LING 388: Language and Computers

Sandiway Fong
Lecture 18
Administrivia

• Next week, Ben Martin will be teaching in place of me ...
Last Time

• We implemented subject-verb agreement:
  – *John eats/*eat cheese

Combine information from the subject NP and the verb

– NP information
  • e.g. John Person:3rd, Number: singular

– Verb inflectional endings

<table>
<thead>
<tr>
<th></th>
<th>Form</th>
<th>POS tag</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>eat</td>
<td>vb</td>
<td>*not 3rd person singular, present</td>
</tr>
<tr>
<td>2.</td>
<td>eats</td>
<td>vbz</td>
<td>3rd person singular, present</td>
</tr>
<tr>
<td>3.</td>
<td>ate</td>
<td>vbd</td>
<td>past</td>
</tr>
<tr>
<td>4.</td>
<td>eaten</td>
<td>vbn</td>
<td>past participle (passive, perfective)</td>
</tr>
<tr>
<td>5.</td>
<td>eating</td>
<td>vbg</td>
<td>gerund (progressive)</td>
</tr>
</tbody>
</table>
• Constraint table:
  - \textit{table of Person Number Tag possible combinations}
  - check(3, plural, vb).
  - check(3, plural, vbd).
  - check(3, singular, vbz).
  - check(3, singular, vbd).
\[ g17.pl \]

- **Sentence rules:**

  \[
  \text{s(s(NP,VP))} \rightarrow \text{np(NP,Person,Number)}, \ \text{vp(VP,Tag)}, \\
  \{\text{check(Person,Number,Tag)}\}. \\
  \text{objrel_s(s(NP,VP))} \rightarrow \text{np(NP,Person,Number)}, \ \text{objrel_vp(VP,Tag)}, \\
  \{\text{check(Person,Number,Tag)}\}. \\
  \text{subjrel_s(s(NP,VP))} \rightarrow \text{empty_np(NP,Person,Number)}, \ \text{vp(VP,Tag)}, \\
  \{\text{check(Person,Number,Tag)}\}. \\
  \]

1. \{\text{check(Person,Number,Tag)}\} \quad \text{Prolog code}
2. \text{np(Parse,Person,Number)} \quad \text{nonterminal}
3. \text{vp(Parse,Tag)} \quad \text{nonterminal}
g17.pl

• VP rules (with updated calls to np):
  \[ \text{vp(vp(V,NP),Tag)} \rightarrow \text{verb(V,Tag), np(NP,_,_).} \]
  \[ \text{objrel_vp(vp(V,NP),Tag)} \rightarrow \text{verb(V,Tag), empty_np(NP,_,_).} \]

1. vp(Parse,Tag)
2. verb(Parse,Tag)
3. np(Parse,Person,Number)
4. empty_np(Parse,Person,Number)
g17.pl

• NP rules:

\[
\text{np(np(DT,NN),Person,Number) } \rightarrow \text{ dt(DT,Number), nn(NN,Number,Person).}
\]

\[
\text{np(np(np(DT,NN),SBAR),Person,Number) } \rightarrow \\
\quad \text{ dt(DT,Number), nn(NN,Number,Person), objrel_sbar(SBAR).}
\]

\[
\text{np(np(np(DT,NN),SBAR),Person,Number) } \rightarrow \\
\quad \text{ dt(DT,Number), nn(NN,Number,Person), subjrel_sbar(SBAR).}
\]

1. np(Parse,Person,Number)
2. nn(Parse,Number,Person)
g17.pl

• nn (common noun) rules:
  nn(nn(man),singular,3) --> [man].
  nn(nn(men),plural,3) --> [men].
  nn(nn(rat),singular,3) --> [rat].
  nn(nn(cat),singular,3) --> [cat].
  nn(nn(Root-Suffix),Number,Person) --> [Word],
    {atom_concat(Root,Suffix,Word),
     suffix(Number,[Suffix],[[]]),
     nn(_,singular,Person,[Root],[[]])}.
  nn(nn(cheese),mass,3) --> [cheese].
  suffix(plural) --> [s].

