Syntactic Attributes

Morphology, heads, gaps, etc.

Note: The properties of nonterminal symbols are often called “features.” However, we will use the alternative name “attributes.”

(We’ll use “features” to refer only to the features that get weights in a machine learning model, e.g., a log-linear model.)
3 views of a context-free rule

- generation (production): \[ S \rightarrow NP \ VP \]
- parsing (comprehension): \[ S \leftarrow NP \ VP \]
- verification (checking): \[ S = NP \ VP \]

Today you should keep the third, declarative perspective in mind.

- Each phrase has
  - an interface (\(S\)) saying where it can go
  - an implementation (\(NP \ VP\)) saying what’s in it

To let the parts of the tree coordinate more closely with one another, enrich the interfaces:
\[ S[attributes...] = NP[attributes...] \ VP[attributes...] \]
Examples

Verb $\rightarrow$ thrills

VP $\rightarrow$ Verb NP

S $\rightarrow$ NP VP

A roller coaster thrills every teenager

S
  └── NP
  └── VP
    └── Verb
    └── NP
3 Common Ways to Use Attributes

morphology of a single word:
Verb[head=thrill, tense=present, num=sing, person=3,...] → thrills

projection of attributes up to a bigger phrase
VP[head=α, tense=β, num=γ...] → V[head=α, tense=β, num=γ...]. NP
provided α is in the set TRANSITIVE-VERBS

agreement between sister phrases:
S[head=α, tense=β] → NP[num=γ,...] VP[head=α, tense=β, num=γ...]
3 Common Ways to Use Attributes

Verb[head=thrill, tense=present, num=sing, person=3,...] →  thrills
VP[head=α, tense=β, num=γ,...] → V[head=α, tense=β, num=γ,...] NP
S[head=α, tense=β] → NP[ num=γ,...] VP[head=α, tense=β, num=γ,...]

(A roller coaster thrills every teenager)
3 Common Ways to Use Attributes

Verb[head=thrill, tense=present, num=sing, person=3,…] → \textbf{thrills}

VP[head=\alpha, tense=\beta, num=\gamma,…] → V[head=\alpha, tense=\beta, num=\gamma,…] NP

S[head=\alpha, tense=\beta] → NP[num=\gamma,…] VP[head=\alpha, tense=\beta, num=\gamma,…]

(\textit{comprehension perspective})

A roller coaster \textbf{thrills} every teenager
The plan to swallow Wanda has been thrilling Otto.
The plan to swallow Wanda has been thrilling Otto.
The plan to swallow Wanda has been thrilling Otto.

NP[\(n=\alpha\)] → Det N[\(n=\alpha\)]
NP[\(n=\alpha\)] → N[\(n=\alpha\)] VP
N[\(n=\alpha\)] → N[\(n=\alpha\)] VP
N[\(n=1\)] → plan

VP[\(n=\alpha\)] → V[\(n=\alpha\)] VP
V[\(n=1\)] → has

S → NP[\(n=\alpha\)] VP[\(n=\alpha\)]
The plan to swallow has been thrilling.

\[
\text{NP}[h=\alpha] \rightarrow \text{Det} \ N[h=\alpha]
\]
\[
\text{N}[h=\alpha] \rightarrow \text{N}[h=\alpha] \ VP
\]
\[
\text{N}[h=\text{plan}] \rightarrow \text{plan}
\]
The plan to swallow Wanda has been thrilling Otto.
Why use heads?

- **Morphology** (e.g., word endings)
  - $N[h=\text{plan}, n=1] \rightarrow \text{plan}$
  - $N[h=\text{plan}, n=2+] \rightarrow \text{plans}$
  - $N[h=\text{thrill}, \text{tense}=\text{prog}] \rightarrow \text{thrilling}$
  - $N[h=\text{thrill}, \text{tense}=\text{past}] \rightarrow \text{thrilled}$
  - $N[h=\text{go}, \text{tense}=\text{past}] \rightarrow \text{went}$
- Why use heads?

