Towards Wide-Coverage Semantics

Richard Moot
(CNRS, LIRMM)

Question:

How can we parse arbitrary text?

Question:

How can we parse arbitrary text *and assign it a representation of its meaning*?

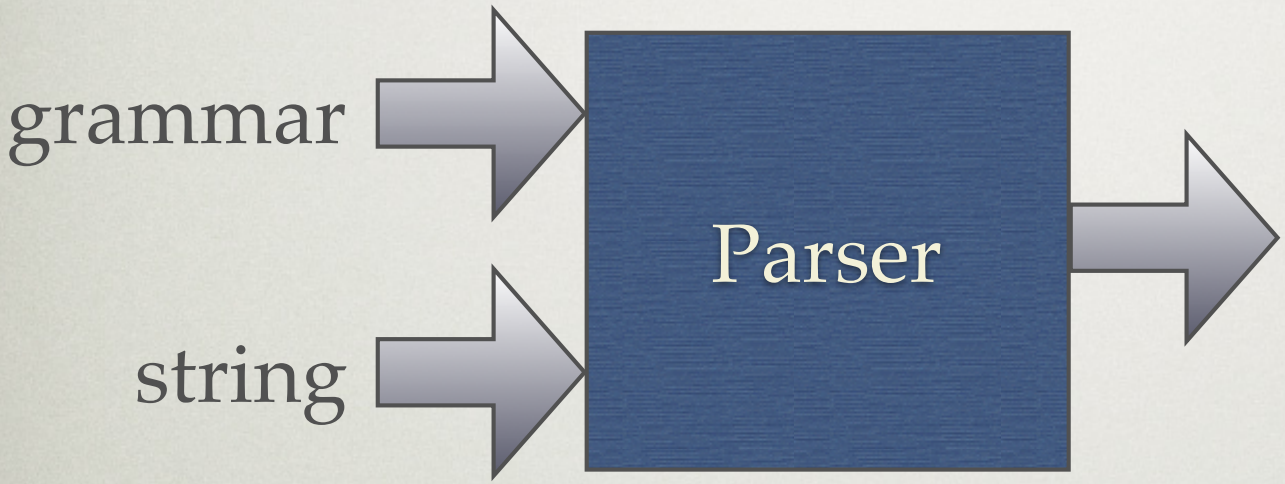
Two Solutions

Writing grammars by hand is very useful and very important, but not well-suited to wide-coverage parsing

1. Program a grammar by hand
2. Machine Learning

Treebanks contain a lot of information we can exploit. So we use treebank transformations to make them as close as possible to what we really want.

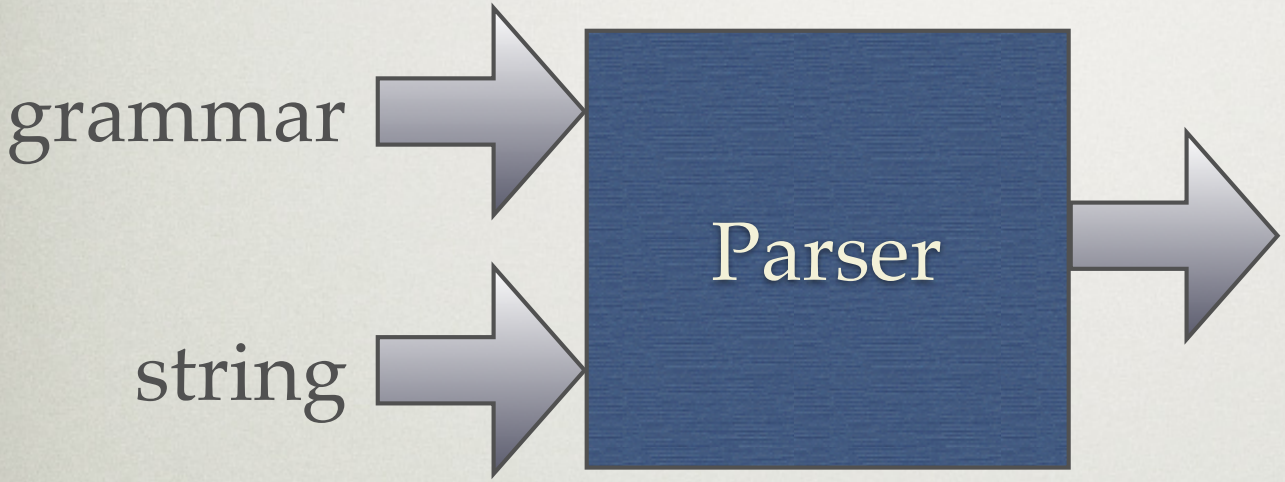
Parsing and Semantics



parse tree

$$\dfrac{\dfrac{\dfrac{\text{de}}{(np\backslash s_{di})/(np\backslash s_i)}\,[Lex] \quad \dfrac{\dfrac{\text{s'}}{cl_r}\,[Lex] \quad \dfrac{\dfrac{\text{attaquer}}{(cl_r\backslash(np\backslash s_i))/pp_{\grave{a}}}\,[Lex] \quad \dfrac{}{\text{p}_0 \vdash pp_{\grave{a}}}\,[Hyp]_1}{\text{a} \circ \text{p}_0 \vdash cl_r\backslash(np\backslash s_i)}\,[/E]}{\text{s'} \circ (\text{a} \circ \text{p}_0) \vdash np\backslash s_i}\,[\backslash E]}{\text{de} \circ (\text{s'} \circ (\text{a} \circ \text{p}_0)) \vdash np\backslash s_i}\,[/E]$$

Parsing and Semantics



parse tree

$$\dfrac{\dfrac{\text{de}}{(np\backslash s_{di})/(np\backslash s_i)}\,[Lex] \quad \dfrac{\dfrac{\text{s'}}{cl_r}\,[Lex] \quad \dfrac{\dfrac{\text{attaquer}}{(cl_r\backslash(np\backslash s_i))/pp_{\grave{a}}}\,[Lex] \quad \dfrac{}{\text{p}_0 \vdash pp_{\grave{a}}}\,[Hyp]_1}{\text{a} \circ \text{p}_0 \vdash cl_r\backslash(np\backslash s_i)}\,[/E]}{\text{s'} \circ (\text{a} \circ \text{p}_0) \vdash np\backslash s_i}\,[\backslash E]}{\text{de} \circ (\text{s'} \circ (\text{a} \circ \text{p}_0)) \vdash np\backslash s_i}\,[/E]$$

semantics

$$(\lambda x.\exists y.(x\ y)\ \lambda z.((\lambda v.\lambda w.\textbf{\textit{shot}}(v,w)\ z)\ \textbf{\textit{vito}}))$$

$$\begin{array}{|l}\hline e_1\ y_1 \\ \hline y_1 : \begin{array}{|l}\hline e_2\ e_3\ x_3 \\ \hline x_3 = ? \\ \textbf{aider\_à}(e_2, x_0, x_3, e_3) \\ \textbf{partir}(e_3, x_3) \\ \hline \end{array} \\ \textbf{demander}(e_1, y_0, x_0, y_1) \\ \hline \end{array}$$

Many applications benefit from Meaning

- Textual entailment
- Question answering
- Anaphora resolution

Entailment example (from Johnson-Laird)

T None of the artists is a beekeeper.
All the beekeepers are chemists.
(There is at least one beekeeper.)

H … artists … chemists.

Entailment example (from Johnson-Laird)

T None of the artists is a beekeeper.
All the beekeepers are chemists.
(There is at least one beekeeper.)

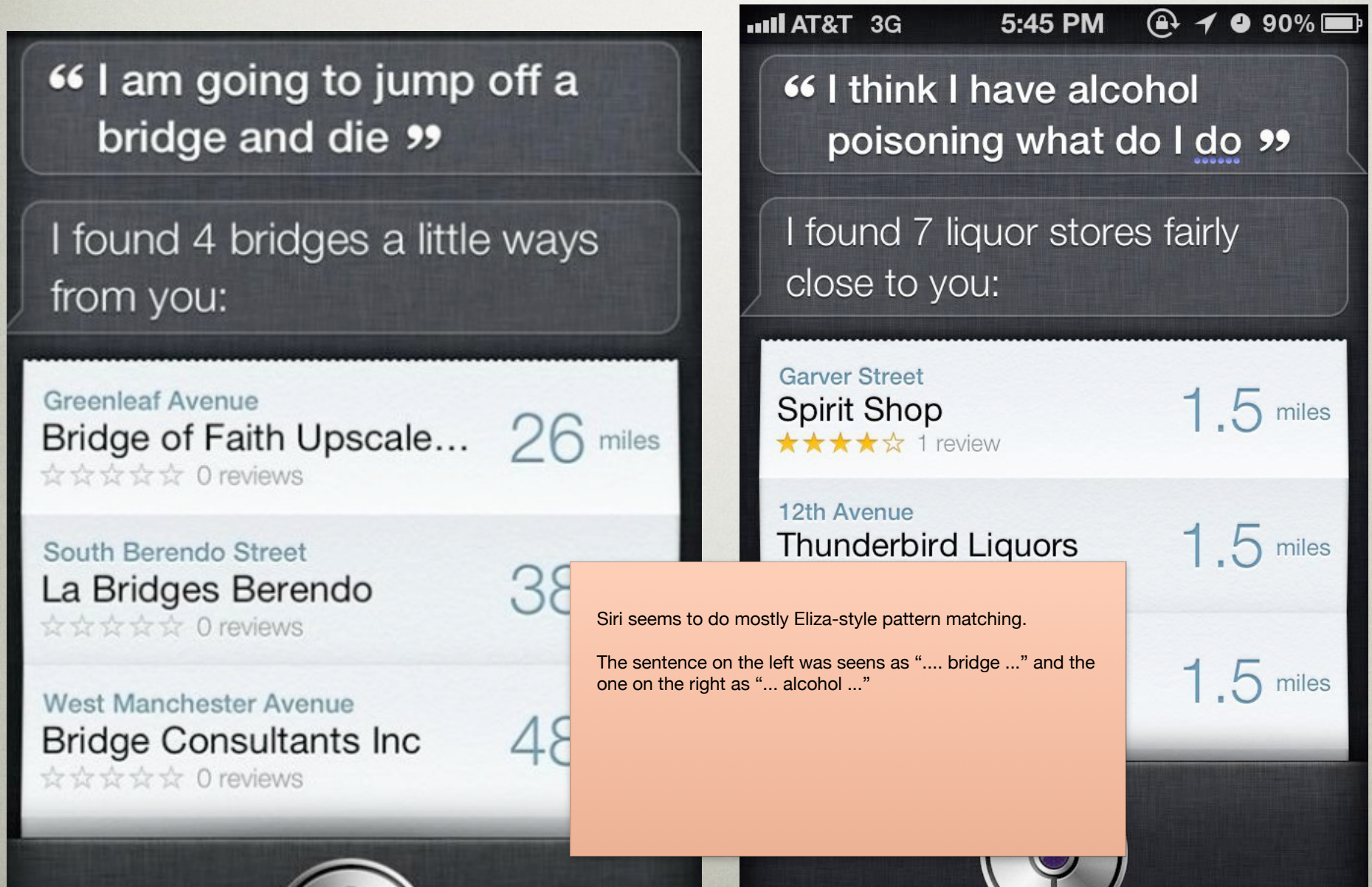
H Some of the chemists are not artists

Entailment example (From RTE)

T Eating lots of foods that are a good source of fiber may keep your blood glucose from rising fast after you eat.

H Fiber improves blood sugar control.

Question Answering
Natural Language Interface

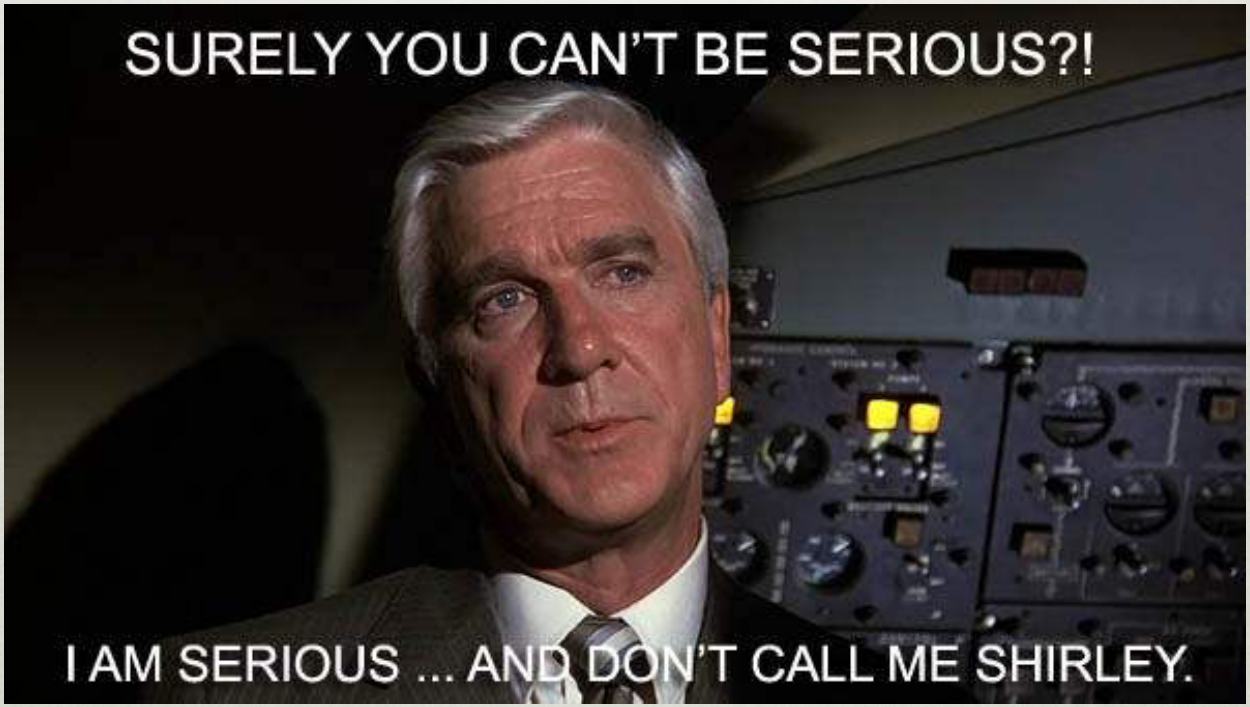


Siri seems to do mostly Eliza-style pattern matching.

The sentence on the left was seens as “.... bridge ...” and the one on the right as “... alcohol ...”

Question Answering
Natural Language Interface

Surely, You can't be serious



Question Answering Natural Language Interface

Wanted for general evilness; last seen at the tower of Barad-Dur; it's a giant eye, folks, kinda hard to miss

Watson does use deep parsing and rather successfully

Question Answering
Natural Language Interface

U.S. Cities

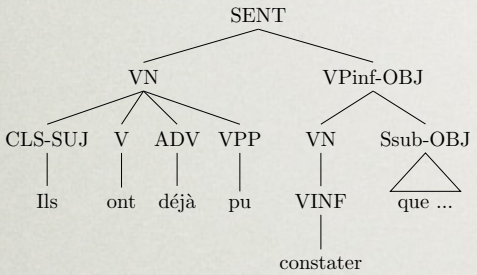
Its largest airport is named for a World War II hero; its second largest, for a World War II battle

Still, there seems to be room for improvement in the parser component: here the second sentence should be interpreted as
“its second largest [airport] [is named] for a World War II battle.

Note: this is very difficult!
np pp -> np
s ; np -> s

Outline

Treebank



Grammar Extraction

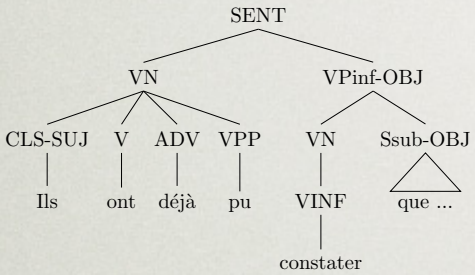
$$\dfrac{\dfrac{\text{de}}{(np\backslash s_{di})/(np\backslash s_i)}\,[Lex] \quad \dfrac{\dfrac{\text{s'}}{cl_r}\,[Lex] \quad \dfrac{\dfrac{\text{attaquer}}{(cl_r\backslash(np\backslash s_i))/pp_{\grave{a}}}\,[Lex] \quad \dfrac{}{\text{p}_0 \vdash pp_{\grave{a}}}\,[Hyp]_1}{\text{a} \circ \text{p}_0 \vdash cl_r\backslash(np\backslash s_i)}\,[/E]}{\text{s'} \circ (\text{a} \circ \text{p}_0) \vdash np\backslash s_i}\,[\backslash E]}{\text{de} \circ (\text{s'} \circ (\text{a} \circ \text{p}_0)) \vdash np\backslash s_i}\,[/E]$$

Applications

e_1 y_1	
y_1 :	e_2 e_3 x_3
	$x_3 =$ **?**
	aider_à(e_2, x_0, x_3, e_3)
	partir(e_3, x_3)
demander(e_1, y_0, x_0, y_1)	

Outline

Treebank



Grammar Extraction

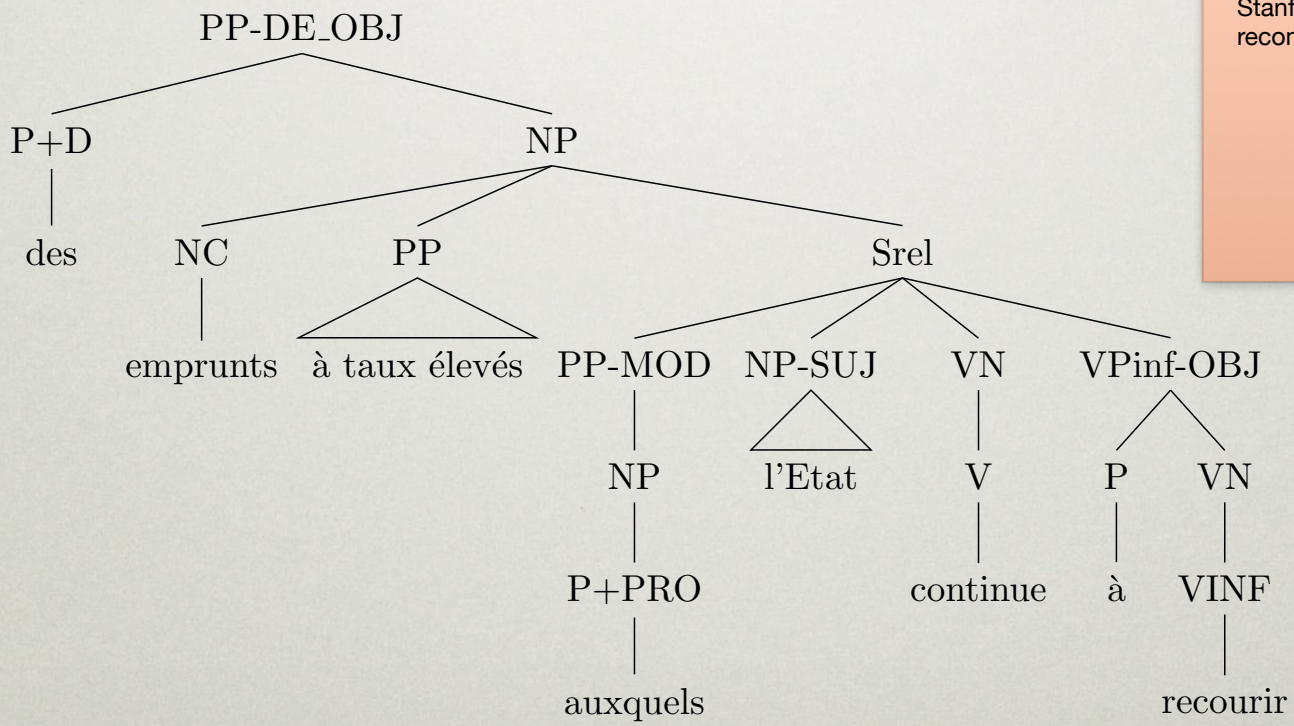
$$\dfrac{\dfrac{\text{de}}{(np\backslash s_{di})/(np\backslash s_i)}\,[Lex] \quad \dfrac{\dfrac{\text{s'}}{cl_r}\,[Lex] \quad \dfrac{\dfrac{\text{attaquer}}{(cl_r\backslash(np\backslash s_i))/pp_{\grave{a}}}\,[Lex] \quad \dfrac{}{\text{p}_0 \vdash pp_{\grave{a}}}\,[Hyp]_1}{\text{a} \circ \text{p}_0 \vdash cl_r\backslash(np\backslash s_i)}\,[/E]}{\text{s'} \circ (\text{a} \circ \text{p}_0) \vdash np\backslash s_i}\,[\backslash E]}{\text{de} \circ (\text{s'} \circ (\text{a} \circ \text{p}_0)) \vdash np\backslash s_i}\,[/E]$$

