A Mention-Synchronous Coreference Resolution Algorithm Based on the Bell Tree

Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL2004)

Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, A Kambhatla, & Salim Roukos.

IBM Research| Watson Research Center

Reporter: Yi-Ting Huang
Date: 2010/10/6
Overview

M={ “Mr. Clinton_1”, “Clinton_2”, “she_3”}
Outline

1. Introduction
2. Bell Tree: From Mention to Entity
3. Coreference Model
4. Search Issues
5. Experiments
6. Related Work
7. Conclusion
Coreference resolution in this context is defined as partitioning mentions into entities.

A mention is an instance of reference to an object, and the collection of mentions referring to the same object in a document form an entity.

“The American Medical Association voted yesterday to install the heir apparent as its president-elect, rejecting a strong, upstart challenge by a District doctor who argued that the nation’s largest physicians’ group needs stronger ethics and new leadership.”
1. Introduction (2/4)

- Early work of anaphora resolution focuses on finding antecedents of pronouns (Hobbs, 1976; Ge et al., 1998; Mitkov, 1998),

- while recent advances (Soon et al., 2001; Yang et al., 2003; Ng and Cardie, 2002; Ittycheriah et al., 2003) employ statistical machine learning methods and try to resolve reference among all kinds of noun phrases (NP), be it a **name**, **nominal**, or **pronominal phrase** – which is the scope of this paper as well.

1. Introduction (3/4)

One common strategy shared by (Soon et al., 2001; Ng and Cardie, 2002; Ittycheriah et al., 2003) is that a statistical model is trained to measure how likely a pair of mentions coreference; then a greedy procedure is followed to group mentions into entities.

\[ M = \{ \text{“Mr. Clinton}_1$$, \text{“Clinton}_2$$, \text{“she}_3$$ \} \]

Entity-mention model:
\[ E = \{ \{\text{“Mr. Clinton}_1$$, \text{“Clinton}_2$$\}, \text{“she}_3$$ \} \]

- (she, [Mr. Clinton, Clinton])
- (she, Mr. Clinton)
- (she, Clinton)
1. Introduction (4/4)

- In this paper, we propose to use the **Bell tree** to represent the process of forming entities from mentions.
  - The Bell tree represents the search space of the coreference resolution problem – each leaf node corresponds to a possible coreference outcome.

- We choose to **Maximum Entropy model** the process from mentions to entities represented in the Bell tree, and the problem of coreference resolution is cast as finding the “best” path from the root node to leaves.
Let us consider traversing mentions in a document from beginning (left) to end (right). The process of forming entities from mentions can be represented by a tree structure.
2. Bell Tree: From Mention to Entity (2/8)

- A figure illustrates how the Bell tree is created for a document with three mentions.
- The root node is the initial state of the process, which consists of a partial entity containing the first mention of a document.
The second mention is added in the next step by either linking to the existing partial entity (b1), or starting a new entity (b2). A second layer of nodes are created to represent the two possible outcomes.

* Means the mention becomes active.
2 in (a) is active mention
[1 2] in (b1) is in-focus entity
Subsequent mentions are added to the tree in the same manner. The process is *mention-synchronous* in that each layer of tree nodes are created by adding one mention at a time.
2. Bell Tree: From Mention to Entity (5/8) - Bell Number

- Since the number of tree leaves is the number of possible coreference outcomes and it equals the Bell Number (Bell, 1934), the tree is called the *Bell tree*.

  - Bell tree is the number of partitions of a set with n members.
  - The first few Bell numbers are: 1, 1, 2, 5, 15, 52, 203, 877, 4140, ...
  - For example, B3 = 5 because the 3-element set \{a, b, c\} can be partitioned in 5 distinct ways:

    - \{\{a\}, \{b\}, \{c\}\}
    - \{\{a\}, \{b, c\}\}
    - \{\{b\}, \{a, c\}\}
    - \{\{c\}, \{a, b\}\}
    - \{\{a, b, c\}\}
2. Bell Tree: From Mention to Entity (6/8) - Bell Number

- The Bell Number $B(n)$ is the number of ways of partitioning $n$ distinguishable objects (i.e., mentions) into non-empty disjoint subsets (i.e., entities).

