Unsupervised Models for Coreference Resolution

Vincent Ng
Human Language Technology Research Institute
University of Texas at Dallas
Coreference

- Identify the noun phrases (or *mentions*) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Coreference

- Identify the noun phrases (or mentions) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Coreference

- Identify the noun phrases (or mentions) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Coreference

- Identify the noun phrases (or *mentions*) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Coreference

- Identify the noun phrases (or mentions) that refer to the same real-world entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Coreference

- Identify the noun phrases (or *mentions*) that refer to the same real-world entity

Queen Elizabeth set about transforming *her* husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help *the King* overcome *his* speech impediment...

- Lots of prior work on *supervised* coreference resolution
  - Soon et al. (2001), Strube et al. (2002), Yang et al. (2003), Luo et al. (2004), Denis and Baldridge (2007), …
Unsupervised Coreference Resolution

Perform coreference resolution using little or no annotated data
Previous Work

- Apply a weakly supervised or unsupervised learning algorithm to **pronoun resolution**
  - **co-training** (Müller et al., 2002)
  - **self-training** (Kehler et al., 2004)
  - **EM** (Cherry and Bergsma, 2005)
Previous Work

- Apply a weakly supervised or unsupervised learning algorithm to **pronoun resolution**
  - co-training (Müller et al., 2002)
  - self-training (Kehler et al., 2004)
  - EM (Cherry and Bergsma, 2005)

- A **nonparametric fully-Bayesian approach** to unsupervised coreference resolution (Haghighi and Klein, 2007)
Goals

- Design a new model for unsupervised coreference resolution
- Improve Haghighi and Klein’s model with three modifications
Unsupervised Coreference as EM Clustering

- Design a generative model that can be used to induce a clustering of the mentions in a given document
Representing a Clustering

- A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent
Representing a Clustering

- A **clustering** $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.
Representing a Clustering

A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.
A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.
A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.

Don’t care about diagonal entries
Representing a Clustering

- A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.

Don’t care about entries below the diagonal.
A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.
A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.

Valid
A clustering $C$ of $n$ mentions is an $n \times n$ Boolean matrix, where $C_{ij} = 1$ iff mentions $i$ and $j$ are coreferent.

**Valid**

**Invalid**
The Generative Model

- Given a document $D$,
  - generate a clustering $C$ according to $P(C)$
  - generate $D$ given $C$

\[ P(D, C) = P(C) P(D|C) \]
The Generative Model

- Given a document $D$,
  - generate a clustering $C$ according to $P(C)$
  - generate $D$ given $C$

$$P(D, C) = P(C)P(D|C)$$

How to generate $D$ given $C$?
The Generative Model

- Given a document D,
  - generate a clustering C according to P(C)
  - generate D given C

\[ P(D, C) = P(C) \cdot P(D|C) \]

How to generate D given C?
- Assume that D is represented by its mention pairs
The Generative Model

- Given a document D,
  - generate a clustering C according to \( P(C) \)
  - generate D given C

\[
P(D, C) = P(C)P(D|C)
\]

How to generate D given C?
- Assume that D is represented by its mention pairs
- To generate D, generate all pairs of mentions in D
  - (Queen Elizabeth, her), (Queen Elizabeth, husband),
    (Queen Elizabeth, King George VI), …
The Generative Model

- Given a document D,
  - generate a clustering C according to P(C)
  - generate D given C

\[
P(D, C) = P(C) P(D|C) = P(C) P(mp_{12}, mp_{13}, mp_{14}, ..., |C)
\]
The Generative Model

- Given a document D,
  - generate a clustering C according to P(C)
  - generate D given C

\[ P(D, C) = P(C) P(D | C) = P(C) P(mp_{12}, mp_{13}, mp_{14} \ldots | C) \]

\( mp_{ij} \) is the pair formed from mention i and mention j
The Generative Model

