Auxiliaries cont: NICE
Overview

• Brief review of our analysis so far
• NICE properties of auxiliaries
• The auxiliary *do*
• NICE properties (lexical rules)
• Reading questions
## Descriptive Summary of the NICE Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Negation</strong></td>
<td>Sentences are negated by putting <em>not</em> after the first auxiliary verb; they can be reaffirmed by putting <em>too</em> or <em>so</em> in the same position.</td>
</tr>
<tr>
<td><strong>Inversion</strong></td>
<td>Questions are formed by putting an auxiliary verb before the subject NP.</td>
</tr>
<tr>
<td><strong>Contraction</strong></td>
<td>Auxiliary verbs take negated forms, with <em>n’t</em> affixed.</td>
</tr>
<tr>
<td><strong>Ellipsis</strong></td>
<td>Verb phrases immediately following an auxiliary verb can be omitted.</td>
</tr>
</tbody>
</table>
Negation (and Reaffirmation)

- Polar adverbs (sentential *not*, *so*, and *too*) appear immediately following an auxiliary
  
  *Pat will not leave*
  *Pat will SO leave*
  *Pat will TOO leave*

- What about examples like *Not many people left*?

- What happens when you want to deny or reaffirm a sentence with no auxiliary?
  
  *Pat left*
  *Pat did not leave*
  *Pat did TOO leave*
The Auxiliary *do*

- Like modals, auxiliary *do* only occurs in finite contexts:
  \[*Pat continued to do not leave*

- Unlike modals, *do* cannot be followed by other auxiliaries:
  \[*Pat did not have left*
The ADV\textsubscript{pol}-Addition Lexical Rule

\[
\begin{align*}
\text{INPUT} & \quad \langle X, & \\
\quad \text{ARG-ST} & \quad \langle \text{INDEX} s_{1} \rangle \oplus \mathbb{A} & \\
\quad \text{SEM} & \quad \langle \text{INDEX} s_{1} \rangle & \\
\end{align*}
\]

\[
\begin{align*}
\quad \text{SYN} & \quad \langle \text{HEAD} \left[ \begin{array}{c} \text{FORM} \text{ fin} \\ \text{POL} - \\ \text{AUX} + \end{array} \right] \rangle & \\
\quad \text{val} & \quad \langle \text{POL} + \rangle & \\
\quad \text{val} & \quad \langle \text{SPR} \langle Z \rangle \rangle & \\
\text{OUTPUT} & \quad \langle Y, & \\
\quad \text{ARG-ST} & \quad \langle \text{INDEX} s_{2} \rangle \oplus \mathbb{A} & \\
\quad \text{SEM} & \quad \langle \text{INDEX} s_{2} \rangle & \\
\end{align*}
\]

\[
\begin{align*}
\quad \text{adv}_{\text{pol}} & \quad \langle \text{INDEX} s_{2} \rangle \oplus \mathbb{A} & \\
\quad \text{rest} & \quad \langle \text{ARG} s_{1} \rangle & \\
\end{align*}
\]
What does the type *pi-rule* mean?

- It maps words to words (hence, “post-inflectional”)
- It preserves MOD values, HEAD values as a default, and (like other lexical rule types) SEM values as a default
Why doesn’t $\text{ADV}_{pol}$-Addition LR mention VAL?

\[
\begin{align*}
\text{INPUT} & \quad \langle X, \quad \langle \text{SYN} \langle X \rangle, \langle \text{ARG-ST} \langle 1 \rangle \rangle, \langle \text{SEM} \langle \text{INDEX } s_1 \rangle \rangle \rangle \\
\text{OUTPUT} & \quad \langle Y, \quad \langle \text{SYN} \langle \text{HEAD} \langle \text{POL } + \rangle \rangle \rangle,\langle \text{ARG-ST} \langle 1 \rangle \rangle, \langle \text{SEM} \langle \text{INDEX } s_2 \rangle \rangle \rangle \rangle
\end{align*}
\]
What is the role of these indices?

\[\text{pi-rule}\]

INPUT \(\langle X, \text{syn}, \text{head}, \text{pol}, \text{aux}, \text{verb} \rangle\)

ARG-ST \(\langle 1 \rangle \oplus [A]\)

SEM \([\text{INDEX} \ s_1]\)

OUTPUT \(\langle Y, \text{syn}, \text{head}, \text{pol}, \text{val}, \text{spring} \rangle\)

