Towards Wide-Coverage Semantics for French

Richard Moot

LaBRI (CNRS), SIGNES (INRIA) & U. Bordeaux

Introduction

Bridge between statistical NLP and syntax/ semantics the way I (and many people here) like it!
Don't worry, this will not be a talk of the style I improved on task X from Y% to Y+. 2%
There will be some percentages, but just to show we are up to the level of some of the statistical NLP guys.

- Many wide-coverage parsers for French exist (witness the participation of the Easy and Passage campaigns)
- My goal is not directly to compete with them, but to move towards a wide-coverage parser which produces structures which are more interesting (at least to me!) than shared forests

Introduction

- I will talk about my current research on a wide-coverage categorial grammar for French.
- I will start by giving a short introduction to categorial grammars, showing how a categorial parse corresponds to a lambda-term in the simply typed lambda calculus.
- Since the work of Montague, we know that the simply typed lambda calculus forms a solid base for the semantic analysis of fragments of natural language.

Introduction

- However, we are by no means limited to Montague semantics: Muskens (1994) and de Groote (2006) show that the semantics of categorial grammars are compatible with modern theories of dynamic semantics (DRT in the case of Muskens, and a continuation-based approach in the case of de Groote)

Introduction

- In this talk I will present the Grail parser and the development of a wide-coverage grammar of French as well as the development of a prototype semantic lexicon producing DRSs, highlighting the treatment of presupposition.
- Wide-coverage semantics in this sense is a relatively new field, which was pioneered for English by Bos e.a. (2004)

Overview

- Categorial Grammars and Lambda Calculus Semantics
- Grammar Extraction
  - converting a corpus into categorial grammar
  - how to use this grammar for parsing
- Wide-Coverage Semantics

Categorial Grammars

Lexicalized grammars and lambda calculus semantics

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 "/"

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

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

Categorial Grammars

Example

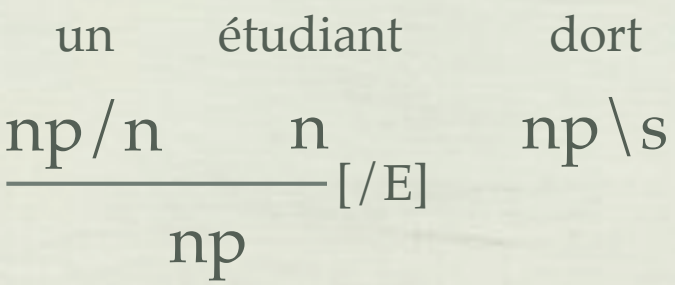
un	étudiant	dort
np/n	n	np\s

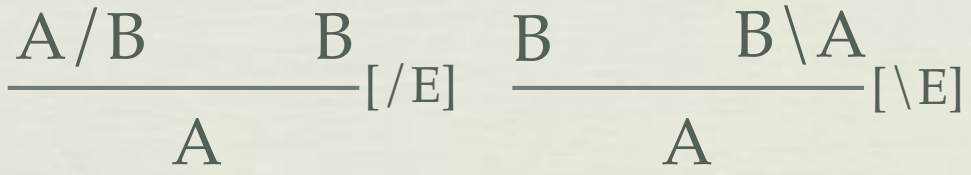
$$\frac{A/B \qquad B}{A}\,[/E] \qquad \frac{B \qquad B\backslash A}{A}\,[\backslash E]$$

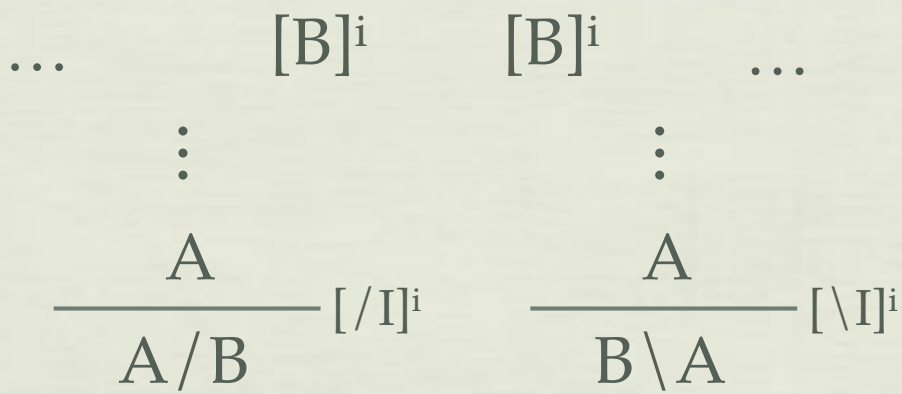
$$\frac{\begin{array}{c}\ldots \qquad [B]^i \\ \vdots \\ A\end{array}}{A/B}\,[/I]^i \qquad \frac{\begin{array}{c}[B]^i \qquad \ldots \\ \vdots \\ A\end{array}}{B\backslash A}\,[\backslash I]^i$$

Categorial Grammars

Example

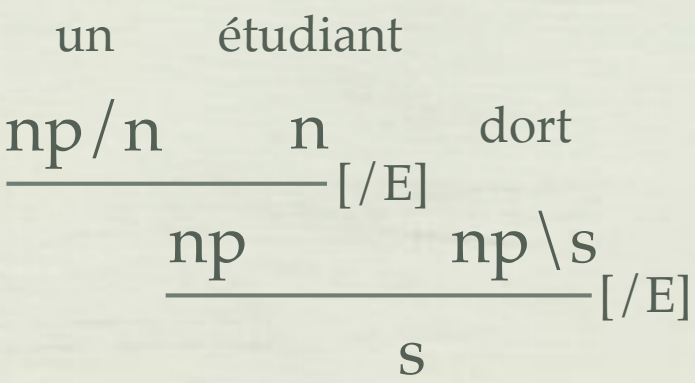


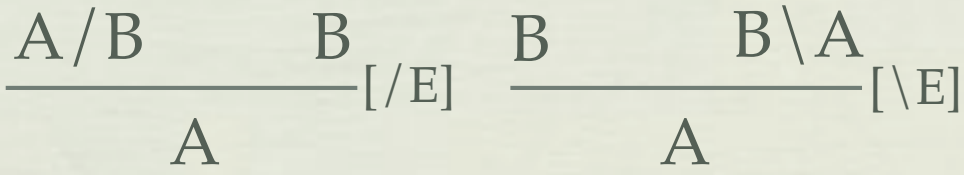


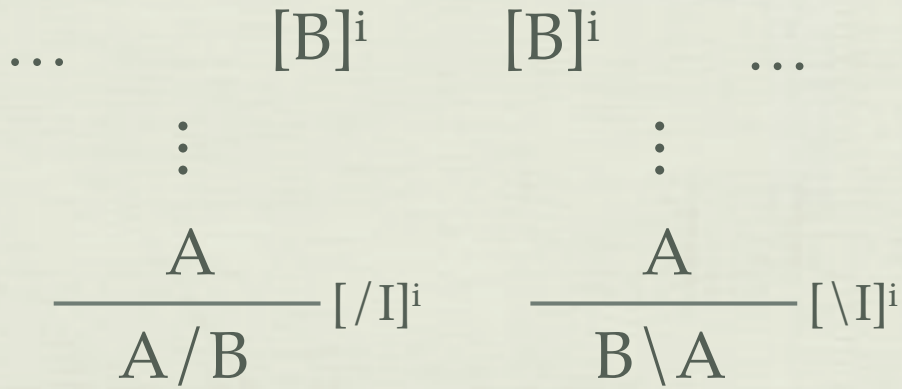


Categorial Grammars

Example







Lamda calculus

Beta reduction

$$(\lambda x.t)\ u\ = t[x:=u]$$

that is t, but with all occurrences of the variable x replaced by the term u
(renaming variables when necessary)

- $(\lambda x.\ x+x)\ \ 2 = 2+2$
- $(\lambda y.(y\ x))\ (\lambda z.\ (f\ z)) = (\lambda z.\ (f\ z))\ x = f\ x$
- $(\lambda y.(y\ x))\ (\lambda z.\ (f\ z)) = (\lambda z.\ (f\ z))\ x = f\ x$

Lamda calculus
Types

Inductive Definition

- Basic types e (for entity) and t (for truth value)
- If α and β are types, then $\alpha \rightarrow \beta$ is a type

Lamda calculus

Terms

- For each type α, there is a (countably infinite) number of variables x, y, z, ... which are terms of type α.
- For each type α, there is a (countably infinite) number of constants a, b, c, ... which are terms of type α.
- If x is term of type $\alpha \rightarrow \beta$ and y is a term of type α then (x y) is a term of type β.
- If y is a term of type β and x is a variable of type α, then $\lambda x.y$ is a term of type $\alpha \rightarrow \beta$

Lamda calculus
Terms

- Some useful constants

$\wedge, \vee, \rightarrow$	$t \rightarrow (t \rightarrow t)$
$\forall, \exists$	$(e \rightarrow t) \rightarrow t$
ι	$(e \rightarrow t) \rightarrow e$

Lamda calculus
Notational conventions

- These notational conventions are useful for people familiar with predicate logic and Montague grammar.
- We will write $(x \wedge y)$ instead of $((\wedge\ y)\ x)$ and adopt a similar convention for the other boolean connectives
- We will write $\iota x.t$ instead of $\iota(\lambda x.t)$ and adopt a similar convention for $\forall$ and $\exists$
- We will write $p(x,y)$ instead of $((p\ y)\ x)$ for constants p (in general $p(x_1,...,x_n)$ for $(...(p\ x_n)...x_1)$

Lamda calculus
Notational conventions

- $\lambda z.\lambda y.\lambda x.(((\mathrm{f}\ z)\ y)\ x) \equiv\ \lambda z.\lambda y.\lambda x.\mathrm{f}(x,y,z)$
- $\forall(\lambda x.\exists(\lambda y.((\mathrm{love}\ y)\ x)) \equiv \forall x.\exists y.\mathrm{love}(x,y)$
- $\forall(\lambda x.(\rightarrow(\mathrm{man}\ x))\ \mathrm{sleep}\ x)) \equiv \forall x.\mathrm{man}(x) \rightarrow \mathrm{sleep}(x)$

Lamda calculus

Formulas as Types

- This is the s… f assigning ty… s in categoria…
- Exception 1: np is lifted from e to (e→t)→t
- Exception 2: pp is assigned the same type as np

In many of the examples, I will assign e to both np and vp, I will leave the changes required to turn these into terms of type (e->t)->t as an exercise

Formula	Type
type(np)	(e→t)→t
type(pp)	(e→t)→t
type(n)	e→t
type(s)	t
type(B/A)	type(A) → type(B)
type(A\B)	type(A) → type(B)

Lamda calculus

Formulas as Types

- Exception 1: from e to (e—

- Exception 2: the same type as np

- To simplify the examples which follow, I will use e as the type of np and pp in the examples (exercise: give the correct lambda terms)

In many of the examples, I will assign e to both np and vp, I will leave the changes required to turn these into terms of type (e->t)->t as an exercise

Formula	Type
type(np)	e
type(pp)	e
type(n)	e→t
type(s)	t
type(B/A)	type(A) → type(B)
type(A \ B)	type(A) → type(B)

Lamda calculus

Proofs as terms

- Proofs in categorial grammar correspond to lambda terms
- These lambda terms abstract away from the directions of the implications.

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

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

Lamda calculus

Proofs as terms

- Type lifting is now a simple consequence of the logical rules.

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

$$x:np$$

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

Lamda calculus

Proofs as terms

- Type lifting is now a simple consequence of the logical rules.

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

$$\frac{x{:}np \qquad y{:}(np\backslash s)}{(y\ x){:}s}$$

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

Lamda calculus

Proofs as terms

- Type lifting is now a simple consequence of the logical rules.

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

$$\frac{x{:}np \qquad [y{:}(np\backslash s)]}{\dfrac{(y\ x){:}s}{\lambda y.(y\ x){:}s/(np\backslash s)}}$$

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

Traces

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

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

$$\dfrac{\text{que}\quad \dfrac{(n\backslash n)/(s/\Diamond\Box np)}{\qquad}\quad \dfrac{\dfrac{s}{s/\Diamond\Box np}\,[/I]}{\qquad}\,[/E]}{n\backslash n}$$

$$\dfrac{\text{l'Iran et l'Arabie saoudite}\quad np \qquad \dfrac{\text{chercheront à}\quad (np\backslash s)/(np\backslash s_{inf}) \qquad \dfrac{\text{sauvegarder}\quad (np\backslash s_{inf})/np \quad [np]}{\qquad}\,[/E]}{\qquad}\,[/E]}{np\backslash s}\,[/E]$$

Traces

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

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

que
$(n\backslash n)/(s/\Diamond\Box np)$

l'Iran et
l'Arabie saoudite
np

chercheront à
$(np\backslash s)/(np\backslash s_{inf})$

$$\frac{\overset{\text{sauvegarder}}{(np\backslash s_{inf})/np} \qquad \overset{\varepsilon}{np}}{np\backslash s_{inf}}\,[/E]$$

Traces

… 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 : $(n\backslash n)/(s/\Diamond\Box np)$

$$\dfrac{\text{l'Iran et l'Arabie saoudite} : np \qquad \dfrac{\text{chercheront à} : (np\backslash s)/(np\backslash s_{inf}) \qquad \dfrac{\text{sauvegarder} : (np\backslash s_{inf})/np \qquad \varepsilon : np}{np\backslash s_{inf}}[/E]}{np\backslash s}[/E]$$

Traces

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

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

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

$$\dfrac{\text{l'Iran et l'Arabie saoudite} : np \qquad \dfrac{\text{chercheront à} : (np\backslash s)/(np\backslash s_{inf}) \qquad \dfrac{\text{sauvegarder} : (np\backslash s_{inf})/np \qquad \varepsilon : np}{np\backslash s_{inf}}[/E]}{np\backslash s}[/E]}{s}$$

Traces

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

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

$$\dfrac{\text{que} \quad \dfrac{\dfrac{\text{l'Iran et l'Arabie saoudite} \quad \text{np} \qquad \dfrac{\text{chercheront à} \quad (np\backslash s)/(np\backslash s_{inf}) \qquad \dfrac{\text{sauvegarder} \quad (np\backslash s_{inf})/np \qquad \dfrac{\varepsilon \quad [np]^1}{}[/E]}{np\backslash s_{inf}}[/E]}{np\backslash s}[/E]}{s}[/I]^1}{(n\backslash n)/(s/\Diamond\Box np)}$$

