Feature Structures and Unification Grammars

11-711 Algorithms for NLP
15 November 2016 – Part II
Linguistic features

• (Linguistic “features” vs. ML “features”.)
• Human languages usually include agreement constraints; in English, e.g., subject/verb
  – I often swim
  – He often swims
  – They often swim
• Could have a separate category for each minor type: N1s, N1p, ..., N3s, N3p, ...
  – Each with its own set of grammar rules!
A day without features...

- NP1s → Det-s N1s
- NP1p → Det-p N1p
  ...
- NP3s → Det-s N3s
- NP3p → Det-p N3p
  ...
- S1s → NP1s VP1s
- S1p → NP1p VP1p
- S3s → NP3s VP3s
- S3p → NP3p VP3p
Linguistic features

• *Could* have a separate category for each minor type: N1s, N1p, ... , N3s, N3p, ...
  – *Each* with its own set of grammar rules!

• Much better: represent these regularities using independent *features*: number, gender, person, ...

• Features are typically introduced by lexicon; checked and propagated by constraint equations attached to grammar rules
Feature Structures (FSs)

Having multiple orthogonal features with values leads naturally to **Feature Structures:**

\[
[\text{Det} \\
 [\text{root}: a] \\
 [\text{number}: sg ]]
\]

A feature structure’s values can in turn be FSs:

\[
[NP \\
 [\text{agreement}: [[\text{number}: sg] \\
 [\text{person}: 3rd]]]]
\]

Feature Path: <NP agreement person>
Adding constraints to CFG rules

- \( S \rightarrow NP \ VP \)
  \(<NP \text{ number}> = <VP \text{ number}>\)
- \( NP \rightarrow \text{Det} \ Nominal \)
  \(<NP \text{ head}> = <\text{Nominal head}>\)
  \(<\text{Det head agree}> = <\text{Nominal head agree}>\)
FSs from lexicon, constrs. from rules

Lexicon entry:

[Det
  [root: a]
  [number: sg]]

Rule with constraints:

NP → Det Nominal

<NP number> = <Det number>
<NP number> = <Nominal number>

• Combine to get result:

[NP [Det
  [root: a]
  [number: sg]]
  [Nominal [number: sg] ...]
  [number: sg]]
Similar issue with VP types

Another place where grammar rules could explode:

Jack laughed

\[ VP \rightarrow \text{Verb} \quad \text{for many specific verbs} \]

Jack found a key

\[ VP \rightarrow \text{Verb} \ \text{NP} \quad \text{for many specific verbs} \]

Jack gave Sue the paper

\[ VP \rightarrow \text{Verb} \ \text{NP} \ \text{NP} \quad \text{for many specific verbs} \]
Verb Subcategorization

Verbs have sets of allowed args. Could have many sets of VP rules. Instead, have a SUBCAT feature, marking sets of allowed arguments:

```
+none -- Jack laughed
+np -- Jack found a key
+np+np -- Jack gave Sue the paper
+vp:inf -- Jack wants to fly
+np+vp:inf -- Jack told the man to go
+vp:ing -- Jack keeps hoping for the best
+np+vp:ing -- Jack caught Sam looking at his desk
+np+vp:base -- Jack watched Sam look at his desk
+np+pp:to -- Jack gave the key to the man

+pp:loc -- Jack is at the store
+np+pp:loc -- Jack put the box in the corner
+pp:mot -- Jack went to the store
+np+pp:mot -- Jack took the hat to the party
+adjp -- Jack is happy
+np+adjp -- Jack kept the dinner hot
+sfor -- Jack believed that the world was flat
+sfor -- Jack hoped for the man to win a prize
```

50-100 possible frames for English; a single verb can have several.

*(Notation from James Allen “Natural Language Understanding”)*
Frames for “ask”  
*(in J+M notation)*

<table>
<thead>
<tr>
<th>Subcat</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Quo</em></td>
<td>asked ([Quo \text{ “What was it like?”}])</td>
</tr>
<tr>
<td><em>NP</em></td>
<td>asking ([NP \text{ a question}])</td>
</tr>
<tr>
<td><em>Swh</em></td>
<td>asked ([Swh \text{ what trades you’re interested in}])</td>
</tr>
<tr>
<td><em>Sto</em></td>
<td>ask ([Sto \text{ him to tell you}])</td>
</tr>
<tr>
<td><em>PP</em></td>
<td>that means asking ([PP \text{ at home}])</td>
</tr>
<tr>
<td><em>Vto</em></td>
<td>asked ([Vto \text{ to see a girl called Evelyn}])</td>
</tr>
<tr>
<td><em>NP Sif</em></td>
<td>asked ([NP \text{ him}] [Sif \text{ whether he could make}])</td>
</tr>
<tr>
<td><em>NP NP</em></td>
<td>asked ([NP \text{ myself}] [NP \text{ a question}])</td>
</tr>
<tr>
<td><em>NP Swh</em></td>
<td>asked ([NP \text{ him}] [Swh \text{ why he took time off}])</td>
</tr>
</tbody>
</table>
Adding transitivity constraint