- **Subcategorization** (i.e., transitive vs. intransitive)
  - When is $VP \rightarrow V \ NP$ ok?
    
    $VP[h=\alpha] \rightarrow V[h=\alpha] \ NP$
    
    restrict to $\alpha \in \text{TRANSITIVE\_VERBS}$
  - When is $N \rightarrow N \ VP$ ok?
    
    $N[h=\alpha] \rightarrow N[h=\alpha] \ VP$
    
    restrict to $\alpha \in \{\text{plan, plot, hope,}...\}$

- Why use heads?
- Why use heads?

- **Selectional restrictions**
  - VP[h=α] → V[h=α] NP
  - I.e., VP[h=α] → V[h=α] NP[h=β]
  - Don’t fill template in all ways:
    - VP[h=thrill] → V[h=thrill] NP[h=Otto]
    - *VP[h=thrill] → V[h=thrill] NP[h=plan] leave out, or low prob

- NP[h=α] → Det N[h=α]
- N[h=α] → N[h=α] VP
- N[h=plan] → plan

- The [head=plan]
- plan [head=plan]
- VP [head=swallow]
- to [head=plan] [head=swallow]
- been [head=thrill]
- thriling [head=thrill] [head=Otto]
- swallow [head=swallow] [head=Wanda]
**Log-Linear Models of Rule Probabilities**

- What is the probability of this rule?
  \[ S[\text{head}=\text{thrill}, \text{tense}=\text{pres}, ...] \rightarrow \]
  \[ \text{NP}[\text{head}=\text{plan}, \text{num}=1, \text{animate}=\text{no}...]\]
  \[ \text{VP}[\text{head}=\text{thrill}, \text{tense}=\text{pres}, \text{num}=1, ...]\]

- We have many related rules.

- \[ p(\text{NP}[, \text{VP}[,...] | S[,...]) \]
  \[ = (1/Z) \exp \sum_k \theta_k \cdot f_k(S[,...] \rightarrow \text{NP}[, ...] \text{VP}[,...]) \]

- We are choosing among all rules that expand \( S[,...]. \)

- If a rule has positively-weighted features, they raise its probability. Negatively-weighted features lower it.

- Which features fire will depend on the attributes!
Log-Linear Models of Rule Probabilities

S[head=thrill, tense=pres, ...] →
  NP[head=plan, num=1, animate=no, ...]
  VP[head=thrill, tense=pres, num=1, ...]

- Some features that might fire on this ...
  - The raw rule without attributes is S → NP VP.
    - Is that good? Does this feature have positive weight?
  - The NP and the VP agree in number. Is that good?
  - The head of the NP is “plan.” Is that good?
  - The verb “thrill” will get a subject.
  - The verb “thrill” will get an inanimate subject.
  - The verb “thrill” will get a subject headed by “plan.”
    - Is that good? Is “plan” a good subject for “thrill”?
Post-Processing

- You don’t have to handle everything with tons of attributes on the nonterminals
- Sometimes easier to compose your grammar with a post-processor:
  1. Use your CFG + randsent to generate some convenient internal version of the sentence.
  2. Run that sentence through a post-processor to clean it up for external presentation.
  3. The post-processor can even fix stuff up across constituent boundaries!

We’ll see a good family of postprocessors later: finite-state transducers.
We'll meet Smith, 59, the chief.
CAPS CAPS smith already met -ed me 's child -s.

Smith already met my children.
What Do These Enhancements Give You? And What Do They Cost?

- In a sense, nothing and nothing!
  - Can automatically convert our new fancy CFG to an old plain CFG.

- This is reassuring ...
  - We haven’t gone off into cloud-cuckoo land where “ooh, look what languages I can invent.”
    - Even fancy CFGs can’t describe crazy non-human languages such as the language consisting only of prime numbers.
    - Because we already know that plain CFGs can’t do that.
  - We can still use our old algorithms, randsent and parse.
    - Just convert to a plain CFG and run the algorithms on that.

