Learning Anaphoricity and Antecedent Ranking Features for Coreference Resolution

Sam Wiseman, Alexander M. Rush, Stuart M. Shieber, and Jason Weston

Facebook AI Research
Cadillac posted a 3.2% increase despite new competition from Lexus, the fledgling luxury-car division of Toyota Motor Corp. Lexus sales weren’t available; the cars are imported and Toyota reports their sales only at month-end.
Cadillac posted a 3.2% increase despite new competition from [Lexus, the fledgling luxury-car division of [Toyota Motor Corp]]. [Lexus] sales weren’t available; the cars are imported and [Toyota] reports [their] sales only at month-end.
Mention Ranking

- Model each mention $x$ as having a single “true” antecedent
- Score potential antecedents $y$ of each mention $x$ with a scoring function $s(x, y)$
  - Common to use $s_{\text{lin}}(x, y) \triangleq \mathbf{w}^T \tilde{\phi}(x, y)$ as scoring function
- Predict $y^* = \arg \max_{y \in \mathcal{Y}(x)} s(x, y)$
- If only clusters annotated, “true” antecedent a latent variable when training

\[
\begin{align*}
    s(x, y_1) &= 0.4 \\
    s(x, y_2) &= 0.9
\end{align*}
\]

\begin{itemize}
    \item \[\ldots \text{[Lexus] sales weren’t available} \ldots \text{[Toyota] reports [their]}\]
\end{itemize}
[Cadillac] posted a [3.2% increase] despite [new competition from [Lexus, the fledgling luxury-car division of [Toyota Motor Corp]]]. [[Lexus sales] weren’t available; [the cars] are imported and [Toyota] reports [[their] sales] only at [month-end].
Also score possibility that \( x \) non-anaphoric, denoted by \( y = \epsilon \)

Can still use \( s_{\text{lin}}(x, y) \triangleq w^T \tilde{\phi}(x, y) \) as scoring function

Now \( Y(x) = \{ \text{mentions before } x \} \cup \{ \epsilon \} \)

Again predict \( y^* = \arg \max_{y \in Y(x)} s(x, y) \)

\[
\begin{align*}
\text{... [the cars]} & \quad \text{are imported and} \quad [\text{Toyota}] \quad \text{reports} \quad [\text{their}] \\
s(x, y_1) &= 1.2 \\
s(x, y_2) &= 0.9 \\
s(x, \epsilon) &= -1.8
\end{align*}
\]
Can duplicate features for a more flexible model:

$$s_{\text{lin}+}(x, y) \triangleq \begin{cases} u^T \left[ \frac{x}{(x,y)} \right] & \text{if } y \neq \epsilon \\ v^T(x) & \text{if } y = \epsilon \end{cases}$$

- features on mention context (capture anaphoricity info)
- features on mention, antecedent pair (capture pairwise affinity)
- Above equivalent to model of ?
Cadillac posted a 3.2% increase despite new competition from Lexus, the fledgling luxury-car division of Toyota Motor Corp. Lexus sales weren’t available; the cars are imported and Toyota reports their sales only at month-end.

Misleading Head Matches
[Lexus sales] and [their sales] not coreferent!
Cadillac posted a 3.2% increase despite new competition from Lexus, the fledgling luxury-car division of Toyota Motor Corp. Lexus sales weren’t available; the cars are imported and Toyota reports their sales only at month-end.

Misleading Number Matches
[the cars] and [their] not coreferent!
Simple Antecedent/Pairwise Features Not Discriminative

E.g., is [Lexus sales] the antecedent of [their sales]?

- Common antecedent features: String/Head Match, Sentences Between, Mention-Antecedent Numbers/Heads/Genders, etc.

$$\phi_p([\text{their sales}],[\text{Lexus sales}]) = \begin{cases} 
\text{string-match}=\text{false} \\
\text{head-match}=\text{true} \\
\text{sentences-between}=0 \\
\text{ment-ant-numbers}=\text{plur., plur.} \\
\vdots
\end{cases}$$
Finding discriminative features a major challenge for coreference systems [??]