- Given a document D,
  - generate a clustering C according to P(C)
  - generate D given C

\[ P(D, C) = P(C) P(D | C) = P(C) P(mp_{12}, mp_{13}, mp_{14}, \ldots | C) \]

Let’s simplify this term
The Generative Model

- Given a document D,
  - generate a clustering C according to P(C)
  - generate D given C

\[ P(D, C) = P(C) P(D|C) \]

\[ = P(C) P(mp_{12}, mp_{13}, mp_{14} \ldots | C) \]

Let’s simplify this term
- assume that each mention pair mp_{ij} is generated conditionally independently given C_{ij}
The Generative Model

- Given a document $D$,
  - generate a clustering $C$ according to $P(C)$
  - generate $D$ given $C$

\[
P(D, C) = P(C) P(D|C)
\]
\[
= P(C) P(mp_{12}, mp_{13}, mp_{14} \ldots | C)
\]
\[
= P(C) \prod_{Pairs(D)} P(mp_{ij}|C_{ij})
\]
The Generative Model

- Given a document $D$,
  - generate a clustering $C$ according to $P(C)$
  - generate $D$ given $C$

$$P(D, C) = P(C) P(D | C)$$

$$= P(C) P(mp_{12}, mp_{13}, mp_{14}, ..., | C)$$

$$= P(C) \prod_{Pairs(D)} P(mp_{ij} | C_{ij})$$

How to represent a mention pair $mp_{ij}$?
## Features

- Use 7 linguistic features divided into 3 groups

| Strong Coreference Indicators | String match  
|                             | Appositive  
|                             | Alias (one is an acronym or abbreviation of the other)  
| Linguistic Constraints       | Gender agreement  
|                             | Number agreement  
|                             | Semantic compatibility  
| Mention Type Pairs           | \((t_i, t_j), \text{ where } t_i, t_j \in \{ \text{Pronoun, Name, Nominal} \} \)  

The Generative Model

- Given a document $D$,
  - generate a clustering $C$ according to $P(C)$
  - generate $D$ given $C$

$$P(D, C) = P(C) \cdot P(D \mid C)$$

$$= P(C) \cdot P(mp_{12}, mp_{13}, mp_{14}..., \mid C)$$

$$= P(C) \prod_{\text{Pairs}(D)} P(mp_{ij} \mid C_{ij})$$
The Generative Model

- Given a document D,
  - generate a clustering C according to P(C)
  - generate D given C

\[
P(D, C) = P(C) P(D | C)
\]

\[
= P(C) P(mp_{12}, mp_{13}, mp_{14}, ..., | C)
\]

\[
= P(C) \prod_{Pairs(D)} P(mp_{ij} | C_{ij})
\]

\[
= P(C) \prod_{Pairs(D)} P(mp_{ij}^1, mp_{ij}^2, ..., mp_{ij}^7 | C_{ij})
\]

7 feature values
The Generative Model

- Given a document $D$,
  - generate a clustering $C$ according to $P(C)$
  - generate $D$ given $C$

$$P(D, C) = P(C) P(D | C)$$
$$= P(C) P(mp_{12}, mp_{13}, mp_{14} \ldots | C)$$
$$= P(C) \prod_{Pairs(D)} P(mp_{ij} | C_{ij})$$
$$= P(C) \prod_{Pairs(D)} P(mp_{ij}^1, mp_{ij}^2, \ldots, mp_{ij}^7 | C_{ij})$$

Let’s simplify this term
The Generative Model

- Given a document $D$,
  - generate a clustering $C$ according to $P(C)$
  - generate $D$ given $C$

\[
P(D, C) = P(C) P(D | C)
\]

\[
= P(C) P(mp_{12}, mp_{13}, mp_{14}, \ldots | C)
\]

\[
= P(C) \prod_{P\text{airs}(D)} P(mp_{ij} | C_{ij})
\]

\[
= P(C) \prod_{P\text{airs}(D)} P(mp_{ij}^1, mp_{ij}^2, \ldots, mp_{ij}^7 | C_{ij})
\]

