CEAF metrics
Neural Mention Ranking Model

Standard feed-forward neural network (Clark and Manning, 2016)
Features

• Word Embeddings
  • Previous two words, first word, last word, **head word** of each mention
  • Groups of words as average of vectors for each word in the group

• Also
  • Distance
  • String Matching
  • Document Genre
  • Speaker Information

• Separate network for anaphrocity scores
## Evaluation

<table>
<thead>
<tr>
<th>Method</th>
<th>CoNLL 2012 English Test Data</th>
<th>CoNLL 2012 Chinese Test Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wiseman et al. (2016)</td>
<td>77.49</td>
<td>69.75</td>
</tr>
<tr>
<td>Clark &amp; Manning (2016)</td>
<td>79.91</td>
<td>69.30</td>
</tr>
<tr>
<td>Heuristic Loss</td>
<td>79.63</td>
<td>70.25</td>
</tr>
<tr>
<td>REINFORCE</td>
<td>80.08</td>
<td>69.61</td>
</tr>
<tr>
<td>Reward Rescaling</td>
<td>79.19</td>
<td>70.44</td>
</tr>
<tr>
<td>B³</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Björkelund &amp; Kuhn (2014)</td>
<td>69.39</td>
<td>62.57</td>
</tr>
<tr>
<td>Clark &amp; Manning (2016)</td>
<td>74.45</td>
<td>64.73</td>
</tr>
<tr>
<td>Heuristic Loss</td>
<td>72.20</td>
<td>66.51</td>
</tr>
<tr>
<td>REINFORCE</td>
<td>74.05</td>
<td>65.38</td>
</tr>
<tr>
<td>Reward Rescaling</td>
<td>73.64</td>
<td>65.62</td>
</tr>
</tbody>
</table>

**Table 1:** Comparison of the methods together with other state-of-the-art approaches on the test sets.
Error Breakdown: Avoiding Costly Mistakes

- Reward-Rescaling makes more errors in total!
  - However, the errors are less severe

<table>
<thead>
<tr>
<th>Model</th>
<th>FN</th>
<th>FA</th>
<th>WL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heuristic Loss</td>
<td>1719</td>
<td>1956</td>
<td>1258</td>
</tr>
<tr>
<td>Reward Rescaling</td>
<td>1725</td>
<td>1994</td>
<td>1247</td>
</tr>
</tbody>
</table>

Table 2: Number of “false new,” “false anaphoric,” and “wrong link” errors produced by the models on the English CoNLL 2012 test set.
Comparison with Heuristic Loss

- High variance in costs for a given error type
  - Distribution of “False New” cost is spread out, so using fixed penalty for an error-type is insufficient

**Figure 1:** Density plot of the costs $\Delta_r$ associated with different error types on the English CoNLL 2012 test set.
Example Improvement: Proper Nouns

- Fewer “false new” errors with proper nouns

<table>
<thead>
<tr>
<th>Class of Mentions</th>
<th>Average Cost $\overline{\Delta_r}$</th>
<th># Heuristic Loss Errors</th>
<th># Reward Rescaling Errors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FN  FA  WL</td>
<td>FN  FA  WL</td>
<td>FN  FA  WL</td>
</tr>
<tr>
<td>Proper nouns</td>
<td>0.90 0.38 1.02</td>
<td>403 597 221</td>
<td>334 660 233</td>
</tr>
<tr>
<td>Pronouns in phone conversations</td>
<td>0.86 0.39 1.21</td>
<td>82 85 81</td>
<td>90 78 67</td>
</tr>
</tbody>
</table>

Table 3: Examples of classes of mention on which the reward-rescaling loss significantly improves upon the heuristic loss due to its reward-based cost function. Reported numbers are from the English CoNLL 2012 test set.
Conclusion

Heuristic Loss < REINFORCE < Reward-Rescaling

• Why?
  • Benefit of Max-Margin Loss
  • Directly optimizes coref metrics rather than heuristic cost function

• Advantages:
  • Does not require hyperparameter training
  • Small boost in accuracy with fewer costly mistakes
Caveats

• Reward metric needs to be fast since it will be computed many times!

• May overfit for evaluation metric
Thank You

Any Questions?