Applications

e_1 y_1	
y_1 :	e_2 e_3 x_3
	$x_3 = ?$
	aider_à(e_2, x_0, x_3, e_3)
	partir(e_3, x_3)
demander(e_1, y_0, x_0, y_1)	

The French Treebank

Example PP from the corpus (slightly simplified)

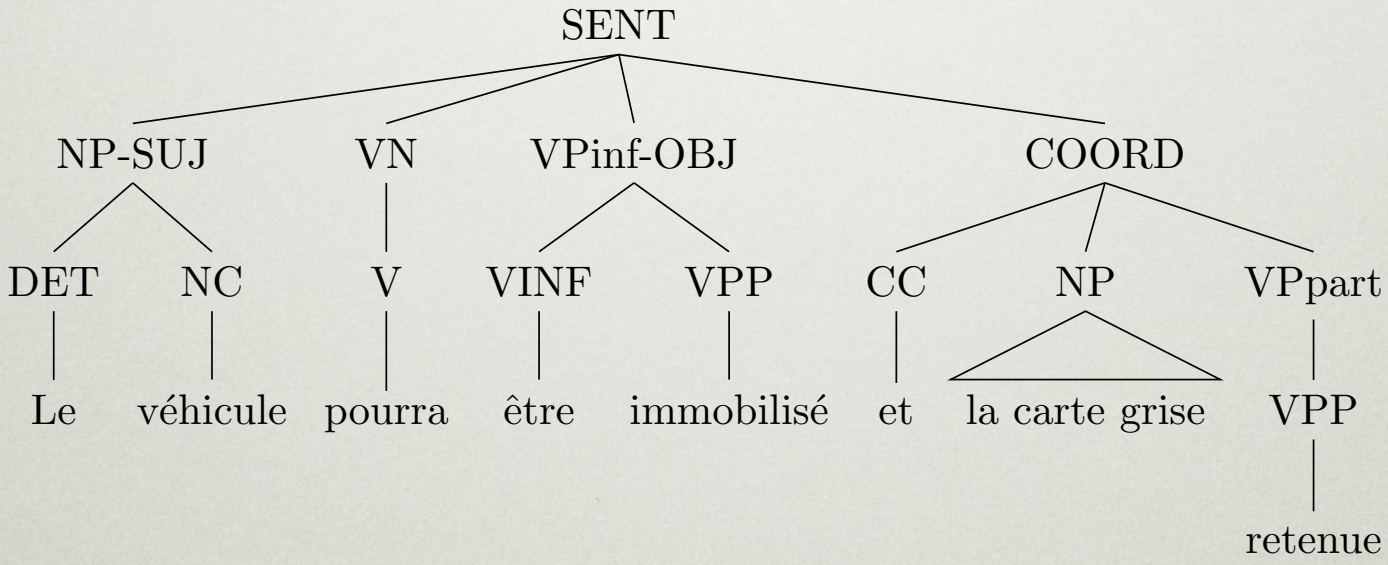


This is the FTB converted to Stanford TregEx (a tool I highlt recommend)

≈ continue_to_resort_to(state,loans)

The French Treebank

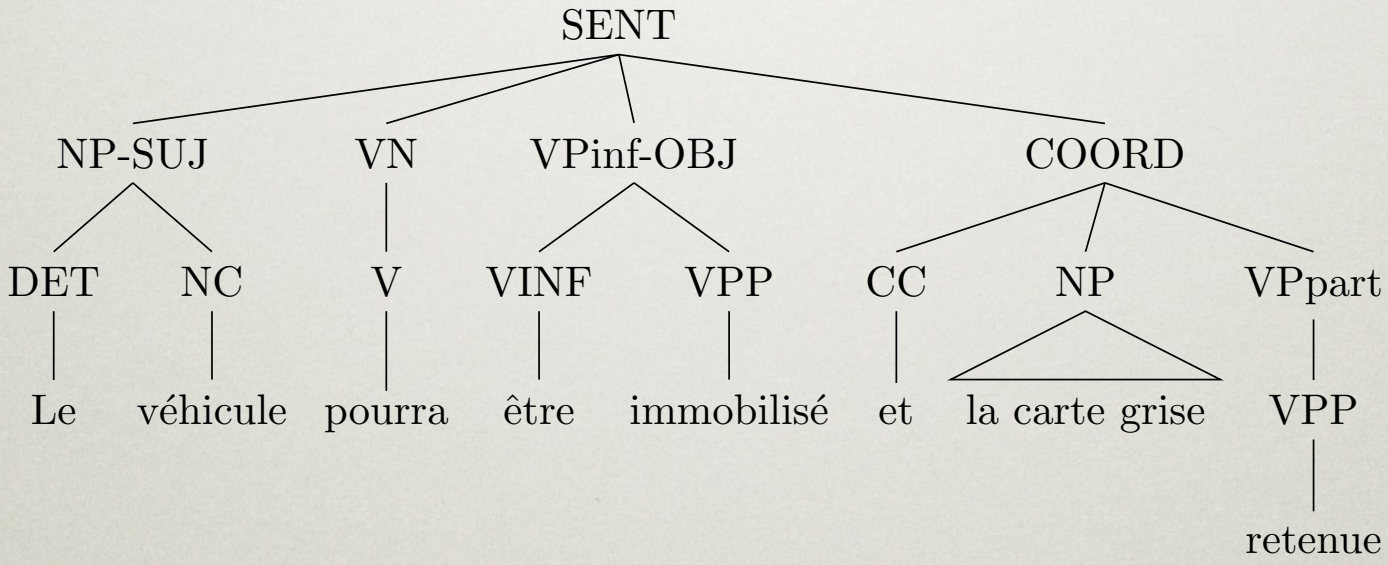
Example from the corpus (slightly simplified)



"The car could be immobilized and
the car registration (could be) taken"

The French Treebank

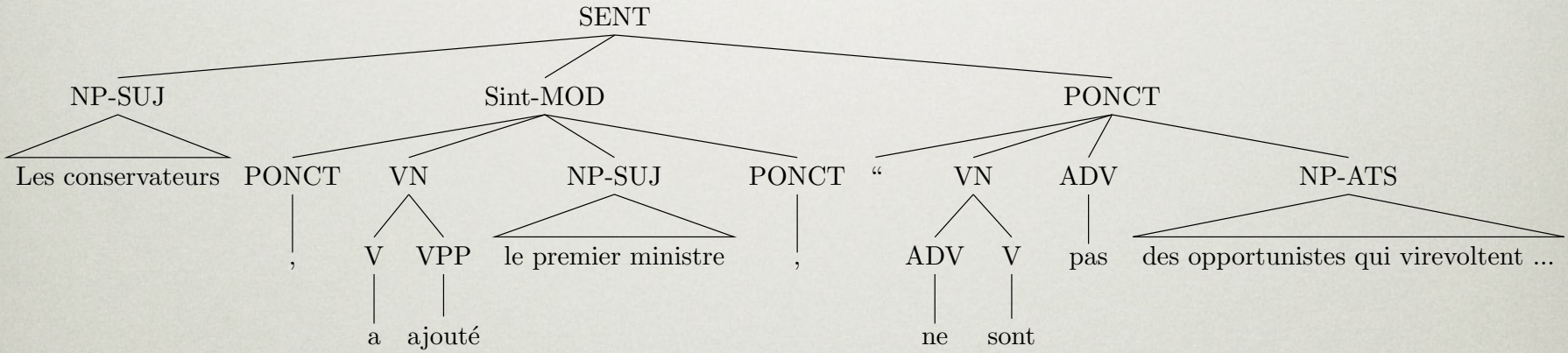
Example from the corpus (slightly simplified)



$\approx \exists x.\text{pouvoir}(\text{immobiliser}(x,\text{véhicule})) \wedge$
$\exists y.\text{pouvoir}(\text{retenir}(y,\text{cart\_grise}))$

The French Treebank

Example from the corpus (slightly simplified)



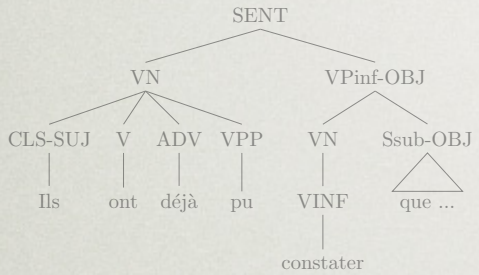
The conservative party, added the prime minister, “are not opportunists who flip-flop...”

FTB: The Good, The Bad and The Ugly

- Context-free grammar
- Verb group as a constituent
- Treatment of coordination
- No annotation of passives, long-distance dependencies, ...

Outline

French Treebank



Grammar Extraction

$$\dfrac{\dfrac{\text{de}}{(np\backslash s_{di})/(np\backslash s_i)}\,[Lex] \quad \dfrac{\dfrac{\text{s'}}{cl_r}\,[Lex] \quad \dfrac{\dfrac{\text{attaquer}}{(cl_r\backslash(np\backslash s_i))/pp_{\grave{a}}}\,[Lex] \quad \dfrac{}{\text{p}_0 \vdash pp_{\grave{a}}}\,[Hyp]_1}{\text{a} \circ \text{p}_0 \vdash cl_r\backslash(np\backslash s_i)}\,[/E]}{\text{s'} \circ (\text{a} \circ \text{p}_0) \vdash np\backslash s_i}\,[\backslash E]}{\text{de} \circ (\text{s'} \circ (\text{a} \circ \text{p}_0)) \vdash np\backslash s_i}\,[/E]$$

Applications

e_1 y_1	
y_1 :	e_2 e_3 x_3
	$x_3 = ?$
	aider_à(e_2, x_0, x_3, e_3)
	partir(e_3, x_3)
demander(e_1, y_0, x_0, y_1)	

Categorial Grammars

Formulas and corresponding expressions

- np
- n
- s
- np\s
- np/n
- (np\s)/np

- Jean, l'étudiant, …
- étudiant, économie, …
- Jean dort, Jean aime Marie
- dort, aime Marie
- un, chaque, l'
- aime, étudie

Categorial Grammars

Rules

Lambek categorial grammars have only four rules: an elimination and an introduction rule for both "\" and "/"

$$\frac{A/B \qquad B}{A}\,[/E] \qquad \frac{B \qquad B\backslash A}{A}\,[\backslash E]$$

$$\frac{\begin{array}{c}\cdots \\ A\end{array}}{A/B}\,[/I]^i \qquad \frac{\begin{array}{c}[B]^i \\ \vdots \\ A\end{array}}{A/B}\,[/I]^i \qquad \frac{\begin{array}{c}[B]^i \\ \vdots \\ A\end{array}}{B\backslash A}\,[\backslash I]^i \qquad \frac{\begin{array}{c}\cdots \\ A\end{array}}{B\backslash A}\,[\backslash I]^i$$

Categorial Grammars

Example

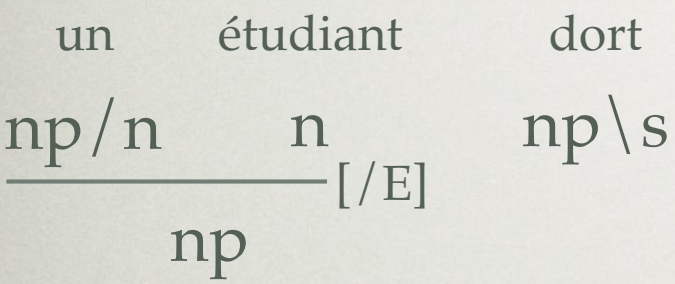
un	étudiant	dort
np/n	n	np\s

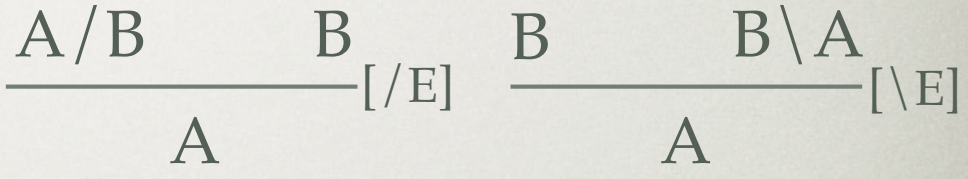
$$\frac{A/B \qquad B}{A}\,[/E] \qquad \frac{B \qquad B\backslash A}{A}\,[\backslash E]$$

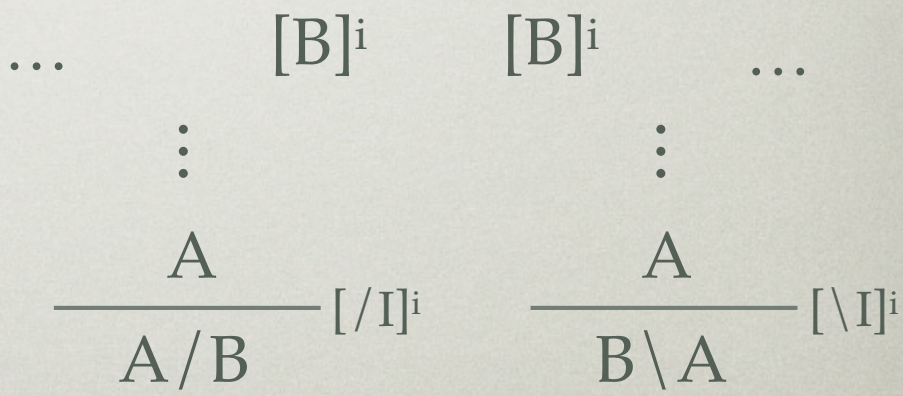
$$\frac{\begin{array}{c}\ldots \\ \vdots \\ A\end{array}}{A/B}\,[/I]^i \qquad \frac{\begin{array}{c}[B]^i \\ \vdots \\ A\end{array}}{B\backslash A}\,[\backslash I]^i$$

Categorial Grammars

Example

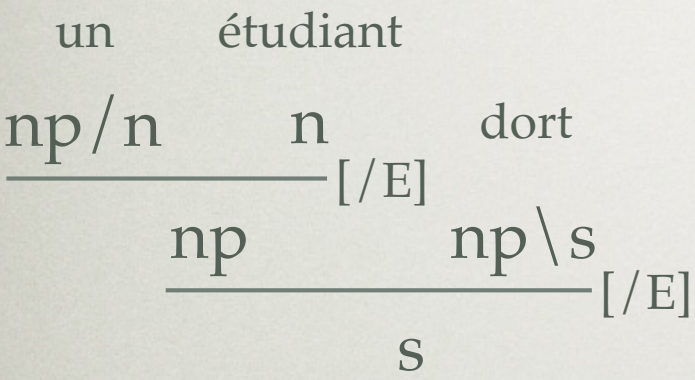


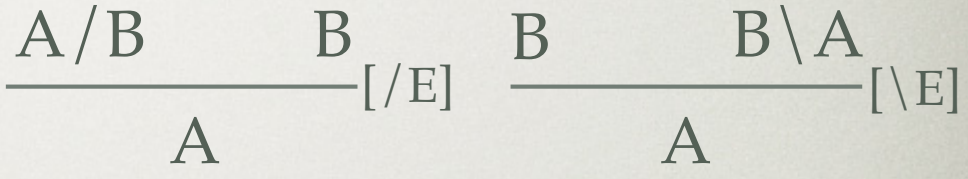


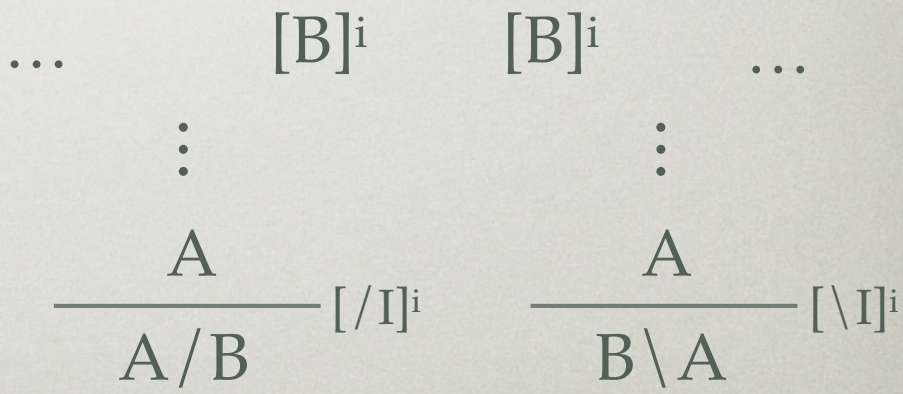


Categorial Grammars

Example



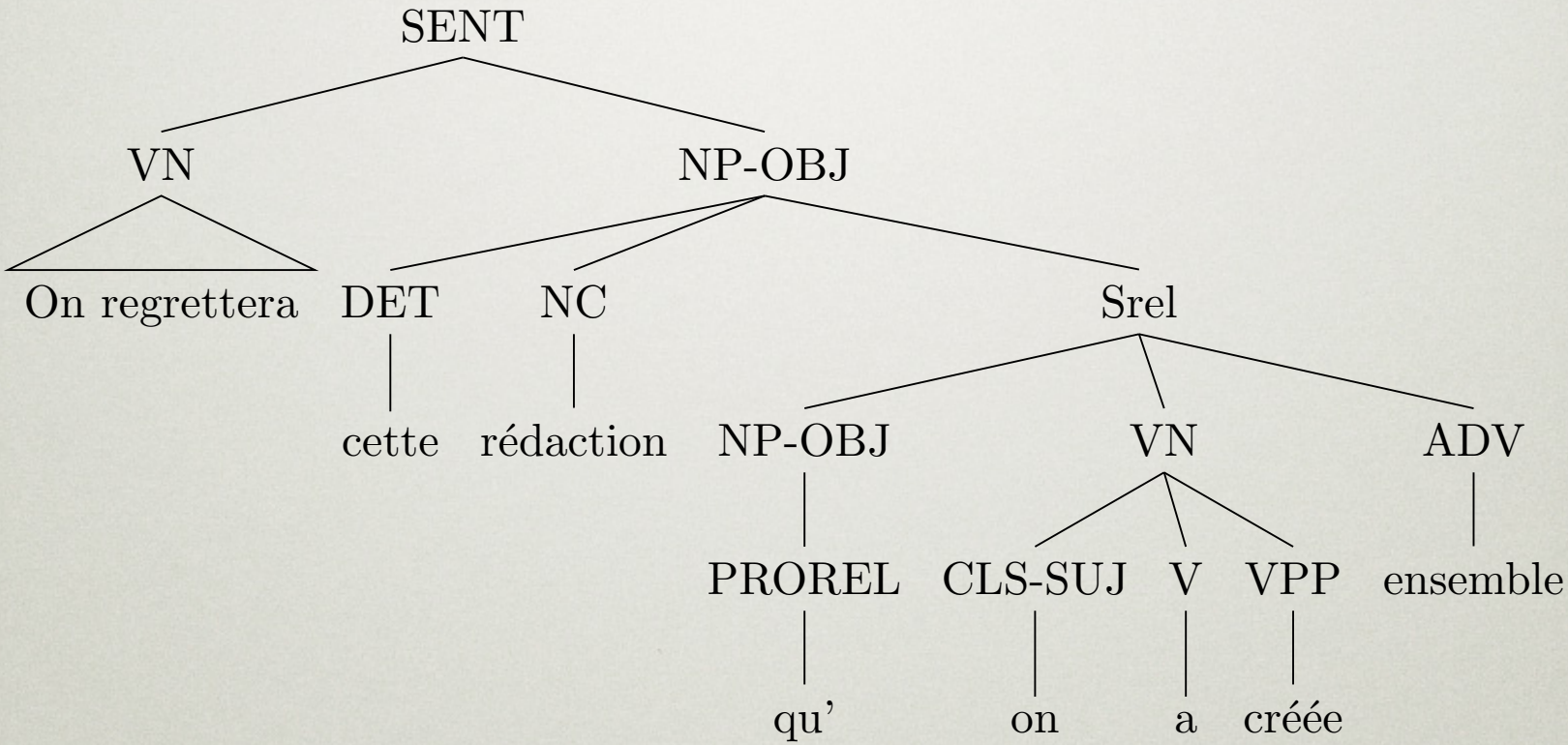




Lambek grammars and beyond

- getting the semantics right requires a somewhat richer system than AB grammars
- introduction rules ("traces" or the original "slash categories" and their semantics)
- structural rules ("movement" or "head wrap", essentially restricted tree rewrite operations)

