Wide-Coverage Parsing With Type-Logical Grammars

Richard Moot

LaBRI CNRS and INRIA Futurs

Richard.Moot@labri.fr

Introduction

- type-logical grammars are most often used for analysing a precise natural language phenomenon using a small grammar fragment
- we will look at automatically extracting a type-logical grammar from a corpus, with the goal of parsing unrestricted text
- we will study the challenges to our parsing algorithms posed by the size of these grammars

Outline

Outline

Outline

Outline

The Spoken Dutch Corpus

- 9 million words of contemporary spoken Dutch with different types of annotation
- orthographic transcription and part-of-speechs tags have been provided for all words
- a core corpus of 1 million words has been provided with syntactic annotation in the form of dependency graphs

The Spoken Dutch Corpus

Syntactic Annotation

wat *is* *er* *zielig* *aan* ?

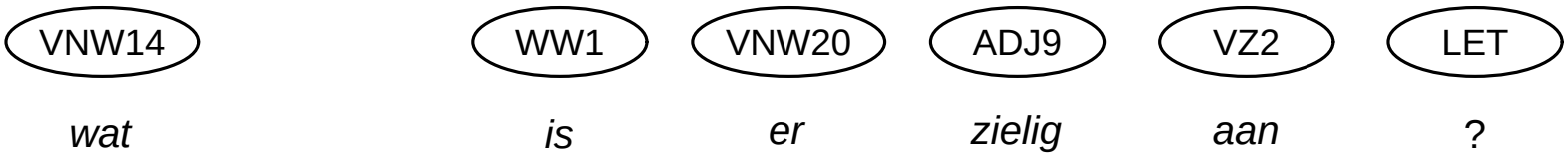
The Spoken Dutch Corpus

Syntactic Annotation

VNW14	WW1	VNW20	ADJ9	VZ2	LET
wat	*is*	*er*	*zielig*	*aan*	?

