Supervised Noun Phrase Coreference Research: The First Fifteen Years

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
My first ACL talk

- “Improving Machine Learning Approaches to Coreference Resolution” (Ng & Cardie, 2002)
  - Proposed linguistic and extra-linguistic extensions to Soon et al.’s (2001) system
  - The mention-pair model
Goal

Survey the major milestones in supervised noun phrase coreference research in the past fifteen years (1994-2009)
Goal

Survey the major milestones in supervised noun phrase coreference research in the past fifteen years (1994-2009)

Focuses on:
- Within-document coreference (no cross-doc coreference)
- Identity coreference (no bridging references, ..)
Areas Covered

- Supervised models
- Linguistic features
- Publicly-available annotated coreference corpora
- Evaluation issues
Areas Covered

- Supervised models
- Linguistic features
- Publicly-available annotated coreference corpora
- Evaluation issues
Noun Phrase Coreference Resolution

- Identify the noun phrases (NPs) in a text that refer to the same real-world entity

- Inherently a clustering problem
  - Goal: produce a partition of the NPs
Standard Supervised Approach

- **Step 1**: Learn a coreference model
  - $CM: NP_i \times NP_j \ [0, 1]$ from **annotated** data

![Diagram showing coreference model with examples like Mr. Clinton, Clinton, she, and their coreference relationships.](image)
Standard Supervised Approach

- **Step 1**: Learn a coreference model
  - \( CM: \text{NP}_i \times \text{NP}_j \quad [0, 1] \) from *annotated* data

- **Step 2**: Apply a clustering algorithm
  - coordinates the pairwise classification decisions
Standard Supervised Approach

- **Step 1**: Learn a coreference model
  - $CM: NP_i \times NP_j \quad [0, 1]$ from annotated data

- **Step 2**: Apply a clustering algorithm
  - coordinates the pairwise classification decisions
Mention-Pair Model

- a classifier that determines whether two NPs are coreferent
- Train the model using any off-the-shelf machine learner
- Apply the model to a test text to determine whether two NPs are coreferent
Mention-Pair Model

- a classifier that determines whether two NPs are coreferent
- Train the model using any off-the-shelf machine learner
- Apply the model to a test text to determine whether two NPs are coreferent

- Need a clustering algorithm to coordinate the pairwise coreference decisions
  - many clustering algorithms have been used
  - three types of clustering algorithms
Really Greedy Clustering Algorithms

- Single-link clustering (Soon et al., 2001)
  - For each NP$_j$, select as its antecedent the closest preceding NP that is determined as coreferent with it.
  - Posit NP$_j$ as non-anaphoric if no preceding NP is coreferent with it.

- Best-first clustering (Ng & Cardie, 2002)
  - Same as single-link clustering, except that we select as the antecedent the NP that has the highest coreference likelihood.
Why are they really greedy?

- Clusters are formed based on a small subset of the pairwise coreference decisions
  - Many pairwise decisions are not used in the clustering process
Why are they really greedy?

- Clusters are formed based on a small subset of the pairwise coreference decisions
  - Many pairwise decisions are not used in the clustering process
Why are they really greedy?

- Clusters are formed based on a small subset of the pairwise coreference decisions
  - Many pairwise decisions are not used in the clustering process
Less Greedy Clustering Algorithms

- Use all the pairwise coreference decisions

- **Graph partitioning algorithms**
  - each text is represented as a graph
    - each vertex corresponds to a NP; edge weight is coref likelihood
    - Goal: partition the graph nodes to form coreference clusters
Less Greedy Clustering Algorithms

- Use all the pairwise coreference decisions

- **Graph partitioning algorithms**
  - each text is represented as a graph
    - each vertex corresponds to a NP; weight of an edge indicates the likelihood that the two NPs are coreferent
    - Goal: partition the graph nodes to form coreference clusters
  - Correlation clustering (e.g., McCallum & Wellner (2004))
    - cluster that respects as many pairwise decisions as possible
  - Minimum-cut-based clustering (Nicolae & Nicolae, 2006)
    - Find the mincut of the graph and partition the graph nodes; repeat until some stopping criterion is reached
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions

Recast as a search problem
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions

```
```

Recast as a search problem

```
```

Expands the most promising paths
Time-Aware Clustering Algorithms

- Later coreference decisions depend on the earlier ones
- Luo et al. (2004): Bell-tree clustering
  - Bell tree: represents the space of possible NP partitions

Recast as a search problem
- Expands the most promising paths
- Scores a path based on pairwise probabilities
Which clustering algorithm is the best?