ARG-ST \(\langle 1 \rangle \oplus [\text{INDEX} \ s_2, \text{ADV}_{pol} \ s_2, \text{restr}, \langle \text{ARG} \ s_1 \rangle] \oplus [A]\)

SEM \([\text{INDEX} \ s_2]\)
Which *nots* does the rule license?

\[ \text{pi-rule} \]

INPUT \( \langle X , \) \( \text{ARG-ST} \) \( \langle 1 \rangle \oplus A \) \( \text{SEM} \) \( \text{INDEX} \ s_1 \) \( \text{SYN} \) \( \text{HEAD} [\text{verb} \] \( \text{FORM} \ fin \) \( \text{POL} - \) \( \text{AUX} + \) \( \text{VAL} \) \( \text{SPR} \langle Z \rangle \) \( \text{OUTPUT} \) \( \langle Y , \) \( \text{ARG-ST} \) \( \langle 1 \rangle \oplus \) \( \text{ADV}_{pol} \) \( \text{INDEX} \ s_2 \) \( \text{RESTR} \) \( \text{INDEX} \ s_1 \) \( \text{SEM} \) \( \text{INDEX} \ s_2 \) \( \rangle \) \( \oplus A \) \( \rangle \)

*Andy must *not* have been sleeping? \( \checkmark \)

*Andy must have *not* been sleeping? \( \times \)

*Andy must have been *not* sleeping? \( \times \)

*Kleptomaniacs cannot* *not* steal. \( \checkmark \)

*Kleptomaniacs cannot* *not* steal. \( \times \)
Negation and Reaffirmation: A Sample Tree

NP

| Leslie

S

| VP

| V

| did

| ADV_{pol}

| so

VP

| eat the whole pizza
Inversion

• Yes-no questions begin with an auxiliary:
  Will Robin win?

• The NP after the auxiliary has all the properties of a subject
  • Agreement: Have they left? vs. *Has they left?
  • Case: *Have them left?
  • Raising: Will there continue to be food at the meetings?

• What happens if you make a question out of a sentence without an auxiliary?
  Robin won
  Did Robin win?
The Inversion Lexical Rule

\[
\begin{aligned}
&\text{INPUT} \quad \langle W , \rangle \\
&\quad \langle \text{SYN} \rangle \\
&\quad \langle \text{ARG-ST} \rangle \\
&\quad \langle \text{SEM} \rangle \\
&\text{OUTPUT} \quad \langle Z , \rangle \\
&\quad \langle \text{SYN} \rangle \\
&\quad \langle \text{ARG-ST} \rangle \\
&\quad \langle \text{SEM} \rangle \\
\end{aligned}
\]
How the Rule Yields Inverted Order

\[
\begin{align*}
\text{INPUT} & \quad \langle W, \quad \text{arg}\_\text{st} \quad A \quad \text{SEM} \quad [\text{mode} \quad \text{prop}] \rangle \\
\text{OUTPUT} & \quad \langle Z, \quad \text{arg}\_\text{st} \quad A \quad \text{SEM} \quad [\text{mode} \quad \text{ques}] \rangle
\end{align*}
\]

...plus the ARP
The Feature INV

• What is the INV value of inputs to the Inversion LR?
  • Perhaps surprisingly, the input is \([\text{INV} +]\)
  • Word-to-word rules (\textit{pi-rules}) have default identity of HEAD features, and no INV value is given on the input

• Then what work is the feature doing?
  • It’s used to mark auxiliaries that can’t or must be inverted

\textit{You better watch out} \hspace{1cm} \textit{vs.} \hspace{1cm} \textbf{*Better you watch out}
\textit{I shall go} (\textit{shall} \sim \textit{‘will’}) \hspace{1cm} \textit{vs.} \hspace{1cm} \textit{Shall I go?} (\textit{shall} \sim \textit{‘should’})
Other Cases of Inversion

• Inversion is not limited to questions
  • Preposed negatives: *Never have I been so upset!*
  • Conditionals: *Had we known, we would have left.*
  • Exclamations: *May your teeth fall out!*

• Does our rule account for these?
  • No. Our rule’s output says [MODE ques]. And each construction has slightly different idiosyncrasies.