Let’s simplify this term

- assume that feature values from different groups are conditionally independent of each other
The Generative Model

- Given a document D,
  - generate a clustering C according to P(C)
  - generate D given C

\[
P(D, C) = P(C) P(D | C) \\
= P(C) P(mp_{12}, mp_{13}, mp_{14} \ldots | C) \\
= P(C) \prod_{\text{Pairs}(D)} P(mp_{ij} | C_{ij}) \\
= P(C) \prod_{\text{Pairs}(D)} P(mp_{ij}^1, mp_{ij}^2, \ldots, mp_{ij}^7 | C_{ij}) \\
= P(C) P(mp_{ij}^1, mp_{ij}^2, mp_{ij}^3 | C_{ij}) \cdot P(mp_{ij}^4, mp_{ij}^5, mp_{ij}^6 | C_{ij}) \cdot P(mp_{ij}^7 | C_{ij})
\]
Model Parameters

\[ P(mp^1, mp^2, mp^3 \mid c) \]
\[ P(mp^4, mp^5, mp^6 \mid c) \]
\[ P(mp^7 \mid c) \]

\( mp^i \) are the feature values
\( c \in \{ \text{Coref, Not Coref} \} \)
Model Parameters

\[ P(mp^1, mp^2, mp^3 \mid c) \]
\[ P(mp^4, mp^5, mp^6 \mid c) \]
\[ P(mp^7 \mid c) \]

\( mp^i \) are the feature values

\( c \in \{ \text{Coref, Not Coref} \} \)
Model Parameters

\[ P(mp^1, mp^2, mp^3 | c) \]

\[ P(mp^4, mp^5, mp^6 | c) \]

\[ P(mp^7 | c) \]

\(mp^i\) are the feature values

\(c \in \{ \text{Coref, Not Coref} \}\)
Model Parameters

\[ P(mp^1, mp^2, mp^3 | c) \]
\[ P(mp^4, mp^5, mp^6 | c) \]
\[ P(mp^7 | c) \]

\( mp^i \) are the feature values
\( c \in \{ \text{Coref, Not Coref} \} \)

**Next step**: use EM to iteratively
- estimate the model parameters
- probabilistically induce a clustering for a document
The Induction Algorithm

- Given a set of unlabeled documents
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to P(C)
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to \( P(C) \)

Initial labelings are presumably noisy
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to $P(C)$
  - estimate the model parameters based on the automatically labeled documents (M-step)
    - maximum likelihood estimation
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to P(C)

- estimate the model parameters based on the automatically labeled documents (M-step)
  - maximum likelihood estimation

- assign a probability to each possible clustering of the mentions for each document (E-step)
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to $P(C)$

- estimate the model parameters based on the automatically labeled documents (M-step)
  - maximum likelihood estimation

- assign a probability to each possible clustering of the mentions for each document (E-step)

3 mentions: 1, 2, 3
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to $P(C)$

- estimate the model parameters based on the automatically labeled documents (M-step)
  - maximum likelihood estimation

- assign a probability to each possible clustering of the mentions for each document (E-step)

3 mentions: 1, 2, 3

+ invalid clusterings
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to P(C)

- estimate the model parameters based on the automatically labeled documents (M-step)
  - maximum likelihood estimation

- assign a probability to each possible clustering of the mentions for each document (E-step)

3 mentions: 1, 2, 3

<table>
<thead>
<tr>
<th>Clustering</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1][2][3]</td>
<td>0.23</td>
</tr>
<tr>
<td>[1][2][3]</td>
<td>0.21</td>
</tr>
<tr>
<td>[1][3][2]</td>
<td>0.11</td>
</tr>
<tr>
<td>[12][3]</td>
<td>0.29</td>
</tr>
<tr>
<td>[1][2][3]</td>
<td>0.05</td>
</tr>
<tr>
<td>+ invalid clusterings</td>
<td></td>
</tr>
</tbody>
</table>

