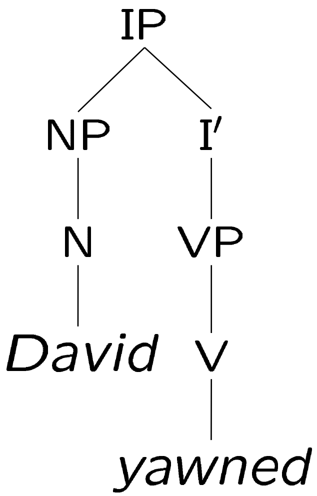
Review: LFG and glue

Linguistics 233B

22 January 2002

David yawned.



$$\begin{bmatrix} \text{PRED} & \text{‘yawn’} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘David’} \\ \text{NUM} & \text{sg} \end{bmatrix} \end{bmatrix}$$

Provide lexical entries and annotated phrase structure rules.

$$\mathrm{IP} \longrightarrow \begin{array}{cc} \mathsf{NP} & \mathrm{I}' \\ (\uparrow \ \mathsf{SUBJ}) = \downarrow & \uparrow = \downarrow \end{array}$$

$$\mathrm{I}' \longrightarrow \begin{array}{c} \mathsf{VP} \\ \uparrow = \downarrow \end{array}$$

$$\mathsf{VP} \longrightarrow \begin{array}{c} \mathsf{V} \\ \uparrow = \downarrow \end{array}$$

$$\mathsf{NP} \longrightarrow \begin{array}{c} \mathsf{N} \\ \uparrow = \downarrow \end{array}$$

$$\begin{array}{lll} \textit{David} & \mathsf{N} & (\uparrow \ \mathsf{PRED}) = \text{‘David’} \\ & & (\uparrow \ \mathsf{NUM}) = \mathsf{sg} \end{array}$$

$$\begin{array}{lll} \textit{yawned} & \mathsf{V} & (\uparrow \ \mathsf{PRED}) = \text{‘yawn}\langle\mathsf{SUBJ}\rangle\text{’} \end{array}$$

Provide the f-description for “David yawned”.

Using mnemonic names for f-structures corresponding to c-structure nodes:

$$ip, i', vp, v : \begin{bmatrix} \textsf{PRED} & \text{`yawn'} \\ \textsf{SUBJ} & np, n : \begin{bmatrix} \textsf{PRED} & \text{`David'} \\ \textsf{NUM} & \textsf{sg} \end{bmatrix} \end{bmatrix}$$

$$\begin{gathered} (ip\ \textsf{SUBJ}) = np \\ ip = i' \\ i' = vp \\ vp = v \\ np = n \\ (n\ \textsf{PRED}) = \text{`David'} \\ (n\ \textsf{NUM}) = \textsf{sg} \\ (v\ \textsf{PRED}) = \text{`yawn}\langle\textsf{SUBJ}\rangle\text{'} \end{gathered}$$

David yawned.

$$ip : \left[\begin{array}{ll} \textsf{PRED} & \textsf{`yawn'} \\ & \\ \textsf{SUBJ} & np : \left[\begin{array}{ll} \textsf{PRED} & \textsf{`David'} \\ \textsf{NUM} & \textsf{sg} \end{array}\right] \end{array}\right]$$

$$\textsf{\textit{yawn}}(\textsf{\textit{David}}) : ip_\sigma$$

Provide augmented lexical entries for “David” and “yawned” incorporating meaning constructors.

David	N	$(\uparrow\ \text{PRED}) = \text{‘David’}$ $(\uparrow\ \text{NUM}) = \text{sg}$ $\textit{David} : \uparrow_\sigma$
yawned	V	$(\uparrow\ \text{PRED}) = \text{‘yawn}\langle\text{SUBJ}\rangle\text{’}$ $\lambda X.\textit{yawn}(X) : (\uparrow\ \text{SUBJ})_\sigma \multimap \uparrow_\sigma$

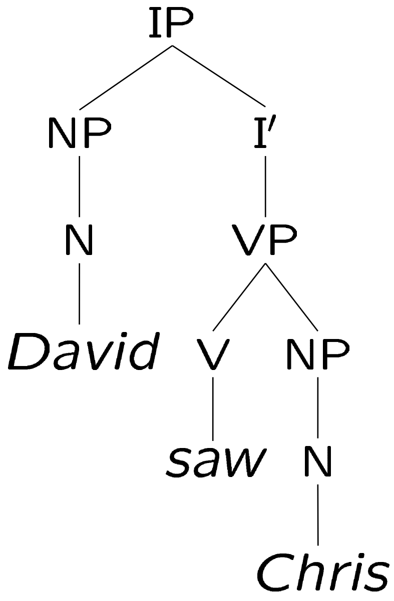
Provide instantiated semantic premises for “David yawned” based on these lexical entries.

$$ip : \left[\begin{array}{ll} \text{PRED} & \text{`yawn'} \\ \text{SUBJ} & np : \left[\begin{array}{ll} \text{PRED} & \text{`David'} \\ \text{NUM} & \text{sg} \end{array}\right] \end{array}\right]$$

$$\textit{David} : np_{\sigma}$$

$$\lambda X.\textit{yawn}(X) : np_{\sigma} \multimap ip_{\sigma}$$

David saw Chris.



$$\begin{bmatrix} \text{PRED} & \text{‘see}\langle\text{SUBJ,OBJ}\rangle\text{’} \\ \text{SUBJ} & \begin{bmatrix}\text{PRED} & \text{‘David’}\end{bmatrix} \\ \text{OBJ} & \begin{bmatrix}\text{PRED} & \text{‘Chris’}\end{bmatrix} \end{bmatrix}$$

