LING/C SC/PSYC 438/538

Lecture 21

Sandiway Fong
Administrivia

• Homework 8 out today
  – Due date next Monday midnight
538 Presentations

- Updated list:

<table>
<thead>
<tr>
<th>Name</th>
<th>Column1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Last</td>
<td>First</td>
</tr>
<tr>
<td>Romero Diaz</td>
<td>Damian</td>
</tr>
<tr>
<td>Muriel</td>
<td>Jorge</td>
</tr>
<tr>
<td>Pike</td>
<td>Ammon Johnson</td>
</tr>
<tr>
<td>Farahnak</td>
<td>Farideh</td>
</tr>
<tr>
<td>Lee</td>
<td>Puay Leng Patricia</td>
</tr>
<tr>
<td>Brown</td>
<td>Rachel</td>
</tr>
<tr>
<td>Sullivan</td>
<td>Trevor</td>
</tr>
<tr>
<td>Xu</td>
<td>Dongfang</td>
</tr>
<tr>
<td>Nie</td>
<td>Xiaowen</td>
</tr>
</tbody>
</table>
s(s(NP, VP)) --> np(NP, Person, Number), vp(VP, TAG), {table(Person, Number, TAG)}.
np(np(NNP), 3, sq) --> nnp(NNP).
np(np(DT, NN), 3, Number) --> dt(DT, Number), nn(NN, Number).
np(Parse, 3, Number) -->
  dt(DT, Number), nn(NN, Number), w_np(Parse, np(DT, NN)).
w_np(Parse, NP) --> pp(PP), w_np(Parse, np(NP, PP)).
w_np(np(NP, PP), NP) --> pp(PP).
nnp(nnp(X)) --> [X], {proper_noun(X)}.
nn(nn(X), sg) --> [X], {common_noun(X)}.
nn(nns(X), pl) --> [X], {common_noun_pl(X)}.
vp(vp(VBF, NP), TAG) --> verb(VBF, TAG), np(NP, _Person, _Number).
vp(Parse, TAG) -->
  verb(VBF, TAG), np(NP, _Person, _Number), w_vp(Parse, vp(VBF, NP)).
w_vp(Parse, VP) --> pp(PP), w_vp(Parse, vp(VP, PP)).
w_vp(vp(VP, PP), VP) --> pp(PP).
dt(dt(the), _Number) --> [the].
dt(dt(a), sg) --> [a].
pp(pp(IN, NP)) --> in(IN), np(NP, _Person, _Number).
in(in(with)) --> [with].
g20.pl

21 proper_noun(X) :- member(X,[john,mary,peter,bill,jill]).
22 common_noun_pl(X) :- member(X,[balls,buckets,men,telescopes,boys]).
23 common_noun(X) :- member(X,[ball,bucket,man,telescope,boy]).
24 verb(vbg(kicking),vbg) --> [kicking].
25 verb(vbd(kicked),vbd) --> [kicked].
26 verb(vbn(kicked),vbn) --> [kicked].
27 verb(vbz(kicks),vbz) --> [kicks].
28 verb(vbp(kick),vbp) --> [kick].
29 verb(vb(kick),vb) --> [kick].
30 verb(vbg(seeing),vbg) --> [seeing].
31 verb(vbd(saw),vbd) --> [saw].
32 verb(vbn(seen),vbn) --> [seen].
33 verb(vbz(sees),vbz) --> [sees].
34 verb(vbp(see),vbp) --> [see].
35 verb(vb(see),vb) --> [see].
36 table(3,sg,vbz).
37 table(3,pl,vbp).
38 table(_,Person,_,Number,vbd).
Parses

- PP-attachment ambiguity example:

```prolog
?- s(Parse, [a, man, saw, the, boy, with, a, telescope], []).
Parse = s(np(dt(a), nn(man)), vp(vbd(saw), np(np(dt(the), nn(boy)), pp(in(with), np(dt(a), nn(telescope)))))),
Parse = s(np(dt(a), nn(man)), vp(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(telescope)))))),
false.
```
Berkeley Parser

(ROOT
  (S
    (NP (DT A) (NN man))
    (VP (VBD saw)
      (NP (DT the) (NN boy))
      (PP (IN with)
        (NP (DT a) (NN telescope))))
  (NP (DT the) (NN boy))
  (PP (IN with)
    (NP (DT a) (NN sandwich))))

(ROOT
  (S
    (NP (DT A) (NN man))
    (VP (VBD saw)
      (NP (DT the) (NN boy))
      (PP (IN with)
        (NP (DT a) (NN sandwich))))
  (NP (DT the) (NN boy))
  (PP (IN with)
    (NP (DT a) (NN sandwich))))
Parses

• PP-attachment ambiguity example:

?- s(Parse, [a, man, saw, the, boy, with, a, ball, with, a, telescope], []).
Parse = s(np(dt(a), nn(man)), vp(vbd(saw), np(np(dt(the), nn(boy)), pp(in(with), np(dt(a), nn(ball)))), pp(in(with), np(dt(a), nn(teleope))))).
Parse = s(np(dt(a), nn(man)), vp(vbd(saw), np(np(dt(the), nn(boy)), pp(in(with), np(dt(a), nn(ball))), pp(in(with), np(dt(a), nn(teleope))))).
Parse = s(np(dt(a), nn(man)), vp(np(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(teleope))))).
Parse = s(np(dt(a), nn(man)), vp(np(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(teleope))))).
Parse = s(np(dt(a), nn(man)), vp(np(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(teleope))))).
 Parse = s(np(dt(a), nn(man)), vp(np(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(teleope))))).
 Parse = s(np(dt(a), nn(man)), vp(np(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(teleope))))).
 Parse = s(np(dt(a), nn(man)), vp(np(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(teleope))))).
 Parse = s(np(dt(a), nn(man)), vp(np(vp(vbd(saw), np(dt(the), nn(boy))), pp(in(with), np(dt(a), nn(teleope))))).
false.