...
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to \( P(C) \)

**Iterate till convergence**
- estimate the model parameters based on the automatically labeled documents \((M\text{-step})\)
  - maximum likelihood estimation
- assign a probability to each possible clustering of the mentions for each document \((E\text{-step})\)

3 mentions: 1, 2, 3

<table>
<thead>
<tr>
<th>Clustering</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>[123]</td>
<td>0.23</td>
</tr>
<tr>
<td>[1][2][3]</td>
<td>0.21</td>
</tr>
<tr>
<td>[13][2]</td>
<td>0.11</td>
</tr>
<tr>
<td>[12][3]</td>
<td>0.29</td>
</tr>
<tr>
<td>[1][23]</td>
<td>0.05</td>
</tr>
<tr>
<td>+ invalid clusterings</td>
<td>...</td>
</tr>
</tbody>
</table>

Iterate till convergence
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to P(C)

**Iterate till convergence**

- estimate the model parameters based on the automatically labeled documents **(M-step)**
  - maximum likelihood estimation
- assign a probability to each possible clustering of the mentions for each document **(E-step)**

3 mentions: 1, 2, 3

<table>
<thead>
<tr>
<th>Clustering</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>[123]</td>
<td>0.23</td>
</tr>
<tr>
<td>[1][2][3]</td>
<td>0.21</td>
</tr>
<tr>
<td>[13][2]</td>
<td>0.11</td>
</tr>
<tr>
<td>[12][3]</td>
<td>0.29</td>
</tr>
<tr>
<td>[1][23]</td>
<td>0.05</td>
</tr>
</tbody>
</table>

+ invalid clusterings...

How to cope with the computational complexity of the E-step?
Approximating the E-step

- Search for the N most probable clusterings only
Approximating the E-step

- Search for the N most probable clusterings only
Approximating the E-step

- Search for the N most probable clusterings only
Approximating the E-step

- Search for the N most probable clusterings only

performs a beam search, expanding the most promising paths
Approximating the E-step

• Search for the N most probable clusterings only
  • using Luo et al.’s (2004) search algorithm

performs a beam search, expanding the most promising paths

scores a path based on pairwise coreference probabilities
The Induction Algorithm

- Given a set of unlabeled documents
  - guess a clustering for each document according to $P(C)$

**Iterate till convergence**

- estimate the model parameters based on the automatically labeled documents (M-step)
  - maximum likelihood estimation
- assign a probability to each possible clustering of the mentions of each document (E-step)
  - use the normalized scores of the 50-best clusterings
Goals

- Design a new model for unsupervised coreference resolution
- Improve Haghighi and Klein’s model with three modifications
Haghighi and Klein’s Model

- Cluster-level model
  - assigns a cluster id to each mention
Haghighi and Klein’s Model

- Cluster-level model
  - assigns a cluster id to each mention

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Haghighi and Klein’s Model

- Cluster-level model
  - assigns a cluster id to each mention

1 Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Haghighi and Klein’s Model

- Cluster-level model
  - assigns a cluster id to each mention

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Haghighi and Klein’s Model

- Cluster-level model
  - assigns a cluster id to each mention

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Haghighi and Klein’s Model

- Cluster-level model
  - assigns a cluster id to each mention

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Haghighi and Klein’s Model

- Cluster-level model
  - assigns a cluster id to each mention
  - ensures transitivity automatically

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. A renowned speech therapist, was summoned to help the King overcome his speech impediment...
Haghighi and Klein’s Generative Story
Haghighi and Klein’s Generative Story

- For each mention encountered in a document,
  - generate a cluster id for the mention (according to some cluster id distribution)
  - generate the head noun of the mention (according to some cluster-specific head distribution)
Haghighi and Klein’s Generative Story

- For each mention encountered in a document,
  - generate a cluster id for the mention (according to some cluster id distribution)
  - generate the head noun of the mention (according to some cluster-specific head distribution)