Traces

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

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

$$\dfrac{\text{que} \qquad \dfrac{\dfrac{\text{l'Iran et l'Arabie saoudite} \quad \text{np} \qquad \dfrac{\text{chercheront à} \quad (np\backslash s)/(np\backslash s_{inf}) \qquad \dfrac{\text{sauvegarder} \quad (np\backslash s_{inf})/np \qquad \text{ε} \quad [np]^1}{np\backslash s_{inf}}[/E]}{np\backslash s}[/E] \qquad \dfrac{s}{s/\Diamond\Box np}[/I]^1}{n\backslash n}[/E] \qquad (n\backslash n)/(s/\Diamond\Box np)}{}$$

Traces

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

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

$\lambda x.\mathit{équilibre}(x) \wedge \mathit{délicat}(x) \wedge \mathit{chercheront\_à\_sauvegarder}(I\&As, x)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\text{que}}}{(n\backslash n)/(s/\Diamond\Box np)} \quad \dfrac{\dfrac{}{s}[/I]^1}{s/\Diamond\Box np}[/E]}{n\backslash n}}{np} \quad \dfrac{\dfrac{}{\text{chercheront à}}}{(np\backslash s)/(np\backslash s_{inf})} \quad \dfrac{\dfrac{\dfrac{}{\text{sauvegarder}}}{(np\backslash s_{inf})/np} \quad \dfrac{}{\varepsilon}}{np}[/E]^1}{np\backslash s_{inf}}[/E]}{np\backslash s}[/E]}{s}[/E]$$

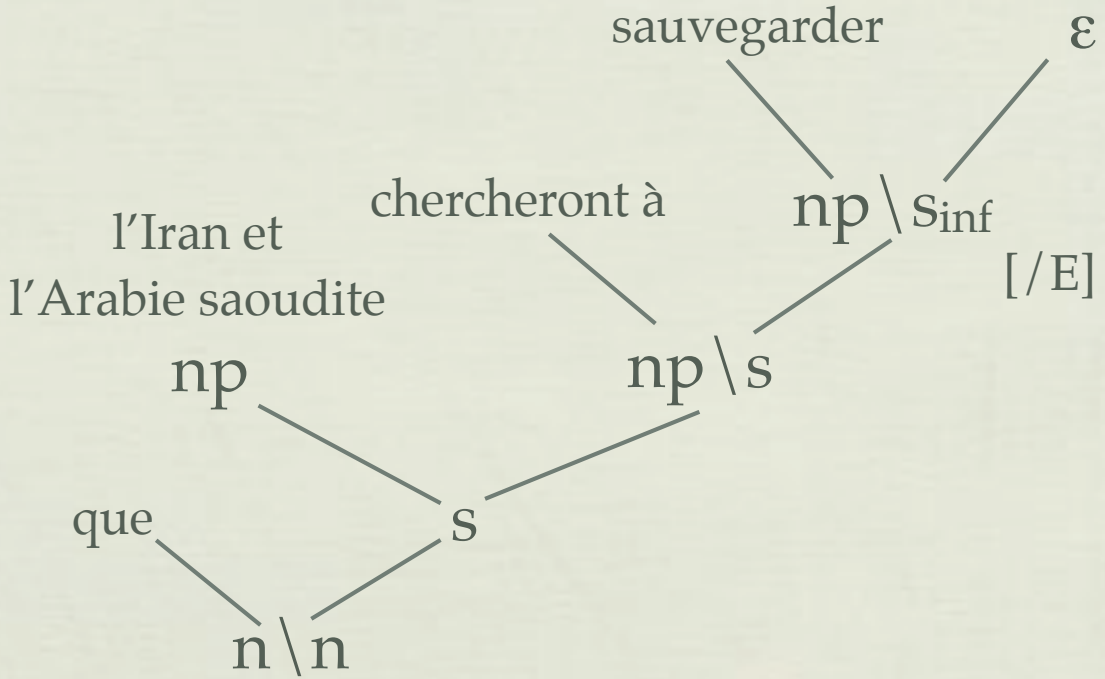
l'Iran et l'Arabie saoudite

Traces

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

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

$$\lambda x.\mathit{équilibre}(x) \wedge \mathit{délicat}(x) \wedge \mathit{chercheront\_à\_sauvegarder}(I\&As,x)$$



Grammar Extraction

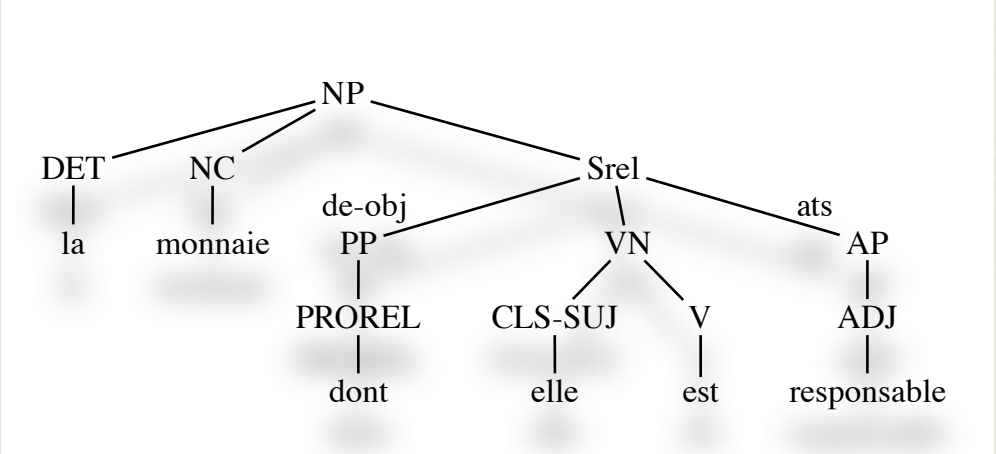
From the Paris VII corpus to a categorial lexicon, while developing several taggers

Grammar Extraction

- Grammar extraction is the conversion of a linguistically annotated corpus (in our case, the Paris VII treebank) into a grammar into a grammar formalism the people doing the conversion really like (in our case, categorial grammar)

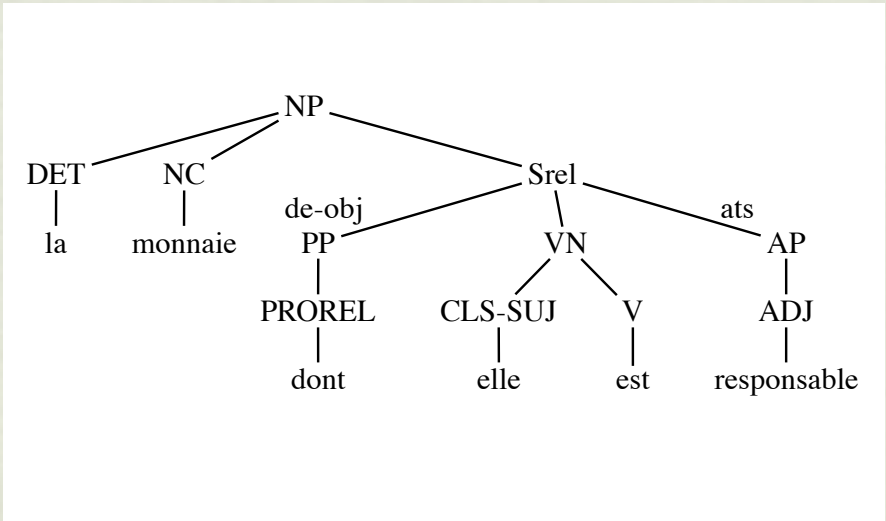
The Paris VII Corpus

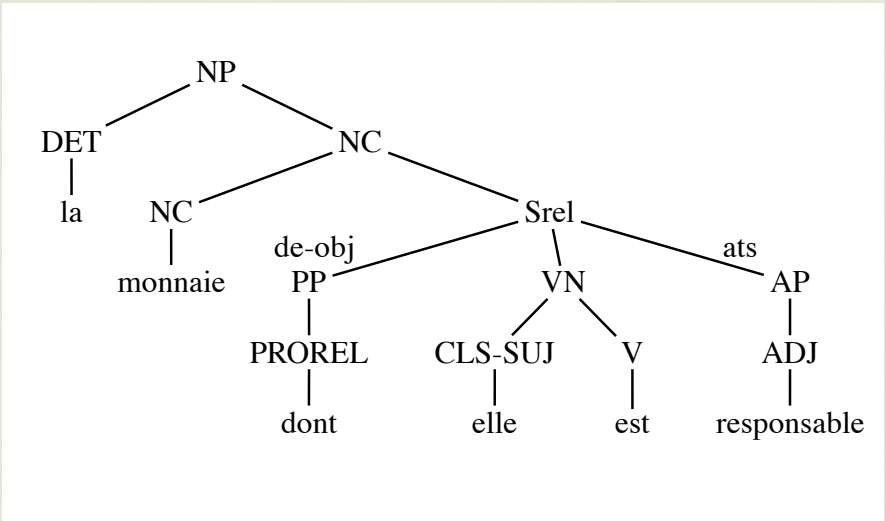
- To the right is a small sentence fragment of the Paris VII corpus, which suffices to illustrate the extraction procedure



