LING 388: Language and Computers

Sandiway Fong
Lecture 16
Administrivia

• Homework 5 was due last night
Homework 5 review

• Implement another kind of recursion:
  – Object relative clauses
    • *the cat that saw the rat that saw the cheese that ...*
    • [{NP the cat [SBAR that [S saw [NP the rat [SBAR that [S saw [NP the cheese that ... ]]]]]]]]
    • *the cheese that the rat ate*
    • *the cheese that the rat that the cat saw ate*
    • *the cheese that the rat that the cat that the dog chased saw ate*
Homework 5 review

Starting point (grammar.pl):
1. sentence(s(X,Y)) → np(X), vp(Y).
2. pp(pp(X,Y)) → in(X), np(Y).
3. in(in(with)) → [with].
4. np(np(X)) → prp(X).
5. np(np(np(X,Y),Z)) → det(X), nn(Y), pp(Z).
6. np(np(D,NN)) → det(D), nn(NN).
7. prp(prp(i)) → [i].
8. prp(prp(me)) → [me].
9. nn(nn(boy)) → [boy].
10. nn(nn(telescope)) → [telescope].
11. vp(vp(V,X)) → verb(V), np(X).
12. vp(vp(V,X,Y)) → verb(V), np(X), pp(Y).
13. verb(vbd(saw)) → [saw].
14. det(dt(the)) → [the].
15. det(dt(a)) → [a].
Last Time

- Introduced the copy/rename technique to restrict the scope of the empty category rule:
  \[- \text{np(np(0))} \rightarrow []\]
Subject relative clauses

subjrel_g.pl
1. sbar(sbar(C,S)) --> complementizer(C), s(S).
2. subjrel_sbar(sbar(C,S)) --> complementizer(C), subjrel_s(S).
3. complementizer(c(that)) --> [that].
4. s(s(NP,VP)) --> np(NP), vp(VP).
5. subjrel_s(s(NP,VP)) --> empty_np(NP), vp(VP).
6. vp(vp(VBD,NP)) --> vbd(VBD), np(NP).
7. np(np(DT,NN)) --> dt(DT), nn(NN).
8. np(np(np(DT,NN),SBAR)) --> dt(DT), nn(NN), sbar(SBAR).
9. np(np(np(DT,NN),SBAR)) --> dt(DT), nn(NN), subjrel_sbar(SBAR).
10. empty_np(np(0)) --> [].
11. dt(dt(the)) --> [the].
12. nn(nn(rat)) --> [rat].
13. nn(nn(cat)) --> [cat].
14. nn(nn(cheese)) --> [cheese].
15. vbd(vbd(saw)) --> [saw].
16. vbd(vbd(chased)) --> [chased].

rules that were copied/renamed are in brown
Copy and Rename

Restricting the scope of the empty category rule:

• subject relative clauses force an empty subject
  - the rat that $\emptyset$ saw the cheese
  - the rat that $\emptyset$ saw $\emptyset$
  - the rat that the rat saw the cheese

\[
\text{np(np(np(D,N),S\text{BAR}))) \rightarrow \text{det}(D), \text{n}(N), \text{subjrel_sbar}(S\text{BAR}).
\]

\[
\text{subjrel_sbar}(s\text{bar}(C,S)) \rightarrow c(C), \text{subjrel_s}(S).
\]

\[
\text{subjrel_s}(s(NP,VP)) \rightarrow \text{empty_np}(NP), \text{vp}(VP).
\]

\[
\text{empty_np}(np(\emptyset)) \rightarrow \square.
\]
Object relative clauses

• Propose similar rule changes for object relative clauses:
  • the cheese that the rat ate Ø
  • the cheese that Ø ate Ø
  • the cheese that the rat saw the cheese
Agreement in English

• Determiner-Noun Agreement

• example
  – English determiner-noun number agreement
  – data
    • the man
    • the men
    • a man
    • *a men
  – lexical features
    • man [singular]
    • men [plural]

Like in the case of want vs. trust:
the [singular] want vbp1
the [plural] want vbp2
a [singular] trust vbp1
Agreement in English

• Method 1: *nonterminal renaming*
  — rename grammar rules to constrain the possibilities for determiner and noun co-occurrence:

\[
\begin{align*}
\text{dt}(\text{dt}(a)) & \rightarrow [a]. \\
\text{dt}(\text{dt}(\text{the})) & \rightarrow [\text{the}].
\end{align*}
\]

\[
\begin{align*}
\text{dt}_\text{singular}(\text{dt}(a)) & \rightarrow [a]. \\
\text{dt}_\text{plural}(\text{dt}(\text{the})) & \rightarrow [\text{the}]. \\
\text{dt}_\text{singular}(\text{dt}(\text{the})) & \rightarrow [\text{the}].
\end{align*}
\]

— write lexical rules for *man/men:*

\[
\begin{align*}
\text{nn}_\text{singular}(\text{nn}(\text{man})) & \rightarrow [\text{man}]. \\
\text{nn}_\text{plural}(\text{nn}(\text{men})) & \rightarrow [\text{men}].
\end{align*}
\]
Agreement in English

- Method 1: *nonterminal renaming*
  - rename grammar rules to constrain the possibilities for determiner and noun co-occurrence:

```plaintext
% NP rules
empty_np(np(0)) --> [].
% np(np(np(D,N),SBAR)) --> dt(D), nn(N), sbar(SBAR).
np(np(np(D,N),SBAR)) --> dt(D), nn(N), objrel_sbar(SBAR).
np(np(np(D,N),SBAR)) --> dt(D), nn(N), subjrel_sbar(SBAR).
np(np(D,N)) --> dt(D), nn(N).
np(np(N)) --> proper_noun(N); pronoun(N).
nn(nn(a(Adj),N)) --> adj(Adj), nn(N).
```

- rewrite existing rules involving nn and dt
Agreement in English

• Need to also rename other lexical rules:

\[
\begin{array}{ll}
38 & nn(nn(bus)) \rightarrow [bus]. \\
39 & nn(nn(cat)) \rightarrow [cat]. \\
40 & nn(nn(dog)) \rightarrow [dog]. \\
41 & nn(nn(rat)) \rightarrow [rat]. \\
42 & nn(nn(cheese)) \rightarrow [cheese]. \\
43 & nn(nn(race)) \rightarrow [race].
\end{array}
\]

How should we handle *cheese*?
Agreement in English

• Method 2: use an extra argument to hold a variable corresponding to the Number feature
  e.g. dt(D,Number), nn(N,Number)
  then the value of the variable Number must match for determiner and noun

  – the [plural/singular]
  – a [singular]
  – man [singular]
  – men [plural]
Agreement in English

• Lexical rules for determiners and nouns must now define the value of Number as a 2\textsuperscript{nd} argument as well

• e.g.
  – nn(nn(man),singular) \rightarrow [\text{man}].
  – dt(dt(a),singular) \rightarrow [a].