• How might we extend our analysis to cover them?
  • Define a type of inversion lexical rules, sharing certain properties, but with some differences.
Inversion: A Sample Tree

S

V

Did

NP

Leslie

VP

eat the entire pizza?
Contraction

• There are several types of contraction in English, but we’re only talking about words ending in *n’t*

• It may seem like just *not* said fast, but there’s more to it

  • Only finite verbs can take *n’t*:
    
    *Terry must haven’t seen us*

• There are morphological irregularities:
  
  won’t, not *willn’t*  %shan’t, not *shalln’t*

  mustn’t pronounced mussn’t
don’t pronounced doesn’t, not dewn’t

  *amn’t*
The Contraction Lexical Rule

\[
\begin{align*}
\text{pi-rule} & \quad \begin{bmatrix}
\text{SYN} & \text{HEAD} & \begin{bmatrix}
\text{verb} \\
\text{FORM} & \text{fin} \\
\text{AUX} & + \\
\text{POL} & -
\end{bmatrix}
\end{bmatrix} \\
\text{INPUT} & \quad \left\langle \begin{bmatrix}
2 \\
\text{ARG-ST} & \text{B} \\
\text{SEM} & \begin{bmatrix}
\text{INDEX} & s_1 \\
\text{RESTR} & \text{A}
\end{bmatrix}
\end{bmatrix} \right\rangle \\
\text{OUTPUT} & \quad \left\langle \begin{bmatrix}
\text{F}_{\text{NEG}}(2) \\
\text{ARG-ST} & \text{B} \\
\text{SEM} & \begin{bmatrix}
\text{INDEX} & s_2 \\
\text{RESTR} & \begin{bmatrix}
\text{RELN} & \text{not} \\
\text{SIT} & s_2 \\
\text{ARG} & s_1
\end{bmatrix}
\end{bmatrix} + \text{A}
\end{bmatrix} \right\rangle
\end{align*}
\]
Most of the work is in the semantics

Why?
What does POL do?

\[ \pi\text{-rule} \]

**INPUT** \[ [2], \]
- SYN
- HEAD
- FORM fin
- AUX +
- POL -
- ARG-ST [B]
- INDEX \[ s_1 \]
- RESTR [A]

**OUTPUT** \[ F_{NEG}(2), \]
- SYN
- HEAD
- POL +
- VAL
- SPR \[ \langle X \rangle \]
- ARG-ST [B]
- INDEX \[ s_2 \]
- RESTR
  - RELN
  - SIT
  - ARG
  - not \[ s_2 \]
  - s_1 \]

\[ \oplus [A] \]

*We can’tn’t stop*

*They won’t TOO mind*
Contraction: Sample Tree

S
  /\  
 NP  VP
  |   /
 Leslie  V
       /\  /
  wouldn’t eat the entire pizza
Ellipsis

• Ellipsis allows VPs to be omitted, so long as they would have been preceded by an auxiliary.

* Pat couldn’t have been watching us, but Chris

• Unlike the other NICE properties, this holds of all auxiliaries, not just finite ones.

• What is the elliptical counterpart to a sentence with no auxiliary?

Whenever Pat watches TV, Chris watches TV
Whenever Pat watches TV, Chris does
The Ellipsis Lexical Rule

\[
\begin{align*}
\text{INPUT} &\quad \langle 1, \begin{bmatrix} \text{auxv-lxm} \\ \text{ARG-ST} \langle 2 \rangle \oplus \text{A} \end{bmatrix} \rangle \\
\text{OUTPUT} &\quad \langle 1, \begin{bmatrix} \text{dervv-lxm} \\ \text{ARG-ST} \langle 2 \rangle \end{bmatrix} \rangle 
\end{align*}
\]

• Note that this is a derivational LR (\textit{d-rule}) -- that is, lexeme-to-lexeme

• This means that SYN and SEM are unchanged, by default
Ellipsis: A Sample Output

\[ \text{could,} \]

\[ \langle \text{NP} \rangle \]

\[ \langle \text{SEM} \rangle \]

\[ \langle \text{RESTR} \rangle \]

\[ \langle \text{INDEX} \rangle \]

\[ \langle \text{MODE} \rangle \]

\[ \langle \text{VAL} \rangle \]

\[ \langle \text{HEAD} \rangle \]

\[ \langle \text{auxv-lxm} \rangle \]
Ellipsis: A Sample Tree

S

NP

Kim

V

could

VP

V

could

VP

have

VP

been

attending the conference
Semantics of Ellipsis

What is the SEM value of the S node of this tree?