- But we do get a benefit!
  - Attributes and post-processing allow simpler grammars.
  - Same log-linear features are shared across many rules.
  - A language learner thus has fewer things to learn.
**Analogy: What Does Dyna Give You?**

- In a sense, nothing and nothing!
  - We can automatically convert our fancy Dyna program to plain old machine code.

- This is reassuring ...
  - A standard computer can still run Dyna. No special hardware or magic wands are required.

- But we do get a benefit!
  - High-level programming languages allow shorter programs that are easier to write, understand, and modify.
What Do These Enhancements Give You? And What Do They Cost?

- In a sense, nothing and nothing!
  - We can automatically convert our new fancy CFG to an old plain CFG.

- Nonterminals with attributes → more nonterminals
  - $S[head=\alpha, \text{tense}=\beta] \rightarrow NP[num=\gamma,...] \ VP[head=\alpha, \text{tense}=\beta, \ num=\gamma,...]$
  - Can write out versions of this rule for all values of $\alpha, \beta, \gamma$
  - Now rename $NP[num=1,...]$ to $NP_{num}_1_...$
  - So we just get a plain CFG with a ton of rules and nonterminals

- Post-processing → more nonterminal attributes
  - Example: Post-processor changes “a” to “an” before a vowel
  - But we could handle this using a “starts with vowel” attribute instead
    - The determiner must “agree” with the vowel status of its Nbar
  - This kind of conversion can always be done! (automatically!)
  - At least for post-processors that are finite-state transducers
  - And then we can convert these attributes to nonterminals as above
**Part of the English Tense System**

<table>
<thead>
<tr>
<th>Tense Type</th>
<th>Present</th>
<th>Past</th>
<th>Future</th>
<th>Infinitive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simple</strong></td>
<td>eats</td>
<td>ate</td>
<td>will eat</td>
<td>to eat</td>
</tr>
<tr>
<td><strong>Perfect</strong></td>
<td>has eaten</td>
<td>had eaten</td>
<td>will have eaten</td>
<td>to have eaten</td>
</tr>
<tr>
<td><strong>progressive</strong></td>
<td>is eating</td>
<td>was eating</td>
<td>will be eating</td>
<td>to be eating</td>
</tr>
<tr>
<td><strong>Perfect+ progressive</strong></td>
<td>has been eating</td>
<td>had been eating</td>
<td>will have been eating</td>
<td>to have been eating</td>
</tr>
</tbody>
</table>
# Tenses by Post-Processing: “Affix-hopping” (Chomsky)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mary jumps</td>
<td>Mary [-s jump]</td>
</tr>
<tr>
<td>Mary has jumped</td>
<td>Mary [-s have] [-en jump]</td>
</tr>
<tr>
<td>Mary is jumping</td>
<td>Mary [-s be] [-ing jump]</td>
</tr>
<tr>
<td>Mary has been jumping</td>
<td>Mary [-s have] [-en be] [-ing jump]</td>
</tr>
</tbody>
</table>

where
- -s denotes “3rd person singular present tense” on following verb (by an -s suffix)
- -en denotes “past participle” (often uses -en or -ed suffix)
- -ing denotes “present participle” Etc.

Could we instead describe the patterns via attributes?
Let’s distinguish the different kinds of VP by tense ...
The plan ... thrilled Otto

Past
- Present tense
The plan …

Past
- Present tense

[V
[head=thrill]
[tense=pres,head=thrill]

NP
[head=plan]

S
[past pres]

VP
[past VP]

eat
[tense=pres,head=thrill]

Thrills
[tense=pres,head=thrill]

Otto
[head=Otto]

Past

Present tense
The plan ...

- Present tense
  (again)
The plan ... has thrilled Otto.

- Present perfect tense
- Present **perfect** tense
Present perfect tense
- The yellow material makes it a perfect tense – what effects?

- not ate – why?
The plan has thrilled Otto.
Present tense
(again)
The plan is thrilling for Otto.

- **Present progressive tense**
The plan was thrilling.
The plan ...

- Present perfect tense (again)
The plan ... has been thrilling Otto.