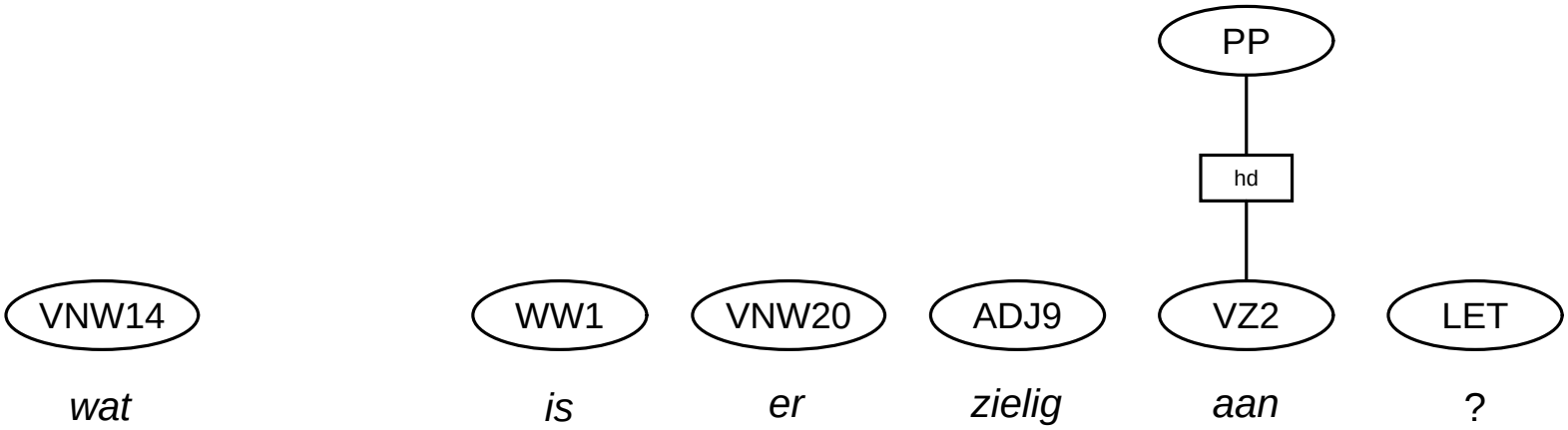
The Spoken Dutch Corpus

Syntactic Annotation



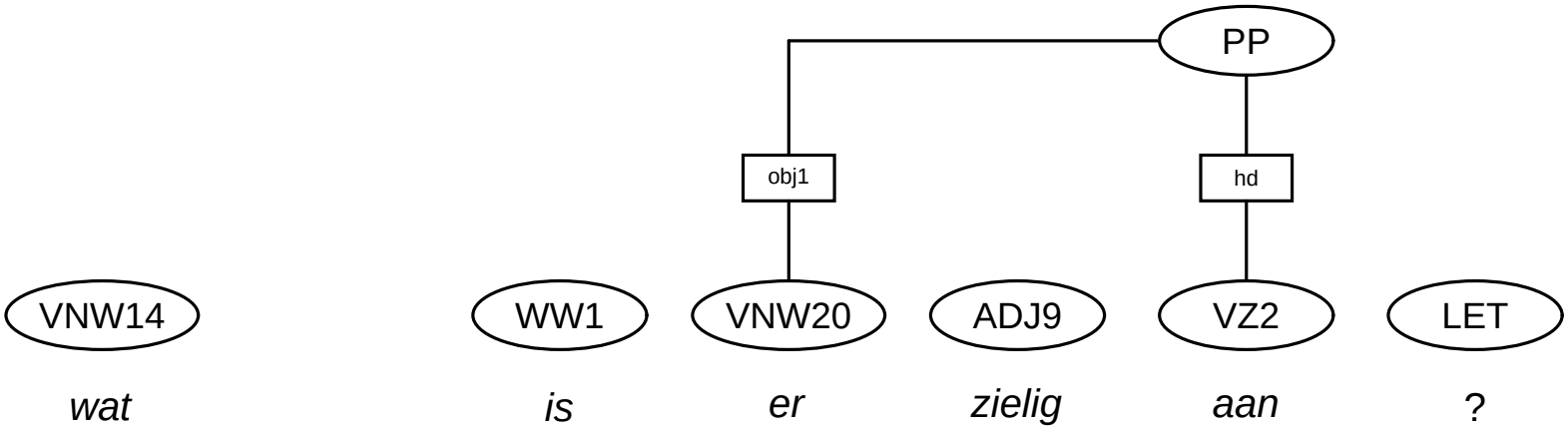
The Spoken Dutch Corpus

Syntactic Annotation



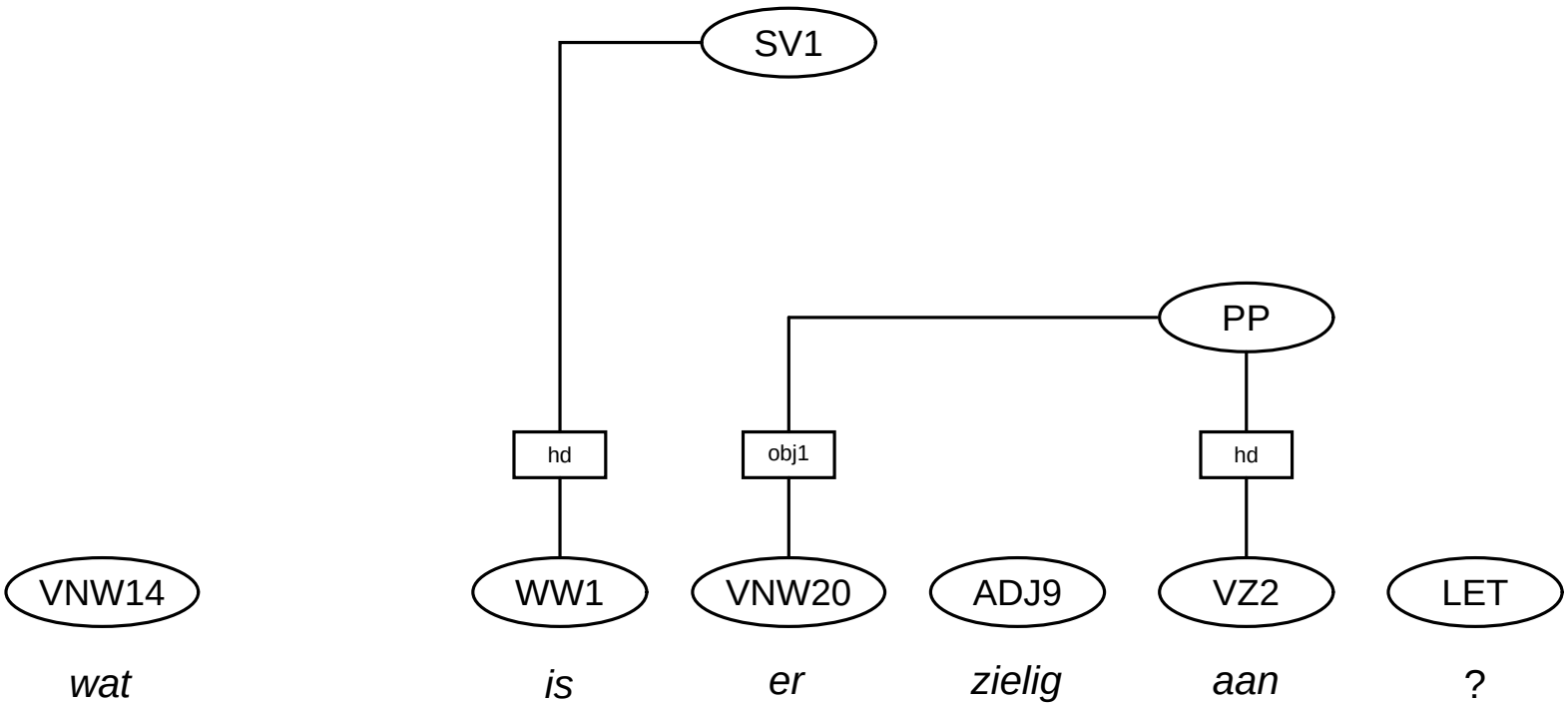
The Spoken Dutch Corpus

Syntactic Annotation



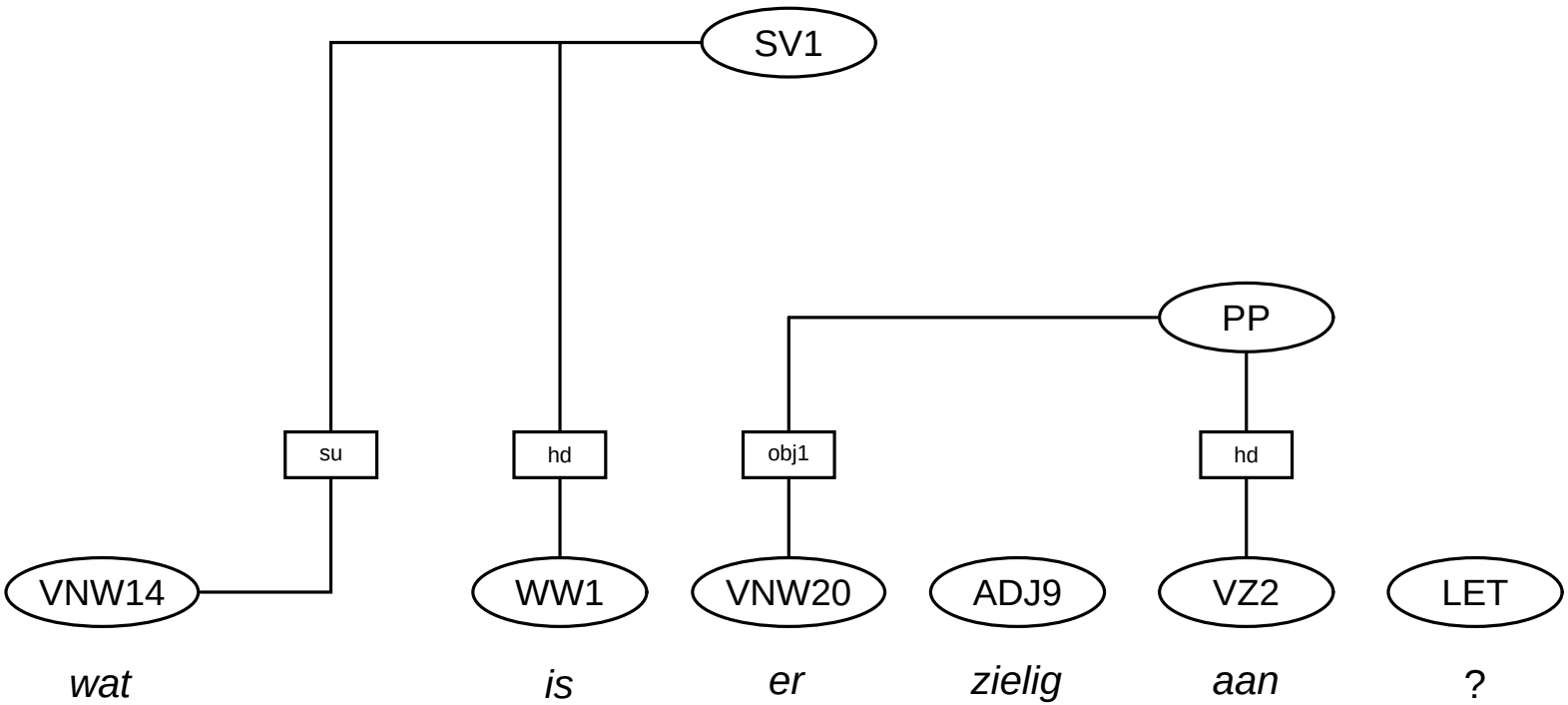
The Spoken Dutch Corpus

Syntactic Annotation