Provide changes and additions to lexical entries and phrase structure rules.

$$\begin{array}{cccc} \text{VP} & \longrightarrow & \text{V} & \text{NP} \\ & & \uparrow = \downarrow & (\uparrow \text{ OBJ}) = \downarrow \end{array}$$

Chris	N	$(\uparrow \text{ PRED}) = \text{`Chris'}$ $(\uparrow \text{ NUM}) = \text{sg}$ $\textit{Chris} : \uparrow_{\sigma}$
saw	V	$(\uparrow \text{ PRED}) = \text{`see}\langle\text{SUBJ,OBJ}\rangle\text{'}$ $\lambda X.\lambda Y.\textit{see}(X, Y) : (\uparrow \text{ SUBJ})_{\sigma} \multimap ((\uparrow \text{ OBJ})_{\sigma} \multimap \uparrow_{\sigma})$

Provide instantiated premises for “David saw Chris”.

$$s : \left[\begin{array}{ll} \text{PRED} & \text{‘see}\langle\text{SUBJ,OBJ}\rangle\text{’} \\ \text{SUBJ} & d : \left[\begin{array}{ll} \text{PRED} & \text{‘David’} \end{array}\right] \\ \text{OBJ} & c : \left[\begin{array}{ll} \text{PRED} & \text{‘Chris’} \end{array}\right] \end{array}\right]$$

$$\mathit{David} : d_\sigma$$

$$\mathit{Chris} : c_\sigma$$

$$\lambda X.\lambda Y.\mathit{see}(X, Y) : d_\sigma \multimap (c_\sigma \multimap s_\sigma)$$

Glue and XLE: alternative notation for premises

Standard style:

$$\textit{David} : d_\sigma$$
$$\lambda X.\textit{yawn}(X) : d_\sigma \multimap y_\sigma$$

XLE style:

$$d_\sigma \leadsto \textit{David}$$
$$\forall X.d_\sigma \leadsto X \multimap y_\sigma \leadsto \textit{yawn}(X)$$

David saw Chris.

$$\begin{bmatrix} \text{PRED} & \text{`see}\langle\text{SUBJ,OBJ}\rangle\text{'} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{`David'} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{`Chris'} \end{bmatrix} \end{bmatrix}$$

$$s_\sigma \rightsquigarrow \mathit{see}(\mathit{David}, \mathit{Chris})$$

Provide lexical entries for “David”, “saw”, “Chris” in alternative notation.

David	N	$(\uparrow \text{ PRED}) = \text{‘David’}$ $\uparrow_{\sigma} \rightsquigarrow \textit{David}$
Chris	N	$(\uparrow \text{ PRED}) = \text{‘Chris’}$ $\uparrow_{\sigma} \rightsquigarrow \textit{Chris}$
saw	V	$(\uparrow \text{ PRED}) = \text{‘see}\langle\text{SUBJ,OBJ}\rangle\text{’}$ $\forall X, Y.(\uparrow \text{ SUBJ})_{\sigma} \rightsquigarrow X \multimap ((\uparrow \text{ OBJ})_{\sigma} \rightsquigarrow Y \multimap \uparrow_{\sigma} \rightsquigarrow \textit{see}(X,Y))$

Provide instantiated premises for “David saw Chris”.

$$\begin{bmatrix} \text{PRED} & \text{‘see}\langle\text{SUBJ,OBJ}\rangle\text{’} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘David’} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{‘Chris’} \end{bmatrix} \end{bmatrix}$$

$$d_\sigma \rightsquigarrow \textit{David}$$

$$c_\sigma \rightsquigarrow \textit{Chris}$$

$$\forall X, Y. d_\sigma \rightsquigarrow X \multimap (c_\sigma \rightsquigarrow Y \multimap s_\sigma \rightsquigarrow \textit{see}(X, Y))$$

Glue and XLE: separate lexical resources

- Glue theory: Lexical entries contain both syntactic and semantic information.

  David N $(\uparrow\ \text{PRED}) = \text{`David'}$
  $\uparrow_\sigma \rightsquigarrow \textit{David}$

- Glue implementation: Syntactic and semantic information is stored in separate files.

  – Syntactic lexicon:

    David N $(\uparrow\ \text{PRED}) = \text{`David'}$

  – Semantic lexicon:

    David $\uparrow_\sigma \rightsquigarrow \textit{David}$

XLE notation: Syntax

```
VP --> V: ^ = !; NP: (^ OBJ) = !.
```

- Need semicolon after annotated category.

- Need period at end of rule.

- "^=!" can be omitted if it is the only annotation on a node.

```
Chris  N  *  (^ PRED) = 'Chris'
             (^ NUM) = sg.
```

XLE documentation available by typing "documentation" to XLE.

XLE notation: Semantics

```
lex_sem(’David’, david, ’N’, name).
sem_template(name, _Name, LogicalConstant, true,

    sigma(^) -~> LogicalConstant

            ).


lex_sem(yawned,   yawn,    ’V’,   v_intrans).
sem_template(v_intrans, _Verb, Pred, true,

    &(X, sigma(p:[^,’SUBJ’]) -~> X  -*   sigma(^) +~> [Pred, X])

            ).
```

Homework: C-structure, f-structure, annotated phrase structure rules, lexical entries for

David thought that Fred saw Chris:

think(*David*, *see*(*Fred*, *Chris*))

David introduced Fred to Chris:

introduce(*David*, *Fred*, *Chris*)

(assume that “to” makes no semantic contribution)