- **Inference:** Gibbs sampling
Haghighi and Klein’s Generative Story

- For each mention encountered in a document,
  - generate a **cluster id** for the mention (according to some cluster id distribution)
  - generate the **head noun** of the mention (according to some cluster-specific head distribution)

- **Inference:** Gibbs sampling

- **Problem with the model: Too simplistic!**
  - mentions with the same head likely to get the same cluster id
Haghighi and Klein’s Generative Story

- For each mention encountered in a document,
  - generate a cluster id for the mention (according to some cluster id distribution)
  - generate the head noun of the mention (according to some cluster-specific head distribution)

- **Inference:** Gibbs sampling

- **Problem with the model:** Too simplistic!
  - mentions with the same head likely to get the same cluster id
    - two occurrences of “she” will likely be posited as coreferent
    - particularly inappropriate for generating pronouns
Haghighi and Klein’s Generative Story

- For each mention encountered in a document,
  - generate a **cluster id** for the mention (according to some cluster id distribution)
  - generate the **head noun** of the mention (according to some cluster-specific head distribution)

- **Inference**: Gibbs sampling

- **Problem with the model**: Too simplistic!
  - mentions with the same head likely to get the same cluster id

- **Extensions**:
  - use a separate “pronoun head model” to generate pronouns
  - incorporate salience
Improving Haghighi and Klein’s Model

- 3 modifications
  - relaxed head generation
  - agreement constraints
  - pronoun-only salience
Modification 1: Relaxed Head Generation

• Motivation
  • H&K’s model is linguistically impoverished
    • does not exploit useful knowledge: alias, appositive, …
Modification 1: Relaxed Head Generation

- Motivation
  - H&K’s model is linguistically impoverished
    - does not exploit useful knowledge: alias, appositives, …

- Goal
  - simple method for incorporating such knowledge sources
Modification 1: Relaxed Head Generation

- pre-process a document by assigning a “head id” to each mention, such that two mentions have the same head id iff
  - they are the same string
  - or they are aliases
  - or they are in an appositive relation
Modification 1: Relaxed Head Generation

- pre-process a document by assigning a “head id” to each mention, such that two mentions have the same head id iff
  - they are the same string
  - or they are aliases
  - or they are in an appositive relation

| International Business Corporation | 1 |
| IBM                              | 1 |
| Charniak                         | 2 |
| ...                              | ... |
Modification 1: Relaxed Head Generation

- pre-process a document by assigning a “head id” to each mention, such that two mentions have the same head id iff
  - they are the same string
  - or they are aliases
  - or they are in an appositive relation

- instead of generating the head noun, generate the head id

| International Business Corporation | 1 |
| IBM | 1 |
| Charniak | 2 |
| ... | ... |
Modification 1: Relaxed Head Generation

- pre-process a document by assigning a “head id” to each mention, such that two mentions have the same head id iff
  - they are the same string
  - or they are aliases
  - or they are in an appositive relation

- instead of generating the head noun, generate the head id
  - the model views “International Business Corporation” and “IBM” as two mentions having the same head
Modification 1: Relaxed Head Generation

- pre-process a document by assigning a “head id” to each mention, such that two mentions have the same head id iff
  - they are the same string
  - or they are aliases
  - or they are in an appositive relation

- instead of generating the head noun, generate the head id
  - the model views “International Business Corporation” and “IBM” as two mentions having the same head
  - encourages the model to put the two into the same cluster
Modification 2: Agreement Constraints

- Motivation
  - gender and number agreement is implemented as a preference, not as a constraint, in H&K’s model
Modification 2: Agreement Constraints

- Motivation
  - gender and number agreement is implemented as a preference, not as a constraint, in H&K’s model
  - while the model favors the assignment of a pronoun to a gender- and number-compatible cluster
  - it also favors the assignment of a pronoun to a large cluster
Modification 2: Agreement Constraints