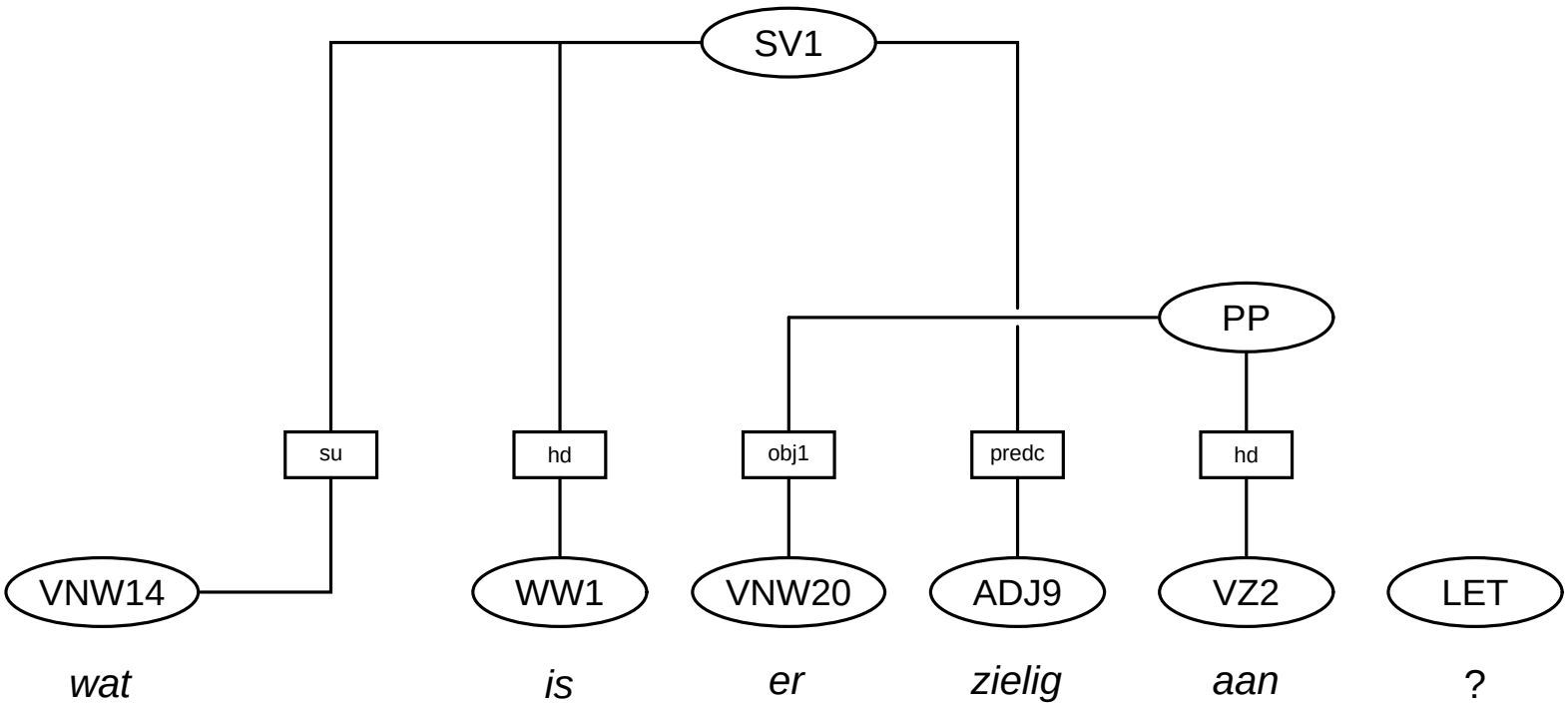
The Spoken Dutch Corpus

Syntactic Annotation



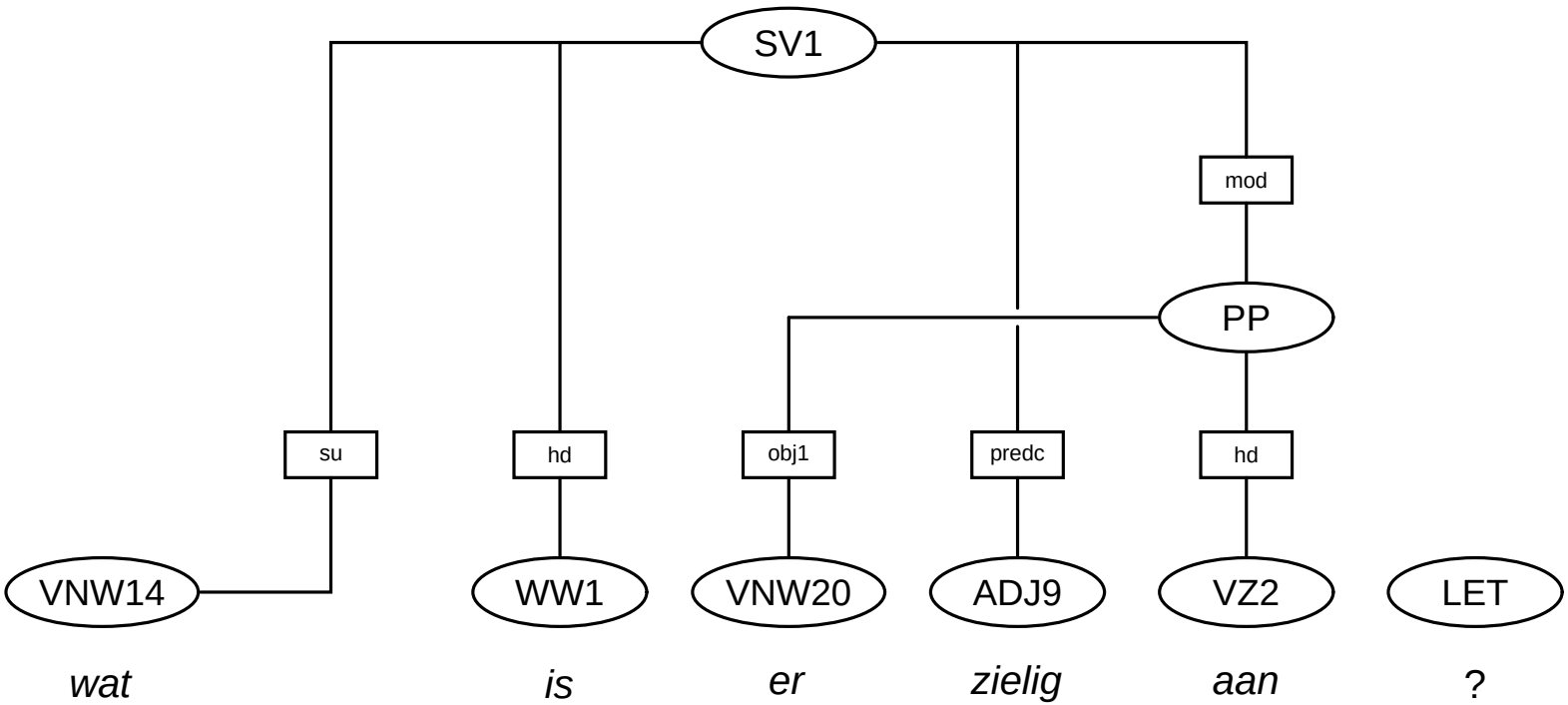
The Spoken Dutch Corpus

Syntactic Annotation



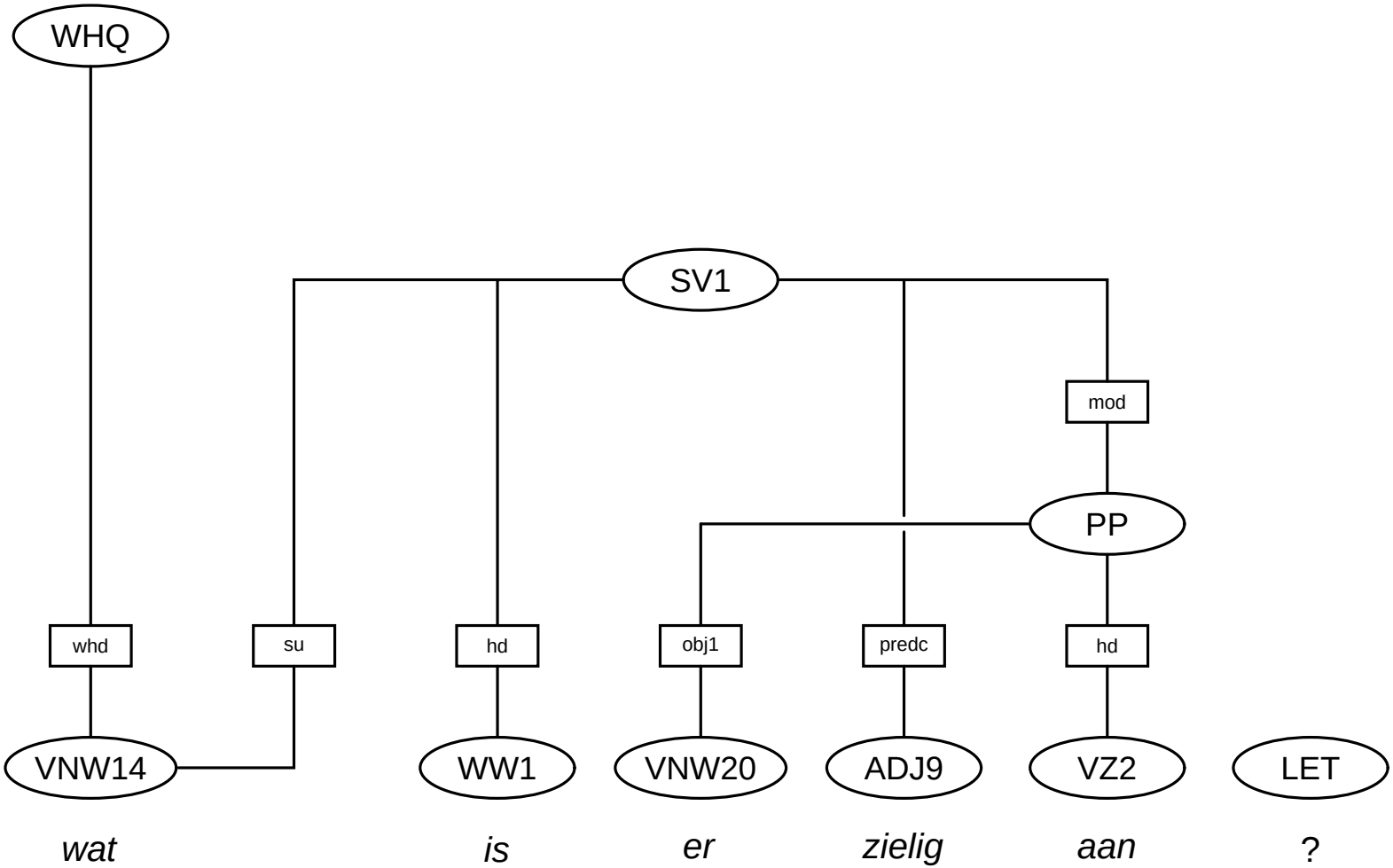
The Spoken Dutch Corpus

Syntactic Annotation



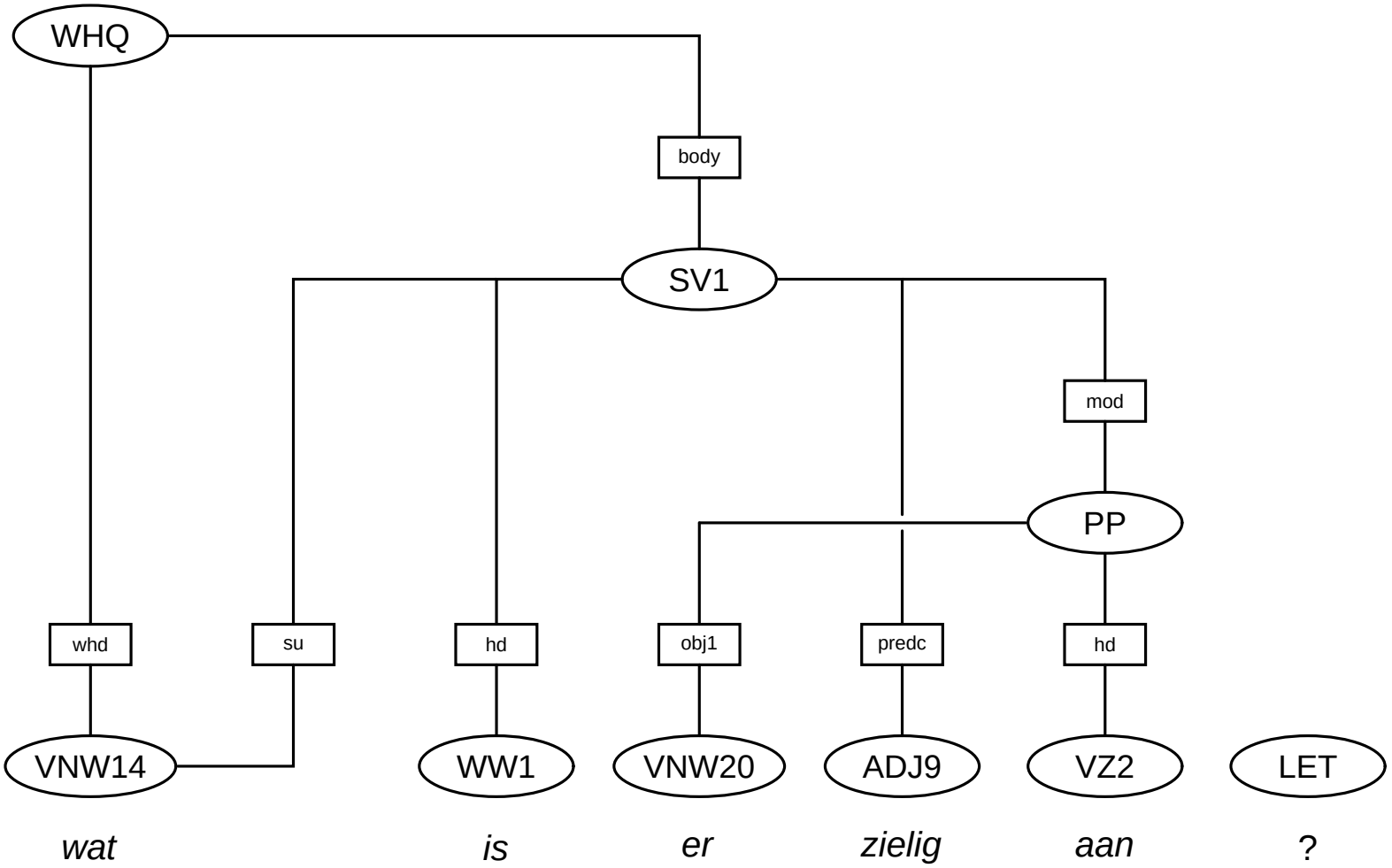
The Spoken Dutch Corpus

Syntactic Annotation