- The Bell Number has a “closed” formula

$$B(n) = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!}$$

and it increases rapidly as $n$ increases:

$$B(20) \approx 5.2 \times 10^{13}!$$
2. Bell Tree: From Mention to Entity (7/8)

- Under the derivation illustrated in the Figure, each leaf node in the Bell tree corresponds to a possible coreference outcome.
- The Bell tree clearly represents the search space of the coreference resolution problem.
- The coreference resolution can therefore be cast equivalently as finding the “best” leaf node.
Since the search space is large (even for a document with a moderate number of mentions), it is difficult to estimate a distribution over leaves directly.

Instead, we choose to model the process from mentions to entities, or in other words, score paths from the root to leaves in the Bell tree.
3. Coreference Model (1/6)

Formally, let \( \{m_i : 1 < i < n\} \) be \( n \) mentions in a document. Mention index \( i \) represents the order it appears in the document. Let \( e_j \) be an entity, and \( g : i \mapsto j \) be the (many-to-one) map from mention index \( i \) to entity index \( j \). For an active mention index \( k (1 \leq k \leq n) \), define

\[
I_k = \{ t : t = g(i), \text{ for some } 1 \leq i \leq k - 1 \},
\]

the set of indices of the partially-established entities to the left of \( m_k \) (note that \( I_1 = \emptyset \)), and

\[
E_k = \{ e_t : t \in I_k \},
\]

the set of the partially-established entities. The link model is then

\[
P(L|M_k, m_k, A_k = t),
\]

the probability linking the active mention \( m_k \) with the in-focus entity \( e_t \). The random variable \( A_k \) takes value from the set \( I_k \) and signifies which entity is in focus; \( L \) takes binary value and is 1 if \( m_k \) links with \( e_t \).

- **M**: mention的集合
- **E**: entity的集合
- **I_k**: mention_k之前所出現的entity集合中的值

**Example:**
- Active mention: 3
- \( E_3 = \{ [1], [2] \} \)
- \( P(L=1|E_3, '3', A_3=2) \)

代表有多大的機會mention3和entity2有連結(coreference)的關係
3. Coreference Model (2/6)

Since starting a new entity means that \( m_k \) does not link with any entities in \( E_k \), the probability of starting a new entity, \( P(L = 0|E_k, m_k) \), can be computed as

\[
P(L = 0|E_k, m_k) = \sum_{t \in I_k} P(L = 0, A_k = t|E_k, m_k) = 1 - \sum_{t \in I_k} P(A_k = t|E_k, m_k).
\]

(3) indicates that the probability of starting an entity can be computed using the linking probabilities \( P(L = 1|E_k, m_k, A_k = t) \), provided that the marginal \( P(A_k = t|E_k, m_k) \) is known. In this paper, \( P(A_k = t|E_k, m_k) \) is approximated as:

\[
P(A_k = t|E_k, m_k) = \begin{cases} 
1 & \text{if } t = \arg \max_{i \in I_k} P(L = 1|E_k, m_k, A_k = i) \\
0 & \text{otherwise}
\end{cases}
\]

With the approximation (4), the starting probability (3) is

\[
P(L = 1|E_k, m_k, A_k = t) = 1 - \max_{t \in I_k} P(L = 1|E_k, m_k, A_k = t).
\]

Since (5) is an approximation, not true probability, a constant \( \alpha \) is introduced to balance the linking probability and starting probability and the starting probability becomes:

\[
P_\alpha(L = 0|E_k, m_k) = \alpha P(L = 0|E_k, m_k).
\]

(6)

If \( \alpha < 1 \), it penalizes creating new entities; Therefore, \( \alpha \) is called start penalty. The start penalty \( \alpha \) can be used to balance entity miss and false alarm.
The score for the path (a)-(b2)-(c4) in the Figure is the product of the start probability from (a) to (b2) and the linking probability from (b2) to (c4).
The model $P(L|E_k, m_k, A_k = t)$ depends on all partial entities $E_k$, which can be very expensive. After making some modeling assumptions, we can approximate it as:

$$
P(L = 1|E_k, m_k, A_k = t) \approx P(L = 1|e_t, m_k) \approx \max_{m \in e_t} P(L = 1|m, m_k).$$

From (7) to (8), entities other than the one in focus, $e_t$, are assumed to have no influence on the decision of linking $m_k$ with $e_t$. → entity-mention model.