- Present perfect progressive tense
The plan has been thrilling.

- Present perfect progressive tense
The plan ... has been thrilling Otto.

- Past
  - Present perfect progressive tense
    - Past perfect tense
The plan ... has been thrilling.

Conditional:
- Present perfect progressive tense
- So what pattern do all progressives follow?
- So what pattern do all progressives follow?

[tense=prog, head=be] [tense=prog, head=be]
[tense=prog, head=eat] [tense=prog, head=eat]
[tense=prog, head=thrill] [tense=prog, head=thrill]
[tense=past, head=eat] [tense=past, head=be]
[tense=pres, head=be] [tense=pres, head=thrill]
Progressive: \( \text{VP}[\text{tense}=\alpha, \text{head}=\beta, \ldots] \rightarrow \text{V}[\text{tense}=\alpha, \text{head}=\text{be \ldots}] \)  
\( \quad \text{VP}[\text{tense}=\text{prog}, \text{head}=\beta \ldots] \)

Perfect: \( \text{VP}[\text{tense}=\alpha, \text{head}=\beta, \ldots] \rightarrow \text{V}[\text{tense}=\alpha, \text{head}=\text{have \ldots}] \)  
\( \quad \text{VP}[\text{tense}=\text{perf}, \text{head}=\beta \ldots] \)

Future or conditional: \( \text{VP}[\text{tense}=\alpha, \text{head}=\beta, \ldots] \rightarrow \text{V}[\text{tense}=\alpha, \text{head}=\text{will \ldots}] \)  
\( \quad \text{VP}[\text{tense}=\text{stem}, \text{head}=\beta \ldots] \)

Infinitive: \( \text{VP}[\text{tense}=\text{inf}, \text{head}=\beta, \ldots] \rightarrow \text{to} \)  
\( \quad \text{VP}[\text{tense}=\text{stem}, \text{head}=\beta \ldots] \)

As well as the “ordinary” rules:  
\( \text{VP}[\text{tense}=\alpha, \text{head}=\beta, \ldots] \rightarrow \text{V}[\text{tense}=\alpha, \text{head}=\beta, \ldots] \text{ NP} \)  
\( \quad \text{VP}[\text{tense}=\text{past}, \text{head}=\text{have \ldots}] \rightarrow \text{had} \)
Gaps ("deep" grammar!)

- Pretend "kiss" is a pure transitive verb.
- Is "the president kissed" grammatical?
  - If so, what type of phrase is it?
- the sandwich that
- I wonder what
- What else has

the president kissed e
Sally said the president kissed e
Sally consumed the pickle with e
Sally consumed e with the pickle
Gaps

- **Object gaps:**
  - the sandwich that
  - I wonder what
  - What else has

- **Subject gaps:**
  - the sandwich that
  - I wonder what
  - What else has

Sally said the president kissed e
Sally consumed the pickle with e
Sally consumed e with the pickle

[how could you tell the difference?]
Gaps

- All gaps are really the same - a missing NP:
  - the sandwich that
  - I wonder what
  - What else has

Phrases with missing NP:

\[ X[\text{missing} = \text{NP}] \]

or just \[ X/\text{NP} \] for short
what was he kissing?

what else could go here?

what else could go here?
what he was wondering was that was kissing the sandwich what else could go here?

what else could go here?

what else could go here?
what /NP he

what else could go here?

was VP/ NP

kissing NP/NP
e

what else could go here?
To indicate what fills a gap, people sometimes “coindex” the gap and its filler.

- Each phrase has a unique index such as “i”.
- In some theories, coindexation is used to help extract a meaning from the tree.
- In other theories, it is just an aid to help you follow the example.

the money$_i$ I spend $e_i$ on the happiness$_j$ I hope to buy $e_j$ which violin$_i$ is this sonata$_j$ easy to play $e_j$ on $e_i$
Lots of attributes (tense, number, person, gaps, vowels, commas, wh, etc...)

- Sorry, that’s just how language is ...
- You know too much to write it down easily!

He has gone