The Spoken Dutch Corpus

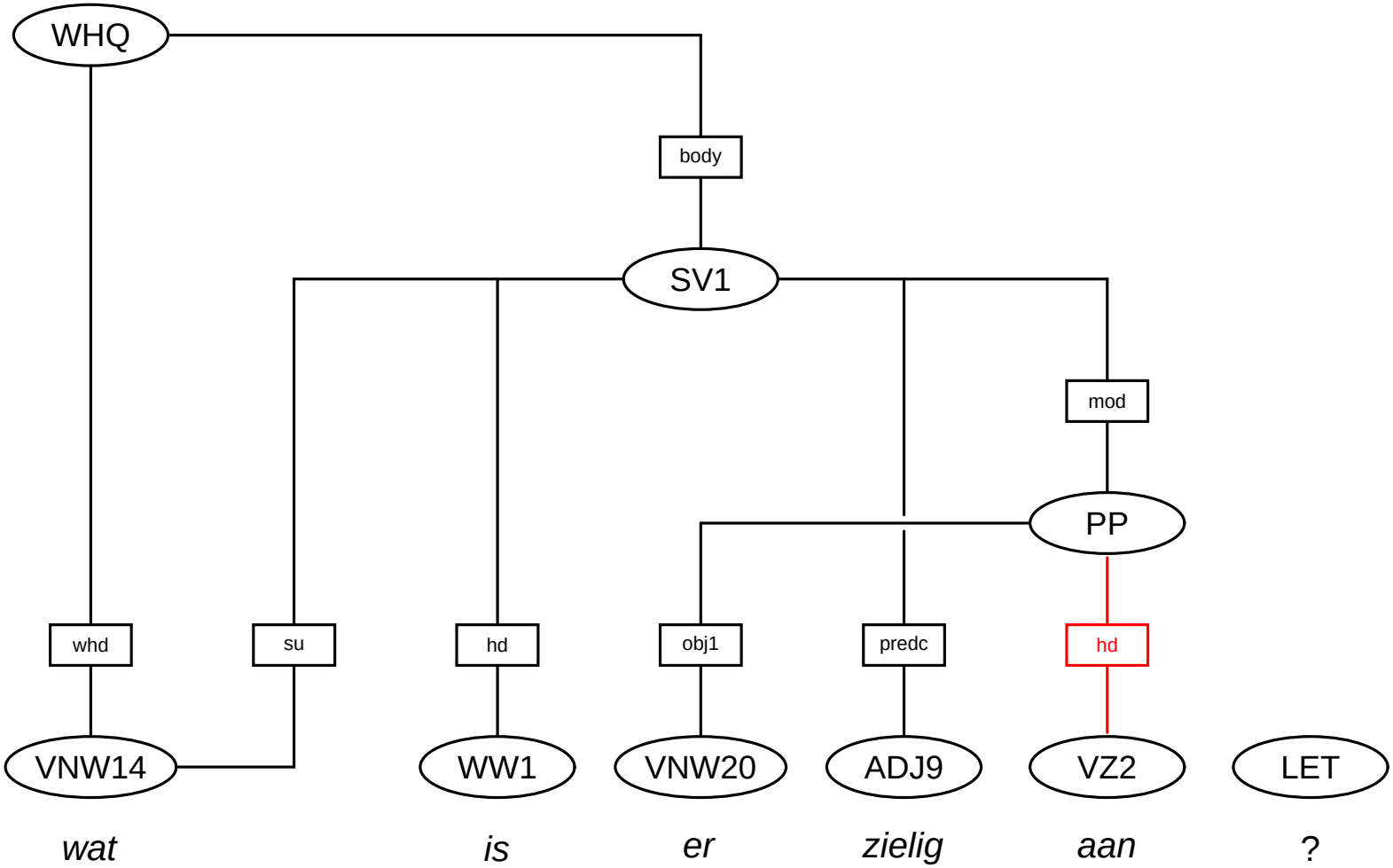
Syntactic Annotation



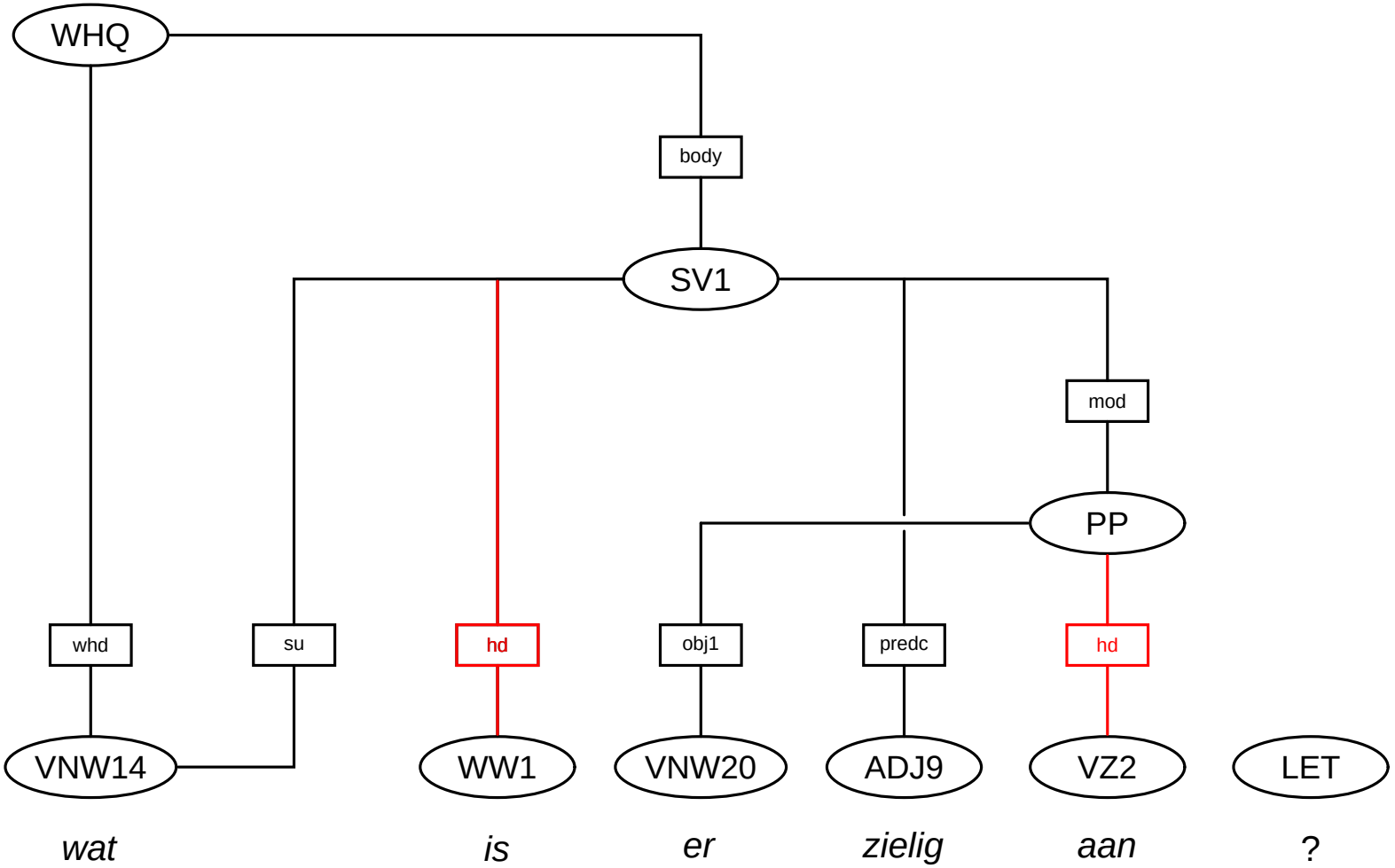
Treebank Extraction

- a function to identify the functor (or head) of every syntactic category
- a function to identify the modifiers of every syntactic category
- a function from syntactic categories to formulas