- Motivation
  - gender and number agreement is implemented as a preference, not as a constraint, in H&K’s model
  - while the model favors the assignment of a pronoun to a gender- and number-compatible cluster
  - it also favors the assignment of a pronoun to a large cluster
  - if a cluster is large enough, the model may assign the pronoun to the cluster even if the two are not compatible
Modification 2: Agreement Constraints

- **Motivation**
  - gender and number agreement is implemented as a preference, not as a constraint, in H&K’s model
  - while the model favors the assignment of a pronoun to a gender- and number-compatible cluster
  - it also favors the assignment of a pronoun to a large cluster
  - if a cluster is large enough, the model may assign the pronoun to the cluster even if the two are not compatible

- **Goal**
  - implement gender and number agreement as a constraint
Modification 2: Agreement Constraints

- disallow the generation of a mention by any cluster where the two are incompatible in number or gender
Modification 3: Pronoun-Only Salience

- In H&K’s model, salience is applied to all types of mentions (pronouns, names and nominals) during cluster assignment.

- Our hypothesis:
  - since names and nominals are less sensitive to salience, the net benefit of applying salience to names and nominals could be negative as a result of inaccurate modeling of salience.

- We restrict the application of salience to pronouns only.
Improving Haghighi and Klein’s Model

- 3 modifications
  - relaxed head generation
  - agreement constraints
  - pronoun-only salience
Evaluation

- EM-based model

- Haghighi and Klein’s model
  - with and without the 3 modifications
Experimental Setup

- The ACE 2003 coreference corpus
  - 3 data sets (Broadcast News, Newswire, Newspaper)
  - each has a training set and a test set; evaluate on test set only
Experimental Setup

• The ACE 2003 coreference corpus
  • 3 data sets (Broadcast News, Newswire, Newspaper)
  • each has a training set and a test set; evaluate on test set only

• Mentions
  • system mentions (mentions extracted by an NP chunker)
  • perfect mentions (mentions extracted from answer key)
Experimental Setup

- The ACE 2003 coreference corpus
  - 3 data sets (Broadcast News, Newswire, Newspaper)
  - each has a training set and a test set; evaluate on test set only

- Mentions
  - system mentions (mentions extracted by an NP chunker)
  - perfect mentions (mentions extracted from answer key)

- Scoring programs: recall, precision, F-measure
Experimental Setup

- The ACE 2003 coreference corpus
  - 3 data sets (Broadcast News, Newswire, Newspaper)
  - each has a training set and a test set; evaluate on test set only

- Mentions
  - system mentions (mentions extracted by an NP chunker)
  - perfect mentions (mentions extracted from answer key)

- Scoring programs: recall, precision, F-measure
  - MUC scoring program (Vilain et al., 1995)
Experimental Setup

- The ACE 2003 coreference corpus
  - 3 data sets (*Broadcast News, Newswire, Newspaper*)
  - each has a training set and a test set; evaluate on test set only

- Mentions
  - system mentions (mentions extracted by an NP chunker)
  - perfect mentions (mentions extracted from answer key)

- Scoring programs: recall, precision, F-measure
  - MUC scoring program (Vilain et al., 1995)
    - under-penalizes partitions where mentions are over-clustered
    - does not reward successful identification of singleton clusters
Experimental Setup

- The ACE 2003 coreference corpus
  - 3 data sets (Broadcast News, Newswire, Newspaper)
  - each has a training set and a test set; evaluate on test set only

- Mentions
  - system mentions (mentions extracted by an NP chunker)
  - perfect mentions (mentions extracted from answer key)

- Scoring programs: recall, precision, F-measure
  - MUC scoring program (Vilain et al., 1995)
  - CEAF scoring program (Luo, 2005)
    - addresses both weaknesses of the MUC scoring program
Experimental Setup