  nn(Parse,Number,Person)
g17.pl

- dt (determiner) rules:
  dt(dt(the),_) --> [the].
  dt(dt(a),singular) --> [a].
• verb rules:
  verb(vb(see),vb) --> [see].
  verb(vbz(sees),vbz) --> [sees].
  verb(vbg(seeing),vbg) --> [seeing].
  verb(vbd(saw),vbd) --> [saw].
  verb(vbn(seen),vbn) --> [seen].

  verb(vb(eat),vb) --> [eat].
  verb(vbz(eats),vbz) --> [eats].
  verb(vbg(eating),vbg) --> [eating].
  verb(vbd(ate),vbd) --> [ate].
  verb(vbn(eaten),vbn) --> [eaten].

  verb(vb(chase),vb) --> [chase].
  verb(vbz(chases),vbz) --> [chases].
  verb(vbg(chasing),vbg) --> [chasing].
  verb(vbd(chased),vbd) --> [chased].
  verb(vbn(chased),vbn) --> [chased].
• Other rules (not affected by subject-verb agreement):
  sbar(sbar(C,S)) --> complementizer(C), s(S).
  objrel_sbar(sbar(C,S)) --> complementizer(C), objrel_s(S).
  subjrel_sbar(sbar(C,S)) --> complementizer(C), subjrel_s(S).
  complementizer(c(that)) --> [that].
• Empty NP rule:
  empty_np(np(0),3,singular) --> []. % for now...

• We can't always assume those Person and Number values. Why?

• Examples:
  1. the rats/rat that ate the cheese
  2. the rats that eat/*eats the cheese
  3. the rat that *eat/eats the cheese

Let's fix g17.pl for the subject relative clause case
English Passivization

• 2\textsuperscript{nd} extra argument verb Tag can also be useful in verb to verb constraints

\begin{align*}
v(vb(eat),vb) & \rightarrow [eat]. \\
v(vbp(eat),vbp) & \rightarrow [eat]. \\
v(vbz(eats),vbz) & \rightarrow [eats]. \\
v(vbg(eating),vbg) & \rightarrow [eating]. \\
v(vbd(ate),vbd) & \rightarrow [ate]. \\
v(vbn(eaten),vbn) & \rightarrow [eaten].
\end{align*}

Stanford parser:

\begin{verbatim}
(ROOT
  (S
    (NP (DT The) (NN sandwich)))
    (VP (VBD was)
      (VP (VBN eaten)))
    ( . . ))
\end{verbatim}

\textbf{Passive morphology}

1. the sandwich was eaten
2. *the sandwich was eat
3. *the sandwich was eats
4. *the sandwich was ate
5. *the sandwich was eating
English Passivization

• Stanford parser:

  S
   NP  VP
   DT  NN  VBD  VBN
   The sandwich was  eaten

  S
   NP  VP
   DT  NN  VBD  ADJP
   The sandwich was  ate

  S
   NP  VP
   DT  NN  VBD  VBG
   The sandwich was  eating

  S
   NP  VP
   DT  NN  VBD  VBG
   The sandwich was  eats

Passive morphology
1. the sandwich was eaten
2. *the sandwich was eat
3. *the sandwich was eats
4. *the sandwich was ate
5. *the sandwich was eating

Non-passive reading...
Let’s implement this passive morphology constraint,
  – i.e. passive auxiliary verb *be*
    requires following verb to have the past participle ending (*vbn, -en*)

\[
\text{vp}(\text{vp}(\text{Vpassive}, \text{vp}(V))) \rightarrow \text{v\_passive}(\text{Vpassive}), \text{v}(V, \text{vbn}).
\]
\[
\text{v\_passive}(\text{aux}(\text{was})) \rightarrow [\text{was}].
\]
\[
\text{v\_passive}(\text{aux}(\text{were})) \rightarrow [\text{were}].
\]
English Passivization

• How about subject verb agreement for passive *be*?
  1. the sandwich was/*were eaten
  2. the sandwiches *was/were eaten