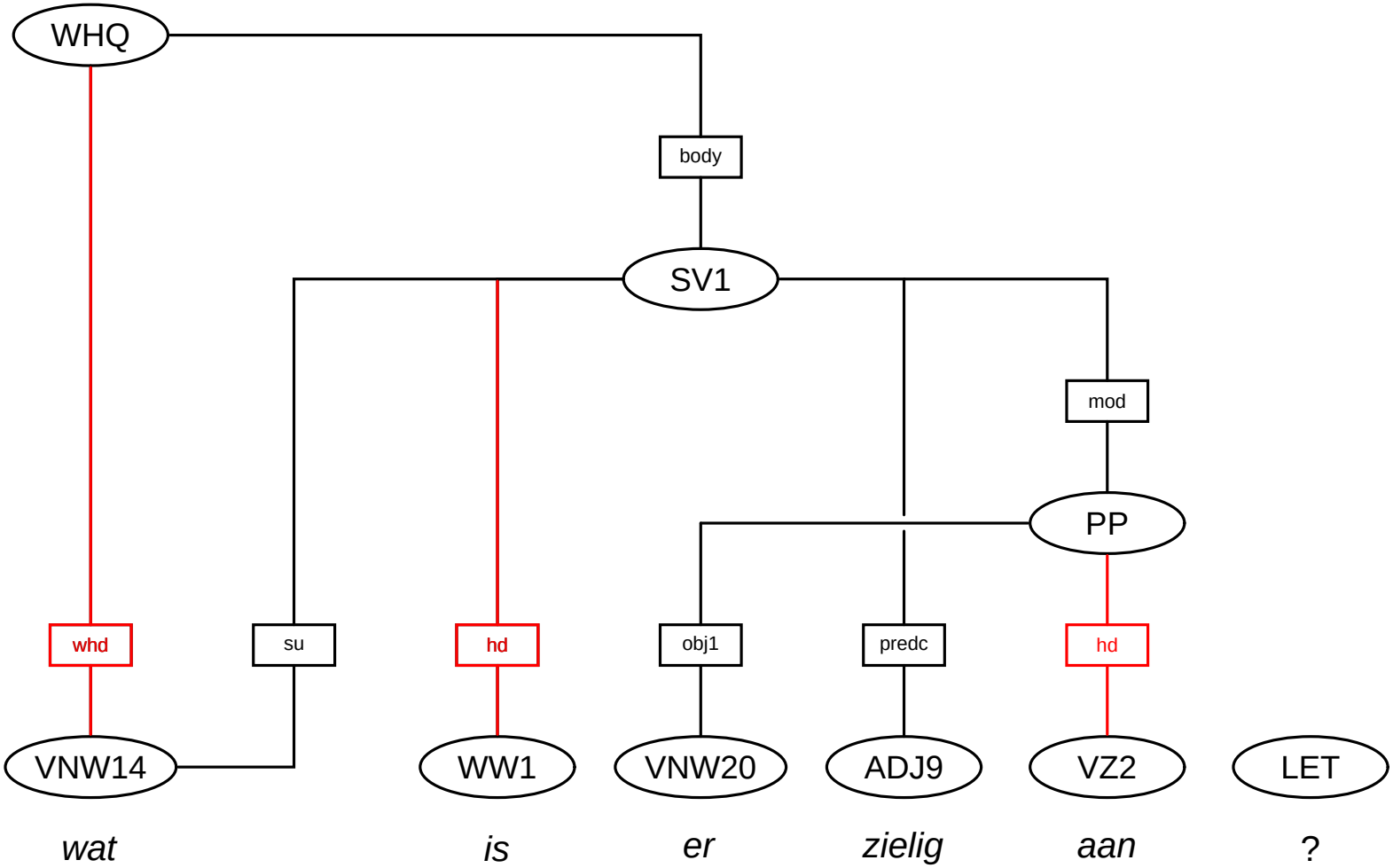
Treebank Extraction: Functors



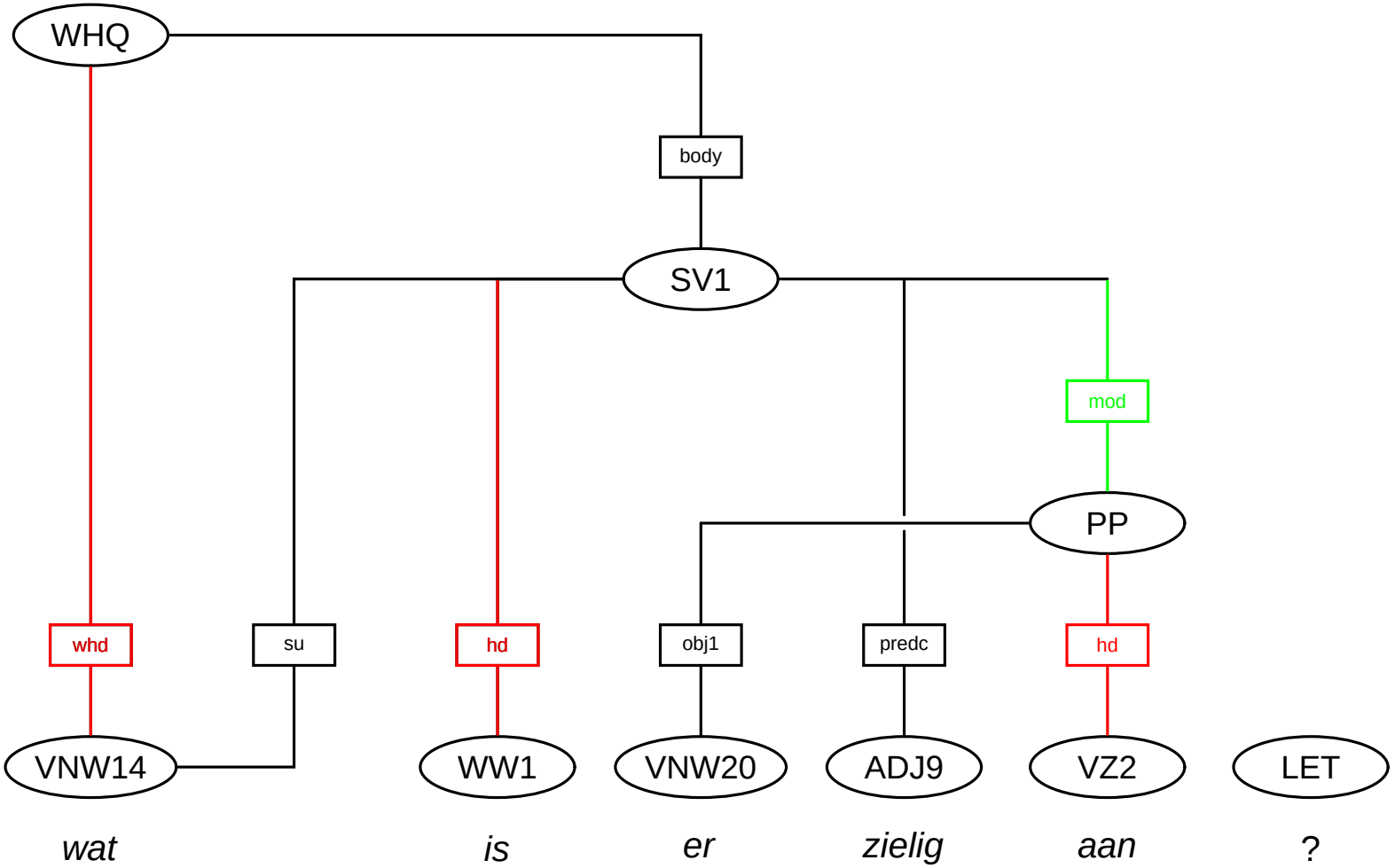
Treebank Extraction: Functors