Typical to define (or search for) feature conjunction-schemes to improve predictive performance [??]. For instance:

- string-match\((x, y) \land \text{type}(x) \land \text{type}(y)\) [??], where

\[
\text{type}(x) = \begin{cases} 
\text{Nom.} & \text{if } x \text{ is nominal} \\
\text{Prop.} & \text{if } x \text{ is proper} \\
\text{citation-form}(x) & \text{if } x \text{ is pronominal}
\end{cases}
\]

- substring-match(head\((x), y\)) \land \text{substring-match}(x, \text{head}(y)) \land \text{coarse-type}(y) \land \text{coarse-type}(x) [??]

Not just a problem for Mention Ranking systems!
Our Approach

**Motivation:** Current conjunction schemes perhaps not optimal, and in any case hard to scale as more features added.

Accordingly, we:

- Develop a model that learns good representations automatically
- Use only raw, unconjoined features
- Introduce pre-training scheme to improve quality of learned representations
Extending the Piecewise Model I

Goal: learn higher order feature representations

We first define the following nonlinear feature representations:

\[ h_a(x) \triangleq \tanh(W_a \phi_a(x) + b_a) \]
\[ h_p(x, y) \triangleq \tanh(W_p \phi_p(x, y) + b_p) \]

- Here, \( \phi_a, \phi_p \) are raw, unconjoined features!
Use the scoring function

\[ s(x, y) \triangleq \begin{cases} 
  u^T g(\begin{bmatrix} h_a(x) \\ h_p(x,y) \end{bmatrix}) + u_0 & \text{if } y \neq \epsilon \\
  v^T h_a(x) + v_0 & \text{if } y = \epsilon 
\end{cases} \]

**($g_1$)** If $g$ is identity, obtain version of $s_{\text{lin}+}$ with nonlinear features.

**($g_2$)** If $g$ is an additional hidden layer, further encourage nonlinear interactions between $h_a, h_p$. 
To train, we use the following margin-based loss:

$$L(\theta) = \sum_{n=1}^{N} \max_{\hat{y} \in \mathcal{Y}(x_n)} \Delta(x_n, \hat{y})(1 + s(x_n, \hat{y}) - s(x_n, y^\ell_n)) + \lambda ||\theta||_1$$

- Slack-rescale with a mistake-specific cost function $\Delta(x_n, \hat{y})$
- $y^\ell_n$ a latent antecedent: equal to highest scoring antecedent in same cluster (or $\epsilon$) [???]
- Note that even if $s$ were linear, would still be non-convex!
Pre-training Subtasks I

Two very natural subtasks for pre-training $h_a$ and $h_p$

**Antecedent Ranking**
Predict antecedents of known anaphoric mentions with scoring function

$$s_p(x, y) \triangleq u_p^T h_p(x, y) + v_0$$

**Anaphoricity Detection**
Predict anaphoricity of mentions with scoring function

$$s_a(x) \triangleq v_a^T h_a(x) + v_0$$

- We use similar, margin-based objectives for training
Two very natural subtasks for pre-training $h_a$ and $h_p$

### Antecedent Ranking

Predict antecedents of known anaphoric mentions with scoring function

$$s_p(x, y) \triangleq u_p^T h_p(x, y) + v_0$$

### Anaphoricity Detection

Predict anaphoricity of mentions with scoring function

$$s_a(x) \triangleq v_a^T h_a(x) + v_0$$

- We use similar, margin-based objectives for training
**Pre-training Subtasks I**