Note: $s_2$ has to be filled in by context.
Infinitival *to* Revisited

• VP Ellipsis can occur after *to*:

*We didn’t find the solution, but we tried to.*

• This is covered by our Ellipsis LR if we say *to* is [AUX +].

• Since AUX is declared on type *verb*, it follows that *to* is a verb.
**do Revisited**

- Chomsky’s old analysis: in sentences w/o auxiliaries...
  - Tense can get separated from the verb in various ways
    - Negation/Reaffirmation inserts something between Tense and the following verb
    - Inversion moves Tense to the left of the subject NP
    - Ellipsis deletes what follows Tense
  - When this happens, *do* is inserted to support Tense

- Our counterpart:
  - NICE properties hold only of auxiliaries
  - *do* is a semantically empty auxiliary, so negated, reaffirmed, inverted, and elliptical sentences that are the semantic counterparts to sentences w/o auxiliaries are ones with *do*. 
Summary

- Our analysis employs straightforward mechanisms
  - Lexical entries for auxiliaries
  - 3 new features (AUX, POL, INV)
  - 4 lexical rules

- We handle a complex array of facts
  - co-occurrence restrictions (ordering & iteration)
  - the NICE properties
  - auxiliary *do*
  - combinations of NICE constructions
Overview

• Brief review of our analysis so far
• NICE properties of auxiliaries
• The auxiliary *do*
• NICE properties (lexical rules)
• Reading questions
Reading Questions

• Is the sentence *Sandy so did not write that.* grammatical? Is the ADV(pol) *not* constituent or sentential negation?

• I definitely use and have heard the construction in (a) with "too," but I'm not sure about the one in (b) with "so." Is the construction in (b) supposed to mean the same thing as that in (a)?

(a) Pat will too leave.
(b) Pat will so leave.
Reading Questions

• I often hear sentences with 'so not' where the 'so' seems to emphasize the 'not' aspect of the sentence.

   Kim is so not happy.
   Sandy is so not mad at you.

• Are these examples of constituent negation? Or are 'so' and 'not' both ADV_pol?

• Is the language changing with respect to "so", and if so, can our grammar keep up?
I can't seem to differentiate between a constituent negation and a negation placed only in the verb. How do you differentiate between them?

Can a sentence like *He can not smoke*. be interpreted as exhibiting either kind of negation, with the following senses:

(i) He is able to not smoke.  [constituent negation]
(ii) He is not able to smoke.  [sentential negation]
Reading Questions

• How will the grammar handle cases like *He'll go downtown tomorrow*? Will it be similar to contraction?

• Wouldn't contractions like "it's" for "it is" or "you're" for "you are" not work with our Contraction Lexical Rule since those contractions aren't negations?
Reading Questions

• I feel like it would be that truer to the morphological history of the word to have some sort of phrasal rule to morph a phrase into a word. Wouldn’t it?

• It strikes me as a little odd to have two separate lexical rules that add negation to auxiliary verbs - both the ADVpol-Addition Lexical Rule and the Contraction Lexical Rule. Intuitively, I feel like I want this to only be a pronunciation difference, outside of the realm of what we're tackling. Does this mean that we also want our grammar to have rules that account for phonological shortenings like "gimme", "wanna", and "dunno"?
Is "better" really an auxiliary verb? I can see it in the sentences, but I've always thought of it as a contraction of some sort. Maybe of "would be better off..."?
Reading Questions

• Why do we have SPR <Z> instead of nothing or SPR <[1]> in the output of the ADVpol Addition LR?

• Is it possible to have an INPUT to these rules with 0 elements on the SPR list, and then... uninvert them?