The extraction algorithm

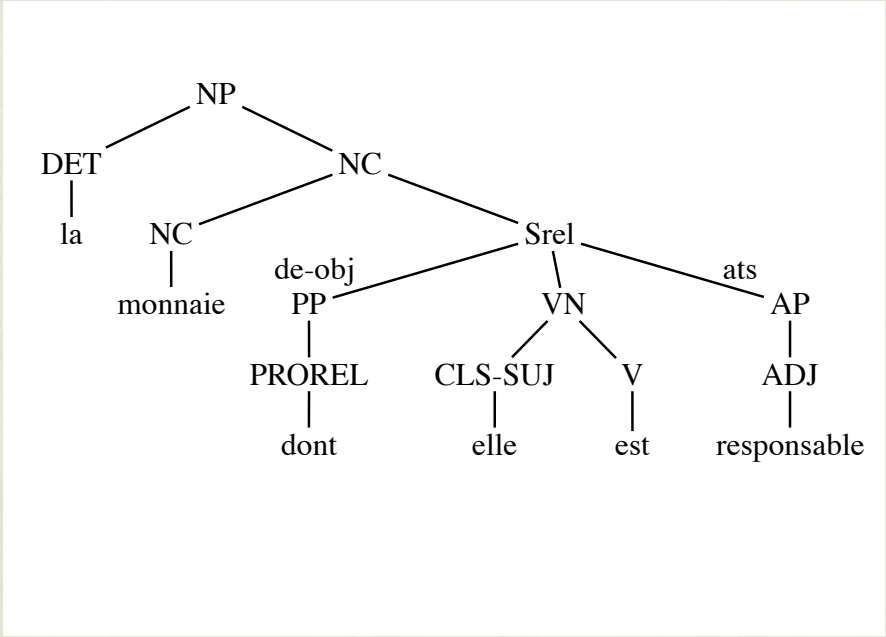
1. Binarize the annotation

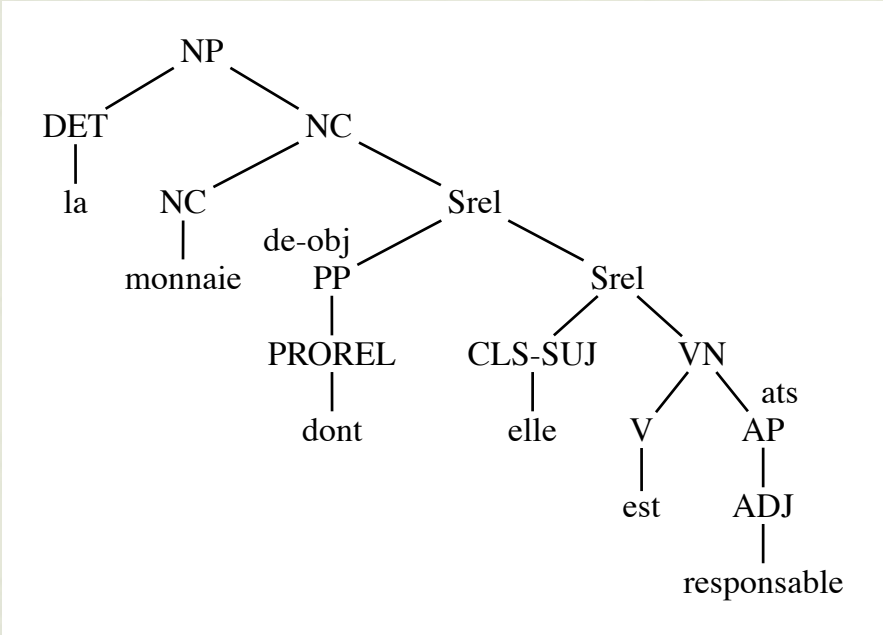




The extraction algorithm

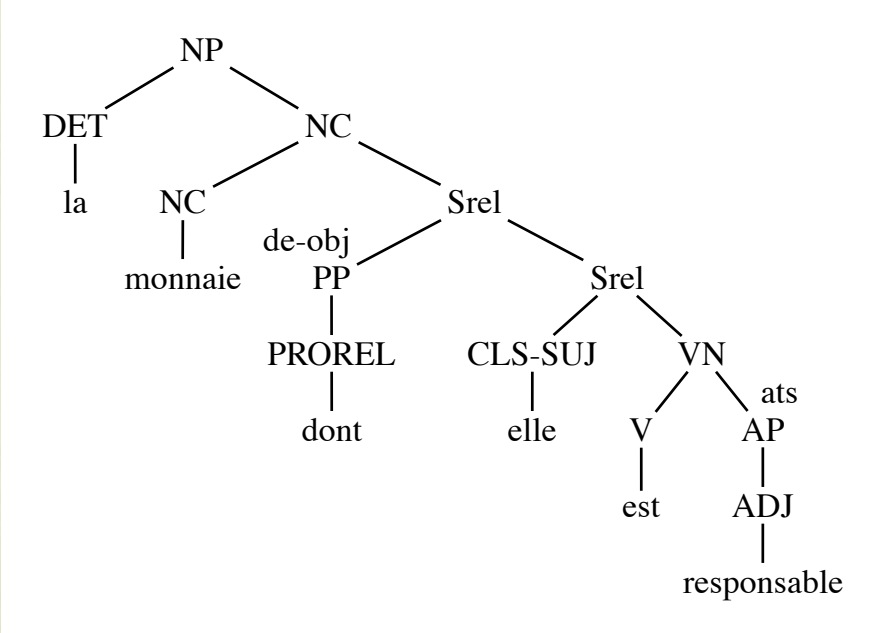
1. Binarize the annotation

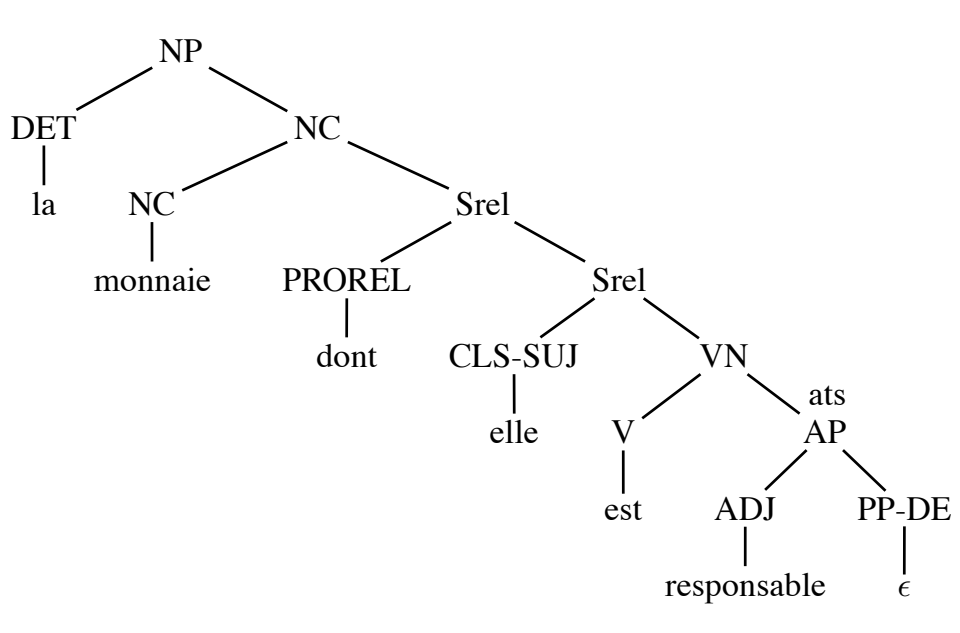




The extraction algorithm

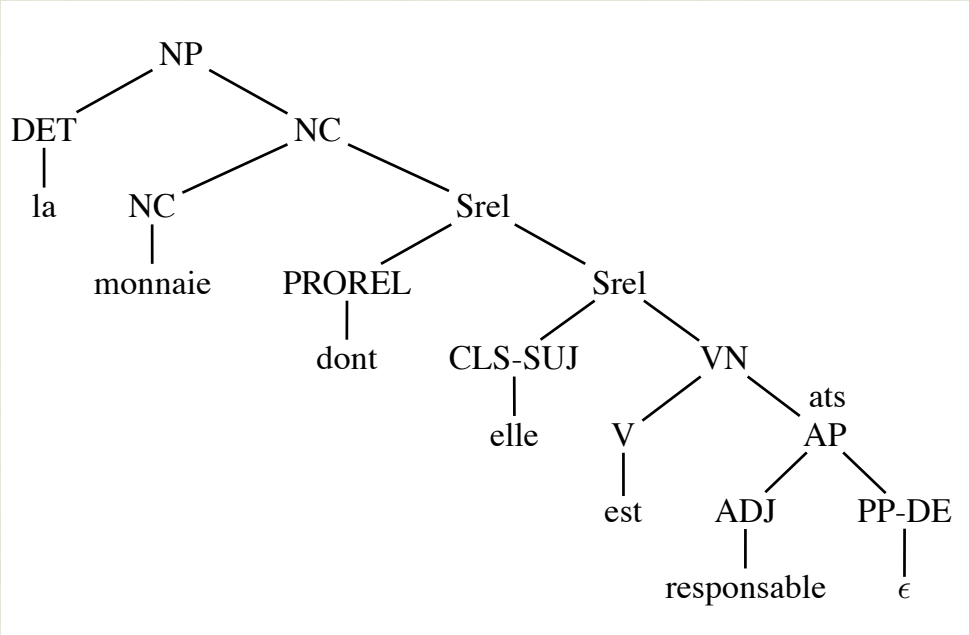
1. Binarize the annotation
inserting traces for wh words