(9) further assumes that the entity-mention score can be obtained by the maximum mention pair score. → the mention-pair model.
3. Coreference Model (5/6)

\begin{align*}
P(L = 1|E_k, m_k, A_k = t) & \quad (7) \\
\approx P(L = 1|e_t, m_k) & \quad (8) \\
\approx \max_{m \in e_t} P(L = 1|m, m_k). & \quad (9)
\end{align*}

\begin{align*}
P(L|e_t, m_k) &= \frac{e^{\left(\sum_k \lambda_k g_k(e_t, m_k, L)\right)}}{Z(e_t, m_k)}, & \quad (11) \\
P(L|m_i, m_k) &= \frac{e^{\left(\sum_k \lambda_k g_k(m_i, m_k, L)\right)}}{Z(m_i, m_k)}, & \quad (10)
\end{align*}

where \(g_k(\cdot, \cdot, L)\) is a feature and \(\lambda_k\) is its weight; \(Z(\cdot, \cdot)\) is a normalizing factor to ensure that (10) or (11) is a probability. Effective training algorithm exists (Berger et al., 1996) once the set of features \(\{g_k(\cdot, \cdot, L)\}\) is selected.

3.2 Features (mention-pair model)

- Features in the lexical category are applicable to non-pronominal mentions only.
- Composite features can be generated by taking conjunction of basic features.
  - For example, a distance feature together with reflexiveness of a pronoun mention can help to capture that the antecedent of a reflexive pronoun is often closer than that of a non-reflexive pronoun.
3.2 Features (entity-mention model)

- The apposition feature are computed by testing any mention in the entity \( et \) against the active mention \( m_k \).
- Distance features are computed by taking minimum between mentions in the entity and the active mention.
- In the entity-mention model, “ncd”, “spell” and “count” features are computed over the canonical mention of the in-focus entity and the active mention.
- Conjunction features are used in the entity-mention model too.
3.2 Features

- The mention-pair model is appealing for its simplicity: features are easy to compute over a pair of mentions; its drawback is that information outside the mention pair is ignored.

- \( E = \{ \text{“Mr. Clinton”}, \text{“Clinton”}, \text{“she”} \} \)
  - \( E = \{ \text{“Mr. Clinton”}, \text{[“Clinton”, “she”]} \} \)
  - \( E = \{ \text{[“Mr. Clinton”, [“Clinton”], “she”]} \} \)
  - \( E = \{ \text{[“Mr. Clinton”, “Clinton”, “she”]} \} \)

- Since gender and number information is propagated at the entity level, the entity-mention model is able to check the gender consistency when considering the active mention “she”.
3. Coreference Model (6/6)

Since starting a new entity means that \( m_k \) does not link with any entities in \( E_k \), the probability of starting a new entity, \( P(L = 0|E_k, m_k) \), can be computed as

\[
P(L = 0|E_k, m_k) = \sum_{t \in I_k} P(L = 0, A_k = t|E_k, m_k)
= 1 - \sum_{t \in I_k} P(A_k = t|E_k, m_k) \cdot P(L = 1|E_k, m_k, A_k = t).
\]  

(3) indicates that the probability of starting an entity can be computed using the linking probabilities \( P(L = 1|E_k, m_k, A_k = t) \), provided that the marginal \( P(A_t = t|E_k, m_k) \) is known. In this paper, \( P(A_k = t|E_k, m_k) \) is approximated as:

\[
P(A_k = t|E_k, m_k) = \begin{cases} 
1 & \text{if } t = \arg \max_{i \in I_k} P(L = 1|E_k, m_k, A_k = i) \\
0 & \text{otherwise}
\end{cases}
\]  

(4)

(4) is not the only way \( P(A_k = t|E_k, m_k) \) can be approximated. For example, one could use a uniform distribution over \( I_k \). We experimented several schemes of approximation, including a uniform distribution, and (4) worked the best and is adopted here. One may consider training \( P(A_k = t|E_k, m_k) \) directly and use it to score paths in the Bell tree. The problem is that 1) the size of \( I_k \) from which \( A_k \) takes value is variable; 2) the start action depends on all entities in \( E_k \), which makes it difficult to train \( P(A_k = t|E_k, m_k) \) directly.
4. Search Issue (1/4)

- The search space of the coreference problem can be represented by the Bell tree.
- Since the Bell number increases rapidly as a function of the number of mentions, pruning is necessary.
- Thus, the search problem reduces to creating the Bell tree while keeping track of path scores and picking the top-N best paths.
4. Search Issue (2/4)