Introduction Rules: Example



Introduction Rules: Example

redaction	qu’	on	a	créée
n	$(n\backslash n)/(s/np)$	np	$(np\backslash s)/(np\backslash s_{ppart})$	$(np\backslash s_{ppart})/np$

Introduction Rules: Example

redaction	qu’	on	a	créée	
n	$(n\backslash n)/(s/np)$	np	$(np\backslash s)/(np\backslash s_{ppart})$	$(np\backslash s_{ppart})/np$	np

Introduction Rules: Example

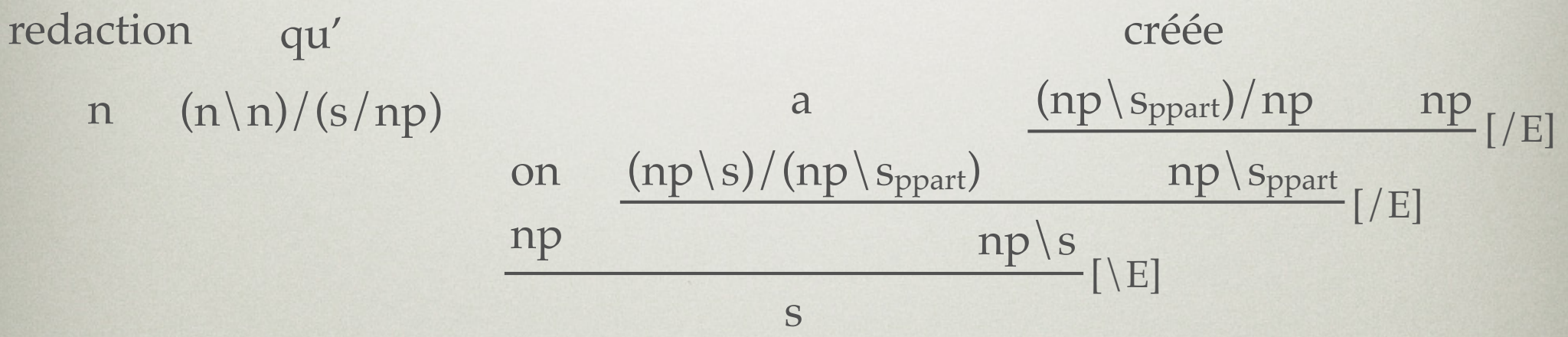
redaction	qu'	on	a	créée	
n	$(n\backslash n)/(s/np)$	np	$(np\backslash s)/(np\backslash s_{ppart})$	$(np\backslash s_{ppart})/np$	np

$$\frac{(np\backslash s_{ppart})/np \qquad np}{np\backslash s_{ppart}}\,[/E]$$

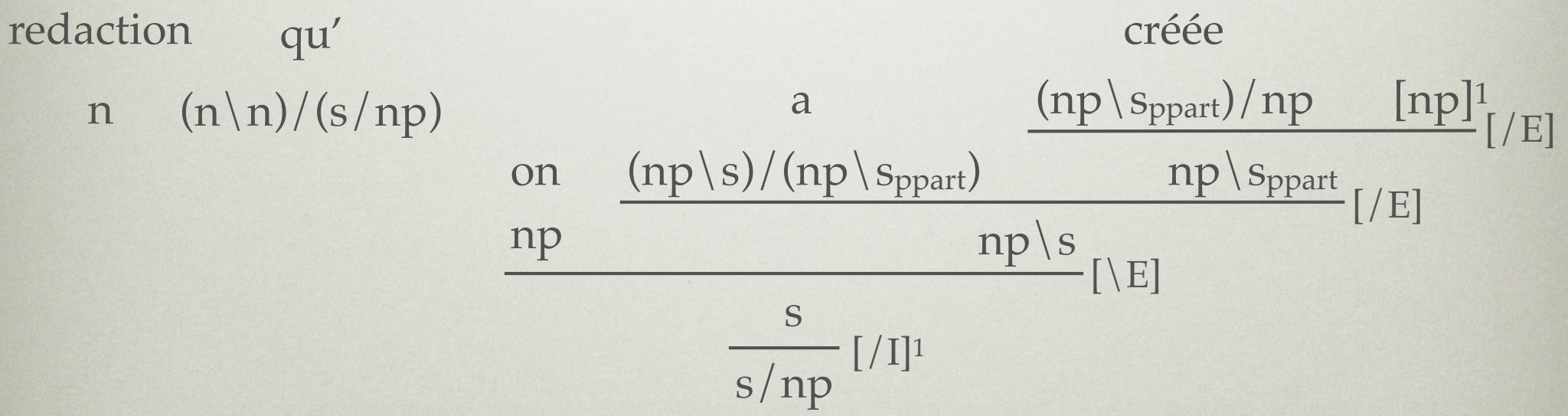
Introduction Rules: Example

redaction	qu'	on	créée		
n	$(n\backslash n)/(s/np)$	np	a	$(np\backslash s_{ppart})/np$	np
			$(np\backslash s)/(np\backslash s_{ppart})$	$np\backslash s_{ppart}$ [/E]	
			$np\backslash s$	[/E]	

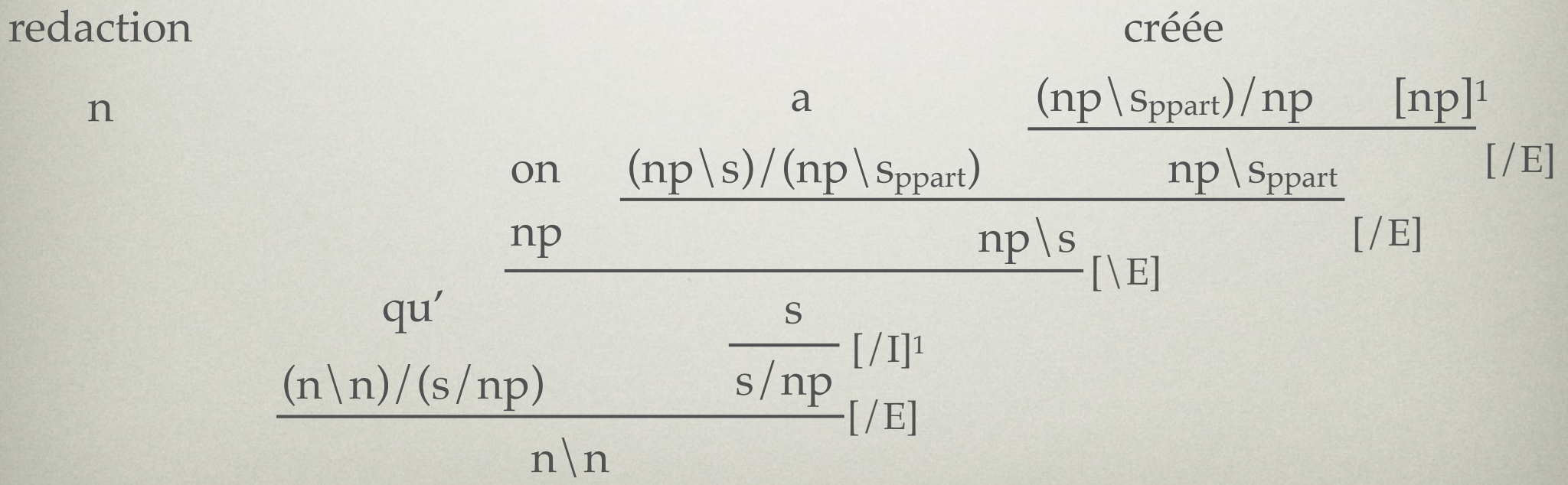
Introduction Rules: Example



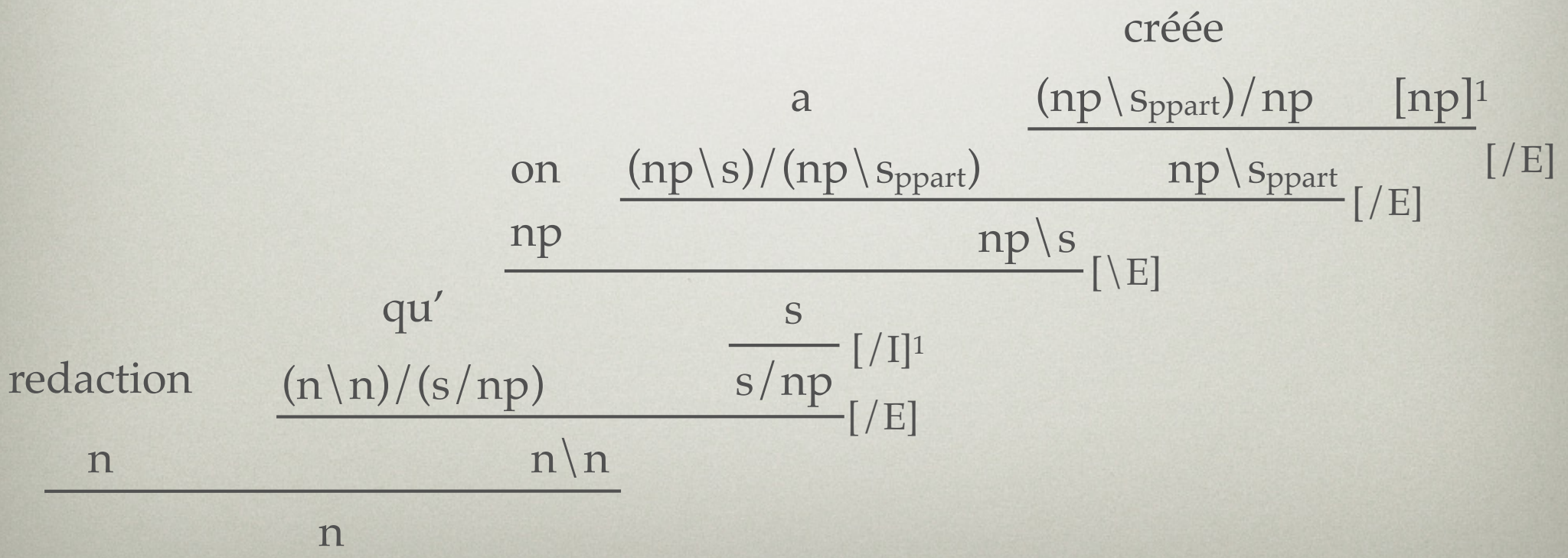
Introduction Rules: Example



Introduction Rules: Example



Introduction Rules: Example



Beyond AB grammars
Introduction Rules

qu’	on	a	créée	ensemble
(n\n)/(s/np)	np	(np\s)/(np\s_{ppart})	(np\s_{ppart})/np	(np\s)\(np\s)

Lambda Calculus and Proofs as terms

- Proofs in categorial grammar correspond to lambda terms
- These lambda terms "forget" the directions of the implications.

$$\frac{t{:}A/B \qquad u{:}B}{(t\ u){:}A} \qquad \frac{u{:}B \qquad t{:}B\backslash A}{(t\ u){:}A}$$

$$\frac{\begin{array}{c}[x{:}B]\\ \vdots \\ t{:}A\end{array}}{A/B{:}\lambda x.t} \qquad \frac{\begin{array}{c}[x{:}B]\\ \vdots \\ t{:}A\end{array}}{B\backslash A{:}\lambda x.t}$$

Computing Semantics

… cet equilibre délicate que l'Iran et l'Arabie saoudite chercheront à sauvegarder ε à Genève.

$$\frac{\ldots\ [x{:}B]\ \ldots \quad \vdots \quad t{:}A}{A/\Diamond\Box B{:}\lambda x.t}$$

que

$w_1{:}(n\backslash n)/(s/\Diamond\Box np)$

l'Iran et
l'Arabie saoudite

$w_2{:}np$

chercheront à

$w_3{:}(np\backslash s)/(np\backslash s_{inf})$

sauvegarder

$w_4{:}(np\backslash s_{inf})/np$

Computing Semantics

… cet equilibre délicate que l'Iran et l'Arabie saoudite chercheront à sauvegarder ε à Genève.

$$\frac{\ldots\ [x{:}B]\ \ldots \quad \vdots \quad t{:}A}{A/\Diamond\Box B{:}\lambda x.t}$$

que

$w_1{:}(n\backslash n)/(s/\Diamond\Box np)$

l'Iran et
l'Arabie saoudite

$w_2{:}np$

chercheront à

$w_3{:}(np\backslash s)/(np\backslash s_{inf})$

sauvegarder

$w_4{:}(np\backslash s_{inf})/np$

ε

$x{:}np$

Computing Semantics

… cet equilibre délicate que l'Iran et l'Arabie saoudite chercheront à sauvegarder ε à Genève.

$$\frac{\ldots\ [x{:}B]\ \ldots \quad \vdots \quad t{:}A}{A/\Diamond\Box B{:}\lambda x.t}$$

que

$w_1{:}(n\backslash n)/(s/\Diamond\Box np)$

l'Iran et l'Arabie saoudite

$w_2{:}np$

chercheront à

$w_3{:}(np\backslash s)/(np\backslash s_{inf})$

$$\frac{\text{sauvegarder}\quad w_4{:}(np\backslash s_{inf})/np \qquad \varepsilon\quad x{:}np}{(w_4\ x){:}np\backslash s_{inf}}$$

Computing Semantics

… cet equilibre délicate que l'Iran et l'Arabie saoudite chercheront à sauvegarder ε à Genève.

$$\frac{\ldots\ [x{:}B]\ \ldots \quad \vdots \quad t{:}A}{A/\Diamond\Box B{:}\lambda x.t}$$

que

$w_1{:}(n\backslash n)/(s/\Diamond\Box np)$

l'Iran et l'Arabie saoudite

$w_2{:}np$

$$\frac{w_3{:}(np\backslash s)/(np\backslash s_{inf}) \qquad \dfrac{\text{sauvegarder}\quad \varepsilon}{w_4{:}(np\backslash s_{inf})/np \quad x{:}np}}{(w_3(w_4\,x)){:}np\backslash s}$$

chercheront à

$(w_4\,x){:}np\backslash s_{inf}$

Computing Semantics

… cet equilibre délicate que l'Iran et l'Arabie saoudite chercheront à sauvegarder ε à Genève.

$$\frac{\ldots\ [x{:}B]\ \ldots \quad \vdots \quad t{:}A}{A/\Diamond\Box B{:}\lambda x.t}$$

que: $w_1{:}(n\backslash n)/(s/\Diamond\Box np)$

$$\frac{\text{l'Iran et l'Arabie saoudite}\ \ w_2{:}np \qquad \dfrac{\text{chercheront à}\ \ w_3{:}(np\backslash s)/(np\backslash s_{inf}) \qquad \dfrac{\text{sauvegarder}\ \ w_4{:}(np\backslash s_{inf})/np \qquad \varepsilon\ \ x{:}np}{(w_4\ x){:}np\backslash s_{inf}}}{(w_3(w_4\ x)){:}np\backslash s}}{((w_3(w_4\ x))\ w_2){:}s}$$

Computing Semantics

… cet equilibre délicate que l'Iran et l'Arabie saoudite chercheront à sauvegarder ε à Genève.

$$\frac{\dots\ [x{:}B]\ \dots \\ \vdots \\ t{:}A}{A/\Diamond\Box B{:}\lambda x.t}$$

$$\frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{w_1{:}(n\backslash n)/(s/\Diamond\Box np)}\ \text{que} \qquad \dfrac{\dfrac{}{w_2{:}np}\ \text{l'Iran et l'Arabie saoudite} \qquad \dfrac{\dfrac{}{w_3{:}(np\backslash s)/(np\backslash s_{inf})}\ \text{chercheront à} \qquad \dfrac{\dfrac{}{w_4{:}(np\backslash s_{inf})/np}\ \text{sauvegarder} \qquad \dfrac{}{x{:}np}\ \varepsilon}{(w_4\ x){:}np\backslash s_{inf}}}{(w_3(w_4\ x)){:}np\backslash s}}{((w_3(w_4\ x))\ w_2){:}s}}{\lambda x.((w_3(w_4\ x))\ w_2){:}s/\Diamond\Box np}}{\dots}$$

Computing Semantics

… cet equilibre délicate que l'Iran et l'Arabie saoudite chercheront à sauvegarder ε à Genève.

$$\frac{\ldots\ [x{:}B]\ \ldots \quad \vdots \quad t{:}A}{A/\Diamond\Box B{:}\lambda x.t}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{w_1{:}(n\backslash n)/(s/\Diamond\Box np)}\ \text{que} \quad \dfrac{\dfrac{\dfrac{}{w_2{:}np}\ \text{l'Iran et l'Arabie saoudite} \quad \dfrac{\dfrac{}{w_3{:}(np\backslash s)/(np\backslash s_{inf})}\ \text{chercheront à} \quad \dfrac{\dfrac{}{w_4{:}(np\backslash s_{inf})/np}\ \text{sauvegarder} \quad \dfrac{}{x{:}np}\ \varepsilon}{(w_4\ x){:}np\backslash s_{inf}}}{(w_3(w_4\ x)){:}np\backslash s}}{((w_3(w_4\ x))\ w_2){:}s}}{\lambda x.((w_3(w_4\ x))\ w_2){:}s/\Diamond\Box np}}{(w_1\ \lambda x.((w_3(w_4\ x))\ w_2)){:}n\backslash n}$$