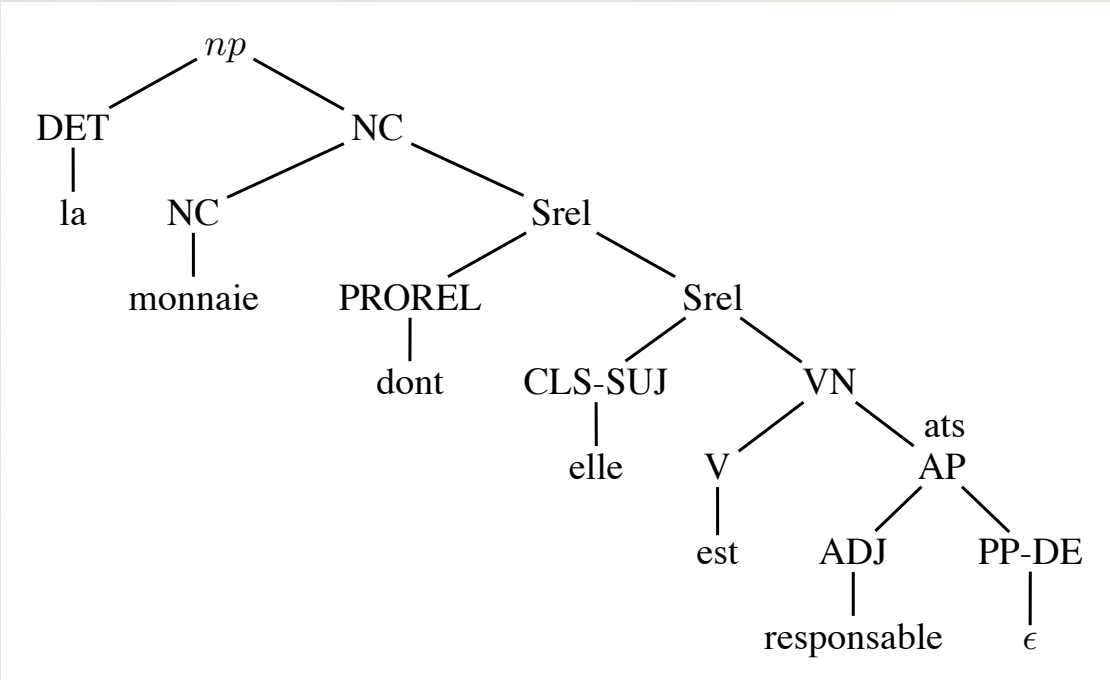
The extraction algorithm

2. Assign formulas



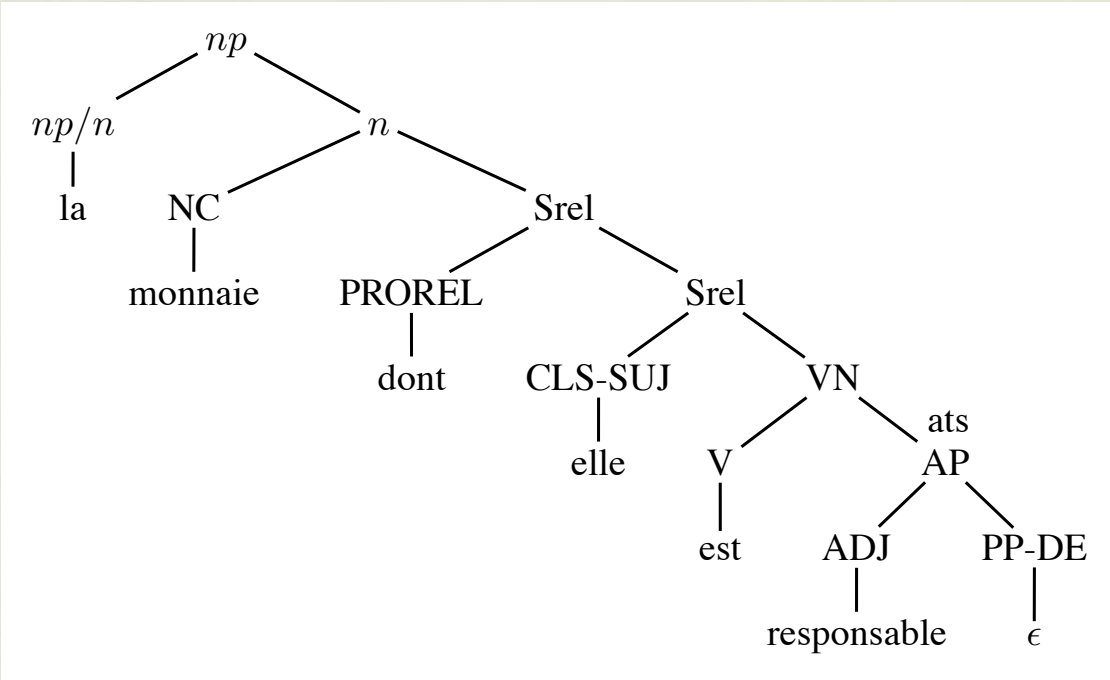
The extraction algorithm

2. Assign formulas



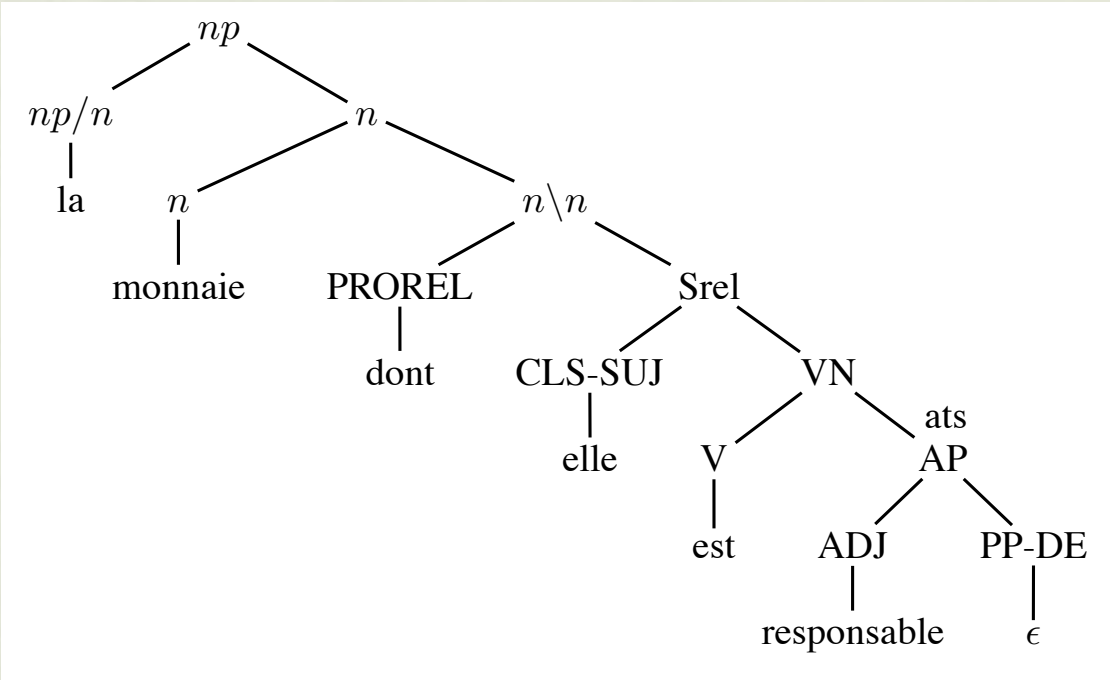
The extraction algorithm

2. Assign formulas



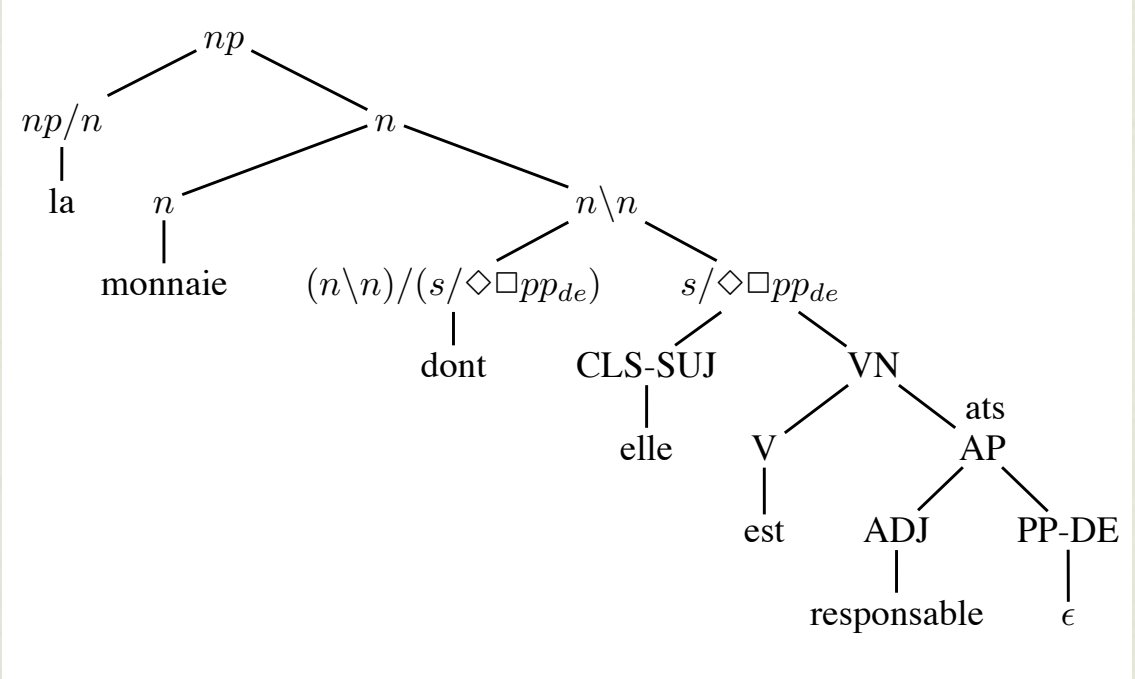
The extraction algorithm

2. Assign formulas



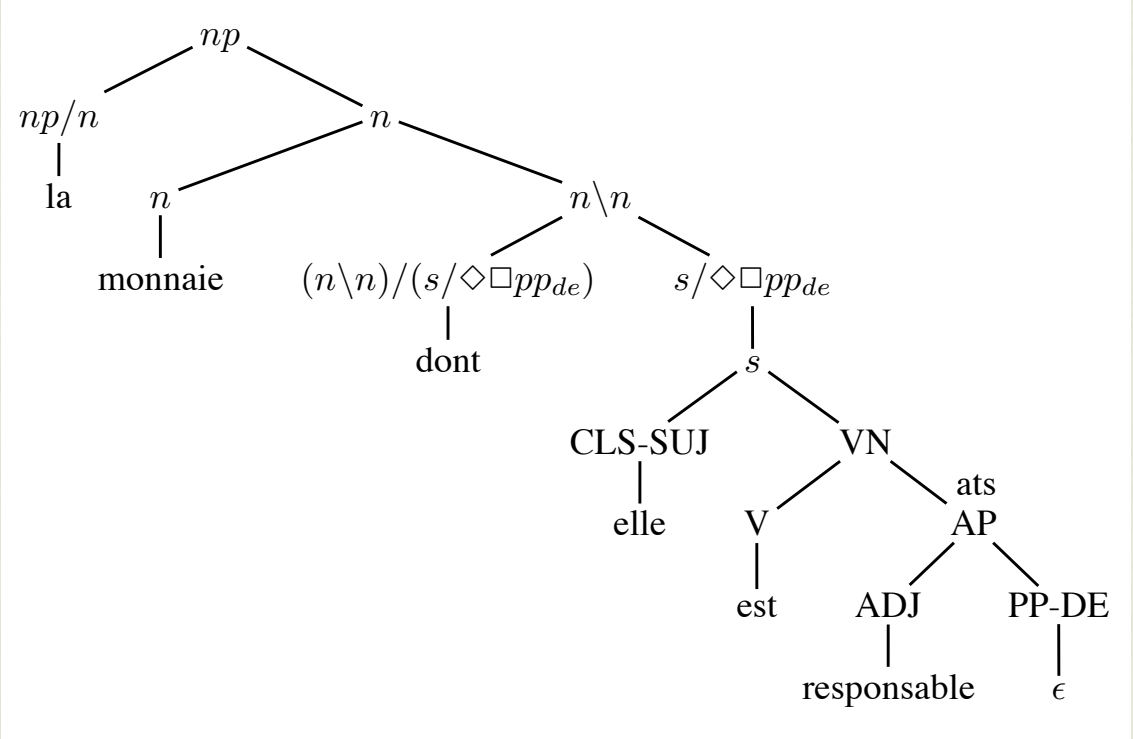
The extraction algorithm

2. Assign formulas



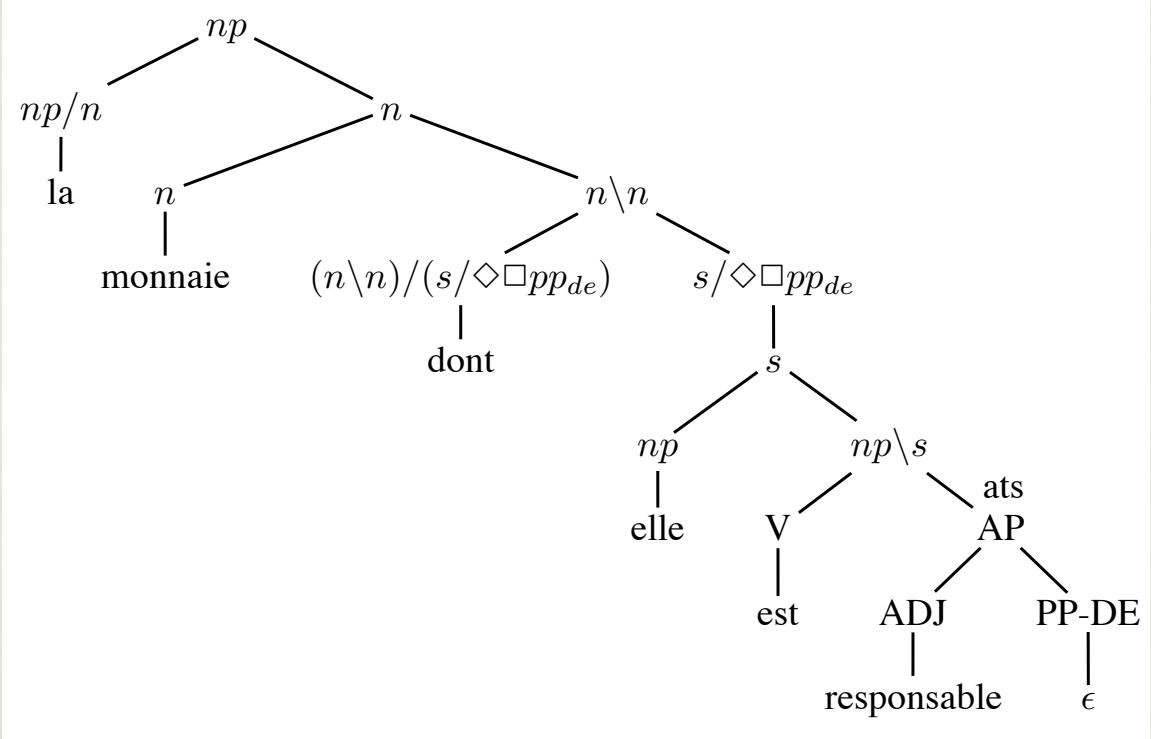
The extraction algorithm

2. Assign formulas



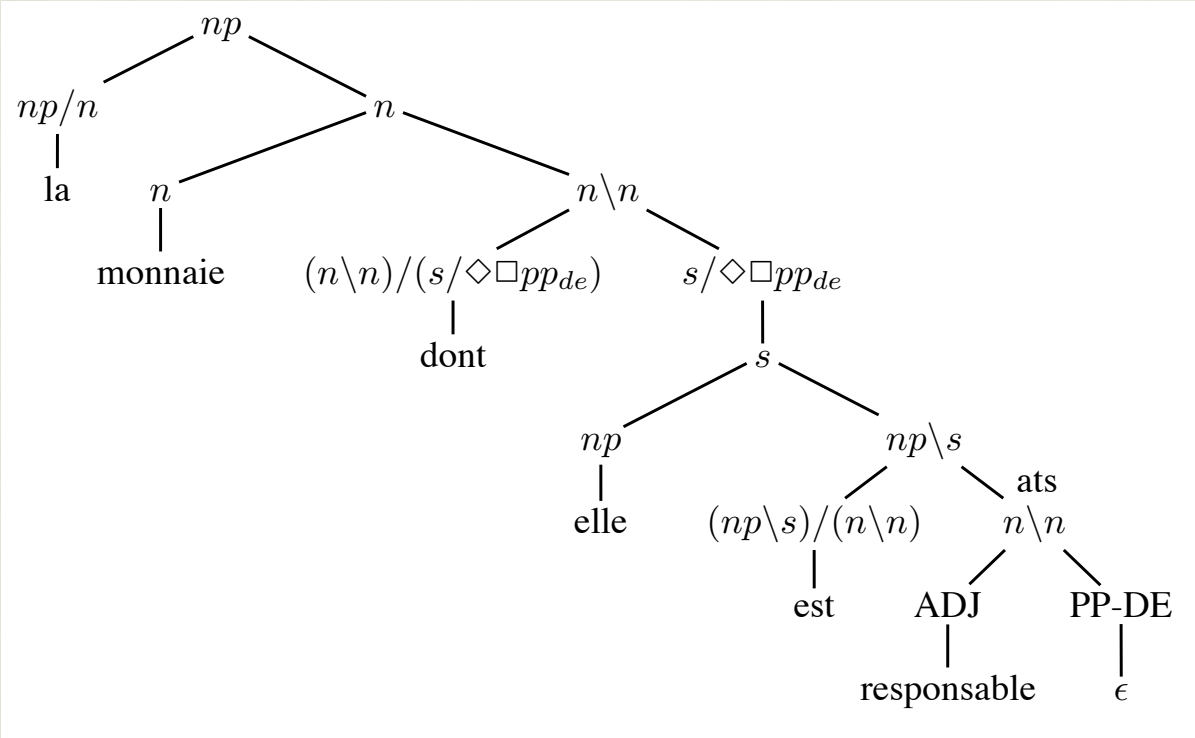
The extraction algorithm

2. Assign formulas



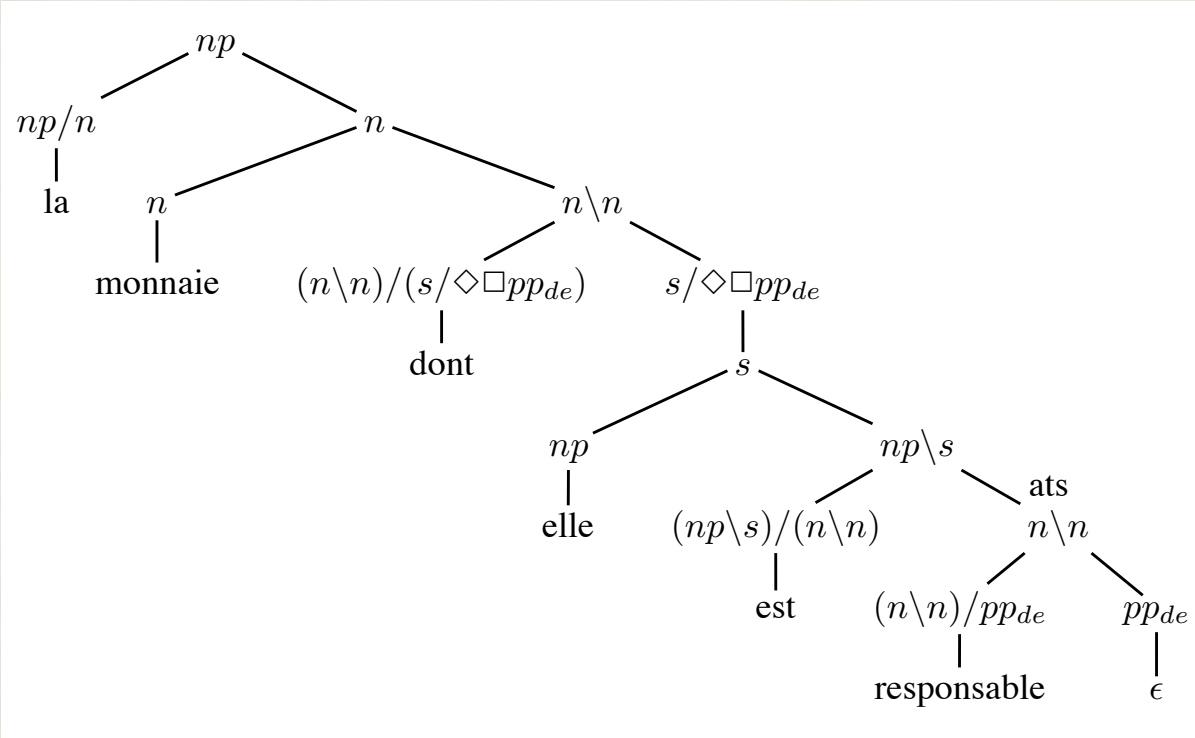
The extraction algorithm

2. Assign formulas



The extraction algorithm

2. Assign formulas



Grammar Extraction

- A lot of useful information (such as the position of “traces” of extracted elements) is not annotated but very useful for the grammar and needs to be added by hand.
- In addition, the extracted grammar has received a very significant amount of manual cleanup

The extracted grammar

- On the basis of the 382.145 words and 12.822 sentence of the treebank, the extraction algorithm extracts 883 different formulas, of which 664 occur more than once.
- Many frequent words are assigned many different formulas
- This is a significant bottleneck for parsing, however we can circumvent this problem using well-known NLP techniques (skip, skip to example)

The extracted grammar

Word	POS	#
et	conj	71
,	ponct	62
à	prp	55
plus	adv	44
ou	conj	42
est	verb	39
être	inf	36
en	prp	34
a	verb	31

POS	#
adv	206
conj	92
prp	149
ponct	89
verb	175

An illustration of some of the most ambiguous words and part-of-speech tags.

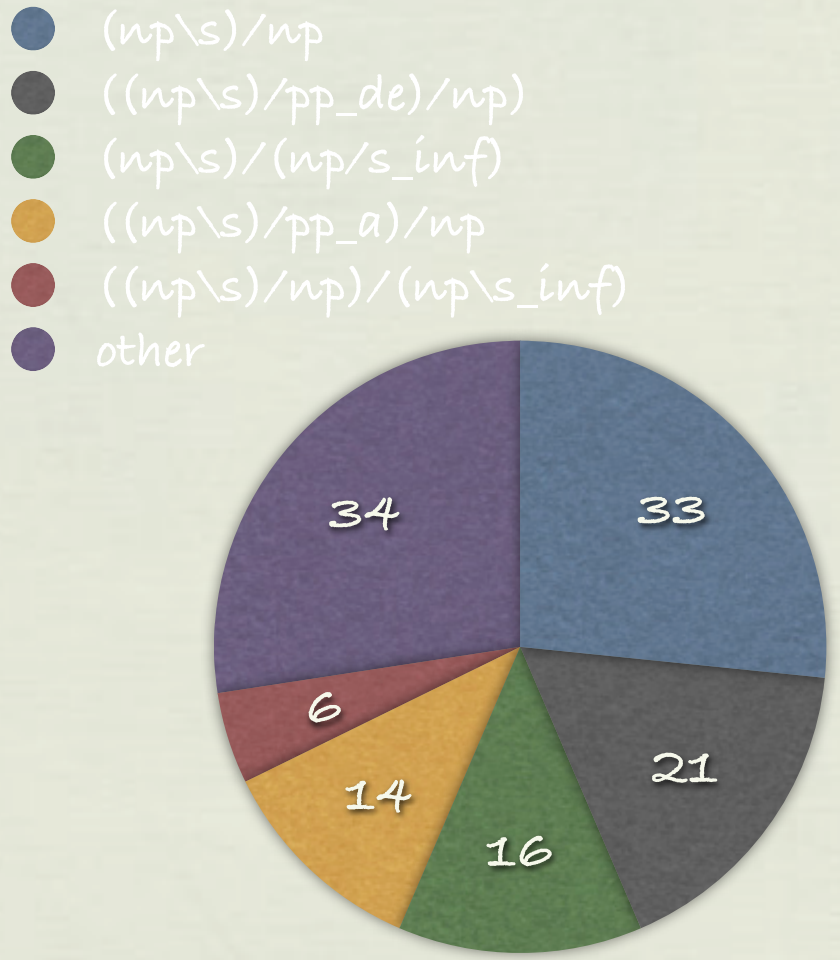
POS tagsets

MElt (Denis & Sagot) and Treetagger (Schmid) are the two main POS-taggers for French. They differ slightly in their tagsets: eg. Treetagger has a single tag "PRP:det" for "du" and "des" (harder for the supertagger), whereas the MElt tagger distinguishes between these words as determiner and as preposition (harder for the POS-tagger)