- $S \rightarrow NP \ VP$
  
  $<NP\ number> = <VP\ number>$

- $NP \rightarrow \text{Det Nominal}$
  
  $<NP\ head> = <\text{Nominal}\ head>$
  $<\text{Det head agree}> = <\text{Nominal head agree}>$

- $VP \rightarrow \text{Verb} \ NP$
  
  $<VP\ head> = <\text{Verb}\ head>$
  $<VP\ head\ subcat> = +np \quad (\text{which means transitive})$
Applying a verb subcat feature

Lexicon entry:

[Verb
  [root: found]
  [head: find]
  [subcat: +np ]]

• Combine to get result:

[VP [Verb
  [root: found]
  [head: find]
  [subcat: +np ]]
[NP ...]
 [head: [find [subcat: +np]]]]

Rule with constraints:

VP → Verb NP
<VP head> = <Verb head>
<VP head subcat> = +np
Relation to LFG constraint notation

- \( VP \rightarrow \text{Verb} \quad \text{NP} \)
  \(<\text{VP head}> = <\text{Verb head}>\)
  \(<\text{VP head subcat}> = +\text{np}\)

  *from JM book is the same as the LFG expression*

- \( VP \rightarrow \text{Verb} \quad \text{NP} \)
  \((↑\text{head}) = (↓\text{head})\)
  \((↑\text{head subcat}) = +\text{np}\)
Unification

• Merging FSs (and failing if not possible) is called **Unification**

• Simple FS examples:

  \[ \text{[number sg]} \sqcup \text{[number sg]} = \text{[number sg]} \]
  \[ \text{[number sg]} \sqcup \text{[number pl]} \text{ **FAILS**} \]
  \[ \text{[number sg]} \sqcup \text{[number [ ]]} = \text{[number sg]} \]
  \[ \text{[number sg]} \sqcup \text{[person 3rd]} = \text{[number sg, person 3rd]} \]
Recap: applying constraints

Lexicon entry:

[Det
  [root: a]
  [number: sg]]

Rule with constraints:

NP → Det Nominal

<NP number> = <Det number>
<NP number> = <Nominal number>

• Combine to get result:

[NP [Det
  [root: a]
  [number: sg]]
[Nominal [number: sg] ...]
[number: sg]]
Turning constraint eqns. into FS

Lexicon entry:

[Det
  [root: a]
  [number: sg ]]

- Combine to get result:

  [NP [Det
    [root: a]
    [number: sg ]]
  [Nominal [number: sg]
    ...
  [number: sg]]]

Rule with constraints:

NP → Det Nominal

\(<\text{NP number}\> = <\text{Det number}>\)
\(<\text{NP number}\> = <\text{Nominal number}>\)

becomes:

[NP [Det [number: (1) ]]
  [Nominal
    [number: (1) ]
    ...
  [number: (1) ]]]
Another example

This (oversimplified) rule:

\[ S \to NP \ VP \]

\[ <S \text{ subject}> = NP \]

\[ <S \text{ agreement}> = <S \text{ subject agreement}> \]

turns into this DAG:

[S [subject (1)
    [agreement (2) ]]
[agreement (2) ]
[agreement (2) ]
[NP (1) ]
[VP ]
Unification example without “EQ“

[agreement [number sg],
 subject [agreement [number sg]]]
⊔[subject [agreement [person 3rd, number sg]]]
= [agreement [number sg],
 subject [agreement [person 3rd, number sg]]]

• <agreement number> is equal to <subject agreement number>, but not EQ
Unification example with “EQ“

[agreement (1), subject [agreement (1)]]

⊔ [subject [agreement [person 3rd, number sg]] = [agreement (1), subject [agreement (1) [person 3rd, number sg]]]

• <agreement number> is <subject agreement number> (EQ), so they are equal
Representing FSs as DAGs

• Taking feature paths seriously
• May be easier to think about than numbered cross-references in text
• [cat NP, agreement [number sg, person 3rd]]
Re-entrant FS as DAGs

- [cat S, head [agreement (1) [number sg, person 3rd], subject [agreement (1)]]]
Seems tricky. Why bother?

