Semantics and Pragmatics of NLP
Pronouns

Alex Lascarides

School of Informatics
University of Edinburgh
Outline

1. Observations of what factors influence the way pronouns get resolved
2. Some algorithms that approximate these influences
Preferences for Pronoun Resolution

Recency: (cf. right-frontier in discourse structure; more later...)

(1) John has a Rover. Bill has a Ford. Mary likes to drive it.

Grammatical Role:

(2) a. John went to the car dealers with Bill. He bought a Rover. [he=John]
   b. Bill went to the car dealers with John. He bought a Rover. [he=Bill]
   c. Bill and John went to the car dealers. He bought a Rover. [he=??]
Repeated Mention: prior discourse focus likely to continue:

(3) John needed a new car. He decided he wanted something sporty. Bill went to the car dealers with him. He bought an MG. [he=John]

Parallelism:

(4) John went to Paris with Bill. Sue went to Toulouse with him. [him=Bill]

cf. Maximising Coherence!
Lexical Semantics:

(5) John telephoned Bill. He lost the pamphlet about MGs [he=John]

(6) John criticised Bill. He lost the pamphlet about MGs. [he=Bill]

General Semantics:

(7) a. John can open Bill’s safe. He knows the combination. [he=John]

b. John can open Bill’s safe. He now fears theft. [he=Bill]

cf. Maximise coherence!
Thematic Roles:

(8)  
  a. John seized the MG pamphlet from Bill. He loves reading about cars.  
      [Goal=John, Source=Bill]
  b. John passed the MG pamphlet to Bill. He loves reading about cars.  
      [Goal=Bill, Source=John]
  c. The car dealer admired John. He knows about MGs inside and out.  
      [Stimulus=John, Experience=dealer]
  d. The car dealer impressed John. He knows about MGs inside and out.  
      [Stimulus=dealer, Experience=John]

cf. Maximising Coherence!
Although a principle of interpreting discourse so as to maximise its (rhetorical) coherence captures an important generalisation, it’s not possible to implement it (currently). So we’ll look at some algorithms that approximate the predictions of the above preferences.
Algorithm 1: Lappin and Leass (1994)

(Simplified to handle just third person non-reflexive pronouns).

- Looks at recency and syntactic preferences, but not semantics.
- Weights assigned to preferences for pronoun resolution.
  - Weights make predictions about which preference wins when they conflict.
- Two operations: discourse update and pronoun resolution
Discourse Update

When you encounter an NP that evokes a new entity:

1. Add it to the discourse model, and
2. assign it a *salience value* = sum of weights given by *salience factors*.

- The Salience factors encodes degree of salience according to *syntax* the salience of the referent based on the properties of the NP that introduced it.
The Salience Factors

sentence recency: 100
subject emphasis: 80  An MG is parked outside.
Existential emphasis: 70  There is an MG parked outside
Direct object emphasis: 50  John drove an MG
Indirect obj. and oblique compl. emphasis: 40  John gave an MG a paint job
Non-adverbial emphasis: 50  John ate his lunch inside his MG > Inside his MG, John ate his lunch.
Head noun emphasis: 80  An MG is parked outside >
The manual for an MG is on the desk.

- Multiple mentions of a referent in the context potentially increase its salience (use highest weight for each factor).
Resolving Pronouns

First, factor in two more salience factors:

Role Parallelism: 35
Cataphora: -175

Then:
1. Collect potential referents (up to 4 sentences back)
2. Remove candidates where agreement etc. violated
3. Add above salience values to existing ones
4. Select referent with highest value.
(9)  a. John saw a beautiful MG at the dealership.
    b. He showed it to Bob.
    c. He bought it.

First sentence:

John:   100 (Rec) + 80 (subj) + 50 (non-adv) + 80 (head)  =  310  
MG:     100 (Rec) + 50 (obj) + 50 (non-adv) + 80 (head)  =  280  
dealership: 100 (Rec) + 50 (non-adv) + 80 (head)  =  230

No pronouns, so on to next sentence, degrading above by 2.
He showed it to Bob

John = 155; MG = 140; dealership = 115

*He:* MG and dealers ruled out (agreement); so John wins, and score increases (see below).

*it:* John (and he) ruled out (agreement, reflexive); MG wins, and score increases (see below).