Word	MElt	Tt
Numerals	DET,ADJ,NC,NPP	NUM
du/des	DET, P+D	PRP:det
Verbs	V	VER:{simp,impf,...}

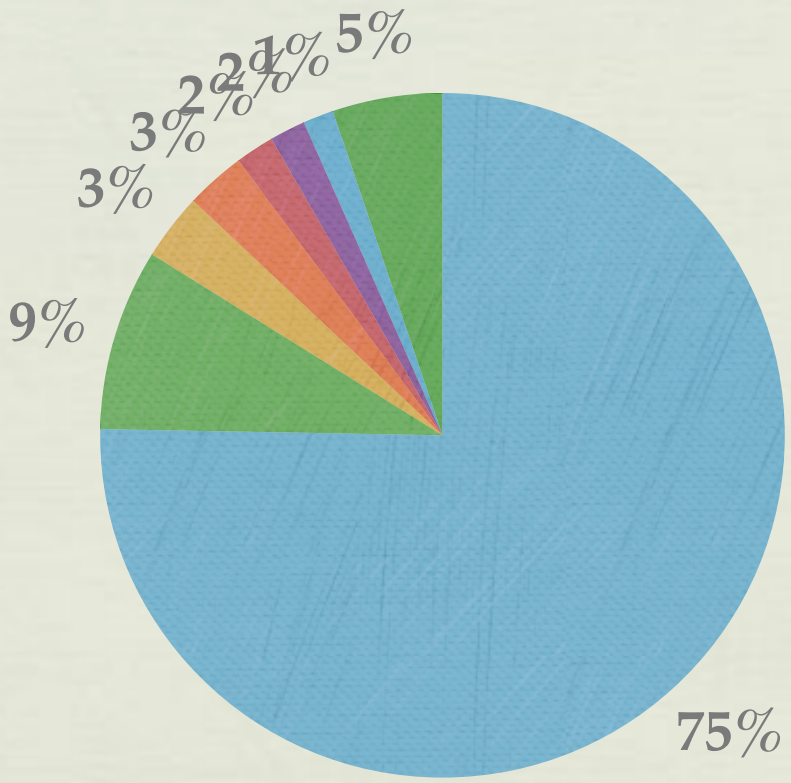
The extracted grammar

- Formula assignments to the present tense form “fait”
- 124 occurrences in the corpus, with 19 different formulas assigned to it.



The extracted grammar

- Formula assignments to the comma “,”
- 21,398 occurrences, 62 different formulas.
  - no formula
  - (np\np)/np
  - (n\n)/n
  - (np\np)/n
  - (s\s)/s
  - ((np\s)\(np\s))/(np\s))
  - ((n\n)\(n\n))\(n\n))
  - other



The extracted grammar

- The sum up, we have produced a categorial grammar for French, which is essentially a very big lexicon.
- The size of this lexicon, coupled with high lexical ambiguity, makes direct exploitation for parsing difficult.
- A fairly standard solution is to use a supertagger to estimate the most likely sequence of formulas for the given words.

Supertagging

- Supertagging is essentially part-of-speech tagging but with richer structure hence "super" tags.
- Like part-of-speech tagging, we use superficial contextual information and statistical estimation to decide the most likely tag.

Supertagging

- So what is the context for a supertagger?
- Typically, it consists of the current word, the surrounding words, the current and surrounding POS tags and the previous supertags.

Context for "de"

np/n	n	?		
DET	NC	P	NPP	NPP
la	voiture	de	Prince	Charles

Supertagging

- The basic procedure for finding the sequence of formulas then becomes
  - Find the correct POS tag sequence
  - Find the correct supertag sequence

Context for "de"

np/n	n	?		
DET	NC	P	NPP	NPP
la	voiture	de	Prince	Charles

Supertagging

- Estimation is done using maximum entropy models
- Very standard and easy to modify (ie. we can add any information we think is useful and let the estimation algorithm decide which ones really are).
- Good performance and

Context for "de"

np/n	n	?		
DET	NC	P	NPP	NPP
la	voiture	de	Prince	Charles

Any information which we can easily obtain, of course. If we think a word having an even number of letters is useful, we can add it.

POS/Supertagging

- Note, that, though Part-of-Speech tagging helps, an incorrect POS-tag can actually hurt the supertagger.
- Errors in DET-N versus CLO-V POS-tags are difficult for the supertagger to recover from.

np/n	n	(np\s)/np	np/n	n
DET	NC	V	DET	N
la	petite	brise	la	glace

np/n	n/n	n	(np\s)/((np\s)/np)	(np\s)/np
DET	ADJ	NC	CLO	V
la	petite	brise	la	glace

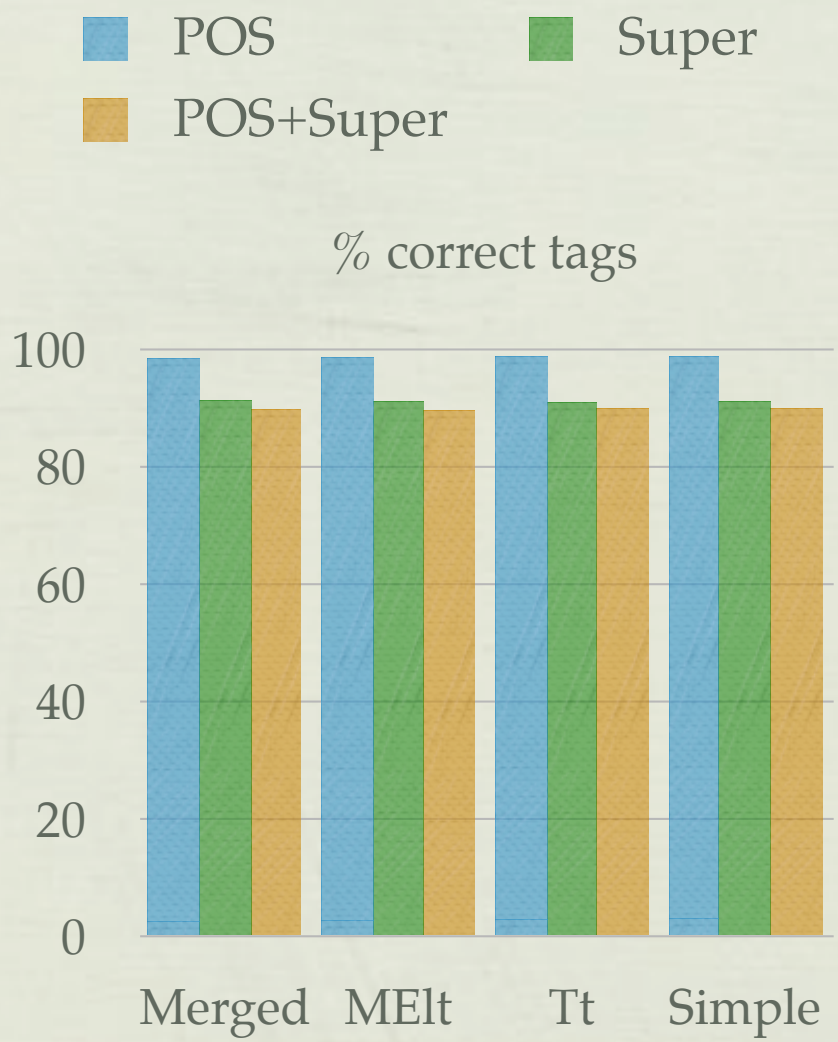
POS/Supertagging

- Other difficult words for the POS-tagger include “que” (which can be a conjunction, an adverb or a relative pronoun)
- However, in general, the POS-tag information helps (as we will see)

np/n	n	(np\s)/np	np/n	n
DET	NC	V	DET	N
la	petite	brise	la	glace

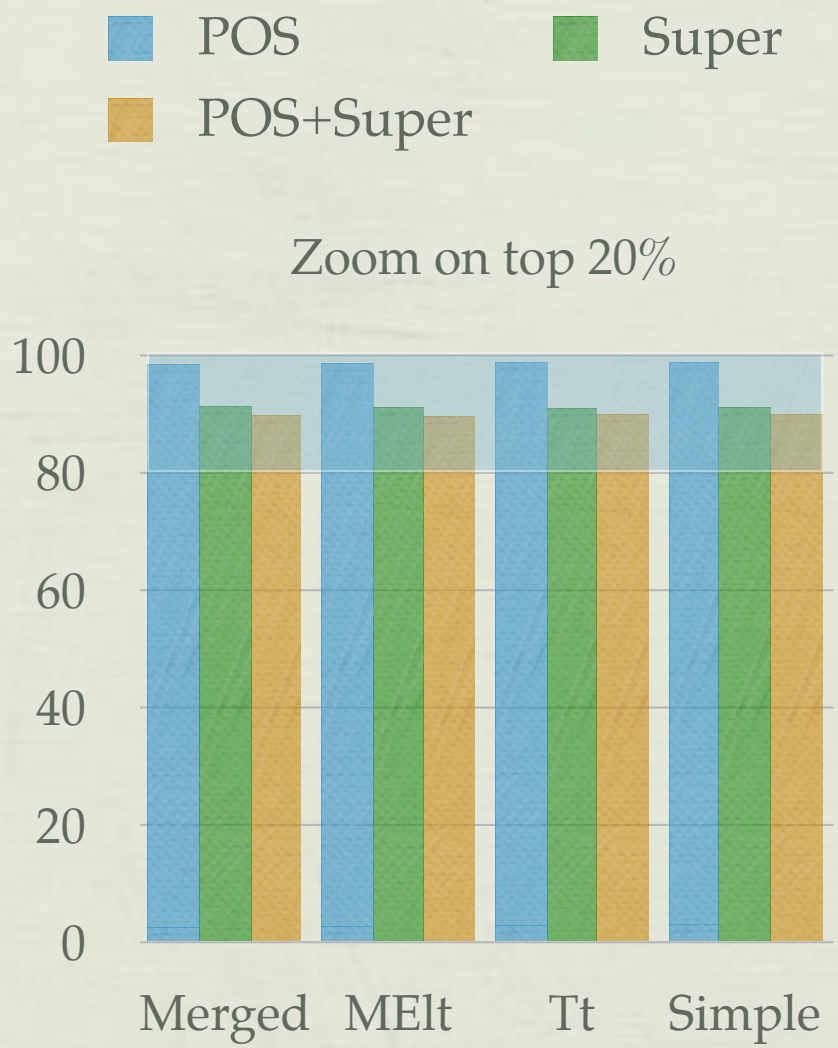
np/n	n/n	n	(np\s)/((np\s)/np)	(np\s)/np
DET	ADJ	NC	CLO	V
la	petite	brise	la	glace

POS/Supertagger



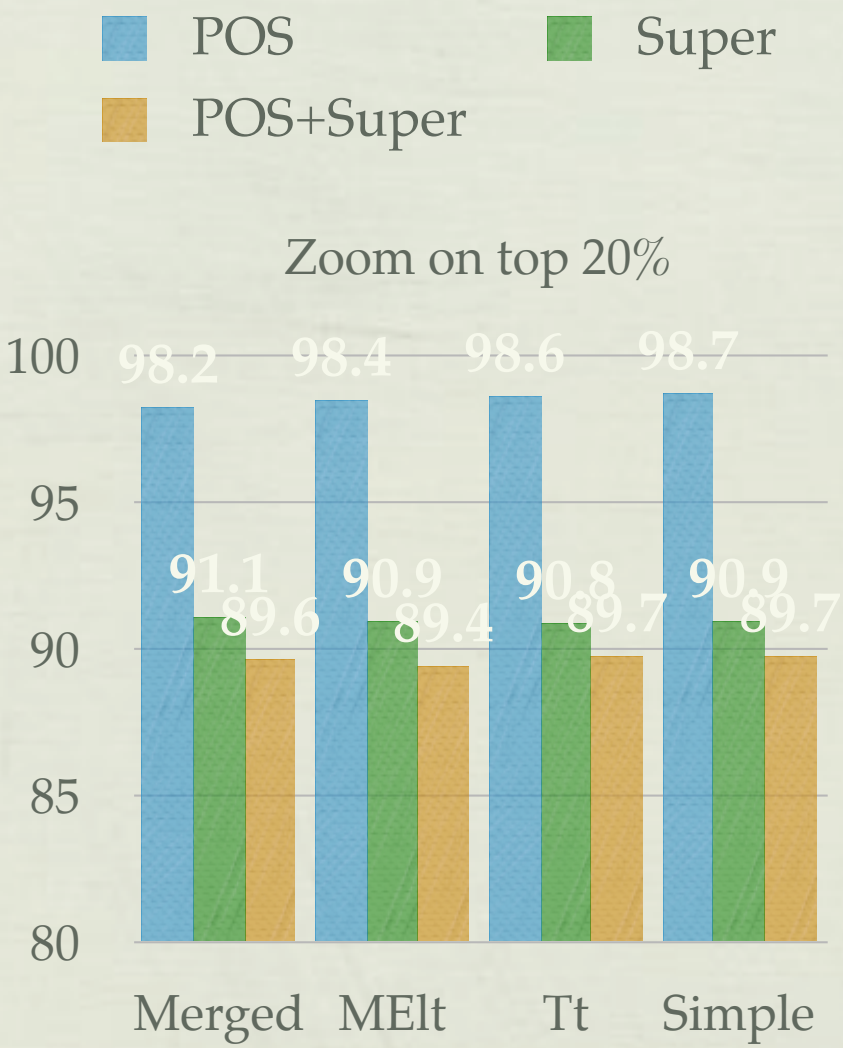
- A plot of POS/Supertagger results for the four different tagsets.
- **POS+Super** gives the % correct supertags given the POS-tag assigned by the tagger, **Super** is the correct supertag given

POS/Supertagger



- A plot of POS/Supertagger results for the four different tagsets.
- **POS+Super** gives the % correct supertags given the POS-tag assigned by the model, **Super** is the correct supertag given

POS/Supertagger



- A plot of POS/Supertagger results for the four different tagsets.
- **POS+Super** gives the % correct supertags given the POS-tag assigned by the model, **Super** is the correct supertag given

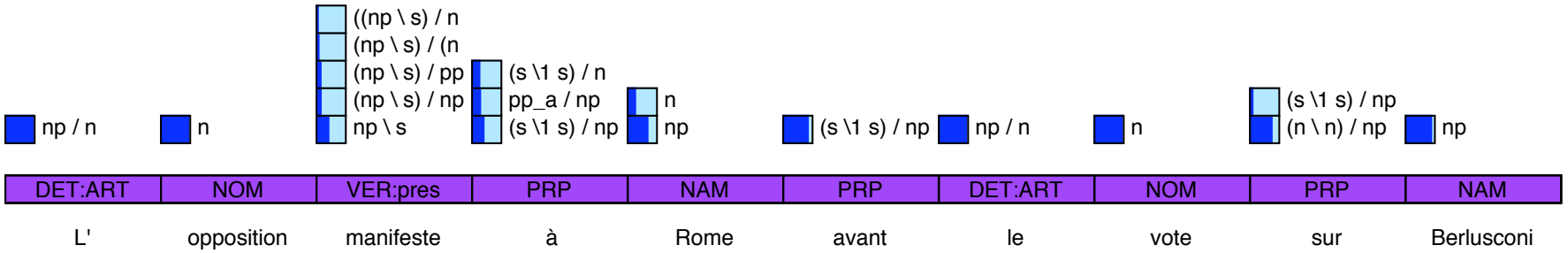
Multiple Solutions

- Though these results are comparable to the best supertaggers for English, in practice, even at around 91% correct supertags, we do not cover enough sentences of the corpus.
- A standard solution is to look at supertags within a range depending on the best supertag.
- This is called the β value.