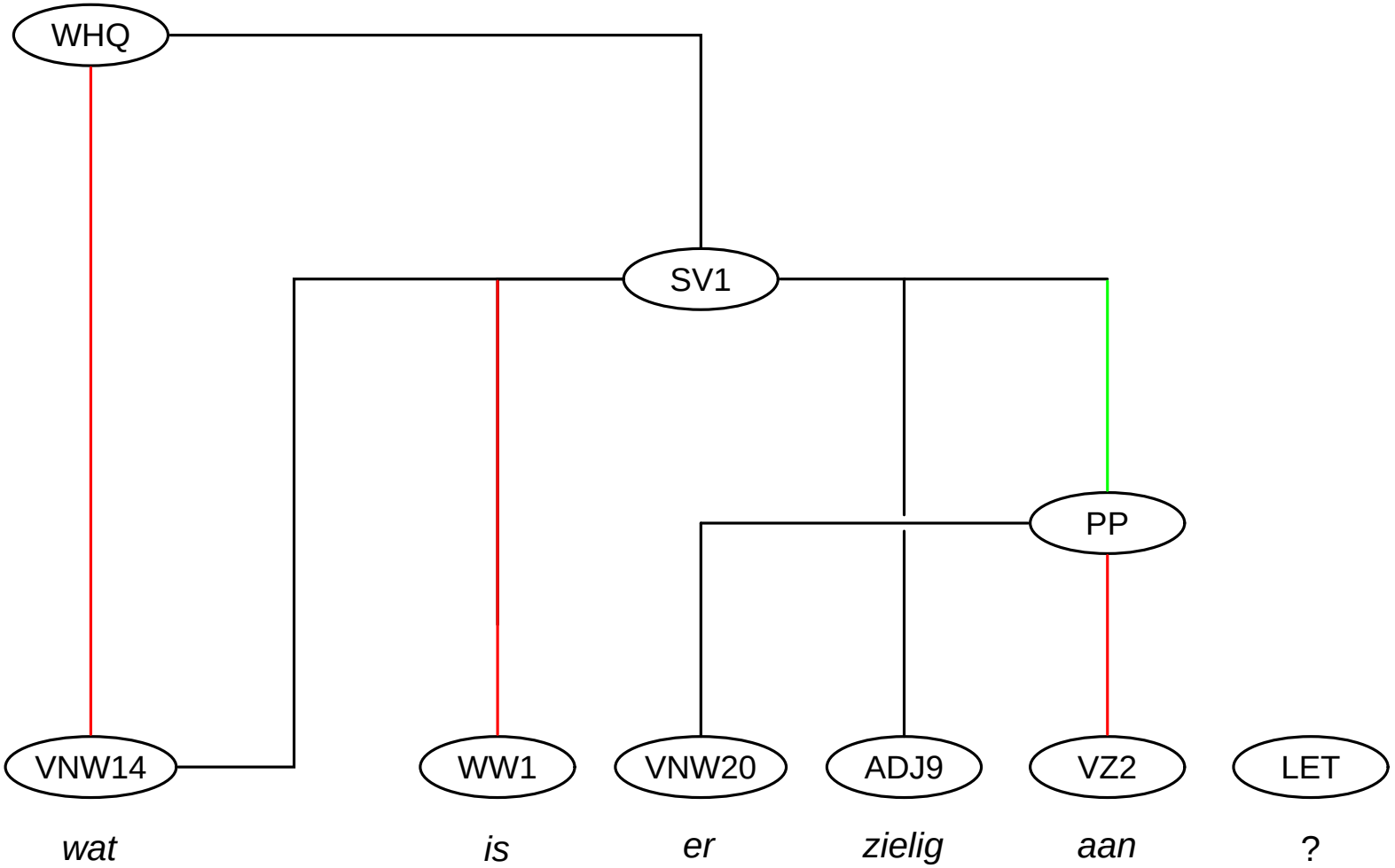
Treebank Extraction: Functors



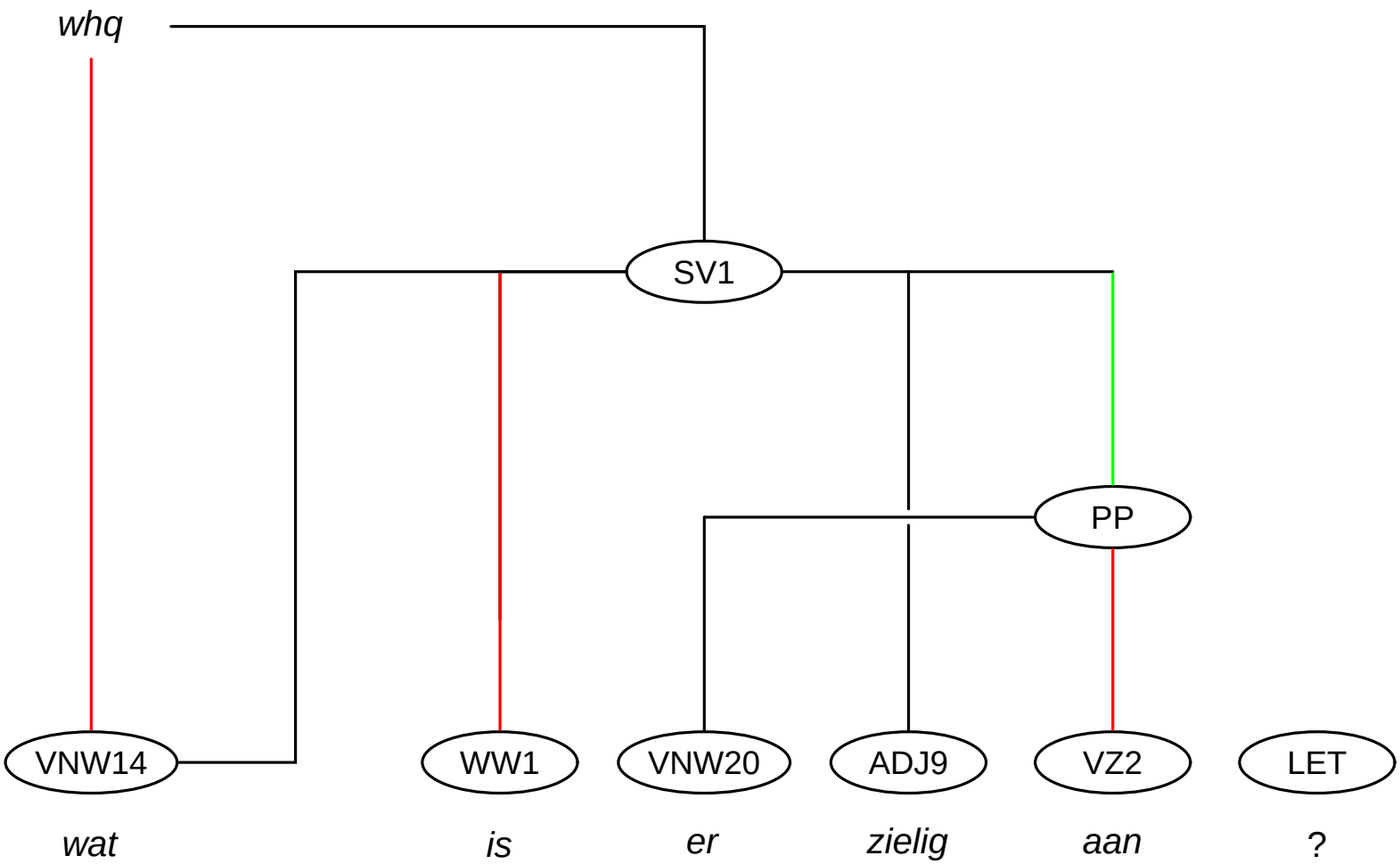
Treebank Extraction: Modifiers