Two very natural subtasks for pre-training $h_a$ and $h_p$

### Antecedent Ranking

Predict antecedents of known anaphoric mentions with scoring function

$$s_p(x, y) \triangleq u_p^T h_p(x, y) + v_0$$

### Anaphoricity Detection

Predict anaphoricity of mentions with scoring function

$$s_a(x) \triangleq v_a^T h_a(x) + v_0$$

- We use similar, margin-based objectives for training
Pre-training Subtasks I

Two very natural subtasks for pre-training $h_a$ and $h_p$

Antecedent Ranking
Predict antecedents of known anaphoric mentions with scoring function

$$s_p(x, y) \triangleq u_p^T h_p(x, y) + v_0$$

Anaphoricity Detection
Predict anaphoricity of mentions with scoring function

$$s_a(x) \triangleq v_a^T h_a(x) + v_0$$

- We use similar, margin-based objectives for training
Antecedent ranking of known anaphoric mentions very similar to “gold mention” version of coreference task (but slightly easier)

Anaphoricity/Singleton detection has a long history in coreference resolution, generally as an initial step in a pipeline
Subtask Performance

Figure: Anaphoricity Detection $F_1$ Score

- Subtask performance itself not crucial, but want to see that networks can learn good representations

Figure: Antecedent Ranking Accuracy
Experimental Setup

- Used standard CoNLL 2012 English dataset experimental split
- Results scored with CoNLL 2012 scoring script v8.01
- Used Berkeley Coreference System for mention extraction
- All optimization with Composite Mirror-Descent flavor of AdaGrad
- All hyperparameters (learning rates and regularization coefficients) tuned with grid-search on development set
Main Results

Figure: Results on CoNLL 2012 English test set. We compare with (in order) ?, ?, ?, and ?. F₁ gains are significant ($p < 0.05$) compared with both B&K and D&K for all metrics.
**Main Results (Full Table)**

<table>
<thead>
<tr>
<th></th>
<th>MUC</th>
<th></th>
<th></th>
<th>B$^3$</th>
<th></th>
<th></th>
<th>CEAF$_e$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F$_1$</td>
<td>P</td>
<td>R</td>
<td>F$_1$</td>
<td>P</td>
<td>R</td>
<td>F$_1$</td>
</tr>
<tr>
<td>BCS</td>
<td>74.89</td>
<td>67.17</td>
<td>70.82</td>
<td>64.26</td>
<td>53.09</td>
<td>58.14</td>
<td>58.12</td>
<td>52.67</td>
<td>55.27</td>
</tr>
<tr>
<td>Ma et al.</td>
<td>81.03</td>
<td>66.16</td>
<td><strong>72.84</strong></td>
<td>66.90</td>
<td>51.10</td>
<td>57.94</td>
<td>68.75</td>
<td>44.34</td>
<td>53.91</td>
</tr>
<tr>
<td>B&amp;K</td>
<td>74.30</td>
<td>67.46</td>
<td>70.72</td>
<td>62.71</td>
<td>54.96</td>
<td>58.58</td>
<td>59.40</td>
<td>52.27</td>
<td>55.61</td>
</tr>
<tr>
<td>D&amp;K</td>
<td>72.73</td>
<td>69.98</td>
<td>71.33</td>
<td>61.18</td>
<td>56.60</td>
<td>58.80</td>
<td>56.20</td>
<td>54.31</td>
<td>55.24</td>
</tr>
<tr>
<td>NN($g_2$)</td>
<td>76.96</td>
<td>68.10</td>
<td>72.26</td>
<td>66.90</td>
<td>54.12</td>
<td>59.84</td>
<td>59.02</td>
<td>53.34</td>
<td>56.03</td>
</tr>
<tr>
<td>NN($g_1$)</td>
<td>76.23</td>
<td>69.31</td>
<td>72.60</td>
<td>66.07</td>
<td>55.83</td>
<td><strong>60.52</strong></td>
<td>59.41</td>
<td>54.88</td>
<td><strong>57.05</strong></td>
</tr>
</tbody>
</table>

**Table:** Results on CoNLL 2012 English test set. We compare with (in order) ?, ?, ?, and ?. F$_1$ gains are significant ($p < 0.05$ under the bootstrap resample test $?$) compared with both B&K and D&K for all metrics.
## Model Ablations