Computing Semantics

w_1:$\lambda P \lambda Q \lambda x (Px) \wedge (Qx)$
w_2:*I&As*
w_3:*chercher_à*
w_4:*sauvegarder*

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{w_1:(n\backslash n)/(s/\Diamond\Box np)}\ \text{que} \quad \dfrac{\dfrac{\dfrac{}{w_2:np}\ \text{l'Iran et l'Arabie saoudite} \quad \dfrac{}{w_3:(np\backslash s)/(np\backslash s_{inf})}\ \text{chercheront à} \quad \dfrac{\dfrac{}{w_4:(np\backslash s_{inf})/np}\ \text{sauvegarder} \quad \dfrac{}{x:np}\ \varepsilon}{(w_4\ x):np\backslash s_{inf}}}{(w_3(w_4\ x)):np\backslash s}}{((w_3(w_4\ x))\ w_2):s}}{\lambda x.((w_3(w_4\ x))\ w_2):s/\Diamond\Box np}}{(w_1\ \lambda x.((w_3(w_4\ x))\ w_2)):n\backslash n}$$

Computing Semantics

w_1:$\lambda P \lambda Q \lambda y(Py) \wedge (Qy)$
w_2:*I&As*
w_3:*chercher_à*
w_4:*sauvegarder*

$(w_1\ \lambda x.((w_3(w_4\ x))\ w_2))$

↓

$((\lambda P \lambda Q \lambda y(Py) \wedge (Qy))$
$\lambda x.((chercher\_à(sauvegarder\ x))\ I\&As))$

↓

$(\lambda Q \lambda y(\lambda x.((chercher\_à(sauvegarder\ x))\ I\&As)\ y) \wedge (Qy)))$

Computing Semantics

w_1:$\lambda P \lambda Q \lambda y(Py) \wedge (Qy)$
w_2:*I&As*
w_3:*chercher_à*
w_4:*sauvegarder*

$(w_1\ \lambda x.((w_3(w_4\ x))\ w_2))$

↓

$((\lambda P \lambda Q \lambda y(Py) \wedge (Qy))$
$\lambda x.((chercher\_à(sauvegarder\ x))\ I\&As))$

↓

$(\lambda Q \lambda y(\lambda x.((chercher\_à(sauvegarder\ x))\ I\&As)\ y) \wedge (Qy)))$

↓

$(\lambda Q \lambda y((chercher\_à(sauvegarder_y))\ I\&As)) \wedge (Qy)))$

↓

$\lambda x.équilibre(x) \wedge délicat(x) \wedge$
$chercher\_à(I\&As, (sauvegarder\ x))$

A Tale of Two Categorial Grammars

Logic and Combinators

combinatorial logic

combinators		logic
SKI, BCK, BCI	⟷	IIL, IAL, ILL

Logics and Combinators

combinatorial logic

combinators		logic
SKI, BCK, BCI	⟷	IIL, IAL, ILL

CCG

combinators (?)	⟶	logic???

Lambek/TLG

combinators???	⟵	logic

CCG and Type-Logical Grammar

Both complicate locally and simplify globally but with a different notion of locality

	CCG	TLG
Sequence	Local	Non-local
Connective	Non-local	Local

CCG: each category can be traced back to a continuous sequence of the input categories
TLG: a formula can correspond to a discontinuous sequence of lexical formulas

CCG: Rules can refer to multiple connectives
TLG: logical rules operate on one connective at a time (subformula property, independence of

Crossed Composition in CCG

qu’	on	a	créée	ensemble
(n\n)/(s/np)	np	(np\s)/(np\s_{ppart})	(np\s_{ppart})/np	(np\s)\(np\s)

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B \qquad \frac{B/C \quad B\backslash A}{A/C}\,Bx$$

$$\dfrac{\text{qu'} \quad \dfrac{\text{on} \quad \dfrac{\text{a} \quad \text{créée} \quad \text{ensemble}}{(np\backslash s)/(np\backslash s_{ppart}) \quad (np\backslash s_{ppart})/np \quad (np\backslash s)\backslash(np\backslash s)}}{(np\backslash s)/np}}{(n\backslash n)/(s/np) \quad \dfrac{np}{s/(np\backslash s)} \quad (np\backslash s)/np}$$

$$s/np$$

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B \qquad \frac{B/C \quad B\backslash A}{A/C}\,Bx$$

$$\dfrac{\text{qu'} \quad \dfrac{\text{on} \quad \dfrac{\text{a} \quad \dfrac{\text{créée} \quad \text{ensemble}}{(np\backslash s)\backslash(np\backslash s)}}{(np\backslash s)/np}}{(np\backslash s)/(np\backslash s_{ppart})}}{(np\backslash s)/np}$$

$$\dfrac{(n\backslash n)/(s/np) \quad \dfrac{np}{s/(np\backslash s)}}{s/np}$$

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B \qquad \frac{B/C \quad B\backslash A}{A/C}\,Bx$$

$$\dfrac{\text{qu'} \quad \dfrac{\text{on} \quad \dfrac{\text{a} \quad \text{créée} \quad \dfrac{\text{ensemble}}{(np\backslash s)\backslash(np\backslash s)}}{(np\backslash s)/np}}{(np\backslash s)/(np\backslash s_{ppart}) \quad (np\backslash s_{ppart})/np}}{(n\backslash n)/(s/np) \qquad \dfrac{np}{s/(np\backslash s)} \qquad (np\backslash s)/np}$$

$$s/np$$

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B \qquad \frac{B/C \quad B\backslash A}{A/C}\,Bx$$

$$\dfrac{\text{qu'} \quad \dfrac{\text{on} \quad \dfrac{\text{a} \quad \text{créée} \quad \dfrac{\text{ensemble}}{(np\backslash s)\backslash(np\backslash s)}}{(np\backslash s)/np}}{(np\backslash s)/np}}{s/np}$$

$$(n\backslash n)/(s/np) \qquad np \qquad (np\backslash s)/(np\backslash s_{ppart}) \quad (np\backslash s_{ppart})/np$$

$$s/(np\backslash s)$$

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B \qquad \frac{B/C \quad B\backslash A}{A/C}\,Bx$$

$$\dfrac{\text{qu'} \quad \dfrac{\text{on} \quad \dfrac{\text{a} \quad \text{créée} \quad \text{ensemble}}{(np\backslash s)/(np\backslash s_{ppart}) \quad (np\backslash s_{ppart})/np \quad (np\backslash s)\backslash(np\backslash s)}}{(np\backslash s)/np}}{(n\backslash n)/(s/np) \quad \dfrac{np}{s/(np\backslash s)} \qquad (np\backslash s)/np}$$

$$s/np$$

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T$$

harmless

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B$$

harmless

mostly harmless

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B \qquad \frac{B/C \quad B\backslash A}{A/C}\,Bx$$

harmless

mostly harmless

danger!

Crossed Composition in CCG

$$\frac{A}{B/(A\backslash B)}\,T \qquad \frac{A/B \quad B/C}{A/C}\,B \qquad \frac{B/C \quad B\backslash A}{A/C}\,Bx$$

harmless

mostly harmless

danger!

$$\frac{\dfrac{A \quad [A\backslash B]^i}{B}\,\backslash E}{B/(A\backslash B)}\,/I_i$$

Crossed Composition in CCG

$$\frac{\mathrm{A}}{\mathrm{B/(A\backslash B)}}\,\mathrm{T} \qquad \frac{\mathrm{A/B} \quad \mathrm{B/C}}{\mathrm{A/C}}\,\mathrm{B} \qquad \frac{\mathrm{B/C} \quad \mathrm{B\backslash A}}{\mathrm{A/C}}\,\mathrm{Bx}$$

harmless

mostly harmless

$$\frac{\dfrac{A \quad [A\backslash B]^i}{B}\,\backslash E}{B/(A\backslash B)}\,/I_i \qquad \frac{A/B \quad \dfrac{B/C \quad [C]^i}{B}\,/E}{\dfrac{A}{A/C}\,/I^i}\,/E$$

Crossed Composition in CCG

$$\frac{\mathrm{A}}{\mathrm{B/(A\backslash B)}}\,\mathrm{T} \qquad \frac{\mathrm{A/B} \quad \mathrm{B/C}}{\mathrm{A/C}}\,\mathrm{B} \qquad \frac{\mathrm{B/C} \quad \mathrm{B\backslash A}}{\mathrm{A/C}}\,\mathrm{Bx}$$

harmless

mostly harmless

$$\frac{\dfrac{x : A \quad [f : A\backslash B]^i}{(f\ x) : B}\,\backslash E}{\lambda f.(f\ x) : B/(A\backslash B)}\,/I_i \qquad \frac{f : A/B \quad \dfrac{g : B/C \quad [x : C]^i}{(g\ x) : B}\,/E}{(f\ (g\ x)) : A}\,/E}{\lambda x.(f\ (g\ x)) : A/C}\,/I^i$$

Crossed Composition

$$\dfrac{\dfrac{B/C \quad [C]^i}{B}\,/E \qquad B\backslash A}{\dfrac{A}{A/C}\,?I_i}\,\backslash E$$

Crossed Composition

$$\dfrac{\dfrac{B/C \quad [C]^i}{B}\,/E \qquad B\backslash A}{\dfrac{A}{A/C}\,/I_i}\,\backslash E$$

Crossed Composition

$$\dfrac{\dfrac{B/C \quad [C]^i}{B}\,/E \qquad B\backslash A}{\dfrac{A}{A/\Diamond\Box C}\,/\Diamond\Box_i}\,\backslash E$$

Crossed Composition

$$\dfrac{\dfrac{\dfrac{g : B/C \quad [x : C]^i}{(g\ x) : B}\ /E \qquad f : B\backslash A}{(f\ (g\ x)) : A}\ \backslash E}{\lambda x.(f\ (g\ x)) : A/C}\ ?I_i$$

How to Extend or Modify a Logic

- Suppose we are unhappy with our logic and we want to change it, then what is our job?

1. reprove formal results (normalisation, subformula propertry)
2. verify we keep what worked and can do some things which didn't work before (possibly by direct translation, but this is not always possible)

Beyond Lambek grammars
New Introduction Rules

$$.... \; [B]^i \; ....$$

$$\frac{A}{A/\Diamond\Box B}\,[/I]^i$$

qu’	on	a	créée	ensemble
$(n\backslash n)/(s/\Diamond\Box np)$	np	$(np\backslash s)/(np\backslash s_{ppart})$	$(np\backslash s_{ppart})/np$	$(np\backslash s)\backslash(np\backslash s)$

Beyond Lambek grammars
New Introduction Rules

qu’	on	a	créée	ensemble
$(n\backslash n)/(s/\Diamond\Box np)$	np	$(np\backslash s)/(np\backslash s_{ppart})$	$(np\backslash s_{ppart})/np$	$(np\backslash s)\backslash(np\backslash s)$

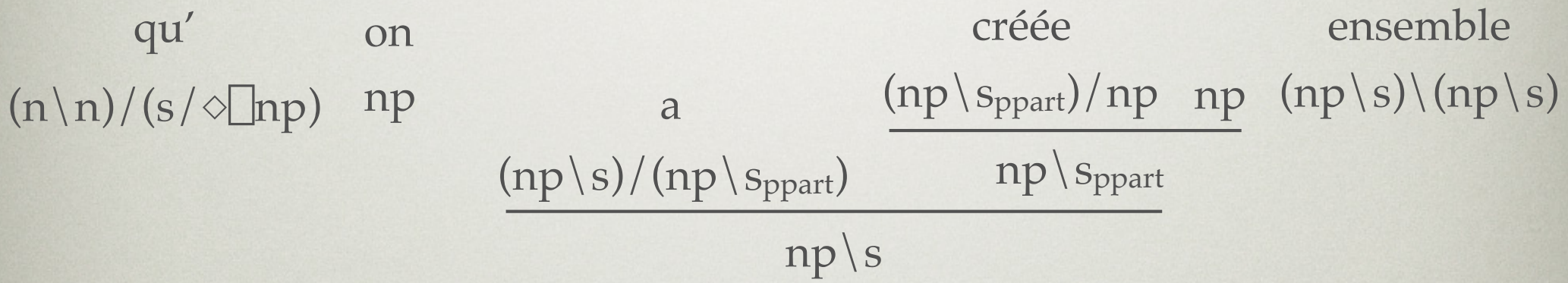
Beyond Lambek grammars
New Introduction Rules

qu’	on	a	créée		ensemble
$(n\backslash n)/(s/\Diamond\Box np)$	np	$(np\backslash s)/(np\backslash s_{ppart})$	$(np\backslash s_{ppart})/np$	np	$(np\backslash s)\backslash(np\backslash s)$

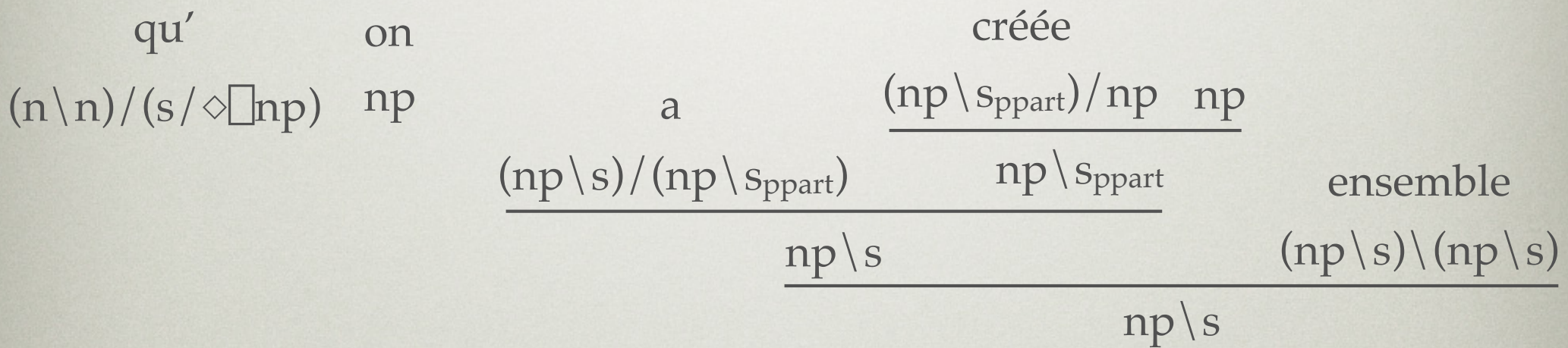
Beyond Lambek grammars
New Introduction Rules

qu’	on	a	créée		ensemble
$(n\backslash n)/(s/\Diamond\Box np)$	np	$(np\backslash s)/(np\backslash s_{ppart})$	$(np\backslash s_{ppart})/np$	np	$(np\backslash s)\backslash(np\backslash s)$
			$np\backslash s_{ppart}$		

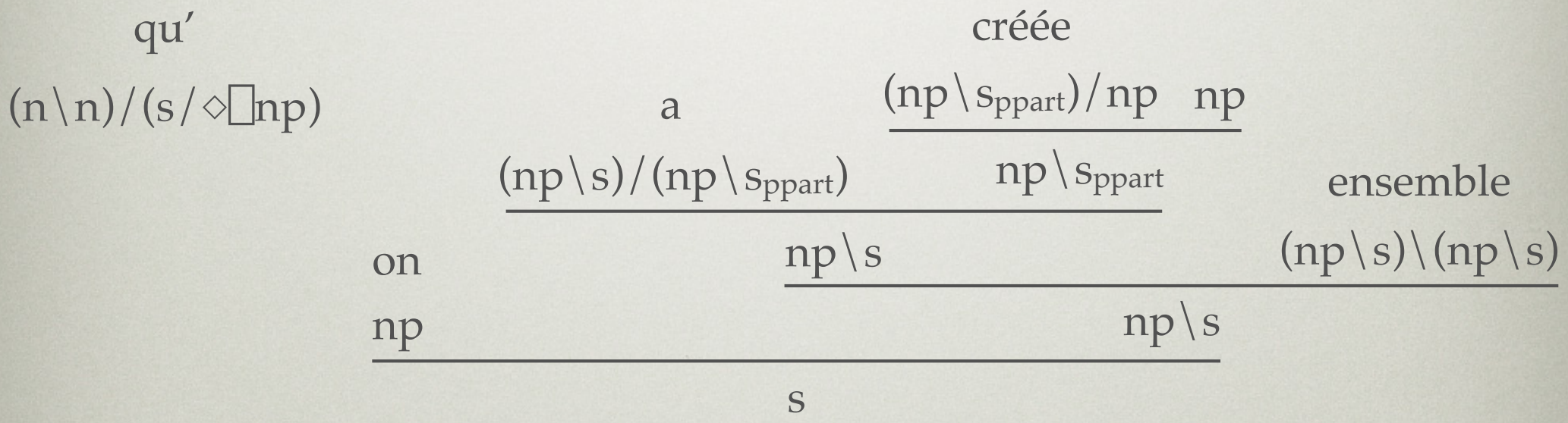
Beyond Lambek grammars
New Introduction Rules



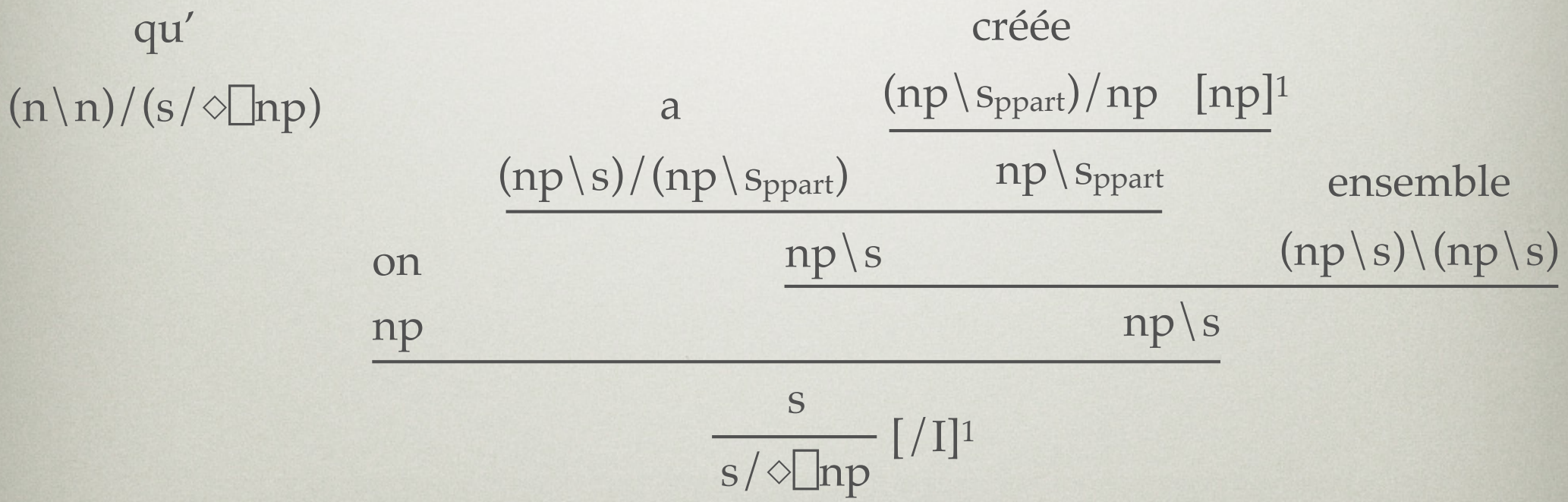
Beyond Lambek grammars
New Introduction Rules