Multiple Solutions

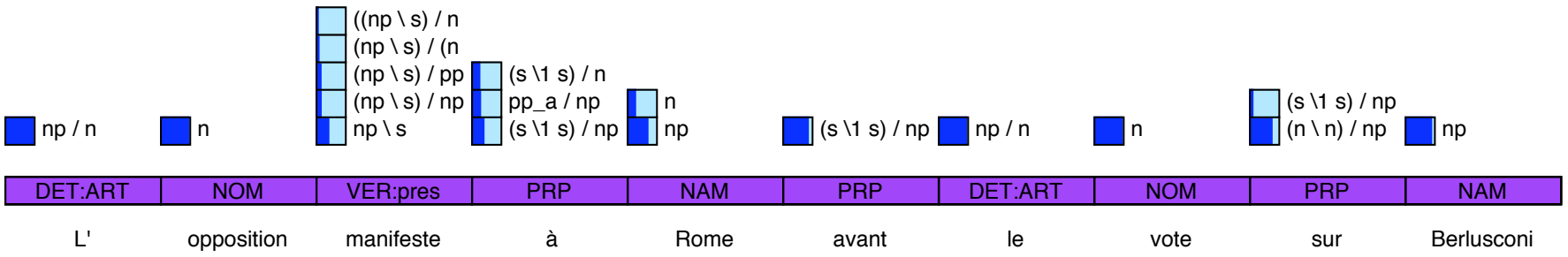
- Roughly speaking: if p is the probability of the best supertag, we will assign all supertags of probability > βp
- So, the less we are sure of our first supertag, the more alternatives we add.
- On average, a β of 0.1 gives 2.7 supertags per word, 0.05 gives 3.1 and 0.01 gives 4.7

Example



- Here is an example with β=0.1
- We can see that many “easy” words get assigned a single supertag whereas difficult words (here: verbs and prepositions) get assigned many tags.

Example

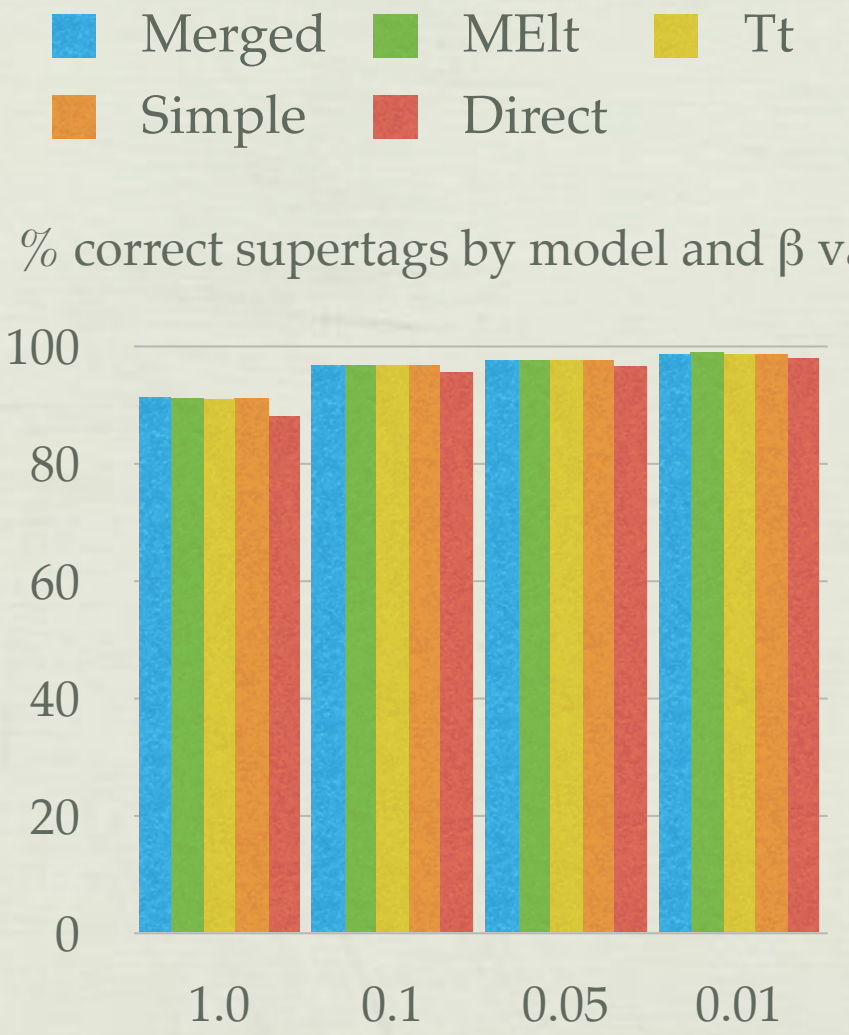


"manifeste"	%
np\s	43.6%
(np\s)/np	15.7%
(np\s)/pp$_a$	15.3%
(np\s)/(np\s$_{ainf}$)	7.7%
((np\s)/np)/pp$_a$)	5.1%

Remark: this is very typical of prepositions, they are either arguments (of verbs, or, more rarely, at least in our analysis, of nouns) or modifiers (of VPs/sentences, so-called adverbial uses, or of nouns)
Adverbial uses are assigned to take scope at the sentence-level instead of at the VP level: this is a simplification, but semantically, we just need the event/state variable of the verb and the subject variable (some adverbs, like "ensemble" or "tous" do clearly need the subject variable, of course!

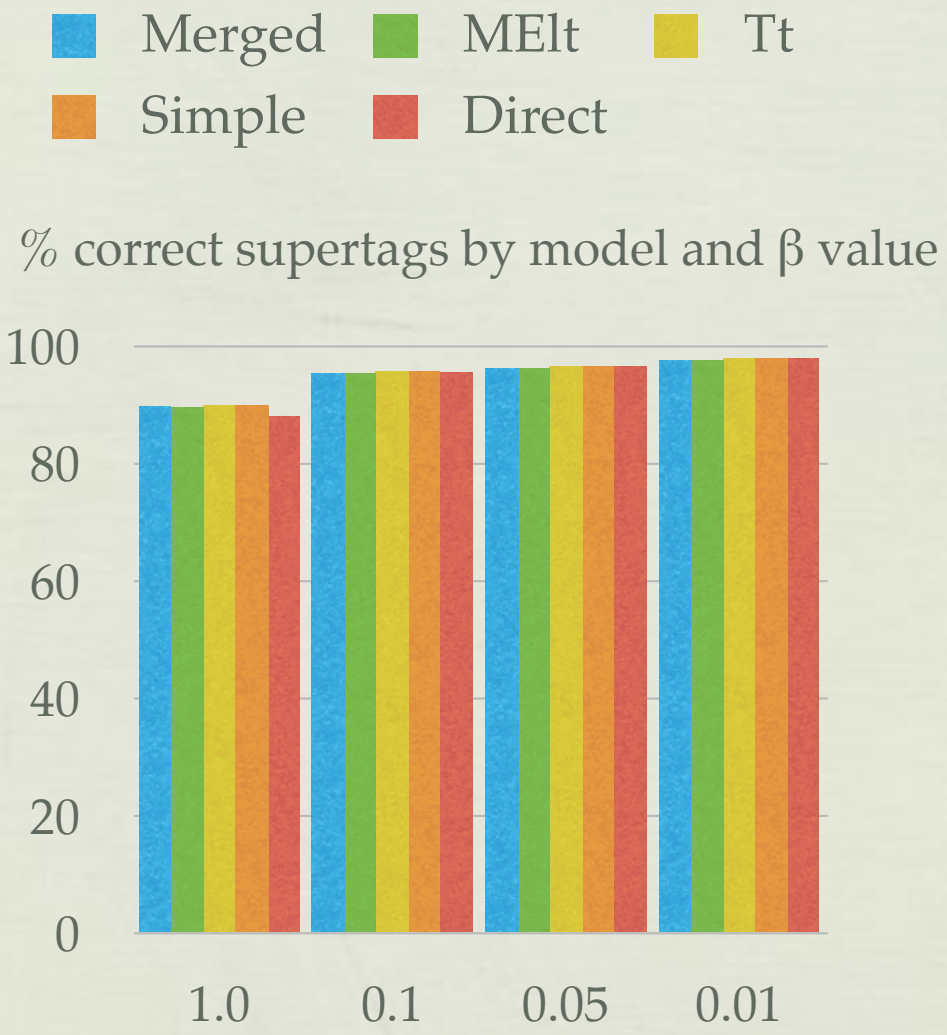
"…"	%
…p	79.1%
…p	9.4%

POS/Supertagger



- Results with the use of different values of β.
- In a sense, the β value allows us to trade coverage for efficiency: at higher values of β, we parse more sentences, but we do so more slowly.

POS/Supertagger

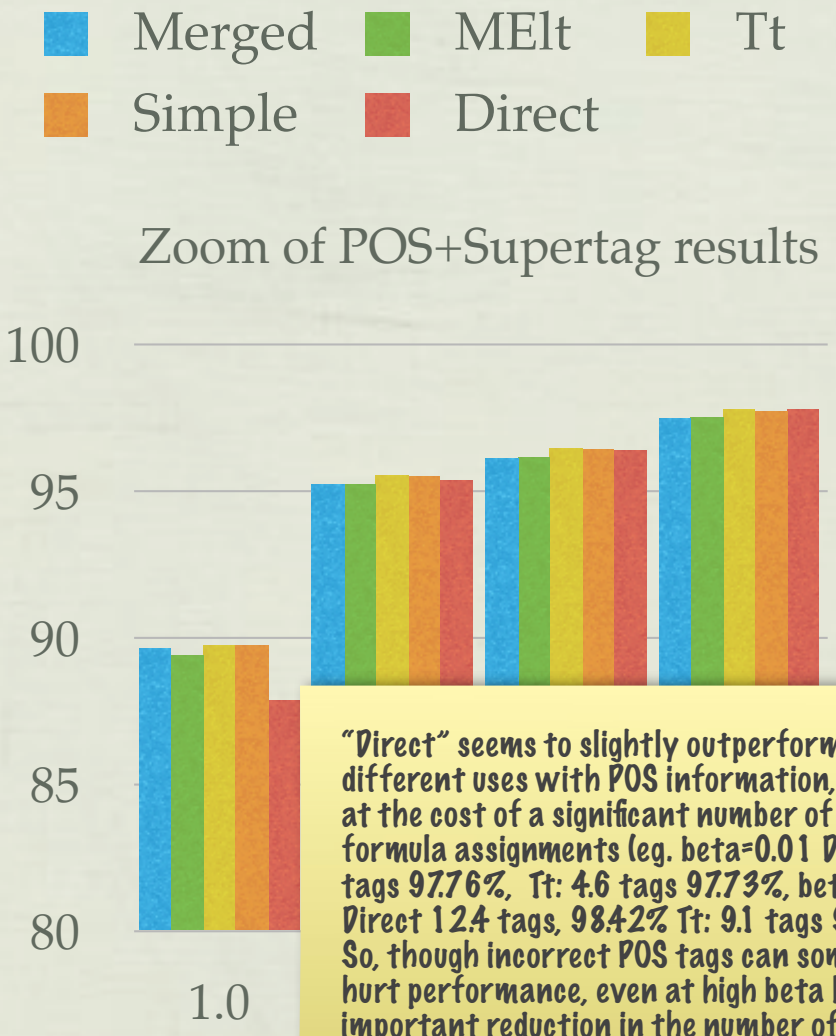


- As before, there is a slight decrease in performance once we switch from "gold" POS tag to tags assigned by the tagger.
- Eg. for the Treetagger tagset, it is -1.0% at β=0.1 and -0.5% at

POS/Supertagger

- A comparison of the Supertagger and the combined POS/ Supertagger.
- Same results as the previous slides, but with a zoom on the top 20 percentile.
- Direct is the result of the

POS/Supertagger

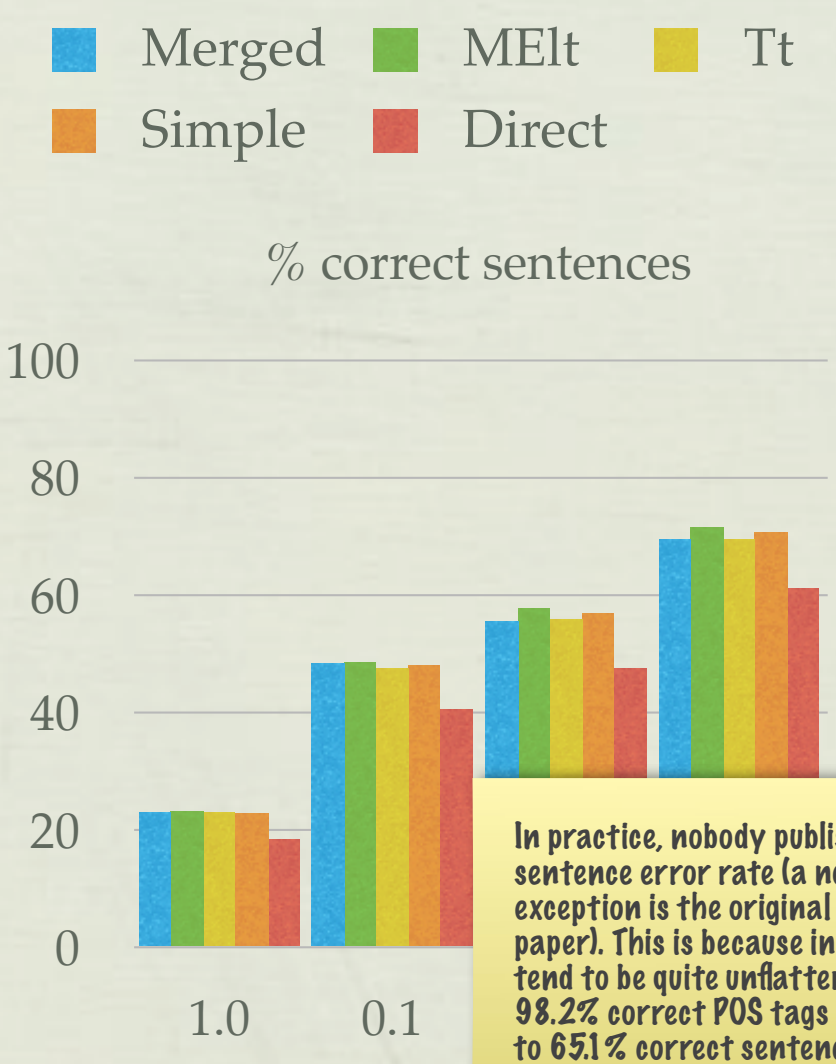


"Direct" seems to slightly outperform the different uses with POS information, but this is at the cost of a significant number of extra formula assignments (eg. beta=0.01 Direct: 5.6 tags 97.76%, Tt: 4.6 tags 97.73%, beta=0.001 Direct 12.4 tags, 98.42% Tt: 9.1 tags 98.40%). So, though incorrect POS tags can sometimes hurt performance, even at high beta levels, the important reduction in the number of tags per word outweighs (IMHO) the slight reduction in correct tags.