<table>
<thead>
<tr>
<th>Model</th>
<th>MUC</th>
<th>B$^3$</th>
<th>CEAF$_e$</th>
<th>CoNLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Layer MLP</td>
<td>71.80</td>
<td>60.93</td>
<td>57.51</td>
<td>63.41</td>
</tr>
<tr>
<td>2 Layer MLP</td>
<td>71.77</td>
<td>60.84</td>
<td>57.05</td>
<td>63.22</td>
</tr>
<tr>
<td>$g_1$</td>
<td>71.92</td>
<td>61.06</td>
<td>57.59</td>
<td>63.52</td>
</tr>
<tr>
<td>$g_1$ + pre-train</td>
<td>72.74</td>
<td>61.77</td>
<td>58.63</td>
<td>64.38</td>
</tr>
<tr>
<td>$g_2$</td>
<td>72.31</td>
<td>61.79</td>
<td>58.06</td>
<td>64.05</td>
</tr>
<tr>
<td>$g_2$ + pre-train</td>
<td>72.68</td>
<td>61.70</td>
<td>58.32</td>
<td>64.23</td>
</tr>
</tbody>
</table>

**Table:** $F_1$ performance on CoNLL 2012 development set

- Top sub-table examines whether separating $h_p$, $h_a$ (in first layer) actually helpful
- Bottom two sub-tables examine whether pre-training is helpful
## Scaling to More Features

<table>
<thead>
<tr>
<th>Model</th>
<th>Features</th>
<th>MUC</th>
<th>B³</th>
<th>CEAF e</th>
<th>CoNLL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lin.</td>
<td></td>
<td>70.44</td>
<td>59.10</td>
<td>55.57</td>
<td>61.71</td>
</tr>
<tr>
<td>NN ($g_2$)</td>
<td><strong>Basic</strong></td>
<td>71.59</td>
<td>60.56</td>
<td>57.45</td>
<td>63.20</td>
</tr>
<tr>
<td>NN ($g_1$)</td>
<td></td>
<td>71.86</td>
<td>60.9</td>
<td>57.90</td>
<td>63.55</td>
</tr>
<tr>
<td>Lin.</td>
<td></td>
<td>70.92</td>
<td>60.05</td>
<td>56.39</td>
<td>62.45</td>
</tr>
<tr>
<td>NN ($g_2$)</td>
<td><strong>Basic+</strong></td>
<td>72.68</td>
<td>61.70</td>
<td>58.32</td>
<td>64.23</td>
</tr>
<tr>
<td>NN ($g_1$)</td>
<td></td>
<td>72.74</td>
<td>61.77</td>
<td>58.63</td>
<td>64.38</td>
</tr>
</tbody>
</table>

**Table:** $F_1$ performance comparison between state-of-the-art linear mention-ranking model and our full models on CoNLL 2012 development set for different feature sets.
Mention Ranking models make error analysis very simple:

- Highest percentage error \( \frac{736}{1000} \) on anaphoric mentions with no previous occurring head-match
  - e.g., [the team] and [the New York Giants]
- Highest number of errors \( \frac{1309}{7300} \) on anaphoric pronouns
  - Almost all were errors on pleonastic pronouns (“it”, “you”). About 2/3 involved incorrectly predicting another instance of same pronoun as antecedent.
- An argument for more structure?
  - 30% of anaphoric pronominal mentions in CoNLL dev data are in pronoun-only clusters!
Summary

(1) Possible to achieve state-of-the-art performance with
- Very simple, local model and powerful scoring function
  - Note most recent state-of-the-art models non-local!
- Only raw, unconjoined features
- Over 1.5 pt increase over previous state-of-the-art in CoNLL score

(2) Separating anaphoricity and antecedent ranking (learned) representations beneficial
- Natural to pre-train on corresponding subtasks
Note that Mention Ranking models make error analysis very simple!