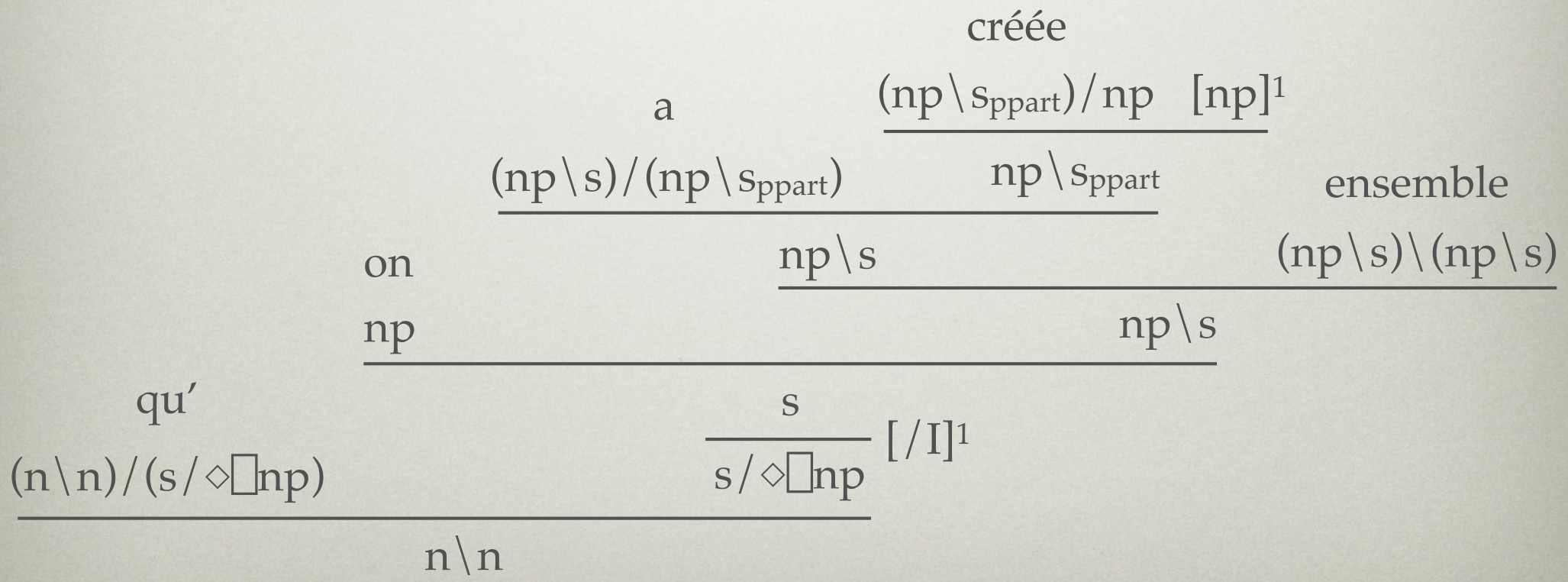
Beyond Lambek grammars
New Introduction Rules



Beyond Lambek grammars
New Introduction Rules



Beyond Lambek grammars
New Introduction Rules



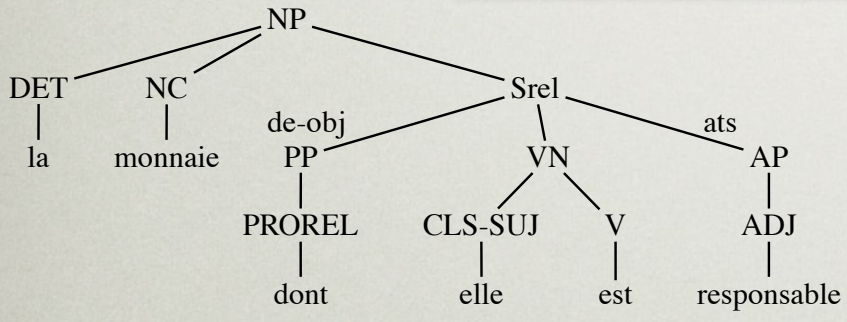
Treebank Extraction

- A rather classical approach. The extraction algorithm is parametric wrt three functions:

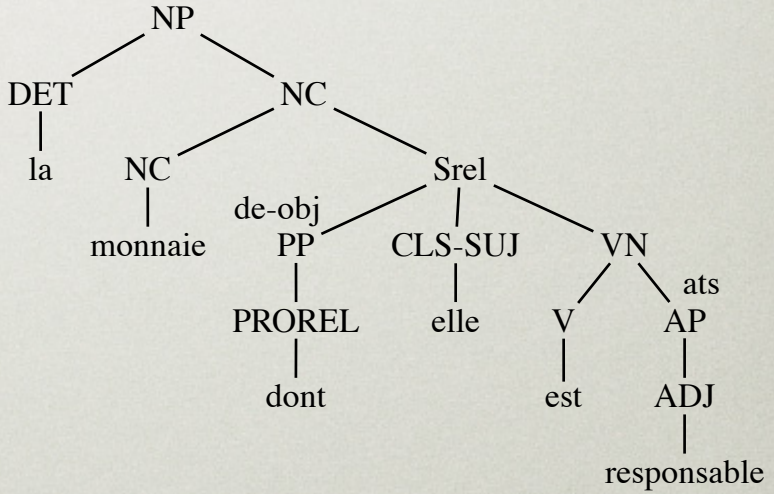
1. a function identifying heads/functors
2. a function identifying modifiers
3. a function given a formula for each syntactic category of the annotation

Treebank Extraction

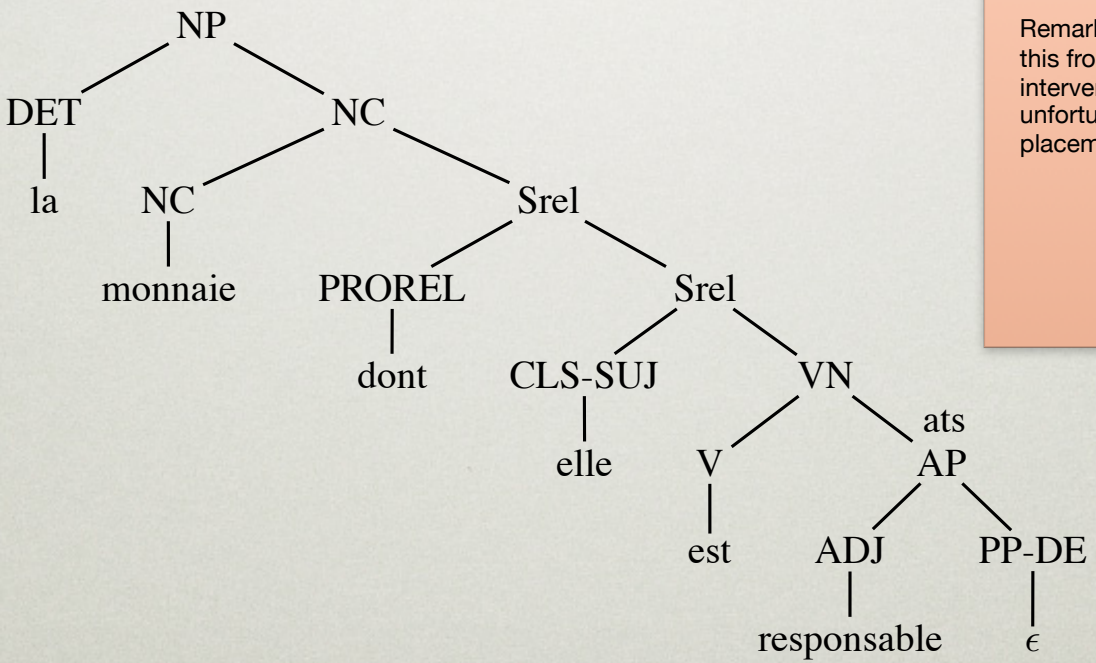
Sentence as we find it in the corpus. “dont” is a relative pronoun like “que” but which selects a sentence missing de “de” preposition (instead of a sentence missing an np like “que”)



Note how “dont” is annotated as a “de-obj” argument, which is useful.



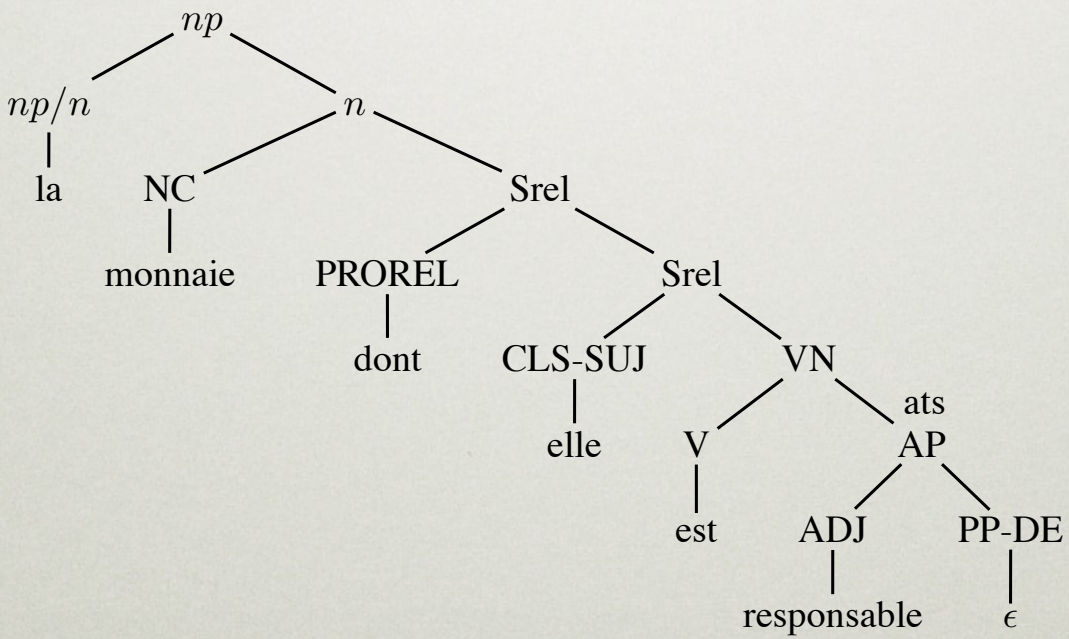
Treebank Extraction



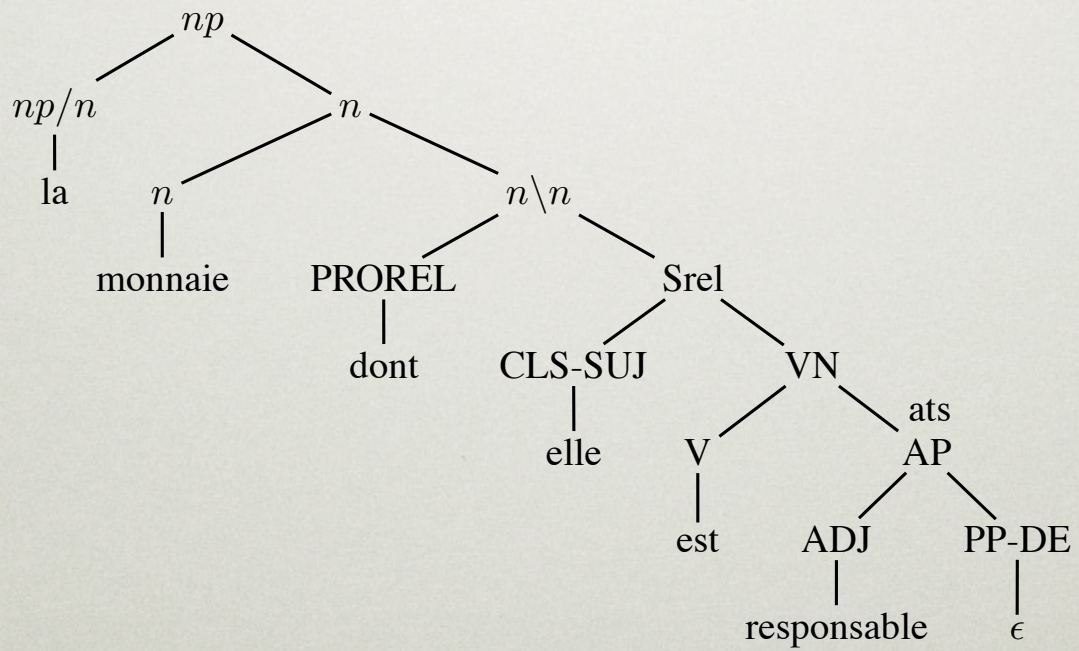
However, the “de” preposition belongs to “responsable” (some adjectives select for prepositions: “responsable de X” functions as an adjective just as “responsable”)

Remark however, that there is no way to derive this from the annotation as it is given. Manual intervention (or at least verification!) is unfortunately necessary to assure the correct placement of the hypothetical preposition.