- A comparison of the Supertagger and the combined POS/Supertagger.
- Same results as the previous slides, but with a zoom on the top 20 percentile.

Direct is the result of the

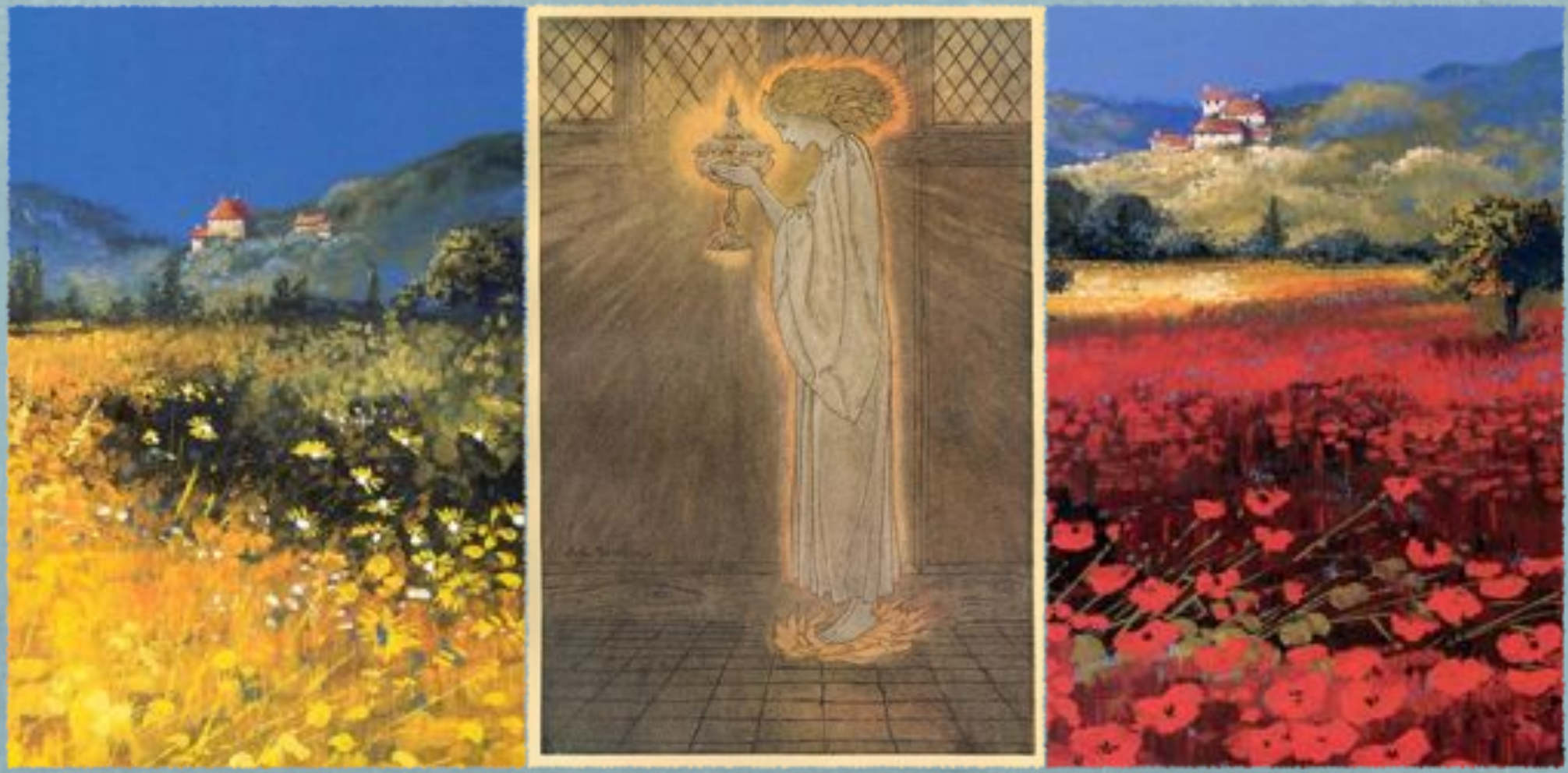
POS/Supertagger



In practice, nobody publishes there per sentence error rate (a notable exception is the original supertagging paper). This is because in general, they tend to be quite unflattering (eg. 98.2% correct POS tags corresponds to 65.1% correct sentences, the figures for beta=0.01 indicate a similar picture)

- Finally, here is the percentage of sentences which are assigned the correct sequence of supertags for the different settings of β and the different POS models.

- Note that we number of sentences for which a parse is found is actually better (around 85% at



Semantics

On the development a wide-coverage semantic lexicon for the extracted categorial grammar

Semantics

- As we have seen, formulas in categorial grammars correspond to types in the simply typed lambda calculus
- Proofs (parses) correspond to lambda terms.
- Thanks to the extracted grammar, we can obtain reasonable accurate parses for French sentences.
- So what is missing to obtain a Montague-style meaning of analysed sentences is a large enough lexicon.

Semantics

- In order to move beyond a simple lexicon listing a limited number of words, it suffices to remark that many of the “open class” words (eg. names, nouns, verbs) follow a general schema to obtain their lexical semantics.
- For example, a noun “n” generally has $\lambda x.n(x)$ as its semantics.

Semantics

- So the basic idea behind wide-coverage semantics is very simple:
  - the lexicon lists words which require special treatment (eg. conjunctions “et” and auxiliary verbs like “être” and “avoir”)
  - other words are assigned a lambda term based on

So the general motto is: if you want to add more information to the semantic lexicon, there are two basic (non-exclusive) solutions: 1) you list the different cases 2) you train a (reliable) tagger
Solution 1 would be an option for distinguishing subjet/object control verbs and Solution 2 would be an option for Named Entities (and their types: persons, places, enterprises), for more complicated semantic distinctions, like events versus states the solution is less clear.

Beyond Montague

- Montague grammar has some well-know limitations.
- I will talk briefly about two of them: anaphora and presuppositions
- I will sketch solutions to both of them, which stay within the framework of the simply typed lambda calculus.

Anaphora

A man walks in. He orders a beer.

- The meaning of this (tiny) discourse is clear: there exists a man, who enters and (presumably) this same man orders a beer.
- In first order logic, this is represented by the following formula:
- $\exists x.$ [enter(x) $\wedge$ $\exists y$ [beer(y) $\wedge$ order(x,y)]]

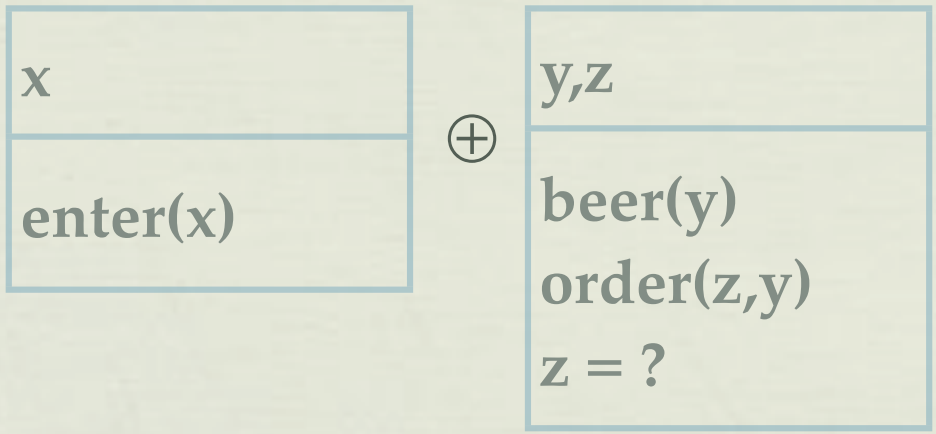
Anaphora

A man walks in. He orders a beer.

- $\exists x.$ [enter(x) $\wedge$ $\exists y$ [beer(y) $\wedge$ order(x,y)]]
- Now what are the formulas we can assign to the two sentences?
- $\exists x.$ [enter(x)]
- $\exists y$ [beer(y) $\wedge$ order(x,y)]

DRT

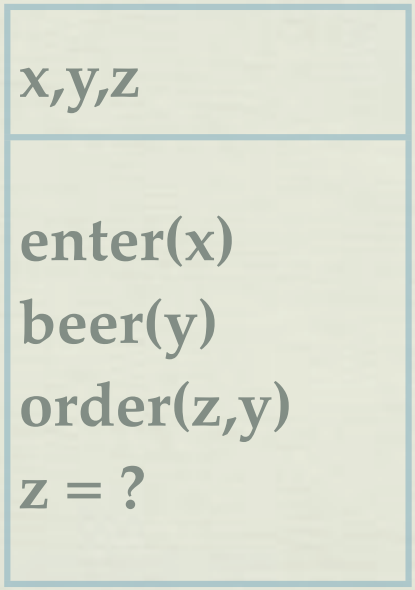
- Kamp (1981) and Kamp & Reyle (1993) propose the following solution:



$\exists x.$ [enter(x)] $\exists y$ [beer(y) $\wedge$ order(z,y)]

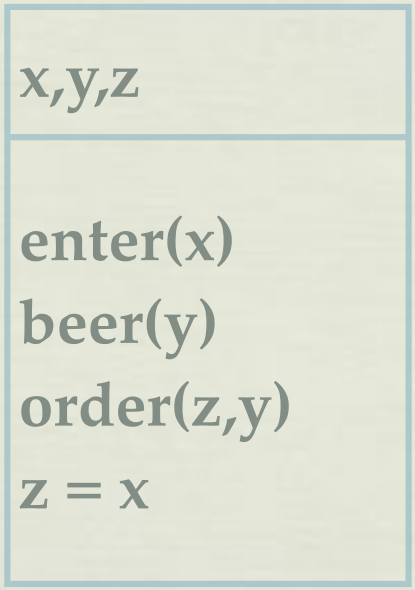
DRT

- Kamp (1981) and Kamp & Reyle (1993) propose the following solution:



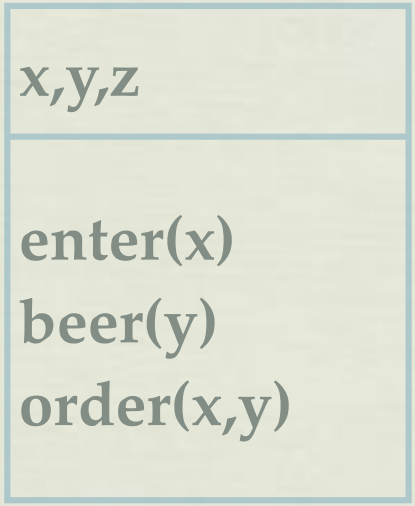
DRT

- Kamp (1981) and Kamp & Reyle (1993) propose the following solution:



DRT

- Kamp (1981) and Kamp & Reyle (1993) propose the following solution:



$\exists x.\ [\ \text{enter}(x) \wedge \exists y\ [\ \text{beer}(y) \wedge \text{order}(x,y)\ ]\ ]$

DRT

- Kamp (1981) and Kamp & Reyle (1993) propose the following solution.
- This solution is compatible with the lambda calculus approach with have been adopting (Muskens 1994)

DRT

Example entries

"dormir" is a state rather than an event, however, the current system does not distinguish between different types of eventualities.

marché : $n - \lambda x_0.$ $\begin{array}{|c|}\hline \\ \hline \mathbf{marché}(x_0) \\ \hline \end{array}$

Marie : $np - \lambda P_0.($ $\begin{array}{|c|}\hline y_0 \\ \hline \mathbf{nommé}(y_0, \mathbf{mary}) \\ \hline \end{array}$ $\oplus P_0(y_0))$

chaque : $np/_0 n - \lambda P_0 Q_0.((\begin{array}{|c|}\hline z_0 \\ \hline \\ \hline \end{array} \oplus P_0(z_0)) \rightarrow Q_0(z_0))$

DRT

Example entries

"dormir" is a state rather than an event, however, the current system does not distinguish between different types of eventualities.

dort : $np\backslash_0 s - \lambda L_0 e_0 . L_0(\lambda z_0 .$

e_0
$\mathbf{event}(e_0)$
$\mathbf{dort}(e_0, z_0)$

)

pousser : $((np\backslash_0 s)/_0(np\backslash_0 s_{ainf}))/_0 np - \lambda x_0 y_0 z_0 x_1 . x_0(\lambda y_1 . z_0(\lambda z_1 .$

d_2
$\mathbf{pousser\_à}(x_1)$
$\mathbf{agent}(x_1, z_1)$
$\mathbf{patient}(x_1, y_1)$
$\mathbf{theme}(x_1, y_2)$
$y_2 : y_0(z_2, x_0)$

))

Presupposition

What is presupposition?