*Bob:* Calculate score as below.

\[
\begin{align*}
\{\text{John, he}_1\} & : & 100 \text{ (Rec)} + 80 \text{ (subj)} + 50 \text{ (non-adv)} + 80 \text{ (head)} + 155 \text{ (prev. score)} & = 465 \\
\{\text{MG, it}_1\} & : & 100 \text{ (rec)} + 50 \text{ (obj)} + 50 \text{ (non-adv)} + 80 \text{ (head)} + 140 \text{ (prev. score)} & = 420 \\
\text{Bob} & : & 100 \text{ (rec)} + 40 \text{ (oblq.)} + 50 \text{ (non-adv)} + 80 \text{ (head)} & = 270 \\
dealership & : & \text{as before} & = 115
\end{align*}
\]
He bought it

\[
\begin{align*}
\{John, he_1\}: & \quad 232.5 \\
Bob: & \quad 135.0 \\
\{MG, it_1\}: & \quad 210.0 \\
dealership: & \quad 57.5
\end{align*}
\]

*He:* MG and dealers ruled out; John is highest score, so its score increases (see below).

*it:* John and bob ruled out; MG is highest score, so its score increases (see below).

\[
\begin{align*}
\{John, he_1, he_2\}: & \quad 100 \text{ (rec)} + 80 \text{ (subj)} + 50 \text{ (non-adv)} + 80 \text{ (head)} + 232.5 \text{ (prev)} = 542.5 \\
\{MG, it_1, it_2\}: & \quad 100 \text{ (rec)} + 50 \text{ (obj)} + 50 \text{ (non-adv)} + 80 \text{ (head)} + 210 \text{ (prev)} = 490.0 \\
Bob: & \quad (as \text{ before}) = 135.0 \\
dealership: & \quad (as \text{ before}) = 57.5
\end{align*}
\]
But How do you Assign Weights?

- These were computed by experimenting on a corpus of computer manuals (manual tuning).
- Algorithm achieves 86% accuracy on unseen test data.
- But accuracy with these weights may decrease for other genres.

Problems:

- Ignores semantics and discourse structure.
  E.g., discourse popping affects anaphora:

(10) To repair the pump, you’ve first got to remove the flywheel.
  ... [*lots of talk about how to do it.*]...
  Right, now let’s see if it works.
A Centering Algorithm

- Also constructs a discourse model, but without weights.
- Assumes there is a single entity being “centered” on at any time.

Forward-looking center $C_f(U_n)$:
- Ordered list of entities mentioned in sentence $U_n$.
  subj > existential > obj > oblique > . . .
  (cf. Lappin and Laess, 1994)

Backward-looking center $C_b(U_{n+1})$: (undefined for $U_1$)

$$C_b(U_{n+1}) =_{def} \text{highest ranked member of } C_f(U_n)$$
  that’s mentioned in $U_{n+1}$

$$C_f(U_n) =_{def} [C_p(U_n)|\text{rest}] \quad (C_p \text{ is preferred center})$$
Four relations based on $C_b$ and $C_p$ relations:

<table>
<thead>
<tr>
<th>$C_b(U_{n+1}) = C_b(U_n)$ or undefined $C_b(U_n)$</th>
<th>$C_b(U_{n+1}) \neq C_b(U_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_b(U_{n+1}) = C_p(U_{n+1})$</td>
<td>Continue</td>
</tr>
<tr>
<td>$C_b(U_{n+1}) \neq C_p(U_{n+1})$</td>
<td>Retain</td>
</tr>
</tbody>
</table>

Rules:

**Rule 1:** If any element of $C_f(U_n)$ is realised by a pronoun in $U_{n+1}$, then $C_b(U_{n+1})$ must be a pronoun too.

*John knows Mary.* ??*John loves her.*

**Rule 2:** Continue $>$ Retain $>$ Smooth-shift $>$ rough-shift
Observations About Data
Algorithms

The Algorithm

1. Generate $C_b - C_f$ combinations for each possible set of reference assignments;
2. Filter by constraints (selectional restrns, centering rules . . .)
3. Rank by orderings in Rule 2.