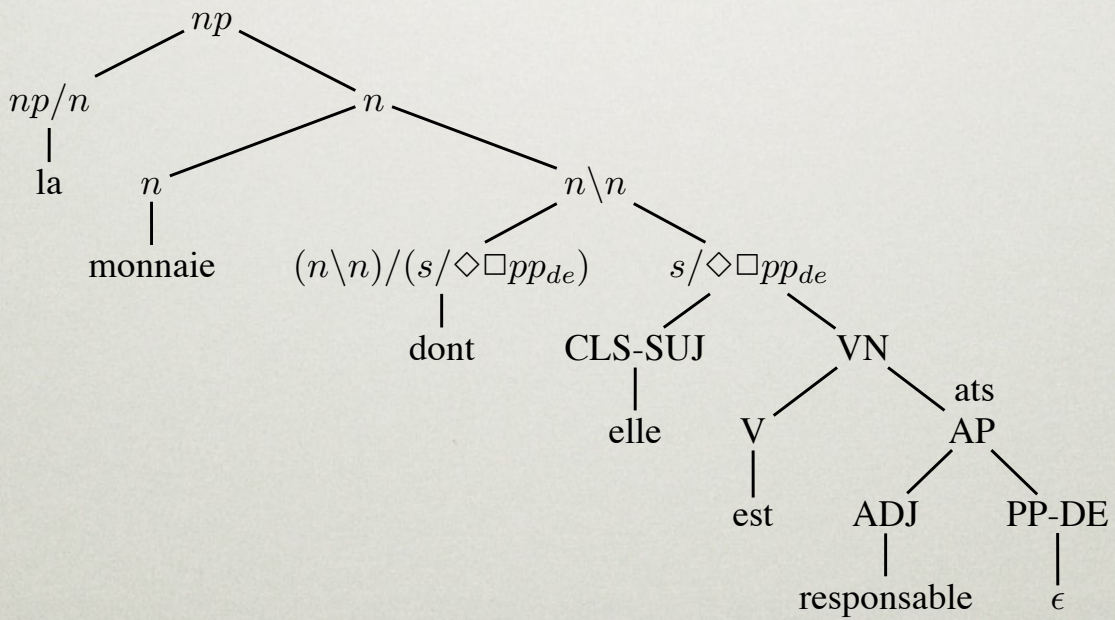
Treebank Extraction



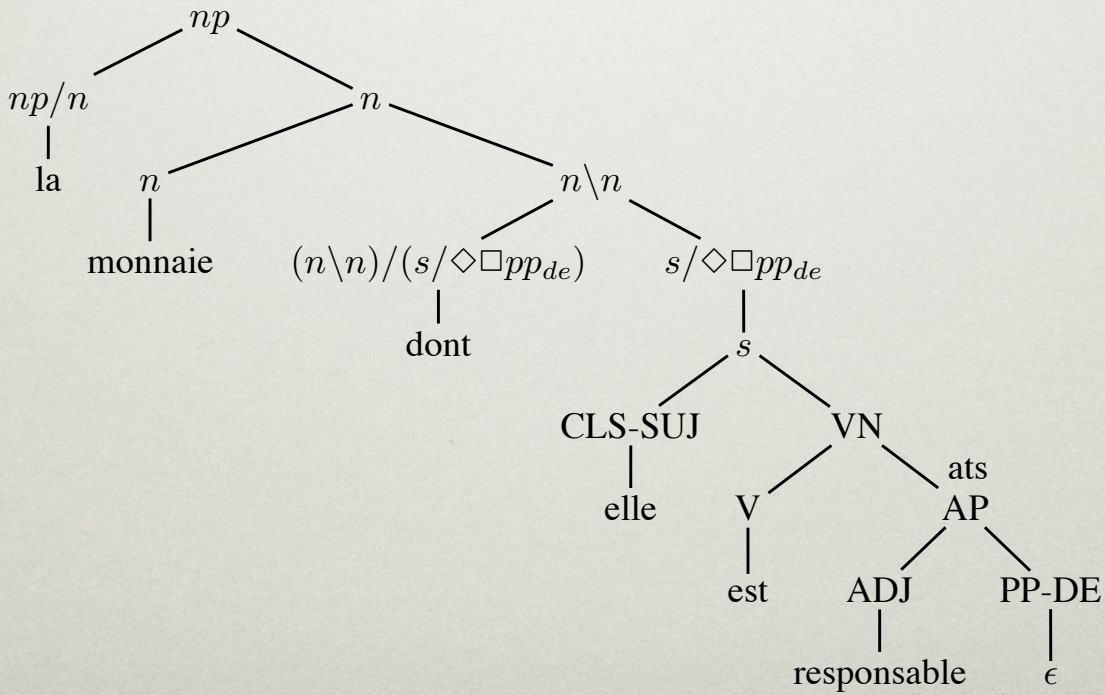
Treebank Extraction



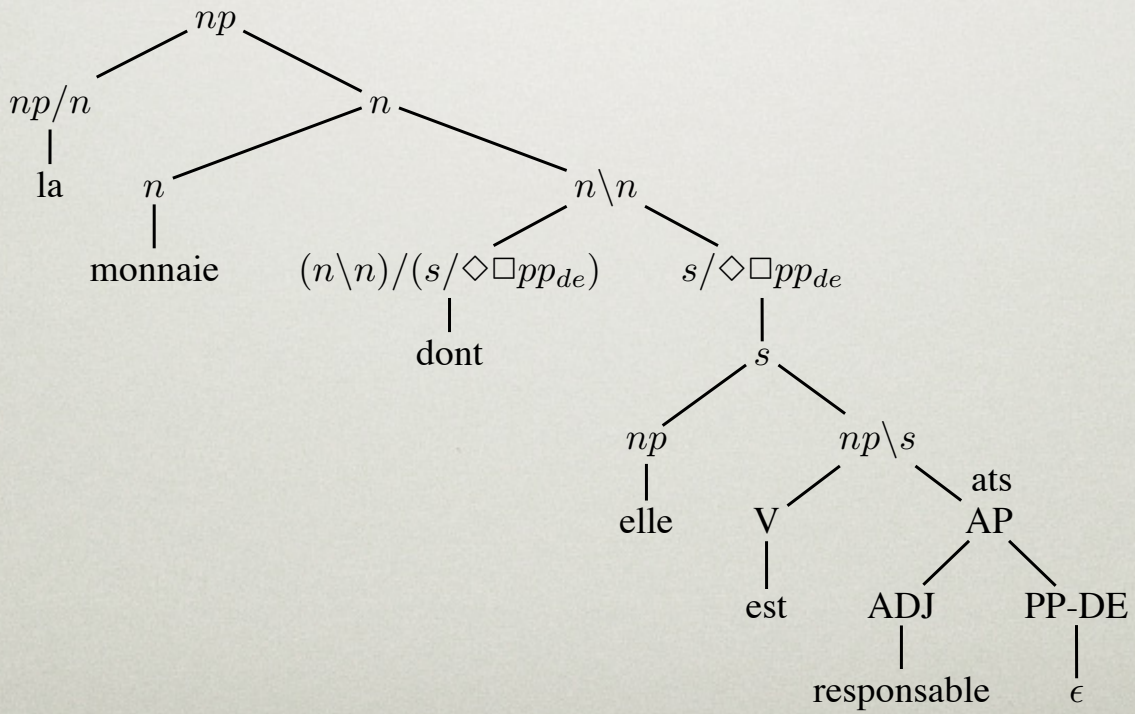
Treebank Extraction



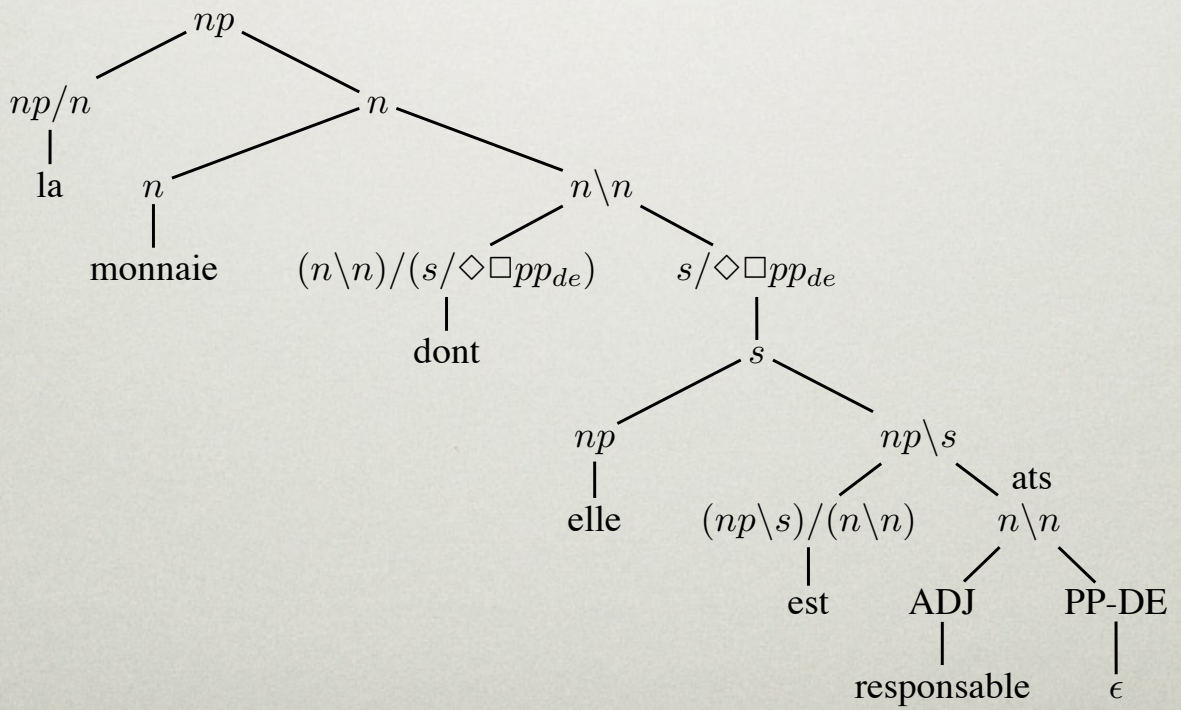
Treebank Extraction



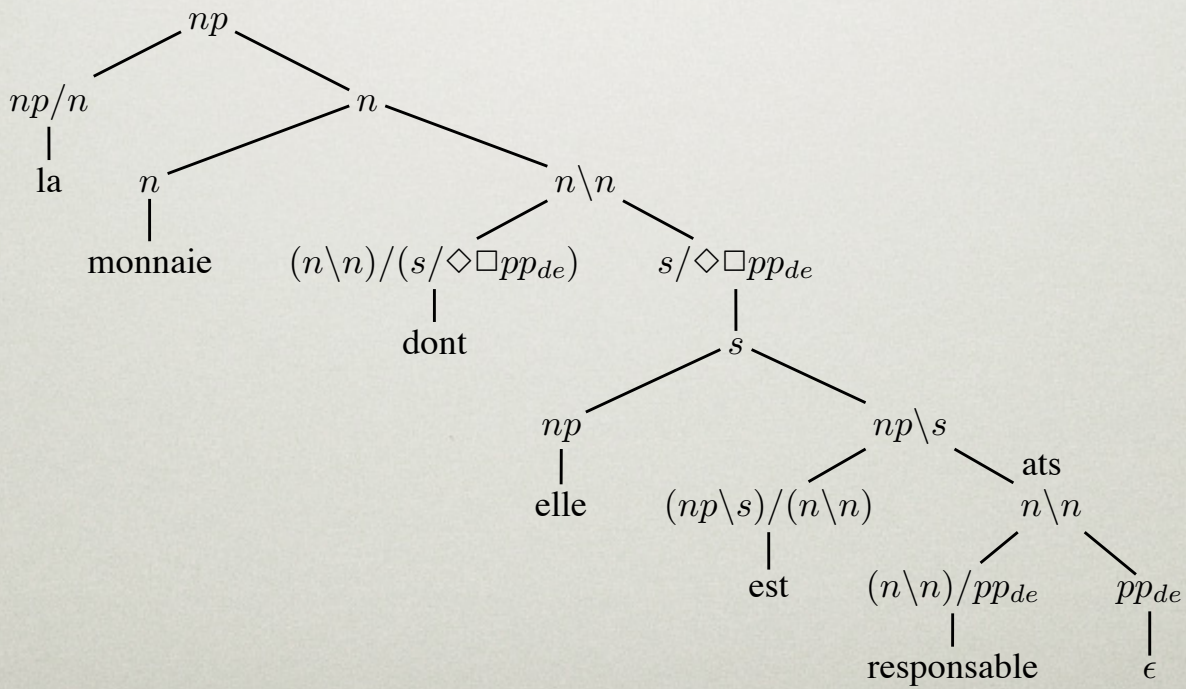
Treebank Extraction



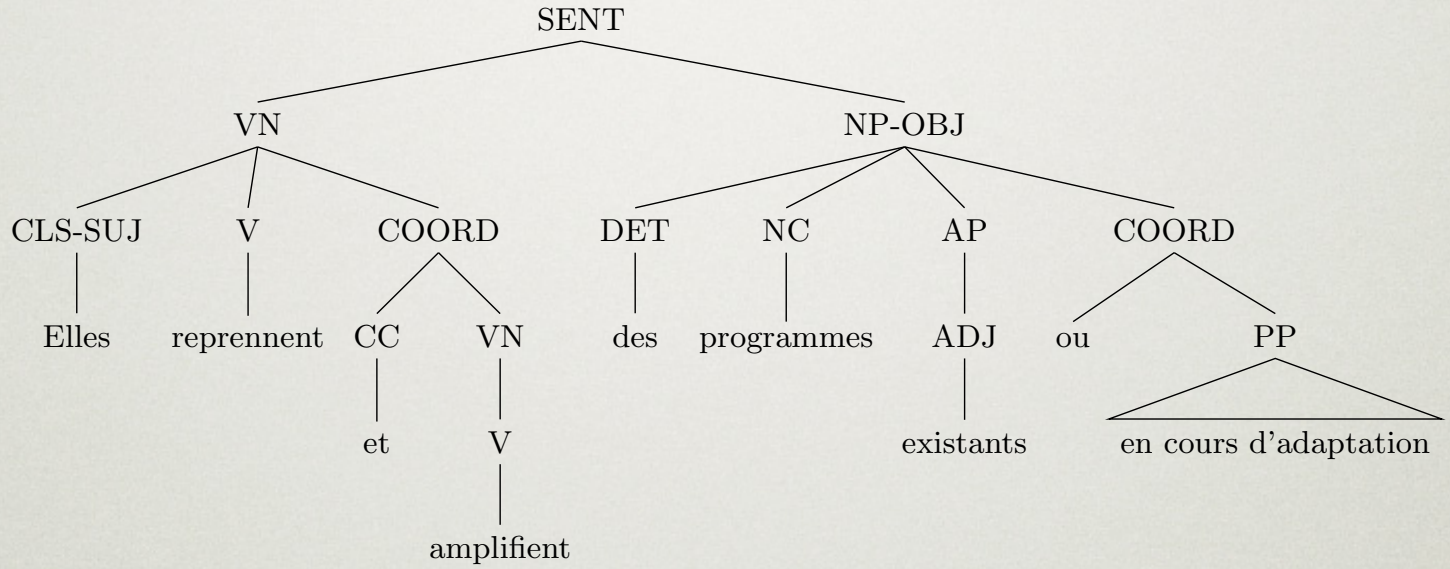
Treebank Extraction



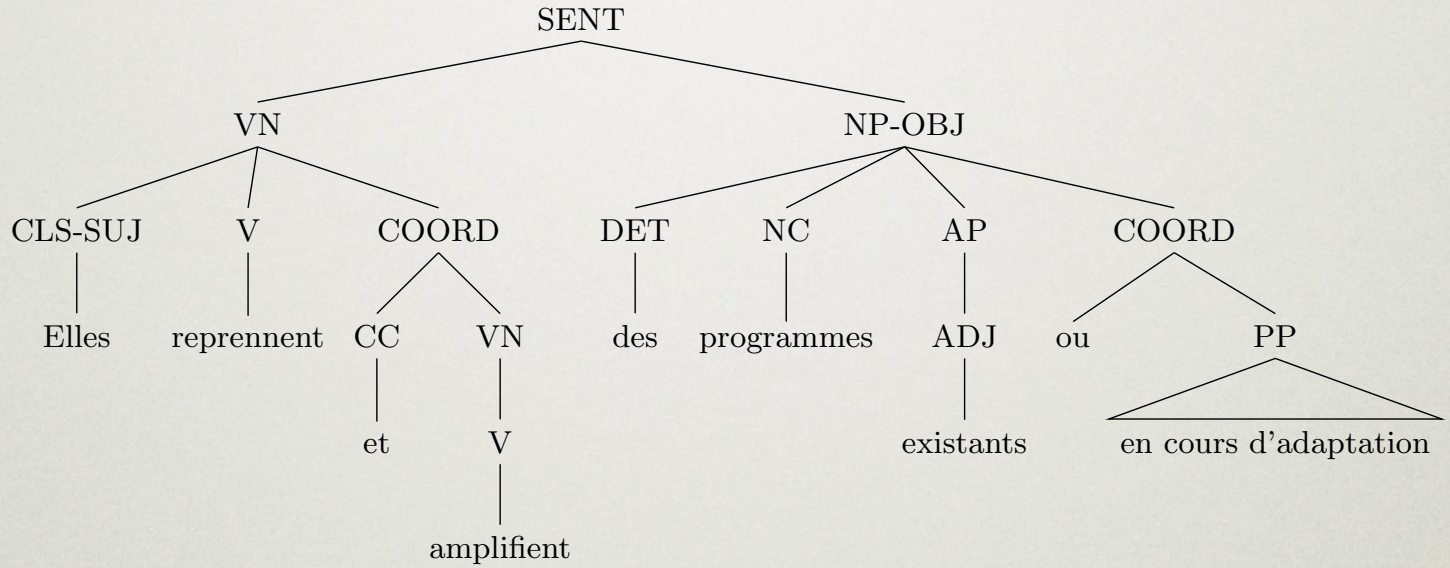
Treebank Extraction



Treebank Extraction: Coordination



Treebank Extraction: Coordination



$$\dfrac{\dfrac{\dfrac{existants}{n\backslash n}\,Lex \quad \dfrac{\dfrac{ou}{((n\backslash n)\backslash(n\backslash n))/(n\backslash n)}\,Lex \quad \dfrac{en\ cours\ldots}{n\backslash n}\,Lex}{(n\backslash n)\backslash(n\backslash n)}\,/E}{n\backslash n}\,\backslash E$$

Statistics About the Extracted French Treebank

- 15,590 sentences 445,918 words
- 43,098 distinct lexical entries
- 859 different formulas
- By comparison: 12,617 CFG rules

Note: these are the statistics after considerable cleanup: the first version had over 4,000 different formulas!

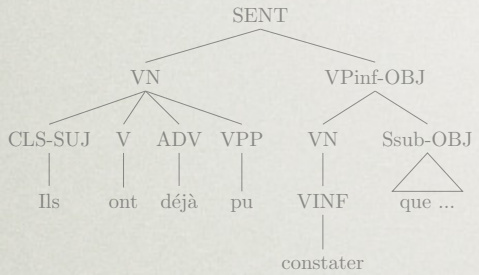
How to Improve an Extracted Treebank

Identify problem and look for clues to the underlying cause:

- annotation error
- need more sophisticated extraction
- need to add information by hand

Outline

French Treebank



Grammar Extraction

$$\dfrac{\dfrac{\text{de}}{(np\backslash s_{di})/(np\backslash s_i)}\,[Lex] \quad \dfrac{\dfrac{\text{s'}}{cl_r}\,[Lex] \quad \dfrac{\dfrac{\text{attaquer}}{(cl_r\backslash(np\backslash s_i))/pp_{\grave{a}}}\,[Lex] \quad \dfrac{}{\text{p}_0 \vdash pp_{\grave{a}}}\,[Hyp]_1}{\text{a} \circ \text{p}_0 \vdash cl_r\backslash(np\backslash s_i)}\,[/E]}{\text{s'} \circ (\text{a} \circ \text{p}_0) \vdash np\backslash s_i}\,[\backslash E]}{\text{de} \circ (\text{s'} \circ (\text{a} \circ \text{p}_0)) \vdash np\backslash s_i}\,[/E]$$

Applications

e_1 y_1	
y_1 :	e_2 e_3 x_3
	$x_3 =$ **?**
	aider_à(e_2, x_0, x_3, e_3)
	partir(e_3, x_3)
demander(e_1, y_0, x_0, y_1)	

Applications

- Wide-coverage parsing for French
- Wide-coverage semantics for French

Wide-Coverage Parsing

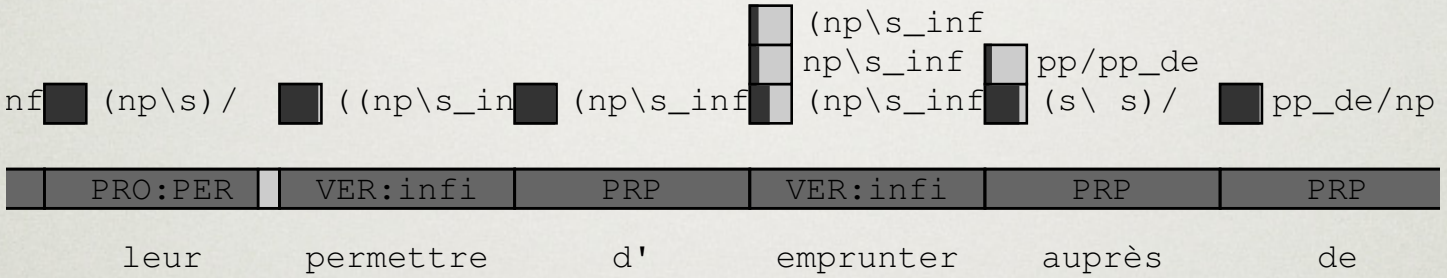
- How can we parse very big grammars efficiently?
- Bottlenecks: lexicon size, grammatical combinatorics

Lexicon Size

- Many frequent words occur with very many different formulas
- Classic solution: supertagging

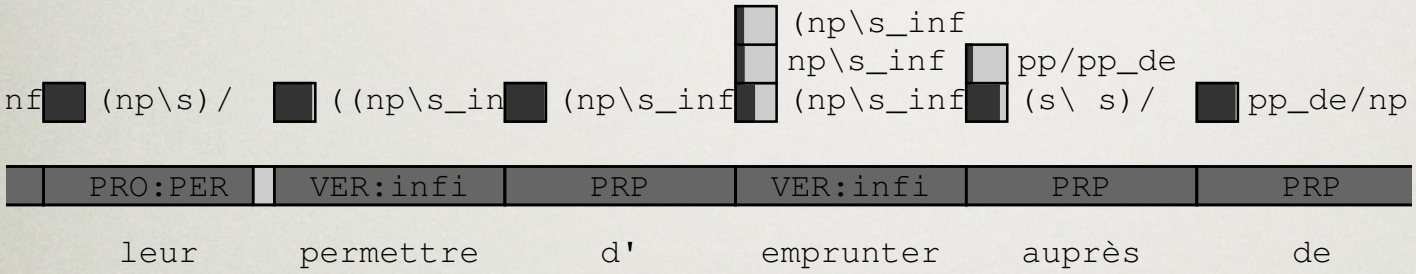
est - "is"	
$(np\backslash s)/np$	23,2 %
$(np\backslash s)/(n\backslash n)$	20,6 %
$(np\backslash s)/(np\backslash s_{pass})$	16,8 %
$(cl_r\backslash(np\backslash s))/(cl_r\backslash(np\backslash s_{ppart}))$	10,8 %
$(np\backslash s)/pp$	8,1 %
$(np\backslash s)/(np\backslash s_{ppart})$	6,3 %
$(np\backslash s)/(np\backslash s_{infX})$	2,8 %
$((np\backslash s)/s_q)/(n\backslash n)$	2,2 %

What Supertagging Does



- Supertagging ≅ statistical approximation of lexical lookup
- Assigns each word the contextually most likely (set of) formulas

What Supertagging Does



emprunter:

47.4 $(np\backslash s_{inf})/np$

20.5 $np\backslash s_{inf}$

18.8 $(np\backslash s_{inf})/pp$

- Supertagging ≅ statistical approximation of lexical lookup
- Assigns each word the contextually most likely (set of) formulas

How to POStag/ Supertag?