• Unification allows the systems that use it to handle many complex phenomena in “simple” elegant ways:
  – There seems to be a dog in the yard.
  – There seem to be dogs in the yard

• Unification makes this work smoothly.
  – Make the Subjects of the clauses EQ:
    <VP subj> = <VP COMP subj>
    [VP [subj: (1)] [COMP [subj: (1)]]]
  – (Ask Lori Levin for LFG details.)
Real Unification-Based Parsing

- $X_0 \rightarrow X_1 X_2$
  
  $<X_0 \text{ cat}> = S$, $<X_1 \text{ cat}> = \text{NP}$, $<X_2 \text{ cat}> = \text{VP}$
  
  $<X_1 \text{ head agree}> = <X_2 \text{ head agree}>$
  
  $<X_0 \text{ head}> = <X_2 \text{ head}>$

- $X_0 \rightarrow X_1 \text{ and } X_2$
  
  $<X_1 \text{ cat}> = <X_2 \text{ cat}>$, $<X_0 \text{ cat}> = <X_1 \text{ cat}>$

- $X_0 \rightarrow X_1 X_2$
  
  $<X_1 \text{ orth}> = \text{how}$, $<X_2 \text{ sem}> = <\text{SCALAR}>$
Complexity

• Earley modification: “search the chart for states whose DAGs unify with the DAG of the completed state”. Plus a lot of copying.

• Unification parsing is “quite expensive”.
  – NP-Complete in some versions.
  – Early AWB paper on Turing Equivalence(!)

• So maybe too powerful?
  (like GoTo or Call-by-Name?)
  – Add restrictions to make it tractable:
    • Tomita’s Pseudo-unification (Tomabechi too)
    • Gerald Penn work on tractable HPSG: ALE
Formalities: subsumption

- Less specific FS1 **subsumes** more specific FS2
  \[ \text{FS1} \sqsubseteq \text{FS2} \quad \text{(Inverse is FS2 extends FS1)} \]
- Subsumption relation forms a **semilattice**, at the top: [ ]

\[
\begin{align*}
&\text{[number sg]} \quad \text{[person 3]} \quad \text{[number pl]} \\
&\text{[number sg, person 3]} \\
\end{align*}
\]

- Unification defined wrt semilattice:
  \[ F \sqcup G = H \quad \text{s.t.} \quad F \sqsubseteq H \text{ and } G \sqsubseteq H \]
  H is the Most General Unifier (MGU)
Hierarchical Types

Hierarchical types allow *values* to unify too (or not):
Hierarchical subcat frames

Many verbs share *subcat* frames, some with more arguments specified than others:
Questions?
## Subcategorization

<table>
<thead>
<tr>
<th>Noun Phrase Types</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>There</strong></td>
<td>nonreferential there</td>
<td><strong>There</strong> is still much to learn</td>
</tr>
<tr>
<td><strong>It</strong></td>
<td>nonreferential it</td>
<td><strong>It</strong> was evident that <strong>my ideas</strong></td>
</tr>
<tr>
<td><strong>NP</strong></td>
<td>noun phrase</td>
<td><strong>As he</strong> was relating his story</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Preposition Phrase Types</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PP</strong></td>
<td>preposition phrase</td>
<td><strong>couch their message in terms</strong></td>
</tr>
<tr>
<td><strong>PPing</strong></td>
<td>gerundive PP</td>
<td><strong>censured him for not having intervened</strong></td>
</tr>
<tr>
<td><strong>PPpart</strong></td>
<td>particle</td>
<td><strong>turn it off</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Verb Phrase Types</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VPbrst</strong></td>
<td>bare stem VP</td>
<td><strong>she could discuss it</strong></td>
</tr>
<tr>
<td><strong>VPto</strong></td>
<td>to-marked infin. VP</td>
<td><strong>Why do you want to know?</strong></td>
</tr>
<tr>
<td><strong>VPwh</strong></td>
<td>wh-VP</td>
<td><strong>it is worth considering how to write</strong></td>
</tr>
<tr>
<td><strong>VPing</strong></td>
<td>gerundive VP</td>
<td><strong>I would consider using it</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Complement Clause Types</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sfin</strong></td>
<td>finite clause</td>
<td><strong>maintain that the situation was unsatisfactory</strong></td>
</tr>
<tr>
<td><strong>Swh</strong></td>
<td>wh-clause</td>
<td><strong>it tells us where we are</strong></td>
</tr>
<tr>
<td><strong>Sif</strong></td>
<td>whether/if clause</td>
<td><strong>ask whether Aristophanes is depicting a</strong></td>
</tr>
<tr>
<td><strong>Sing</strong></td>
<td>gerundive clause</td>
<td><strong>see some attention being given</strong></td>
</tr>
<tr>
<td><strong>Sto</strong></td>
<td>to-marked clause</td>
<td><strong>know themselves to be relatively unhealthy</strong></td>
</tr>
<tr>
<td><strong>Sfrom</strong></td>
<td>for-to clause</td>
<td><strong>She was waiting for him to make some reply</strong></td>
</tr>
<tr>
<td><strong>Sbrst</strong></td>
<td>bare stem clause</td>
<td><strong>commanded that his sermons be published</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other Types</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AjP</strong></td>
<td>adjective phrase</td>
<td><strong>thought it possible</strong></td>
</tr>
<tr>
<td><strong>Quo</strong></td>
<td>quotes</td>
<td><strong>asked “What was it like?”</strong></td>
</tr>
</tbody>
</table>
• (Add an example full parse “he runs”)
  – After “another example” slide?
• Get from F15(?) Recitation notes??