Pronouns

- Reference to an entity already introduced called anaphora.
- Pronoun is licensed by previous mention of an antecedent.
- Pronoun resolution subset of general reference resolution.

Antecedent Game

Constraints on antecedents:
- Number agreement.
  - John his a ball. He threw *them* far.
- Person agreement
  - 1st, 2nd, 3rd person match
- Gender agreement
  - he/she/it

Binding theory constraints:
- John bought himself an ice cream.
- John bought him an ice cream
- John said that Bill bought him an ice cream
- John said that Bill bought himself an ice cream
- He said that he bought Bill an ice cream
- Constraints on meaning of him, himself, be.

Selectional restrictions:
- John ate his sandwich in his office.
  - It was made with roast beef.
  - It was quieter than eating in the snack bar.
- Recency:
  - Lee met Mary for lunch. They saw Sue at the restaurant. She gave Lee a hug.
- Grammatical role: Subject > object
  - Jane saw Sally at the market. She went over to say hello.

Repeated mention:
- John had a long day. He had not gotten much sleep the night before. He and Fred went to the movies that night. He had a hard time staying awake.
- Parallelism
  - Jane helped Mary with her Physics homework. Ellen helped her with her English.
- Verb Semantics:
  - Jane gave Mary the letter.
    - She was excited to receive it.
    - She had received it yesterday.
Algorithms for Pronominal Anaphora Resolution

Hobbs 1978

- Works on parse trees of sentence containing pronoun and of all previous sentences.
- Approximates binding theory, recency, and grammatical role preferences.
- Uses info on gender, person, and number constraints as a final check.

Hobbs

1. Begin at NP immediately dominating the pronoun
2. Go up tree to first NP or S node encountered. Call it X and path to it p.
3. Traverse all branches below X to left of path p in a left-to-right, breadth-first fashion. Propose as antecedent any NP node encountered which has an NP or S node between it and X.
4. If X is highest S node in sentence, traverse parse trees of previous sentences in order of recency, each in a left-to-right, breadth-first manner, and when an NP is encountered, propose as antecedent. If X not highest, go to 5.
5. From X go up to first NP or S. Call new node X and path to it p.
6. If X is NP and p did not pass through Nominal that X immediately dominates, propose X as antecedent.
7. Traverse all branches below X to left of p in left-to-right, breadth-first manner, but do not go below any NP or S node encountered.
8. If X is S node, traverse all branches of X to right of p in left-to-right, breadth-first manner, but do not go below any NP or S node encountered. Propose any NP encountered as antecedent.

Examples

- John saw a beautiful MGB at the dealership.
- He showed it to Bob.
- He bought it.

Final Check

- Parsers generally return number and person info, but usually not gender.
- Check hyponyms in WordNet of head noun.
- Person, living thing indicate animate noun
- female indicates female gender, ...
- Cues in titles: Mr., Ms.
**Centering Algorithm**

- Claim: There is single entity being "centered" on at any point in the discourse.
- Let $U_n$, $U_{n+1}$ be 2 consecutive utterances.
- Backward looking center of $U_n$, written $C_b(U_n)$, represents focus after $U_n$ interpreted.
- Forward looking centers of $U_n$, written $C_f(U_n)$, forms ordered list of entities in $U_n$ that can serve as $C_b(U_{n+1})$.
- $C_b(U_{n+1})$ is highest ranking elt of $C_f(U_n)$ mentioned in $U_{n+1}$.

**Centers**

- Order of entities in $C_f(U_n)$:
  - subject > existential predicate nominal > object > indirect object > demarcated adverbial PP
- Let $C_p(U_{n+1})$ be highest ranked forward looking center

**State-Based Transitions**

<table>
<thead>
<tr>
<th>$C_f(U_{n+1}) = C_f(U_n)$</th>
<th>$C_b(U_{n+1}) = C_b(U_n)$</th>
<th>$C_b(U_{n+1})$</th>
<th>$C_p(U_{n+1})$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_b(U_{n+1}) = C_b(U_n)$</td>
<td>Continue</td>
<td>$C_b(U_{n+1})$</td>
<td>$C_p(U_{n+1})$</td>
</tr>
<tr>
<td>$C_b(U_{n+1}) \neq C_b(U_n)$</td>
<td>Smooth-Shift</td>
<td>$C_b(U_{n+1})$</td>
<td>$C_p(U_{n+1})$</td>
</tr>
<tr>
<td>$C_b(U_{n+1})$</td>
<td>Rough-Shift</td>
<td>$C_b(U_{n+1})$</td>
<td>$C_p(U_{n+1})$</td>
</tr>
</tbody>
</table>

- Rule 1: If any elt of $C_f(U_n)$ is realized by a pronoun in $U_{n+1}$ then $C_b(U_{n+1})$ must be realized as a pronoun also.
- Rule 2: Transition states are ordered. Continue > Retain > Smooth-Shift > Rough-Shift.