?- 
Parses

• PP-attachment ambiguity example: (1)
Parses

- PP-attachment ambiguity example: (2)
Parses

• PP-attachment ambiguity example: (3)
Parses

• PP-attachment ambiguity example: (4)
Parses

• PP-attachment ambiguity example: (5)
Berkeley Parser

(S
  (NP (DT A) (NN man))
  (VP (VBD saw)
   (NP (DT the) (NN boy))
   (PP (IN with)
    (NP
     (NP (DT a) (NN ball))
     (PP (IN with)
      (NP (DT a) (NN telescope)))))))

= Parse #4 above
saw
  boy
  ball
telescope
Parses

- Subject-Verb agreement by feature percolation

```prolog
?- s(Parse,[the,man,kick,ball],[[]]).
false.

?- s(Parse,[the,man,kicks,ball],[[]]).
Parse = s(np(dt(the), nn(man)), vp(vbz(kicks), np(dt(the), nn(ball))));
false.

?- s(Parse,[the,men,kicks,ball],[[]]).
false.

?- s(Parse,[the,men,kick,ball],[[]]).
Parse = s(np(dt(the), nns(men)), vp(vbp(kick), np(dt(the), nn(ball))));
false.

?- s(Parse,[the,men,kicked,ball],[[]]).
Parse = s(np(dt(the), nns(men)), vp(vbd(kicked), np(dt(the), nn(ball))));
false.

?- s(Parse,[the,man,kicked,ball],[[]]).
Parse = s(np(dt(the), nn(man)), vp(vbd(kicked), np(dt(the), nn(ball))));
false.
```
Parses

• Subject-Verb agreement by feature percolation:

```
1 \textit{s(s(NP,VP)) \rightarrow np(NP,Person,Number), vp(VP,TAG),
    \{\textit{table(Person,Number,TAG)}\}}.

37 \textit{table(3,sg,vbz).}
38 \textit{table(3,pl,vbp).}
39 \textit{table(_Person,_Number,vbd).}
```
Writing natural language grammars

• Mechanisms:
  1. Extra argument for a Prolog term representation of a parse
  2. Extra arguments for feature value agreement
  3. Dealing with left recursive rules: grammar transformation

• *Now it's your turn...*
Homework 8

• Question 1: write a grammar with parses for
  1. We awarded several prizes
  2. We will award several prizes
  3. *We will awarded several prizes
  4. Several prizes will be awarded
  5. *Several prizes will be award
  6. *Several prizes will be awarded several prizes

• Note:
  – your program should accept and print a parse representation for sentences marked as being grammatical, and reject the ungrammatical ones.
Homework 8

• Question 1:
  – submit your grammar
  – show output on sentences 1–6.
  – Example:

```prolog
?- s(P,[we, awarded, several, prizes],[[]]).
P = s(np(prp(we)), vp(vbd(awarded)), np(dt(several), nns(prizes)))) ;
false.
?- 
```
Homework 8

3. *We will awarded several prizes*
4. *Several prizes will be award*
5. *Several prizes will be awarded several prizes*

```prolog
[?- s(P,[we, will, awarded, several, prizes],[]).] false.
[?- s(P,[several, prizes, will, be, award],[]).] false.
[?- s(P,[several, prizes, will, be, awarded, several, prizes],[]).] false.
?- ]
```
Homework 8

• Question 2: extend your grammar to include
  7. It is likely that we awarded several prizes
  8. It is likely we awarded several prizes
  9. It is likely that we will award several prizes
 10. It is likely we will award several prizes
 11. It is likely that several prizes will be awarded
 12. It is likely several prizes will be awarded
Homework 8

• Question 2:
  – submit your revised grammar
  – **highlight** the changes you made to the grammar you submitted for Question 1
  – show output on sentences 7–12.
Homework 8

• Question 3: further extend your grammar to include

13. There are likely to be several prizes awarded
14. *There is likely to be several prizes awarded
15. *There are likely to be awarded several prizes
16. *There are likely that we will awarded several prizes
17. *There are likely we will awarded several prizes
18. There is likely to be a prize awarded
19. *There are likely to be a prize awarded
20. *There is likely to be awarded a prize

This is where the grammar programming really starts...
(13) comes from Derivation By Phase (Chomsky, 2001)
Homework 8

• Question 3:
  – submit your revised grammar
  – **highlight** the changes you made to the grammar you submitted for Question 2
Homework 8

• There are likely to be several prizes awarded
• Salient Points:
  1. Existential there (POS tag EX)
  2. Notice the long distance agreement between matrix be and several prizes.
  3. Infinitival to (POS tag TO)
  4. Notice the position of the object several prizes.
     It has been shifted to the left of the main verb award of the embedded clause.
     (Leftwards TH/EX) (Thematization/Extraction.)
13. There are likely to be several prizes awarded

Berkeley Parser: incorrect parse
13. There are likely to be several prizes awarded

?- s(P,Num,[there,are,likely,to,be,several,prizes,awarded],[]).
P = s(ex(there),vp(vbp(are)),adjp(adj(likely)),sbar(s(vp(to(to),vp(vb(be),vp(np(dt(several),nns(prizes))),vp(vbn(awarded))))),Num = pl
true ;
false.
Homework 8

• Question 4: Finally, also include

21. Several prizes are likely to be awarded
22. *Several prizes is likely to be awarded
23. *Several prizes are likely that will be awarded
24. *Several prizes are likely will be awarded
25. A prize is likely to be awarded
26. *A prize are likely to be awarded