- The ACE 2003 coreference corpus
  - 3 data sets (Broadcast News, Newswire, Newspaper)
  - each has a training set and a test set; evaluate on test set only

- Mentions
  - system mentions (mentions extracted by an NP chunker)
  - perfect mentions (mentions extracted from answer key)

- Scoring programs: recall, precision, F-measure
  - MUC scoring program (Vilain et al., 1995)
  - CEAF scoring program (Luo, 2005)
  - CEAF variant
    - same as CEAF, but ignores singleton clusters
Heuristic Baseline

- Simple rule-based system

- Posits two mentions as coreferent if and only if they are
  - the same string
  - aliases
  - in an appositive relation
# Heuristic Baseline: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
</tbody>
</table>
Heuristic Baseline: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
</tbody>
</table>
## Heuristic Baseline: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
</tbody>
</table>
Heuristic Baseline: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
</tbody>
</table>
### Experiments on System Mentions

<table>
<thead>
<tr>
<th></th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Heuristic Baseline</strong></td>
<td>R  30.9</td>
<td>P  44.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F  36.4</td>
</tr>
<tr>
<td></td>
<td>R  36.3</td>
<td>P  53.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F  43.2</td>
</tr>
</tbody>
</table>
EM-Based Model

- Initialize the parameters using one (labeled) document
  - rather than using randomly guessed clusterings
## EM-Based Model: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
</tbody>
</table>
EM-Based Model: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
</tbody>
</table>

- gains in both recall and precision
- F-measure increases by 15%
Duplicated Haghighi and Klein’s Model

- The version that incorporates both salience and the separate model for generating pronouns
- Use the same labeled document as in the EM-based model to learn one of the concentration parameters, $\alpha$
### Duplicated H&K’s Model: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>50.8</td>
<td>40.7</td>
</tr>
</tbody>
</table>
## Duplicated H&K’s Model: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>50.8</td>
<td>40.7</td>
</tr>
</tbody>
</table>

- In comparison to EM-based model
  - precision drops substantially
  - F-measure decreases by 6-16%
## Adding 3 Modifications: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>50.8</td>
<td>40.7</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>48.3</td>
<td>45.7</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>50.4</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.2</td>
<td>53.0</td>
</tr>
</tbody>
</table>
# Adding 3 Modifications: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>50.8</td>
<td>40.7</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>48.3</td>
<td>45.7</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>50.4</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.2</td>
<td>53.0</td>
</tr>
</tbody>
</table>

- In comparison to Duplicated Haghighi and Klein
  - F-measure improves after the addition of each modification
## Adding 3 Modifications: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>50.8</td>
<td>40.7</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>48.3</td>
<td>45.7</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>50.4</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.2</td>
<td>53.0</td>
</tr>
</tbody>
</table>

- In comparison to Duplicated Haghighi and Klein
  - F-measure improves after the addition of each modification
  - modest gain in recall and substantial gain in precision when all modifications are applied (7-9% gain in F-measure)
## Supervised Resolver: MUC Results

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>P</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>30.9</td>
<td>44.3</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>42.4</td>
<td>66.0</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>50.8</td>
<td>40.7</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>48.3</td>
<td>45.7</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>50.4</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.2</td>
<td>53.0</td>
</tr>
<tr>
<td>Fully Supervised Model</td>
<td>53.0</td>
<td>70.3</td>
</tr>
</tbody>
</table>

- Trained using C4.5, entire ACE training set, 34 features
- Outperforms the unsupervised models by 3-8%
## MUC, CEAF, CEAF-Variant F-Scores

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUC</td>
<td>CEAF</td>
</tr>
<tr>
<td><strong>Heuristic Baseline</strong></td>
<td>36.4</td>
<td>48.4</td>
</tr>
<tr>
<td><strong>Our EM-based Model</strong></td>
<td>51.6</td>
<td>55.7</td>
</tr>
<tr>
<td><strong>Duplicated Haghighi and Klein</strong></td>
<td>45.2</td>
<td>45.2</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>47.0</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>48.9</td>
<td>51.4</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.6</td>
<td>54.7</td>
</tr>
<tr>
<td><strong>Fully Supervised Model</strong></td>
<td>60.4</td>
<td>61.8</td>
</tr>
</tbody>
</table>
## MUC, CEAF, CEAF-Variant F-Scores