• Why don't we need to say SPR <X> on the input?
The \( \text{ADV}_{pol} \)-Addition Lexical Rule

\[
\begin{align*}
\text{INPUT} & \quad \left\langle X, \begin{array}{l}
\text{SYN} \\
\text{ARG-ST} \\
\text{SEM}
\end{array} \right\rangle, \\
\text{OUTPUT} & \quad \left\langle Y, \begin{array}{l}
\text{SYN} \\
\text{ARG-ST} \\
\text{SEM}
\end{array} \right\rangle.
\end{align*}
\]
Reading Questions

• The Ellipsis Lexical Rule and the ADVpol-Addition Lexical Rule both involve 'factoring' the ARG-ST. What guarantees that only the first item on the list is taken out? Can the tag ONLY contain one item (i.e. the specifier, or one complement)? Is this also the reason for the first tag being a number (because it is inside of the a list), and the second being a letter (which seems to be used to convey a list, rather than an item on a list)?
• Why there is X and Y difference in the definition of ADVPOL-Addition Lexical Rule (51)? Aren't they supposed to be the same? (It is same index [0] in (52).)
The Inversion Lexical Rule only accounts for yes-no questions, and it seems that we wouldn't want to get rid of the specifier for all cases of inversion. Wouldn't we want to posit a pi-rule that licenses the final argument of the verb, or predp-lxm even, from the INPUT as a wh-pronoun in the SPR position of the OUTPUT?

What are you eating?
Who will you be going to the movies with?
Reading Questions

• Are there similarities of the Inversion Lexical Rule to producing an active or passive sentence from the other? Do we have something like an "Active to Passive Lexical Rule" or vice versa?

• Why did we limit the Inversion Lexical Rule such that a MODE ques verb can not be the input to produce the inverted sentence with MODE prop? i.e. "Is Kim Happy?" -> "Kim is Happy" Which would then provide a Rule that can be reversed on it's own.
Reading Questions

• We need dervv-lxm because the ARG-ST has only one element as compared to the your-SPR-is-my-SPR in the ARG-ST for auxv-lxm. For the Imperative Lexical Rule, we say that there is SPR <NP[PER 2nd]> but it is never realized. Why can't we do that with the dervv-lxm?

• Is limiting the input the only legit way of controlling the order of pi-rules that apply to a word?
Reading Questions

• I'm wondering why it is that the ADVpol-Addition Lexical Rule is formulated as a pi-rule instead of a d-rule (certainly not an i-rule, since the ARG-ST list must be changed). Is one possible reason that we don't want outputs of the ADVpol rule to be able to feed into certain inflectional or derivational rules, and therefore want it to have an output of type word?
Reading Questions

• In footnote 18 in section 13.5.4 it is mentioned that the absent constituent in the elliptical condition is interpreted as a VP but in the AUX alternative what is missing is the part of the AUX plus the following VP. Could you explain what is happening here further?

• This section on elliptical constructions leaves a lot of questions unanswered regarding the semantic relations of elliptical constructions if there's time could you explain a little more about how this is handled in HPSG.
Reading Questions

• Why is POL a feature of verb-lxm rather than auxv-lxm? Does it ever apply to a non-auxiliary verb?

• Where is the aux verb in:

(74)a. We asked them to open the window, and they tried to.

• Is our process for negation compatible with ellipsis, or is it not?
Reading Questions

- The Inversion Lexical Rule makes me wonder about the distinction between specifier and complement. The book defines specifiers as 'subjects of clauses, determiners of noun phrases, and certain other constituents that are neither heads of the phrases they appear in nor complements to the heads' (p. 569), and specifically states that 'subjects... are arguments that are not complements, but specifiers' (p. 557). Cross-linguistically this seems to be how we're dealing with them (that is, whether we call something a 'specifier' is based on its role in the constituent rather than its location). However, the ILR handily dumps the specifier into the COMPS phrase in order to get the proper word order. Does the role of this word change? If not, why not just call the categories 'pre-head args' and 'post-head args' and not have the distinction above at all? If so, how and why?
Reading Questions

• Would it be possible to instead introduce a second Head-Specifier Rule that acts on [INV +] words or phrases? For example:

• \([\text{phrase } \text{SPR }<>] -> \text{H}\{\text{INV }+, \text{SPR }<[1]>\}\) [1]
Reading Questions

- Does the book analyze this question?
- What about ain't? Is it the output of one of these rules?
- How do we handle:

  *Kleptomaniacs can not NOT steal.*
Reading Questions

• Don't we need posit "your specifier is my first-complement"-kind of rule somewhere in Inversion? (Or is it already taken care of with auxv-lxm + ARP?)

• In (60) (word structure for *will* as output of Inversion LR) where does [CASE nom] on the first ARG-ST element come from?
• Lastly, can you give some comments on implementing NICE properties in terms of CFG in the class? It seems like implementing NICE properties with CFG (using AUX) will soon make us to introduce new rules and thing get hairy pretty quickly, I believe.