- Presupposition (like anaphora and their resolution) are a linguistic phenomenon on the semantics/pragmatics interface.
- Its particularity is the it *presupposes* something which may not be directly said.
- So we could say it is a sort of "window" through which we can observe aspects of the abstract notion of "common ground".

Presupposition

What is presupposition?

Some examples

1. Have you stopped beating your wife?

    a. presupposes the listener has been beating his wife

2. George W. Bush would torture again.

    b. presupposes Bush has tortured

3. Obama regrets intervening to save Terry Schiavo.

    c. presupposes Obama intervened to save T. Schiavo

Presupposition

What is presupposition?

How can we decide if something is a presupposition (as opposed to an implicature or an entailment?

1. Presuppositions stay when embedded inside of a negation, a question, or a modal.

2. "Hey, wait a minute, I didn't know that..."

- Obama regrets intervening to save Terry Schiavo.

- Obama doesn't regret intervening to save Terry Schiavo.

- Obama may regret intervening to save Terry Schiavo.

Presupposition

What is presupposition?

How can we decide if something is a presupposition (as opposed to an implicature or an entailment?

1. Presuppositions stay when embedded inside of a negation, a question, or a modal.

2. "Hey, wait a minute, I didn't know that..."

- Hey, wait a minute, I didn't know he intervened to save Terry Schiavo.

- # Hey, wait a minute, I didn't know he regretted doing that.

Presupposition

From Karttunen (1973)
Question: what do the sentences presuppose about the guilt of someone other than Nixon?

1. If Dean told the truth, Nixon is gu
2. If Haldeman is guilty, Nixon is gu
3. If Miss Woods destroyed the missing tapes, Nixon is guilty too.

Just to show that things can get complicated and that inference is sometimes necessary: in (3), if destroying the tapes implies being guilty, then (3) as a whole doesn't imply someone besides Nixon is guilty, since this is part of the antecedent.
Note that with a lot of effort, we could do the same for (1): imagine a strange dictatorship, where it is against the law to speak the truth when it harms the president

Note that there is another possible lecture for "too", which would occur in cases like "Nixon was not just incompetent, he was guilty too".

Presupposition

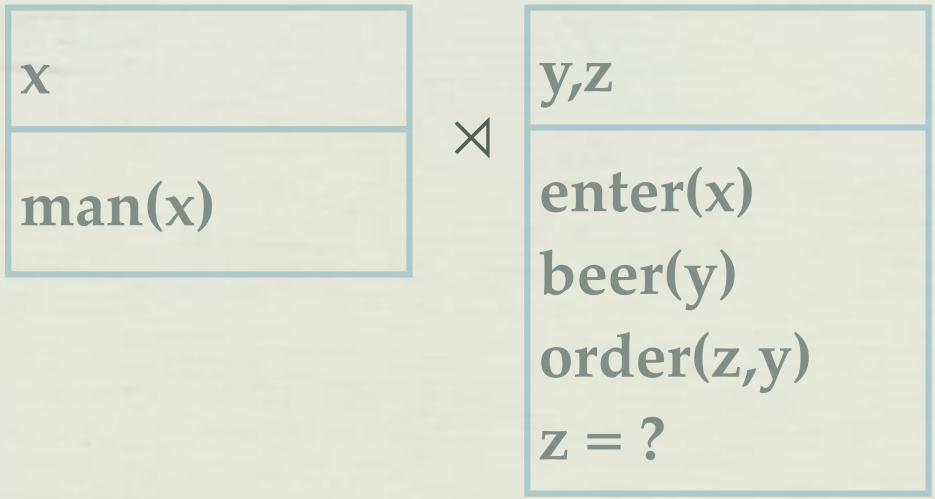
- Presupposition is not an obscure phenomenon, which rarely occurs in a corpus: proper names and definite articles both presuppose existence of (at least an intension of) the name or definite description.
- So, with a quick and approximate calculation, we have an average of around three presuppositions per sentence!
- This means that in order to do wide-coverage semantics, we need to have at least some way of treating presupposition.

Presupposition in DRT

- DRT has been extended to handle presuppositions (Kamp 2001a, Kamp 2001b, Kamp & Reyle, to appear)
- The highly simplified version of it allows a DRS to be a pair of two DRSs, where the first contains the presuppositions of the second.

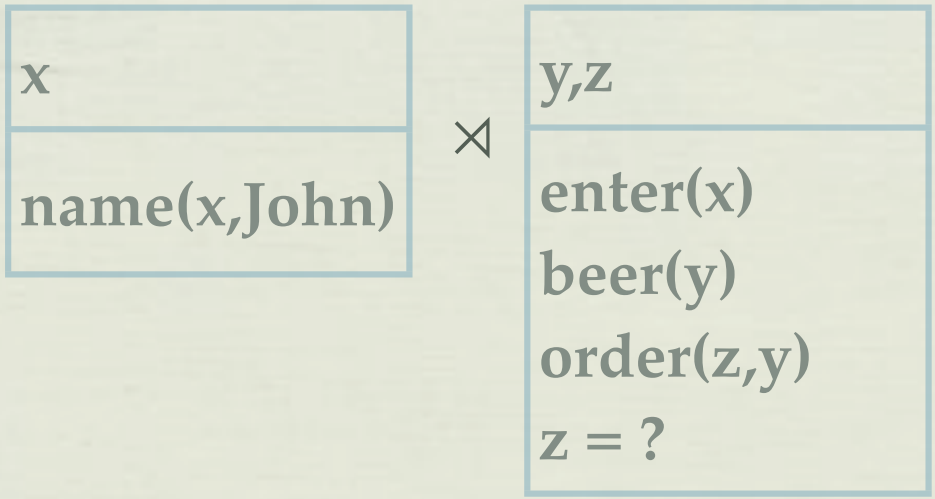
Presupposition in DRT

The man walks in. He orders a beer.

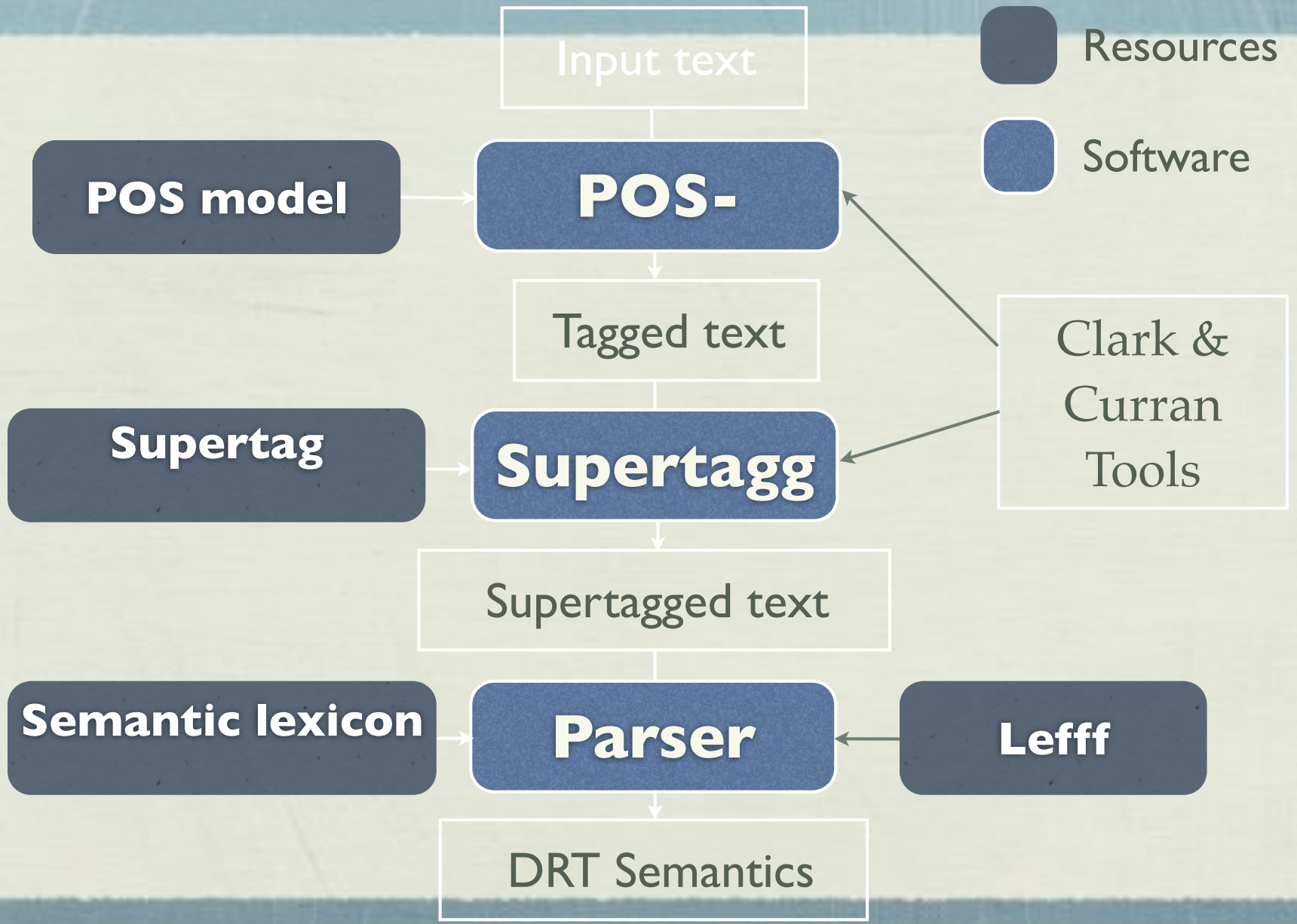


Presupposition in DRT

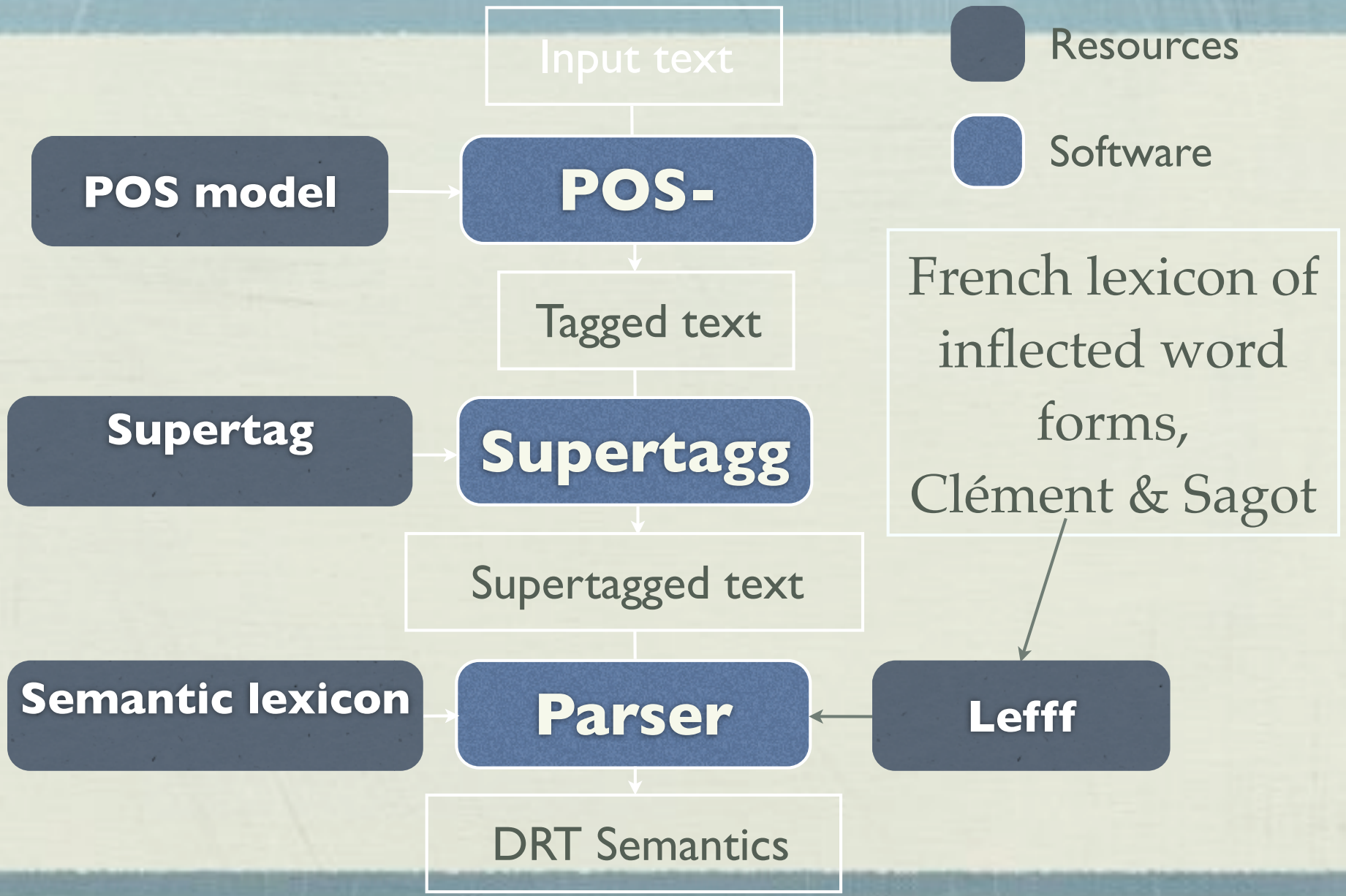
John walks in. He orders a beer.



Grail & Friends



Grail & Friends



Demo

- All talk and no demo make Jack a dull boy.
- All talk and no demo make Jac

Give a demo of the system with today's headlines from "Google Actualités"

Conclusion

- I have described the development of a wide-coverage categorial grammar for French and first steps towards using it for wide-coverage semantics
- All software and resources are available under LGPL (with the unfortunate exception of the annotated corpus, which is bound by the same conditions as the Paris VII treebank).

Future Work

- A very long list, but I will mention some of the more important tasks.

Future Work - Parser

- Improve the accuracy of the extracted grammar and the parser
- Improve the efficiency of parser (eg. by using tree automata)
- Add a component for multi-word expressions.

(as in Noémie-Fleur's talk, of course !)

Future Work - Semantics

- Incorporate a Named-Entity component.
- Incorporate a rudimentary analysis of tense/aspect and discourse structure using the French Timebank (Bittar 2010).
- Word sense disambiguation
- General problem: lack of annotated data

Future Work - Semantics

- Open questions:
  - how “deep” can we go with wide-coverage semantics?
  - what are appropriate evaluation measures?