- Few empirical comparisons
- Luo et al. (2004) didn’t compare their Bell-tree approach against the really greedy algorithms
Which clustering algorithm is the best?

- Few empirical comparisons

- Luo et al. (2004) didn’t compare their Bell-tree approach against the really greedy algorithms
  - Klein (2005, pc): search space is too large, need to apply a lot of heuristics to prune the space, making it a greedy algorithm
Which clustering algorithm is the best?

- Few empirical comparisons

- Luo et al. (2004) didn’t compare their Bell-tree approach against the really greedy algorithms
  - Klein (2005, pc): search space is too large, need to apply a lot of heuristics to prune the space, making it a greedy algorithm
  - Nicolae & Nicolae (2006): not much difference in performance between Bell tree clustering and the really greedy algorithms
Supervised Coreference (Recap)

• **Step 1**: Learn a coreference model

• **Step 2**: Apply a clustering algorithm
Supervised Coreference (Recap)

- **Step 1**: Learn a coreference model
  - Mention-pair model

- **Step 2**: Apply a clustering algorithm
  - Really greedy algorithms
  - Less greedy algorithms
  - Time-aware algorithms
Weaknesses of the Mention-Pair Model

• **Limited expressiveness**
  - information extracted from two NPs may not be sufficient for making an informed coreference decision

• **Can’t determine which candidate antecedent is the best**
  - only determine how good a candidate is relative to NP to be resolved, not how good it is relative to the others
Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two NPs may not be sufficient for making an informed coreference decision

- **Can’t determine which candidate antecedent is the best**
  - only determine how good a candidate is relative to NP to be resolved, not how good it is relative to the others
Want a coreference model that can tell us how likely “she” and a preceding cluster of “she” are coreferent.
The Entity-Mention Model

- a classifier that determines whether (or how likely) an NP belongs to a preceding coreference cluster

- more **expressive** than the mention-pair model
  - can employ **cluster-level** features defined over any subset of NPs in a preceding cluster

- addresses the expressiveness problem

Pasula et al. (2003), Luo et al. (2004), Yang et al. (2004, 2008), Daume & Marcu (2005), Culotta et al. (2007), …
Weaknesses of the Mention-Pair Model

- **Limited expressiveness**
  - information extracted from two NPs may not be sufficient for making an informed coreference decision

- **Can’t determine which candidate antecedent is the best**
  - only determine how good a candidate is relative to NP to be resolved, not how good it is relative to the others
How to address this problem?

- Idea: train a model that imposes a **ranking** on the candidate antecedents for an NP to be resolved
  - so that it assigns the highest rank to the correct antecedent
How to address this problem?

- Idea: train a model that imposes a **ranking** on the candidate antecedents for an NP to be resolved
  - so that it assigns the highest rank to the correct antecedent

- A ranker allows all candidate antecedents to be considered simultaneously and captures competition among them
  - allows us find the best candidate antecedent for an NP

- There is a natural resolution strategy for a ranking model
  - An NP is resolved to the highest-ranked candidate antecedent
How to train a ranking model?

- Convert the problem of ranking $m$ NPs into the a set of pairwise ranking problems
  - Each pairwise ranking problem involves determining which of two candidate antecedents is better for an NP to be resolved
    - Each one is essentially a classification problem
How to train a ranking model?

- Convert the problem of ranking $m$ NPs into a set of pairwise ranking problems
  - Each pairwise ranking problem involves determining which of two candidate antecedents is better for an NP to be resolved
  - Each one is essentially a classification problem

- First supervised coreference model: **Connolly et al. (1994)**
  - Train a decision tree to determine which of the two candidate antecedents of an NP is more likely to be its antecedent
  - During testing, need to heuristically combine the pairwise ranking results to select an antecedent for each NP
Revival of the Ranking Approach

- The ranking model is theoretically better but far less popular than the mention-pair model in the decade following its proposal.

- Rediscovered almost ten years later independently by:
  - Yang et al. (2003): twin-candidate model
  - Iida et al. (2003): tournament model
The Mention-Ranking Model

- Denis & Baldridge (2007, 2008): train the ranker using maximum entropy
  - model outputs a rank value for each candidate antecedent
  - obviates need to heuristically combine pairwise ranking results
The Mention-Ranking Model

- Denis & Baldridge (2007, 2008): train the ranker using maximum entropy
  - model outputs a rank value for each candidate antecedent
  - obviates need to heuristically combine pairwise ranking results
Caveat

- Since a ranker only imposes a ranking on the candidates, it cannot determine whether an NP is anaphoric
  - Need to train a classifier to determine if an NP is anaphoric
Recap

<table>
<thead>
<tr>
<th>Problem</th>
<th>Entity Mention</th>
<th>Mention Ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Limited expressiveness</td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>Cannot determine best candidate</td>
<td>✗</td>
<td>✓</td>
</tr>
</tbody>
</table>

Can we combine the strengths of these two models?
Consider preceding clusters, not candidate antecedents.