### Experiments on System Mentions

<table>
<thead>
<tr>
<th></th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUC</td>
<td>CEAF</td>
</tr>
<tr>
<td><strong>Heuristic Baseline</strong></td>
<td>36.4</td>
<td>48.4</td>
</tr>
<tr>
<td><strong>Our EM-based Model</strong></td>
<td>51.6</td>
<td>55.7</td>
</tr>
<tr>
<td><strong>Duplicated Haghighi and Klein</strong></td>
<td>45.2</td>
<td>45.2</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>47.0</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>48.9</td>
<td>51.4</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.6</td>
<td>54.7</td>
</tr>
<tr>
<td><strong>Fully Supervised Model</strong></td>
<td>60.4</td>
<td>61.8</td>
</tr>
</tbody>
</table>
## MUC, CEAF, CEAF-Variant F-Scores

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUC</td>
<td>CEAF</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>36.4</td>
<td>48.4</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>51.6</td>
<td>55.7</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>45.2</td>
<td>45.2</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>47.0</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>48.9</td>
<td>51.4</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.6</td>
<td>54.7</td>
</tr>
<tr>
<td>Fully Supervised Model</td>
<td>60.4</td>
<td>61.8</td>
</tr>
</tbody>
</table>
### MUC, CEAF, CEAF-Variant F-Scores

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUC</td>
<td>CEAF</td>
</tr>
<tr>
<td><strong>Heuristic Baseline</strong></td>
<td>36.4</td>
<td>48.4</td>
</tr>
<tr>
<td><strong>Our EM-based Model</strong></td>
<td>51.6</td>
<td>55.7</td>
</tr>
<tr>
<td><strong>Duplicated Haghighi and Klein</strong></td>
<td>45.2</td>
<td>45.2</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>47.0</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>48.9</td>
<td>51.4</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.6</td>
<td>54.7</td>
</tr>
<tr>
<td><strong>Fully Supervised Model</strong></td>
<td>60.4</td>
<td>61.8</td>
</tr>
</tbody>
</table>
MUC, CEAF, CEAF-Variant F-Scores

<table>
<thead>
<tr>
<th>Experiments on System Mentions</th>
<th>Broadcast News</th>
<th>Newswire</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MUC</td>
<td>CEAF</td>
</tr>
<tr>
<td>Heuristic Baseline</td>
<td>36.4</td>
<td>48.4</td>
</tr>
<tr>
<td>Our EM-based Model</td>
<td>51.6</td>
<td>55.7</td>
</tr>
<tr>
<td>Duplicated Haghighi and Klein</td>
<td>45.2</td>
<td>45.2</td>
</tr>
<tr>
<td>+ Relaxed Head Generation</td>
<td>47.0</td>
<td>47.5</td>
</tr>
<tr>
<td>+ Agreement Constraints</td>
<td>48.9</td>
<td>51.4</td>
</tr>
<tr>
<td>+ Pronoun-only Salience</td>
<td>52.6</td>
<td>54.7</td>
</tr>
<tr>
<td>Fully Supervised Model</td>
<td>60.4</td>
<td>61.8</td>
</tr>
</tbody>
</table>

- Similar performance trends across the 3 scoring programs
Experiments using Perfect Mentions

- Similar performance trends observed
  - except that the unsupervised models perform comparably to the fully-supervised resolver

- Conclusions drawn from system mentions are not always generalizable to perfect mentions and vice versa
Summary

- Presented an EM-based model for unsupervised coreference resolution that
  - outperforms Haghighi and Klein’s coreference model
  - compares favorably to a modified version of their model