- We prune the search space in the following places:
  - Local pruning:
    if any children with a score below a fixed factor $\delta$ are pruned, otherwise expended:

\[
\begin{align*}
  P(L = 1|E, m_k, A = i) & > \delta p_m \\
  P_x(L = 0|E, m_k) & > \delta p_m.
\end{align*}
\]

- Global pruning: similar to local pruning except that this is done using the cumulative score $S(E)$. 
Algorithm 1 Search Algorithm

Input: mentions $M = \{m_i : 1, \ldots, n\}$; $N$
Output: top $N$ entity results

1: Initialize: $\mathcal{H} := \{E_1 := \{[m_1]\}\}$; $S(E_1) = 1$
2: for $k = 2$ to $n$
3:     foreach node $E \in \mathcal{H}$
4:         compute $p_m$.
5:     foreach $i \in I_k$
6:         if ($P(L = 1|E, m_k, A = i) > \delta p_m$) {
7:             Extend $E$ to $E_i'$ by linking $m_k$ with $e_i$
8:             $S(E_i') := S(E) \cdot P(L = 1|E, m_k, A = i)$
9:         }
10:    if ($P_\alpha (L = 0|E, m_k) > \delta p_m$) {
11:        Extend $E$ to $E'$ by starting $[m_k]$
12:        $S(E') := S(E) \cdot P_\alpha (L = 0|E, m_k)$
13:    }
14: $\mathcal{H} := \{E'\} \cup \{E_i' : i \in I_k\}$.
15: return $\{E_{(1)}, E_{(2)}, \ldots, E_{(N)}\}$

$S(E_{(1)}) \geq S(E_{(2)}) \geq \cdots \geq S(E_{(N)})$. 
Algorithm 1 Search Algorithm

Input: mentions \( M = \{m_i : 1, \ldots, n\}; \ N \)

Output: top \( N \) entity results

1: Initialize: \( \mathcal{H} := \{E_1 := \{[m_1]\}\} ; S(E_1) = 1 \)

2: for \( k = 2 \) to \( n \)
3: \hspace{1em} foreach node \( E \in \mathcal{H} \)
4: \hspace{2em} compute \( p_m \).
5: \hspace{2em} foreach \( i \in I_k \)
6: \hspace{3em} if \( P(L = 1|E, m_k, A = i) > \delta p_m \) \{
7: \hspace{4em} Extend \( E \) to \( E'_i \) by linking \( m_k \) with \( e_i \)
8: \hspace{4em} \hspace{1em} \( S(E'_i) := S(E) \cdot P(L = 1|E, m_k, A = i) \)
9: \hspace{3em} \}
10: \hspace{2em} \}
11: \hspace{1em} if(\( P_\alpha(L = 0|E, m_k) > \delta p_m \) \{
12: \hspace{2em} Extend \( E \) to \( E' \) by starting \([m_k]\) \).
13: \hspace{2em} \hspace{1em} \( S(E') := S(E) \cdot P_\alpha(L = 0|E, m_k) \)
14: \hspace{2em} \}
15: \hspace{1em} \}
16: \hspace{1em} \mathcal{H} := \{E'\} \cup \{E'_i : i \in I_k\} \).

16: return \( \{E(1), E(2), \ldots, E(N)\} \)

\[ S(E(1)) \geq S(E(2)) \geq \cdots \geq S(E(N)). \]
5. Experiment 1

- **Metrics**: ACE-value
- the ACE-value can be interpreted as percentage of value a system has, relative to the perfect system.
- A perfect coreference system will get a 100% ACE-value while a system outputs no entities will get a 0 ACE-value.
- ACE-value is computed by first calculating the weighted cost of entity insertions, deletions and substitutions.
- Weights are designed to emphasize NAME entities, while PRONOUN entities (i.e., an entity consisting of only pronominal mentions) carry very low weights.
5. Experiment 1 (1/11)

- **Metrics:** an entity-constrained mention F-measure (henceforth “ECM-F”) because the ACE-value is an entity-level metric and is weighted heavily toward NAME entities.

- The ECM-F measures the percentage of mentions that are in the “right” entities.

- The metric first aligns the system entities with the reference entities so that the number of common mentions is maximized. Each system entity is constrained to align with at most one reference entity, and vice versa.

- **Example:**
  
  System: \{[m_1, m_2], [m_3], [m_5], [m_6]\}

  True: \{[m_1], [m_2, m_3], [m_4]\}

  \[m_1] \leftrightarrow [m_1, m_2], [m_2, m_3] \leftrightarrow [m_3]

  P:2/5

  R:2/4
5. Experiment 1 (2/11)

- The ACE coreference system is trained with 416 documents (about 190K words) of ACE2002 training dataset.