Mention-ranking model: Rank candidate antecedents.

Entity-mention model:
Consider preceding clusters,
not candidate antecedents

Rank preceding clusters
The Cluster-Ranking Model

Mention-ranking model

- Rank candidate antecedents

Entity-mention model

- Consider preceding clusters, not candidate antecedents
- Rank preceding clusters
The Cluster-Ranking Model (Rahman & Ng, 2009)

- **Training**
  - train a *ranker* to rank preceding clusters

- **Testing**
  - resolve each NP to the highest-ranked preceding cluster
The Cluster-Ranking Model (Rahman & Ng, 2009)

- **Training**
  - train a ranker to rank preceding clusters

- **Testing**
  - resolve each NP to the highest-ranked preceding cluster

Lappin & Leass’s (1994) heuristic pronoun resolver
The Cluster-Ranking Model (Rahman & Ng, 2009)

- As a ranker, the cluster-ranking model cannot determine whether an NP is anaphoric
  - Before resolving an NP, we still need to use an anaphoricity classifier to determine if it is anaphoric
    - yields a **pipeline** architecture

- Potential problem
  - errors made by the anaphoricity classifier will be propagated to the coreference resolver

- Solution
  - **joint learning** for anaphoricity and coreference resolution
Some Empirical Results on ACE 2005

<table>
<thead>
<tr>
<th>Model</th>
<th>B³</th>
<th>CEAF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Mention-Pair Baseline</td>
<td>50.8</td>
<td>57.9</td>
</tr>
<tr>
<td>Entity-Mention Baseline</td>
<td>51.2</td>
<td>57.8</td>
</tr>
<tr>
<td>Mention-Ranking Baseline (Pipeline)</td>
<td>52.3</td>
<td>61.8</td>
</tr>
<tr>
<td>Mention-Ranking Baseline (Joint)</td>
<td>50.4</td>
<td>65.5</td>
</tr>
<tr>
<td>Cluster-Ranking Model (Pipeline)</td>
<td>55.3</td>
<td>63.7</td>
</tr>
<tr>
<td>Cluster-Ranking Model (Joint)</td>
<td>54.4</td>
<td>70.5</td>
</tr>
</tbody>
</table>
Some Empirical Results on ACE 2005

<table>
<thead>
<tr>
<th></th>
<th>B³</th>
<th>CEAF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Mention-Pair Baseline</td>
<td>50.8</td>
<td>57.9</td>
</tr>
<tr>
<td>Entity-Mention Baseline</td>
<td>51.2</td>
<td>57.8</td>
</tr>
<tr>
<td>Mention-Ranking Baseline (Pipeline)</td>
<td>52.3</td>
<td>61.8</td>
</tr>
<tr>
<td>Mention-Ranking Baseline (Joint)</td>
<td>50.4</td>
<td>65.5</td>
</tr>
<tr>
<td>Cluster-Ranking Model (Pipeline)</td>
<td>55.3</td>
<td>63.7</td>
</tr>
<tr>
<td>Cluster-Ranking Model (Joint)</td>
<td>54.4</td>
<td>70.5</td>
</tr>
</tbody>
</table>

- Cluster ranking is better than mention ranking, which in turn is better than the entity-mention model and the mention-pair model.
- Joint models perform better than pipeline models.
Summary

- A lot of progress in supervised coreference modeling
  - the mention-pair model is theoretically unappealing
    - it makes coreference decisions based on only two NPs

- The cluster-ranking model
  - resembles Lappin & Leass’s (1994) heuristic pronoun resolver
  - narrows the gap between the sophistication of heuristic-based coref models and the simplicity of learning-based coref models
Concluding Remarks

- To ensure progress, new coreference results should be compared against a baseline stronger than Soon et al. (2001)
Concluding Remarks

- To ensure progress, new coreference results should be compared against a baseline stronger than Soon et al. (2001)

- Publicly available coreference systems
  - The mention-pair model
    - JavaRAP (Qiu et al., 2004)
    - GuiTAR (Poesio & Kabadjov, 2004)
    - BART (Versley et al., 2008)
    - The Illinois Coreference Package (Bengtson & Roth, 2008)
    - Reconcile (Stoyanov et al., 2010)
  - The mention-ranking model
    - CoRTex (Denis & Baldridge, 2008)
  - The cluster-ranking model
    - CherryPicker (Rahman & Ng, 2009)