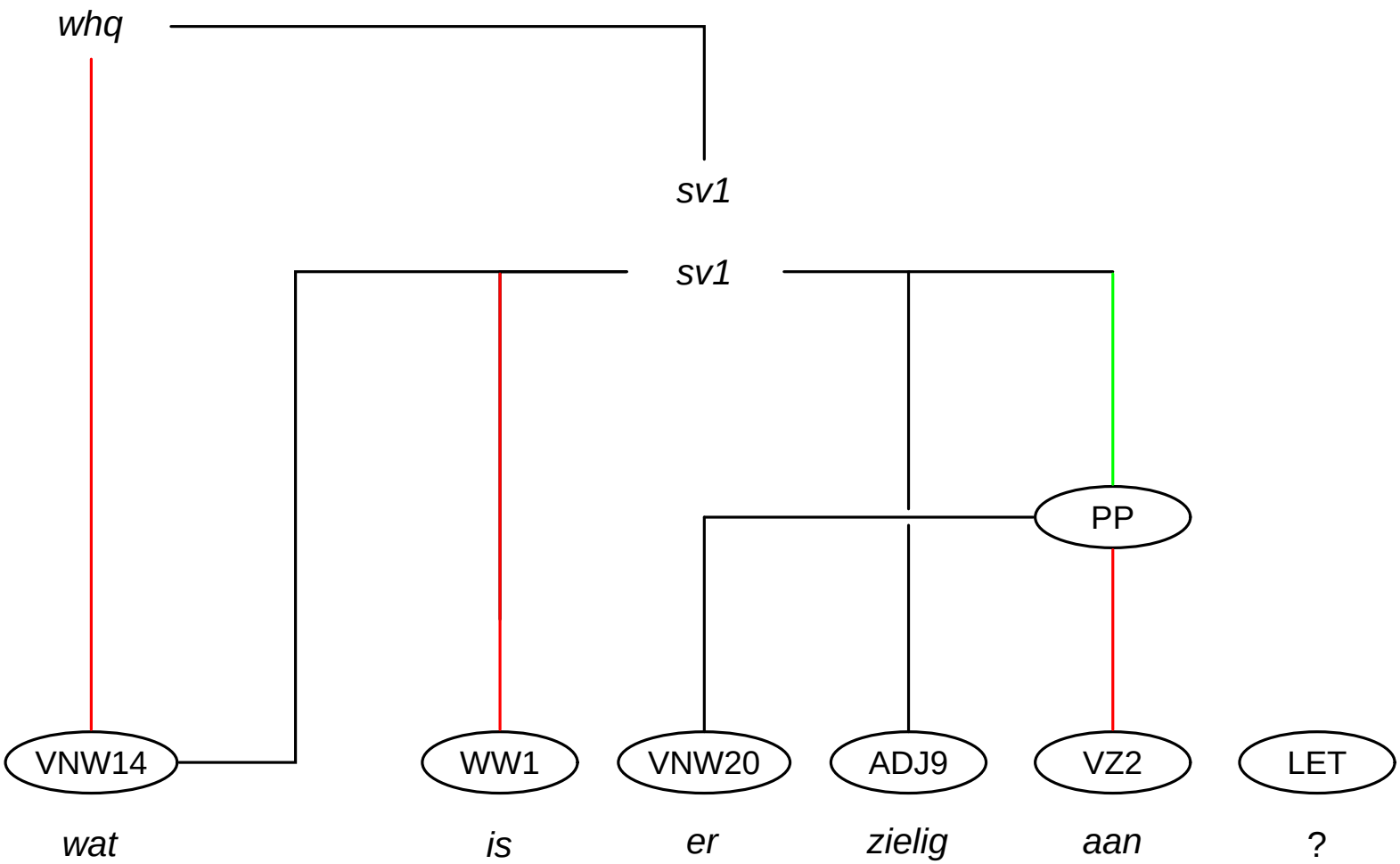
Treebank Extraction



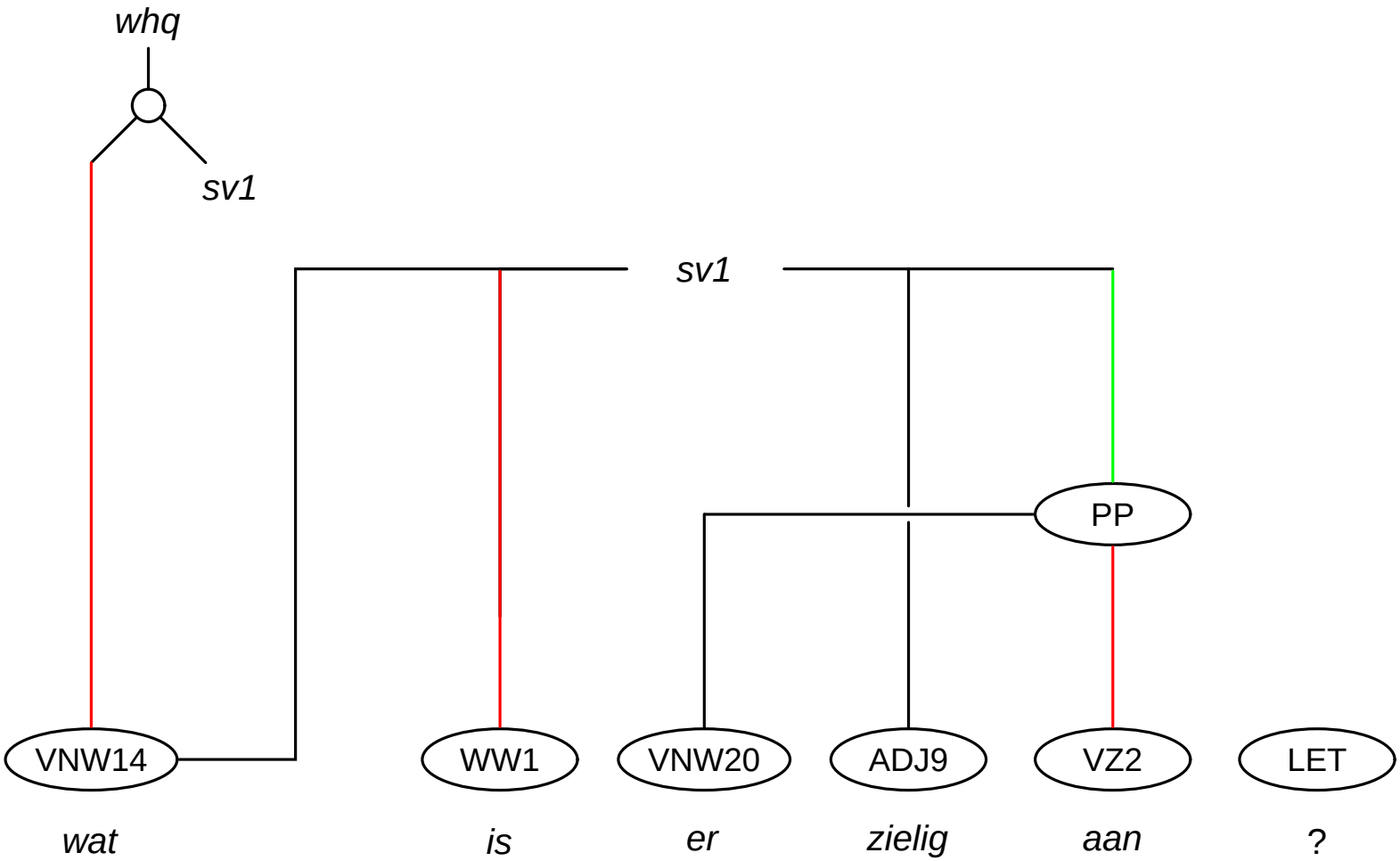
Treebank Extraction



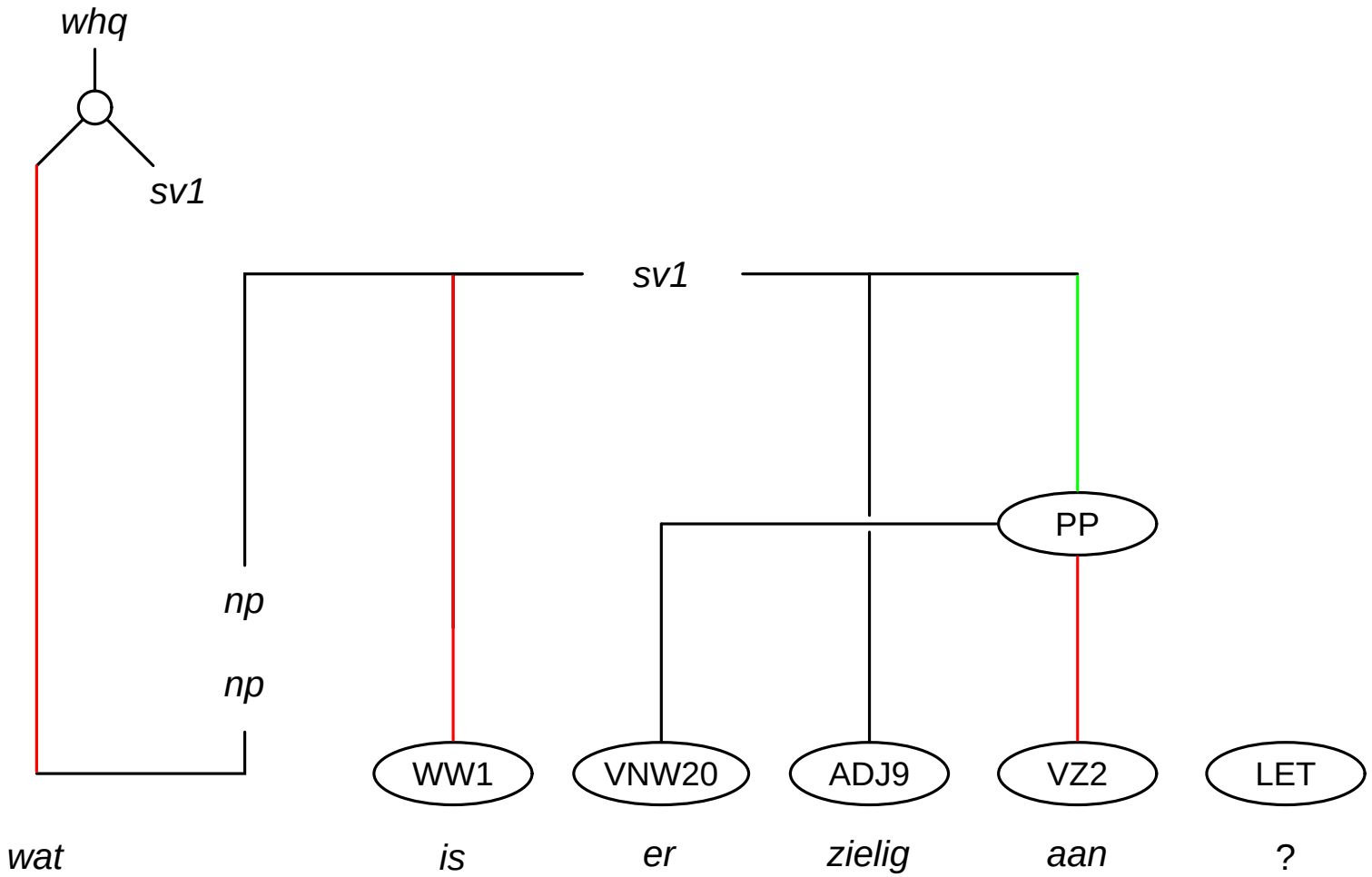