- A separate 90 documents (50K words) is used as the development-test (Devtest) set. In 2002, NIST released two test sets in February (Feb02) and September (Sep02), respectively.

<table>
<thead>
<tr>
<th>TestSet</th>
<th>#-docs</th>
<th>#-words</th>
<th>#-mentions</th>
<th>#-entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Devtest</td>
<td>90</td>
<td>50426</td>
<td>7470</td>
<td>2891</td>
</tr>
<tr>
<td>Feb02</td>
<td>97</td>
<td>52677</td>
<td>7665</td>
<td>3104</td>
</tr>
<tr>
<td>Sep02</td>
<td>186</td>
<td>69649</td>
<td>10577</td>
<td>4355</td>
</tr>
</tbody>
</table>

Table 2: Statistics of three test sets.
5. Experiment 1 (3/11)
Mention-pair model

- For the mention-pair model, training events are generated for all compatible mention-pairs, which results in about 990K events, about 150K of which are positive examples.
- The full mention-pair model uses about 171K features.
For the entity-mention model, events are generated by walking through the Bell tree. Only events on the true path (i.e., positive examples) and branches emitting from a node on the true path to a node not on the true path (i.e., negative examples) are generated.

This scheme generates about 322K events, out of which about 18K are positive training examples.

The full entity-mention model has about 8.4K features, due to less number of conjunction features and training examples.
5. Experiment 1 (5/11)

- We will report coreference results on the **true mentions** of the three test sets, and pruning threshold $\delta = 0.001$.

<table>
<thead>
<tr>
<th></th>
<th>Devtest</th>
<th></th>
<th>Feb02</th>
<th></th>
<th>Sep02</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>ACE-val(%)</td>
<td>ECM-F(%)</td>
<td>ACE-val(%)</td>
<td>ECM-F(%)</td>
<td>ACE-val(%)</td>
<td>ECM-F(%)</td>
</tr>
<tr>
<td>MP</td>
<td>89.8</td>
<td>73.2 (±2.9)</td>
<td>90.0</td>
<td>73.1 (±4.0)</td>
<td>88.0</td>
<td>73.1 (±6.8)</td>
</tr>
<tr>
<td>EM</td>
<td>89.9</td>
<td>71.7 (±2.4)</td>
<td>88.2</td>
<td>70.8 (±3.9)</td>
<td>87.6</td>
<td>72.4 (±6.2)</td>
</tr>
</tbody>
</table>

- The mention-pair model in most cases performs better than the mention-entity model by both ACE-value and ECM-F measure although none of the differences is statistically significant (pair-wise t-test) at p-value 0.05.
The mention-pair model uses 20 times more features than the entity-pair model.

The mention-pair sometimes mistakenly places a male pronoun and female pronoun into the same entity, while the same mistake is avoided in the entity-mention model.
5. Experiment 1 (7/11)

Feature Impact

- To see how each category of features affects the performance, we start with the aforementioned mention-pair model, incrementally remove each feature category, retrain the system and test it on the Devtest set.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACE-val(%)</th>
<th>ECM-F(%)</th>
<th>#-features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>89.8</td>
<td>73.20 (±2.9)</td>
<td>171K</td>
</tr>
<tr>
<td>-syntax</td>
<td>89.0</td>
<td>72.6 (±2.5)</td>
<td>71K</td>
</tr>
<tr>
<td>-count</td>
<td>89.4</td>
<td>72.0 (±3.3)</td>
<td>70K</td>
</tr>
<tr>
<td>-dist</td>
<td>86.7</td>
<td>*66.2 (±3.9)</td>
<td>24K</td>
</tr>
<tr>
<td>-type/level</td>
<td>86.8</td>
<td>65.7 (±2.2)</td>
<td>5.4K</td>
</tr>
<tr>
<td>-spell</td>
<td>86.0</td>
<td>64.4 (±1.9)</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 4: Impact of feature categories. Numbers after ± are the standard deviations. * indicates that the result is significantly (pair-wise t-test) different from the line above at $p = 0.05$. 
5. Experiment 1 (8/11) Feature Impact