- Hidden Markov Model
- Maximum Entropy
- Deep learning (LSTM)

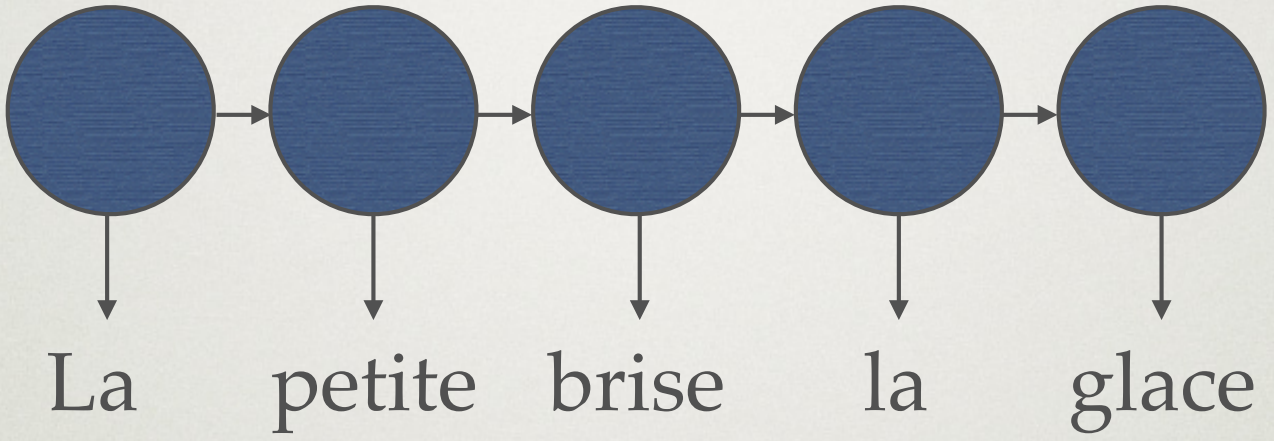
How to POStag/ Supertag?

HMM

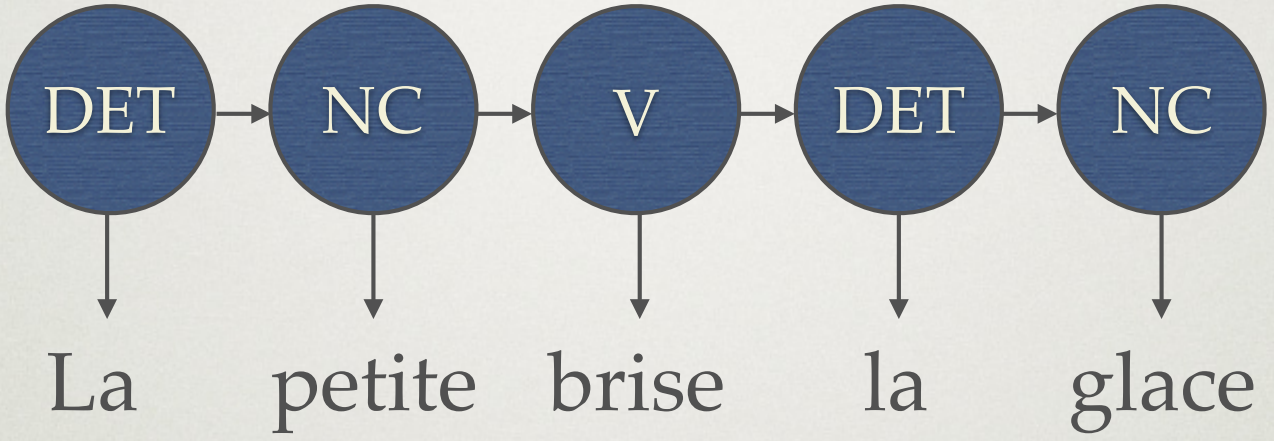
MaxEnt

LSTM

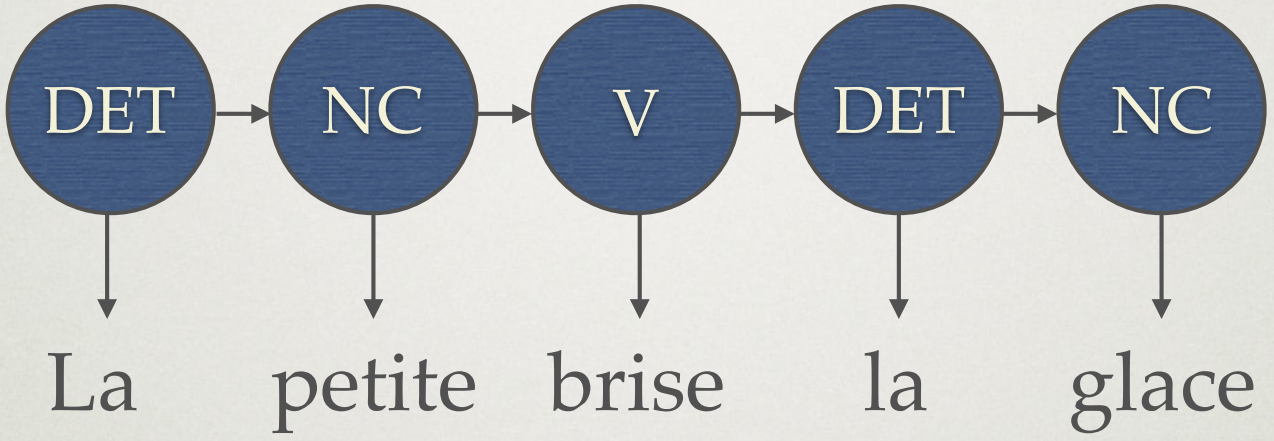
Hidden Markov Models

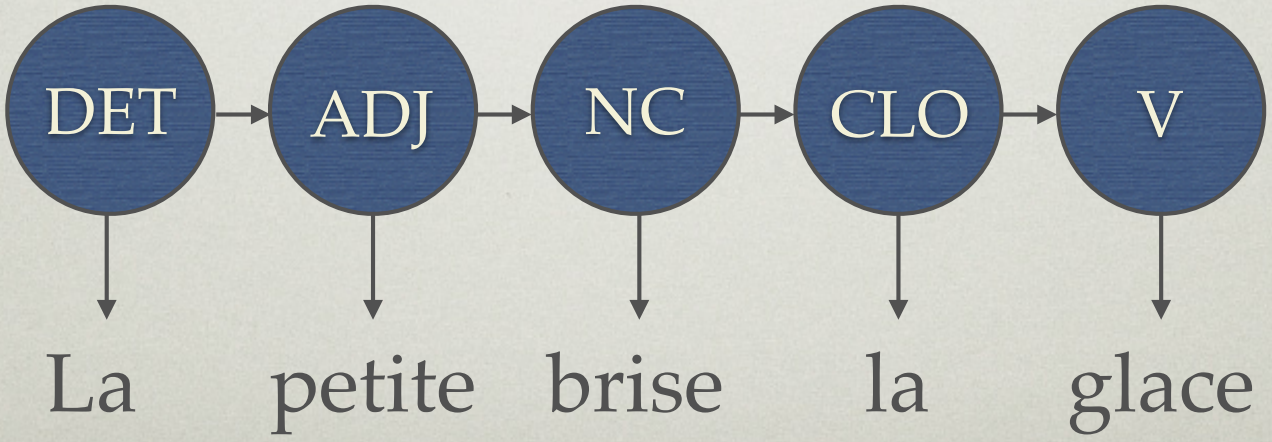


Hidden Markov Models



Hidden Markov Models





What is Maximum Entropy?

- General-purpose statistical modelling tool
- Use any "features" we like with no assumptions of statistical independence
- Slogan: assume the least possible about data we haven't seen

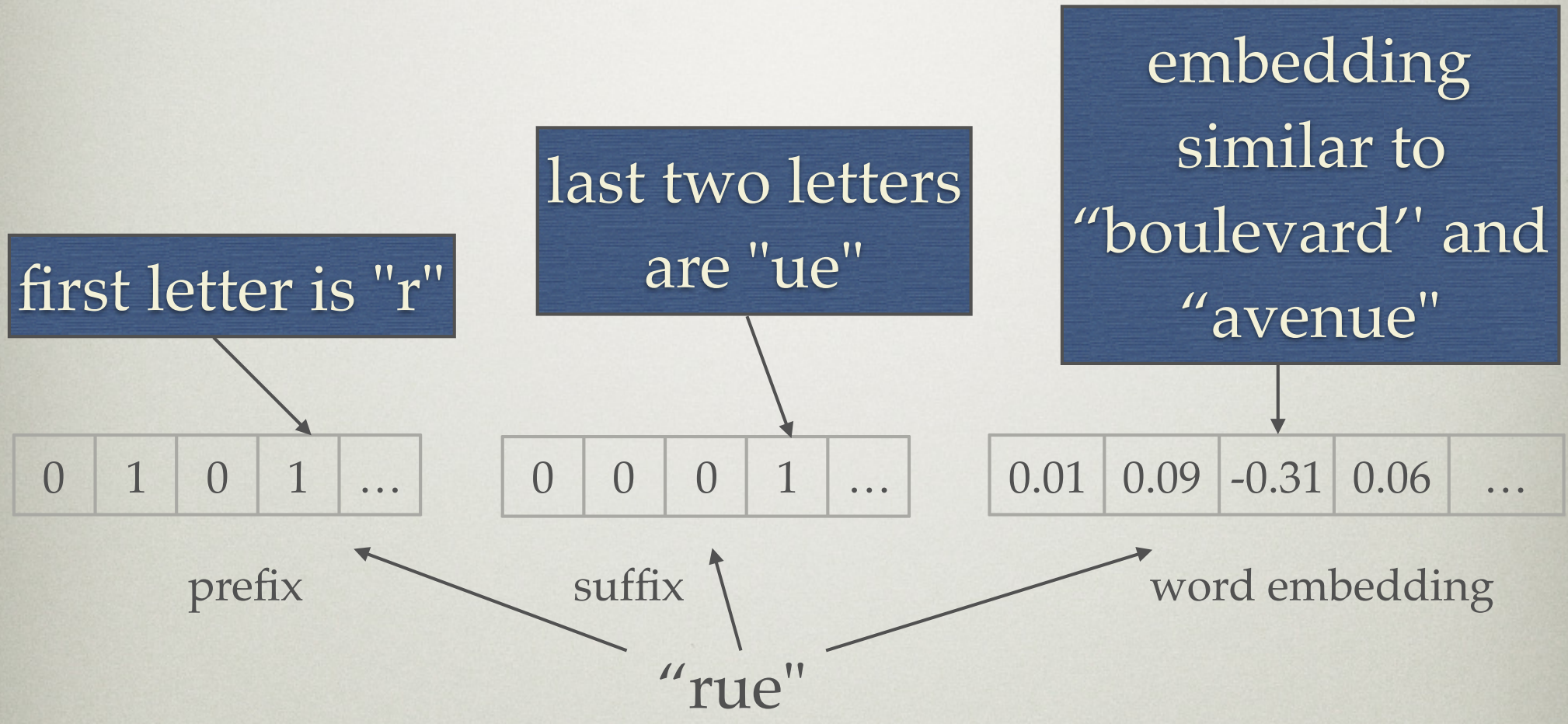
Recursive Neural Networks

- Recursive neural networks can handle long-distance dependencies in theory.
- However, in practice they do not.

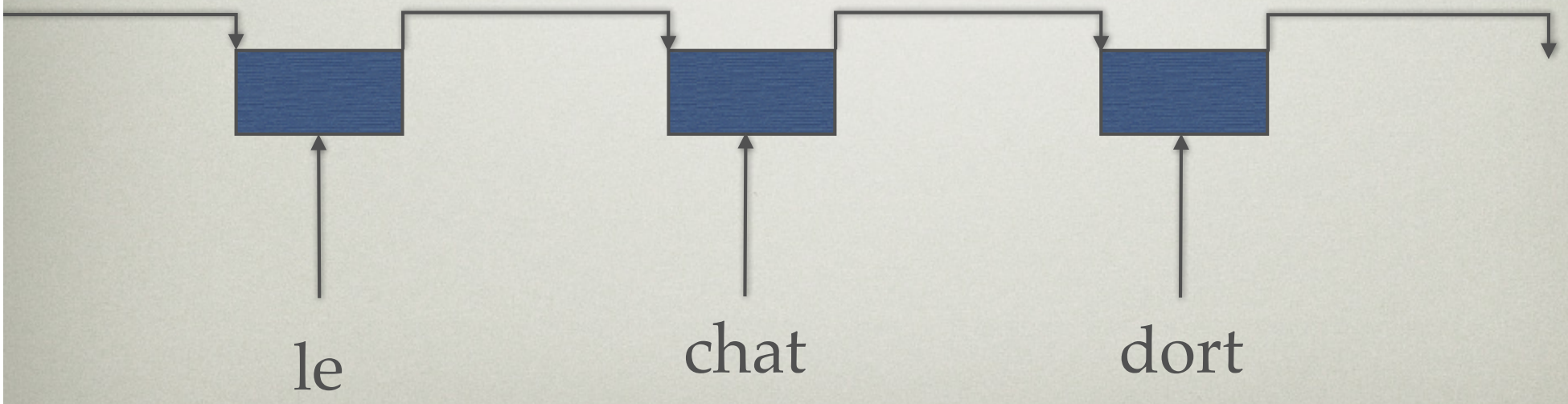
LSTM

- LSTM (long short-term memory) have an explicit memory mechanism
- There are separate network parameters for “remembering/storing” and “forgetting”
- One of the most widely used modern technologies in natural language processing