So the antecedent is assigned to yield the highest ranked relation from Rule 2 that doesn’t result in a violation of Rule 1 and other coreference constraints.
The Example Again

(9)  

a. John saw a beautiful MG at the dealership. \( U_1 \)
b. He showed it to Bob. \( U_2 \)
c. He bought it. \( U_3 \)

\[ C_b(U_1): \text{undefined} \]
\[ C_f(U_1): \{\text{John, MG, dealership}\} \]
\[ C_p(U_1): \text{John} \]

Sentence \( U_2 \):

- \textit{he} must be John because it’s the only choice (gender).
- So John is highest ranked in \( C_f(U_1) \) that’s also in \( U_2 \).
  So \( C_b(U_2) = \text{John} \).
He showed it to Bob

If *it* is MG, then:

\[
C_b(U_2): \text{John} \\
C_f(U_2): \{\text{John, MG, Bob}\} \\
C_p(U_2): \text{John} \\
\text{Result: Continue (because } C_p(U_2) = C_b(U_2); C_b(U_1) \text{ undefined)}
\]

If *it* is dealership, then:

\[
C_b(U_2): \text{John} \\
C_f(U_2): \{\text{John, dealership, Bob}\} \\
C_p(U_2): \text{John} \\
\text{Result: Continue (because } C_p(U_2) = C_b(U_2); C_b(U_1) \text{ undefined)}
\]

So no decision. Assume ties broken by ordering of previous \(C_f\)-list. So *it* = MG.
He bought it

- *it* compatible only with MG (dealership not in $C_f(U_2)$).
- *He* could be John or Bob.

**He=John:**
- $C_f(U_3)$: \{John (because *he*=John), MG\}
- $C_b(U_3)$: John
- $C_p(U_3)$: John

Result: **Continue**

- $(C_b(U_3) = C_p(U_3); C_b(U_3) = C_b(U_2))$

**He=Bob:**
- $C_f(U_3)$: \{Bob (because *he*=Bob), MG\}
- $C_b(U_3)$: Bob
- $C_p(U_3)$: Bob

Result: **Smooth-shift**

- $(C_b(U_3) = C_p(U_3); C_b(U_3) \neq C_b(U_2))$
     b. John took a look at the MGs in his lot.  
     c. He ended up buying one.  

Lappin and Laess: *he* in (11)c is John (exercise).  

Centering: 

\[
C_f(U_1) = \{\text{Bob, dealership}\} \quad C_f(U_2) = \{\text{John, MGs, Bob}\}  
C_p(U_1) = \text{Bob} \quad C_p(U_2) = \text{John}  
C_b(U_1) \text{ undefined} \quad C_b(U_2) = \text{Bob}
\]
Dealing with (11)c

     b. John took a look at the MGs in his lot.
     c. He ended up buying one.

\[ C_f(U_1) = \{Bob, dealership\} \quad C_f(U_2) = \{John, MGs, Bob\} \]
\[ C_p(U_1) = Bob \quad C_p(U_2) = John \]
\[ C_b(U_1) \text{ undefined} \quad C_b(U_2) = Bob \]

If *he* is John:
\[ C_f(U_3) = \{John, MG\} \]
\[ C_p(U_3) = John \]
\[ C_b(U_3) = John \]

If *he* is Bob:
\[ C_f(U_3) = \{Bob, MG\} \]
\[ C_p(U_3) = Bob \]
\[ C_b(U_3) = Bob \]

**Smooth-shift**
\[ (C_b(U_3) = C_p(U_3); C_b(U_2) \neq C_p(U_2)) \]

**Continue**
\[ (C_b(U_3) = C_p(U_3); C_b(U_2) = C_p(U_2)) \]
These methods are designed to handle pronouns where
the antecedent is in the prior sentence.
But they need to be extended to deal with cases where the
antecedent is in the same sentence:

(12) He worries that Glendenning’s initiative could push
his industry over the edge, forcing it to shift
operations elsewhere

*it* refers to *industry*. 
Kehler *et al.* (NAACL 2004), inspired by Lappin and Laess, use MaxEnt to learn from an annotated corpus the weights of candidate antecedents *both within and across sentence boundaries.*

Interestingly, they found that predicate-argument structure didn’t help the model:

- Predicting that *forcing industry* is more likely than *forcing initiative* or *forcing edge* doesn’t help.
Conclusions

- There are tractable algorithms for computing antecedents to pronouns.
- They vary in their predictions.
- But no algorithm clearly wins over the others.
- Errors are sometimes due to ignoring factors concerning discourse coherence.
- But ignoring discourse coherence is a practical necessity (for now).
- We’ll look at discourse coherence next...