- It is striking that the smallest system consisting of only 39 features (string and substring match, acronym, edit distance and number of different capitalized words) can get as much as 86% ACE-value.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACE-val(%)</th>
<th>ECM-F(%)</th>
<th>#-features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>89.8</td>
<td>73.20 (±2.9)</td>
<td>171K</td>
</tr>
<tr>
<td>-syntax</td>
<td>89.0</td>
<td>72.6 (±2.5)</td>
<td>71K</td>
</tr>
<tr>
<td>-count</td>
<td>89.4</td>
<td>72.0 (±3.3)</td>
<td>70K</td>
</tr>
<tr>
<td>-dist</td>
<td>86.7</td>
<td>*66.2 (±3.9)</td>
<td>24K</td>
</tr>
<tr>
<td>-type/level</td>
<td>86.8</td>
<td>65.7 (±2.2)</td>
<td>5.4K</td>
</tr>
<tr>
<td>-spell</td>
<td>86.0</td>
<td>64.4 (±1.9)</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 4: Impact of feature categories. Numbers after ± are the standard deviations. * indicates that the result is significantly (pair-wise t-test) different from the line above at $p = 0.05$. 

[37]
5. Experiment 1 (9/11)  
Feature Impact

- The lexical features and the distance features are the most important.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACE-val(%)</th>
<th>ECM-F(%)</th>
<th>#-features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>89.8</td>
<td>73.20 (±2.9)</td>
<td>171K</td>
</tr>
<tr>
<td>-syntax</td>
<td>89.0</td>
<td>72.6 (±2.5)</td>
<td>71K</td>
</tr>
<tr>
<td>-count</td>
<td>89.4</td>
<td>72.0 (±3.3)</td>
<td>70K</td>
</tr>
<tr>
<td>-dist</td>
<td>86.7</td>
<td>*66.2 (±3.9)</td>
<td>24K</td>
</tr>
<tr>
<td>-type/level</td>
<td>86.8</td>
<td>65.7 (±2.2)</td>
<td>5.4K</td>
</tr>
<tr>
<td>-spell</td>
<td>86.0</td>
<td>64.4 (±1.9)</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 4: Impact of feature categories. Numbers after ± are the standard deviations. * indicates that the result is significantly (pair-wise t-test) different from the line above at \( p = 0.05 \).
5. Experiment 1 (10/11)

- The start penalty $\alpha$ can be used to balance the entity miss and false alarm.
- The ACE-value and ECM-F track each other fairly well.
- Both achieve the optimal when $\log \alpha \approx -0.8$
5. Experiment 2 (1/2)

- **Metrics:** MUC scorer
- **Data Set:** MUC6 data.
- To minimize the change to the coreference system, we first map the MUC data into the ACE style.
  - The original MUC coreference data does not have entity types (i.e., “ORGANIZATION”, “LOCATION” etc), required in the ACE style.
  - Part of entity types can be recovered from the corresponding named-entity annotations.
  - There are **504** out of 2072 mentions of the MUC6 formal test set and **695** out of 2141 mentions of the MUC6 dry-run test set that cannot be assigned labels by this procedure. → “UNKNOWN”
5. Experiment 2 (2/2)

- Two coreference systems are trained on the MUC6 data:
  - one trained with 30 dry-run test documents (henceforth “MUC6-small”);
  - the other trained with 191 “dry-run train” documents.

<table>
<thead>
<tr>
<th>System</th>
<th>MUC F-measure</th>
<th>ECM-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC6-small</td>
<td>83.9%</td>
<td>72.1%</td>
</tr>
<tr>
<td>MUC6-big</td>
<td>85.7%</td>
<td>76.8%</td>
</tr>
</tbody>
</table>
6. Related Work (1/4)

- Soon et al. (2001) uses a decision tree model for coreference resolution on the MUC6 and MUC7 data.
  - Ng and Cardie (2002) expands the feature sets of (Soon et al., 2001).

- Neither (Soon et al., 2001) nor (Ng and Cardie, 2002) searches for the global optimal entity in that they make locally independent decisions during search. In contrast, our decoder always searches for the best result ranked by the cumulative score (subject to pruning), and subsequent decisions depend on earlier ones.

6. Related Work (2/4)

- McCallum and Wellner (2003) proposed to use graphical models for computing probabilities of entities.
- The model is appealing in that it can potentially overcome the limitation of mention-pair model in which dependency among mentions other than the two in question is ignored.
- However, models in (McCallum and Wellner, 2003) compute directly the probability of an entity configuration conditioned on mentions, and it is not clear how the models can be factored to do the incremental search. The Bell tree representation proposed in this paper, however, provides us with a naturally incremental framework for coreference resolution.