Three Kinds of Errors Possible

(Adopting terminology of ??):

- **False Link** errors: predicting a mention to be anaphoric when it is non-anaphoric
- **False New** errors: predicting a mention to be non-anaphoric when it is anaphoric
- **Wrong Link** errors: predicting an incorrect antecedent for an anaphoric mention
### Discussion: What are we getting wrong?

<table>
<thead>
<tr>
<th></th>
<th>Singleton</th>
<th>1&lt;sup&gt;st&lt;/sup&gt; in clust.</th>
<th>Anaphoric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ment. w/ prev. head match</td>
<td>817</td>
<td>147</td>
<td>700 + 318</td>
</tr>
<tr>
<td>Ment. w/o prev. head match</td>
<td>86</td>
<td>41</td>
<td>677 + 59</td>
</tr>
<tr>
<td>Pronominal mentions</td>
<td>948</td>
<td>257</td>
<td>434 + 875</td>
</tr>
</tbody>
</table>

Largest % error on anaphoric mentions with no previous head match:

- The classic “hard” coreference case, presumably requiring knowledge, understanding

But make **most** errors (by far) on pronouns!
Pronoun Problems

Which pronominal mentions are we missing?

- FL and WL pronominal errors almost entirely on pleonastic pronominal mentions (e.g., “it”, “you”)
- Predicted antecedent almost always (another instance of) same pronoun

An argument for non-local inference?

- Note that 30% of anaphoric pronominal mentions in CoNLL development data in pronoun-only clusters
Thanks!
<table>
<thead>
<tr>
<th><strong>All Features</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mention Features ($\phi_a$)</td>
</tr>
<tr>
<td>Mention Head</td>
</tr>
<tr>
<td>Mention First Word</td>
</tr>
<tr>
<td>Mention Last Word</td>
</tr>
<tr>
<td>Word Preceding Mention</td>
</tr>
<tr>
<td>Word Following Mention</td>
</tr>
<tr>
<td># Words in Mention</td>
</tr>
<tr>
<td>Mention Synt. Ancestry</td>
</tr>
<tr>
<td>Mention Type</td>
</tr>
<tr>
<td>Mention Governor</td>
</tr>
<tr>
<td>Mention Sentence Index</td>
</tr>
<tr>
<td>Mention Entity Type</td>
</tr>
<tr>
<td>Mention Number</td>
</tr>
<tr>
<td>Mention Animacy</td>
</tr>
<tr>
<td>Mention Gender</td>
</tr>
<tr>
<td>Mention Person</td>
</tr>
<tr>
<td><strong>Pairwise Features ($\phi_p$)</strong></td>
</tr>
<tr>
<td>$\phi_a$(Mention); $\phi_a$(Antecedent)</td>
</tr>
<tr>
<td>Mentions between Ment., Ante.</td>
</tr>
<tr>
<td>Sentences between Ment., Ante.</td>
</tr>
<tr>
<td>i-within-i</td>
</tr>
<tr>
<td>Same Speaker</td>
</tr>
<tr>
<td>Document Type</td>
</tr>
<tr>
<td>Ante., Ment. String Match</td>
</tr>
<tr>
<td>Ante. contains Ment.</td>
</tr>
<tr>
<td>Ment. contains Ante.</td>
</tr>
<tr>
<td>Ante. contains Ment. Head</td>
</tr>
<tr>
<td>Mention contains Ante. Head</td>
</tr>
<tr>
<td>Ante., Ment. Head Match</td>
</tr>
<tr>
<td>Ante., Ment. Synt. Ancestries; Numbers; Genders; Persons; Entity Types; Heads; Types</td>
</tr>
</tbody>
</table>
Can get up to 83.3462 on dev full task get: received MUC: 75.980000 69.490000 72.590000 ESC received BCUB: 66.490000 58.030000 61.970000 received CEAFe: 61.120000 56.490000 58.710000 received CoNLL: 64.423333