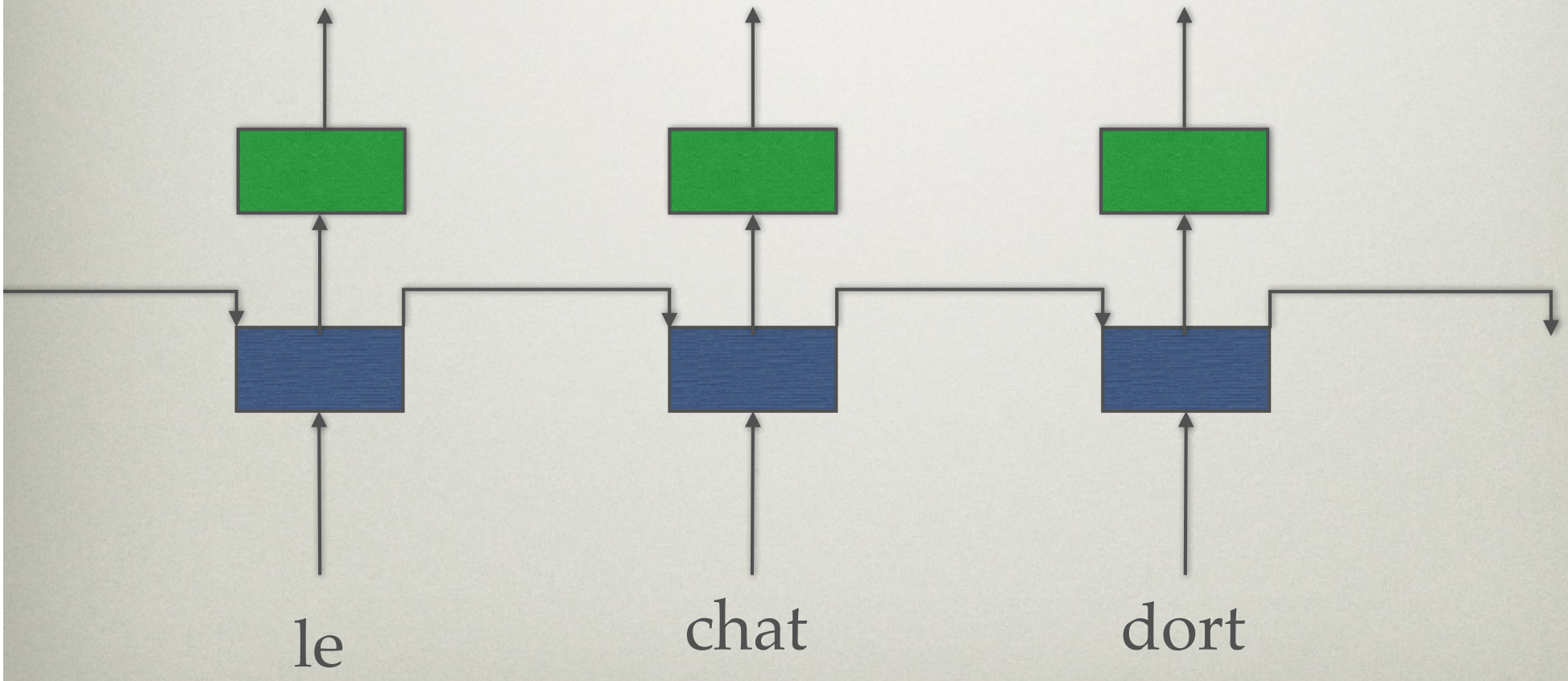
Translating words into Model Inputs



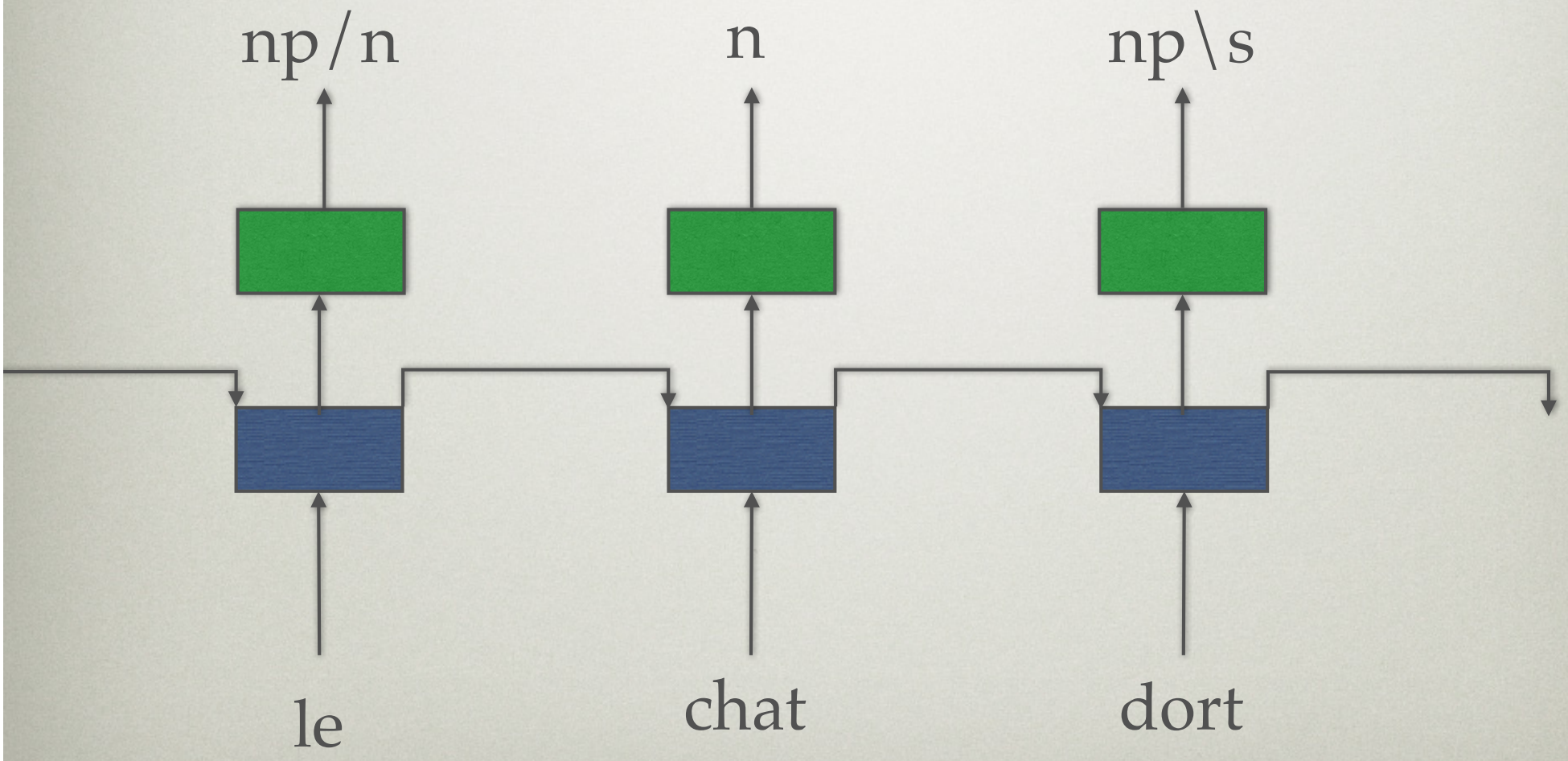
LSTM Tagging/ Supertagging



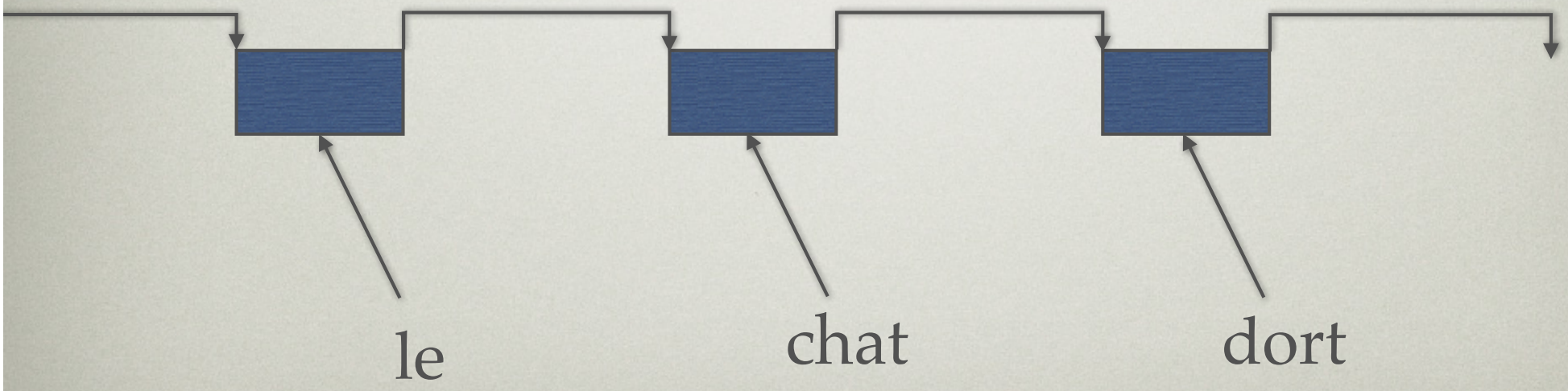
LSTM Tagging/ Supertagging



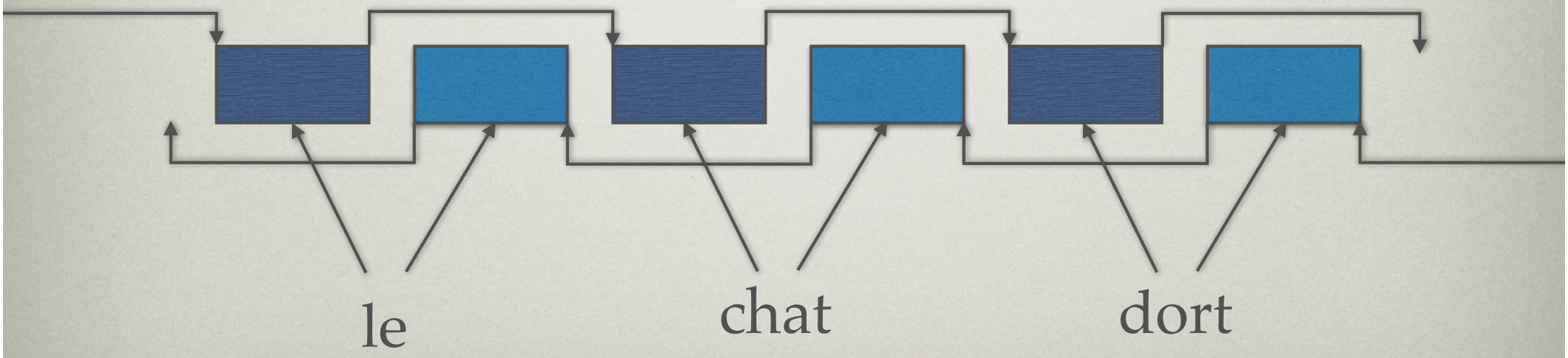
LSTM Tagging/ Supertagging



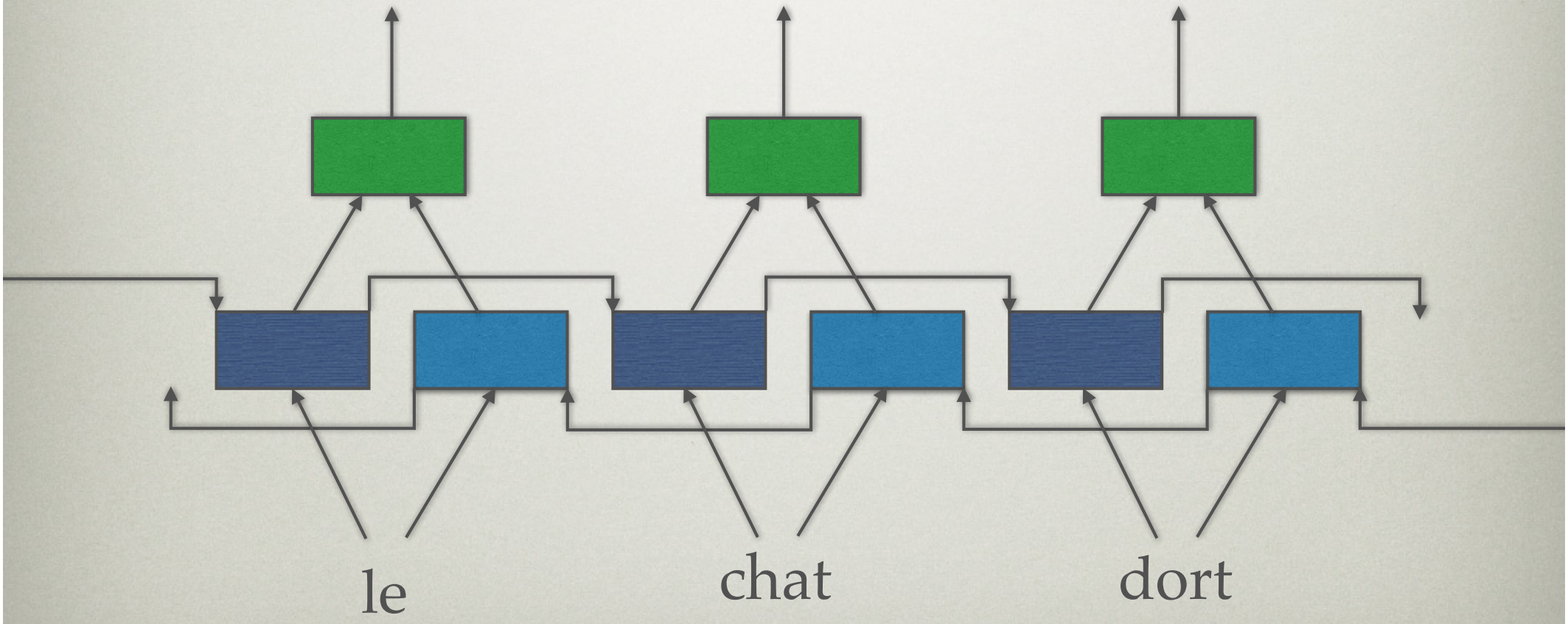
LSTM Tagging/ Supertagging



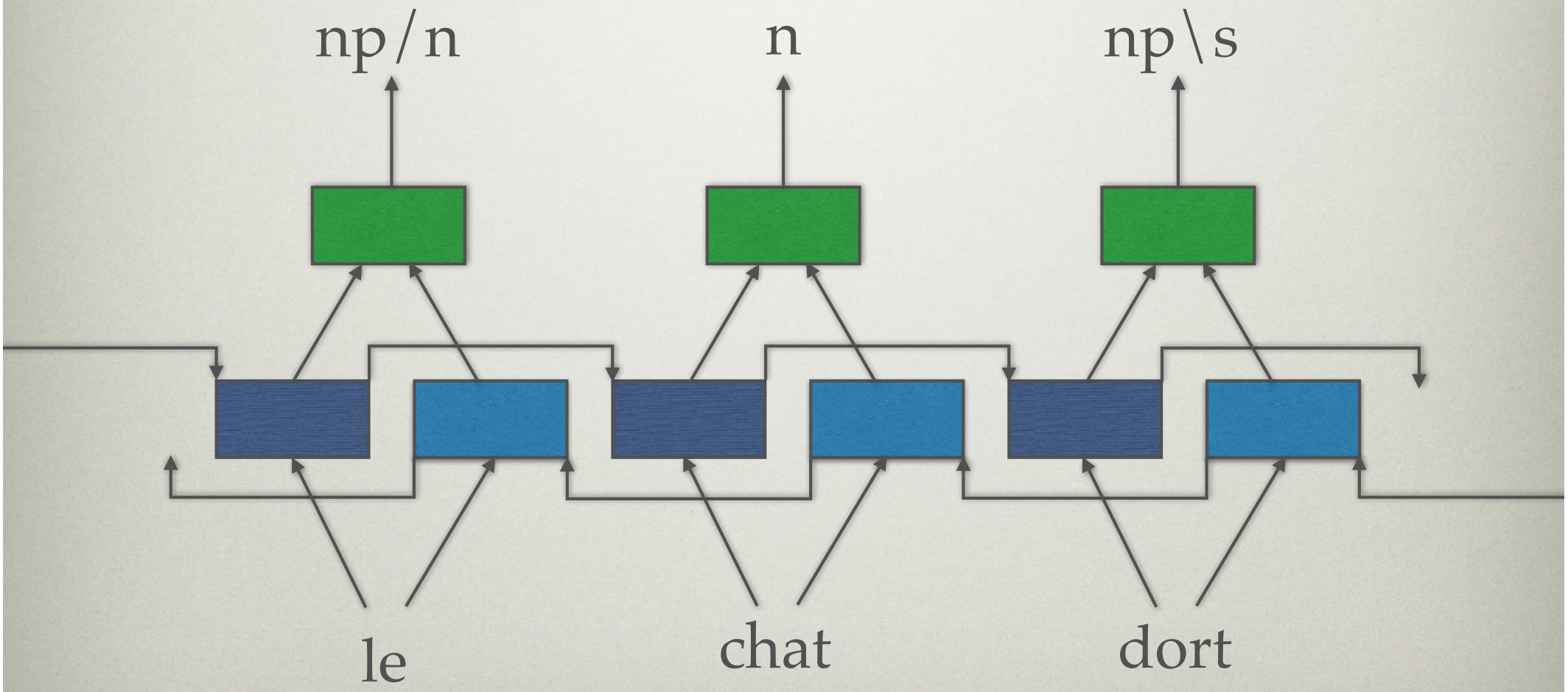
LSTM Tagging/ Supertagging



LSTM Tagging/ Supertagging



LSTM Tagging/ Supertagging



Supertagger Performance (MaxEnt)

Corpus	POS	Super	0,1	0,01	F/w
FTB	97,8 %	90,6 %	96,4 %	98,4 %	2,3

Supertagger Performance (MaxEnt)

Corpus	POS	Super	0,1	0,01	F/w
FTB	97,8 %	90,6 %	96,4 %	98,4 %	2,3
Le Monde 2010	97,3 %	89,9 %	95,8 %	97,9 %	2,2
Sequoia/Annodis	97,3 %	88,1 %	94,8 %	97,6 %	2,4
Itipy/Forbes	95,7 %	86,7 %	93,8 %	97,1 %	2,6

How good is this?

- 90.6% accuracy for the best supertag sounds good, but this is given the correct part-of-speech tag
- When combining POS-tagger with supertagger, accuracy drops to 88.7% (without POS-tagger, we end up at 86.7%, so POS-tagging helps)

Supertagger Performance

Corpus	POS	Super	0,1	0,01	0,001
MaxEnt	97,8	90,6	96,4 (1,4)	98,4 (2,3)	98,8 (4,7)
LSTM	98,4	92,2	95,8 (1,2)	97,9 (1,5)	99,0 (2,4)

with β=0,0001, we have 4,7 formulas per word (same as ME with β=0,001) but accuracy of 99.5%

Chart-Parsing Type-Logical Grammars

- Even when using a supertagger, we there are still > 2 formulas per word
- Simple enumeration is not efficient enough
- Sharing subcomputations for proof nets requires more complex proof-theoretic machinery.

Chart-Parsing Type-Logical Grammars

- Original intention: preprocessing step for proof net parser
- At some point my preprocessor single-handedly parsed 97-98% of my extracted French treebank
- It seemed reasonable to add some extra chart rules to handle the rest.

Chart Parsing

- Categorial grammar parsing in the 90s has studied chart parsing for the Lambek calculus (and some variants)
- Chart parsing for the Lambek calculus turned out to be fairly complicated.
- An important source of complexity is the interaction of hypothetical formulas. Tiede (2001) showed that Lambek calculus proofs are not regular tree languages.

Chart-Parsing Type-Logical Grammars

- Prototype exhaustive chart parser for Type-Logical Grammars, strongly inspired by Shieber e.a. (1995)
- Fine-tuned for grammars in the restricted form as produced by the extraction algorithm
- LaTeX output

Chart-Parsing Type-Logical Grammars

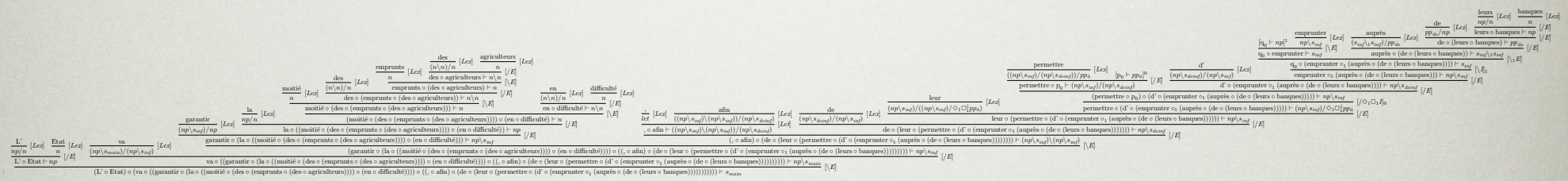


Chart-Parsing Type-Logical Grammars

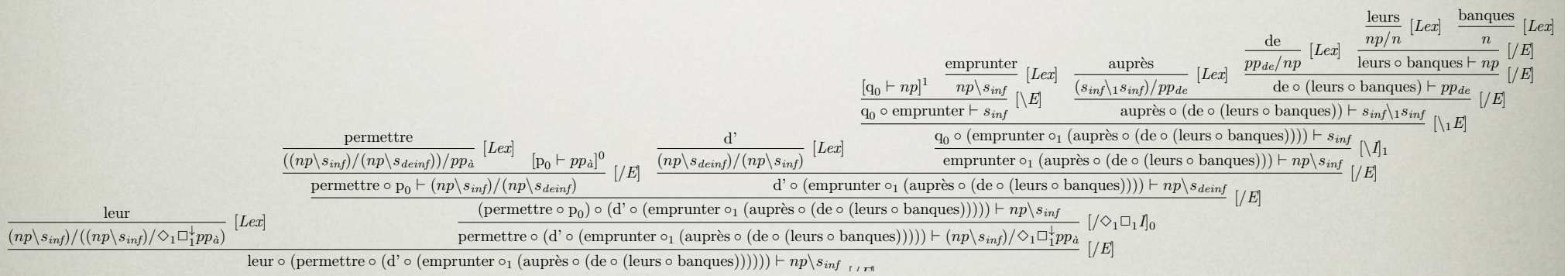
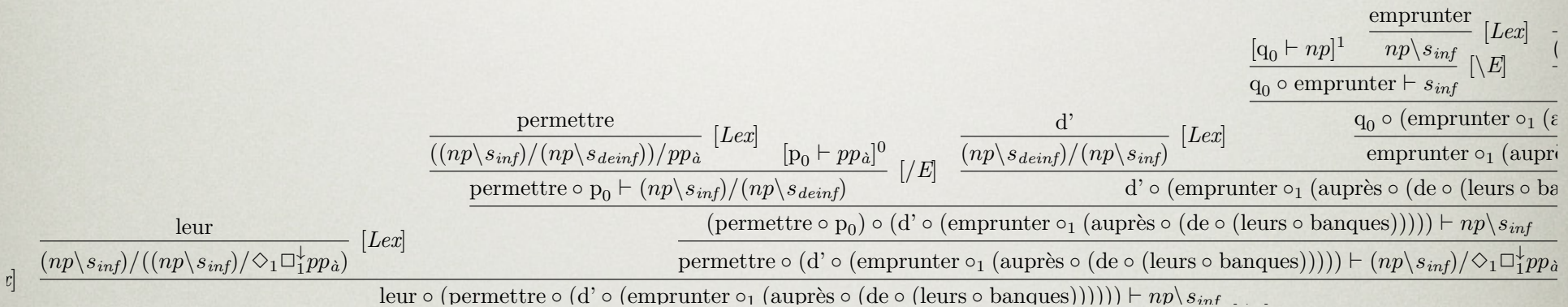


Chart-Parsing Type-Logical Grammars



Semantics

- Type-logical grammar proofs are a subset of intuitionistic proofs, which correspond to lambda-terms.
- Lexical substitution follow by beta normalization gives us the full sentence meaning

Semantics

- Montague-style semantics, where the meaning of *love* is **love'**
- But which has the advantage of being scaleable, since many lexical entries follow a specific pattern
- Uses DRT

Semantics

Example entries (slightly simplified)

marché: $\lambda x.$ [; **marché(x)**]

Marie: $\lambda P.$ [**y** ; **nommé(y,Marie)**] $\oplus$ **(P y)**

chaque: $\lambda P\ \lambda Q.($ [**z** ;] $\oplus$ **(P z))**$\rightarrow$ **(Q z)**

Semantics

Example entries (slightly simplified)

aime:	(np\s)/np	λy. λx.	
			aimer(x,y)
semble:	(np\s)/(np\s_{inf})	λP. λx.	s
			sembler(s) s: (P x)
veut:	(np\s)/(np\s_{inf})	λP. λx.	s
			vouloir(x,s) s: (P x)

Semantics

- 692 words in the lexicon, with idiosyncratic properties
- 337 lexical schemata, eg. (np\s)/np for word w means semantics w' ($\equiv \lambda y.\lambda x.$ $w'(x,y)$)

Future Work

- Connect word meaning to nodes in JeuxDeMots
- Add logical entailment component, possibly à la Natural Logic
- Add multiword units component (eg. “être en train de”, but also “plus X que Y”)

Conclusions

- Type-Logical Grammars factorize grammars in a way permitting an easy connection to formal semantics,
- this ensures a direct link between wide-coverage parsing and wide-coverage *semantics*.