6. Related Work (3/4)

- Kehler (1997) uses a mention-pair maximum entropy model, and two methods are proposed to compute entity scores based on the mention-pair model:
  1) a distribution over entity space is deduced;
  2) the most recent mention of an entity, together with the candidate mention, is used to compute the entity-mention score.

- In contrast, in our mention pair model, an entity-mention pair is scored by taking the maximum score among possible mention pairs.

6. Related Work (4/4)

- Morton (2000) also uses a maximum entropy mention-pair model, and a special “dummy” mention is used to model the event of starting a new entity. Features involving the dummy mention are essentially computed with the single (normal) mention, and therefore the starting model is weak.

- In our model, the starting model is obtained by “complementing” the linking scores. The advantage is that we do not need to train a starting model. To compensate the model inaccuracy, we introduce a “starting penalty” to balance the linking and starting scores.

Conclusion

- We propose to use the Bell tree to represent the process of forming entities from mentions.
- We studied two maximum entropy models, namely the mention-pair model and the entity-mention model, both of which can be used to score entity hypotheses.
- A search algorithm is used to search the best entity result. State-of-the-art performance has been achieved on the ACE coreference data.
5. Experiment 1 (10/11)
Feature Impact

- Sometimes the ACE-value increases after removing a set of features, but the ECM-F measure tracks nicely the trend that the more features there are, the better the performance is.

<table>
<thead>
<tr>
<th>Model</th>
<th>ACE-val(%)</th>
<th>ECM-F(%)</th>
<th>#-features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full</td>
<td>89.8</td>
<td>73.20 (±2.9)</td>
<td>171K</td>
</tr>
<tr>
<td>-syntax</td>
<td>89.0</td>
<td>72.6 (±2.5)</td>
<td>71K</td>
</tr>
<tr>
<td>-count</td>
<td>89.4</td>
<td>72.0 (±3.3)</td>
<td>70K</td>
</tr>
<tr>
<td>-dist</td>
<td>86.7</td>
<td>*66.2 (±3.9)</td>
<td>24K</td>
</tr>
<tr>
<td>-type/level</td>
<td>86.8</td>
<td>65.7 (±2.2)</td>
<td>5.4K</td>
</tr>
<tr>
<td>-spell</td>
<td>86.0</td>
<td>64.4 (±1.9)</td>
<td>39</td>
</tr>
</tbody>
</table>

Table 4: Impact of feature categories. Numbers after ± are the standard deviations. * indicates that the result is significantly (pair-wise t-test) different from the line above at $p = 0.05$. 
5. Experiment 2 (3/5)

- The MUC scorer cannot be used since it inherently favors systems that output fewer number of entities (e.g., putting all mentions of the MUC6 formal test set into one entity will yield a 100% recall and 78.9% precision of links, which gives an 88.2% F-measure).

<table>
<thead>
<tr>
<th>System</th>
<th>MUC F-measure</th>
<th>ECM-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC6-small</td>
<td>83.9%</td>
<td>72.1%</td>
</tr>
<tr>
<td>MUC6-big</td>
<td>85.7%</td>
<td>76.8%</td>
</tr>
</tbody>
</table>
5. Experiment 2 (4/5)

- When measured by the ECM-F measure, the MUC6-small system has the same level of performance as the ACE system, while the MUC6-big system performs better than the ACE system.

<table>
<thead>
<tr>
<th>System</th>
<th>MUC F-measure</th>
<th>ECM-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC6-small</td>
<td>83.9%</td>
<td>72.1%</td>
</tr>
<tr>
<td>MUC6-big</td>
<td>85.7%</td>
<td>76.8%</td>
</tr>
</tbody>
</table>
5. Experiment 2 (5/5)

- The MUC6-small system compares favorably with the similar experiment in Harabagiu et al. (2001) in which an 81.9% F-measure is reported.
- The results show that the algorithm works well on the MUC6 data despite some information is lost in the conversion from the MUC format to the ACE format.

<table>
<thead>
<tr>
<th>System</th>
<th>MUC F-measure</th>
<th>ECM-F</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUC6-small</td>
<td>83.9%</td>
<td>72.1%</td>
</tr>
<tr>
<td>MUC6-big</td>
<td>85.7%</td>
<td>76.8%</td>
</tr>
</tbody>
</table>