**Centering Algorithm**

- Generate possible $C_b$ - $C_f$ combinations for each possible set of reference assignments.
- Filter by constraints (syntactic coreference constraints, selectional, centering rules and constraints).
- Rank by transition orderings
- Assign referents based on Rule 2, if Rule 1 and other constraints not violated.

**Example redux**

- John saw a beautiful MGB at the dealership.
- He showed it to Bob.
- He bought it.

**Example redux**

- John saw a beautiful MGB at the dealership.
  - $C_f(U_1) = [John, MGB, dealership]$ - in order
  - $C_p(U_1) = John$
  - $C_b(U_1)$: undefined (highest ranked from prev $C$)
He showed it to Bob.  \( \text{it} = \text{MGB?} \)
- \( C_f(U_2) = \{ \text{John, MGB, Bob} \} \)
- \( C_p(U_2) = \text{John} \)
- \( C_b(U_2): \text{John} \) 
  - highest from \( C_f(U_1) \)
- Result: continue - \( C_p(U_2) = C_b(U_2), C_b(U_1) \) undefined

He showed it to Bob.  \( \text{it} = \text{dealership?} \)
- \( C_f(U_2) = \{ \text{John, dealership, Bob} \} \)
- \( C_p(U_2) = \text{John} \)
- \( C_b(U_2): \text{John} \)
- Result: continue - \( C_p(U_2) = C_b(U_2), C_b(U_1) \) undefined

Tied, arb pick MGB since 1st in \( C(U_1) \)

He bought it.  \( \text{it} = \text{MGB, he} = \text{John?} \)
- \( C_f(U_3) = \{ \text{John, MGB} \} \)
- \( C_p(U_3) = \text{John} \)
- \( C_b(U_3): \text{John} \) 
  - highest from \( C_f(U_2) \)
- Result: continue - \( C_p(U_3) = C_b(U_3) = C_b(U_2) \)

He bought it.  \( \text{it} = \text{MGB, he} = \text{Bob?} \)
- \( C_f(U_3) = \{ \text{Bob, MGB} \} \)
- \( C_p(U_3) = \text{Bob} \)
- \( C_b(U_3): \text{Bob} \)
- Result: Smooth-Shift - \( C_p(U_3) = C_b(U_3), C_b(U_2) \) != \( C_b(U_1) \)

Pick John as continue > Smooth-shift

**Centering**

- Implicitly incorporates grammatical role, recency, and repeated mention.
- Can get confused.
  - Bob opened a new bike shop last week. John took a look at the road bikes in his shop. He ended up buying one.
  - Incorrectly assigns he to “Bob” because \( C_b(U_2) = \text{Bob} \) so get continue, while “John” gets smooth-shift.

**Machine Learning**

- Train classifier: Log-linear (we skipped) or Naive Bayes.
- Rely on hand-labeled corpus where each pronoun linked to antecedent.
- Present positive and negative results for training.
- Extract features for training.

**Features**

- Commonly used for anaphora resolution:
  - strict gender [boolean]
  - compatible gender [boolean]
  - strict number [boolean]
  - compatible number [boolean]
  - sentence distance [0,1,2,...] from pronoun
  - Hobbs distance [0,1,2,...] # Hobbs NP skipped
  - Grammatical role [subject, object, PP]
  - Linguistic form [proper, definite, indefinite, pronoun]

**Example**

- John saw an MGB at the dealership. (U_1)
- He showed it to Bob. (U_3)
- He bought it. (U_3)
**Training**
- Train on vectors.
- Filter out pleonastic "it" as in "it is raining"
- Results in weights for each of the features and combinations of features.

**Co-reference Resolution**

**Coreferences**
- Extract coreference chains
- Secretary of State Colin Powell, he, Mr. Powell, Powell.
- Condoleeza Rice, she, Rice
- President Bush, Bush
- Can use machine learning classifier as before
  - Process from left to right.
  - For each NP, search backwards for match using classifier

**Need more features**
- Need to recognize that Microsoft is company to make sense of:
  - Microsoft announced record profits today. The company ...
- Jane .... The 50 year old mother of two ...

**Common Features**
- Anaphor edit distance \([0,1,2,...]\):
  \[
  100 \times \frac{m - (s + i + d)}{m}
  \]
  where \(m\) = size of antecedent.
- Antecedent edit distance \([0,1,2,...]\)
  \[
  100 \times \frac{n - (s + i + d)}{n}
  \]
  where \(n\) = size of anaphor

**Common Features**
- alias [true or false]: names equivalent or acronyms.
- appositive [true or false]: Mary, the new student, ...
- linguistic form [proper, definite, indef, pronoun] type of anaphor
Psychological Justification

- Reading time experiments
  - Clark & Sengal found reading time faster when referent for pronoun in most recent clause, rather than 2 or 3 back (for which speeds same)
  - Crawley found subjects identified antecedent of pronoun if subject more often than if object.
  - Smyth found strong impact of parallel placement.
  - Matthews & Chodorow found slower comprehension when pronoun antecedent occupied early syntactically deep position

Any Questions?