Treebank Extraction



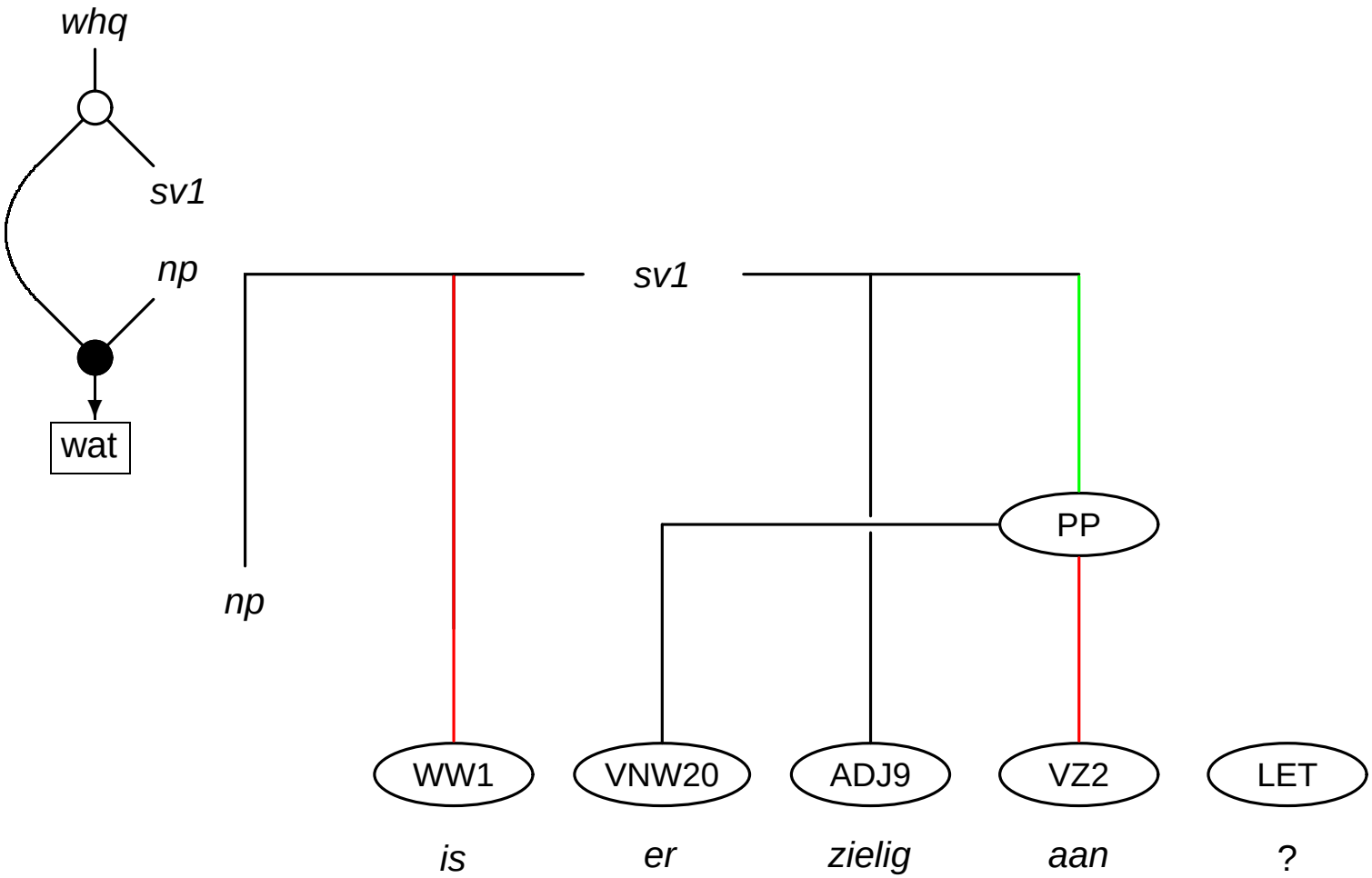
Treebank Extraction



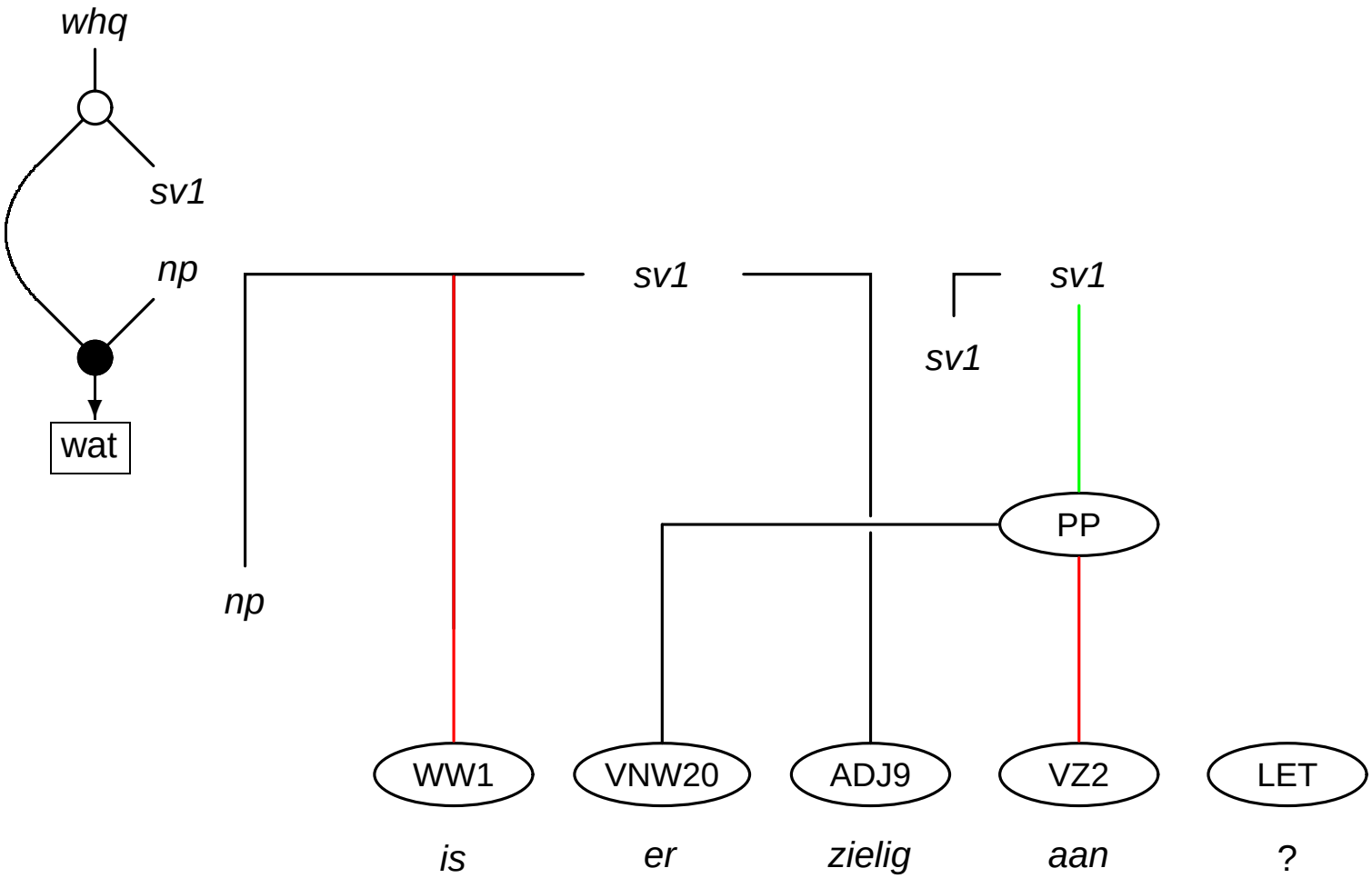
Treebank Extraction



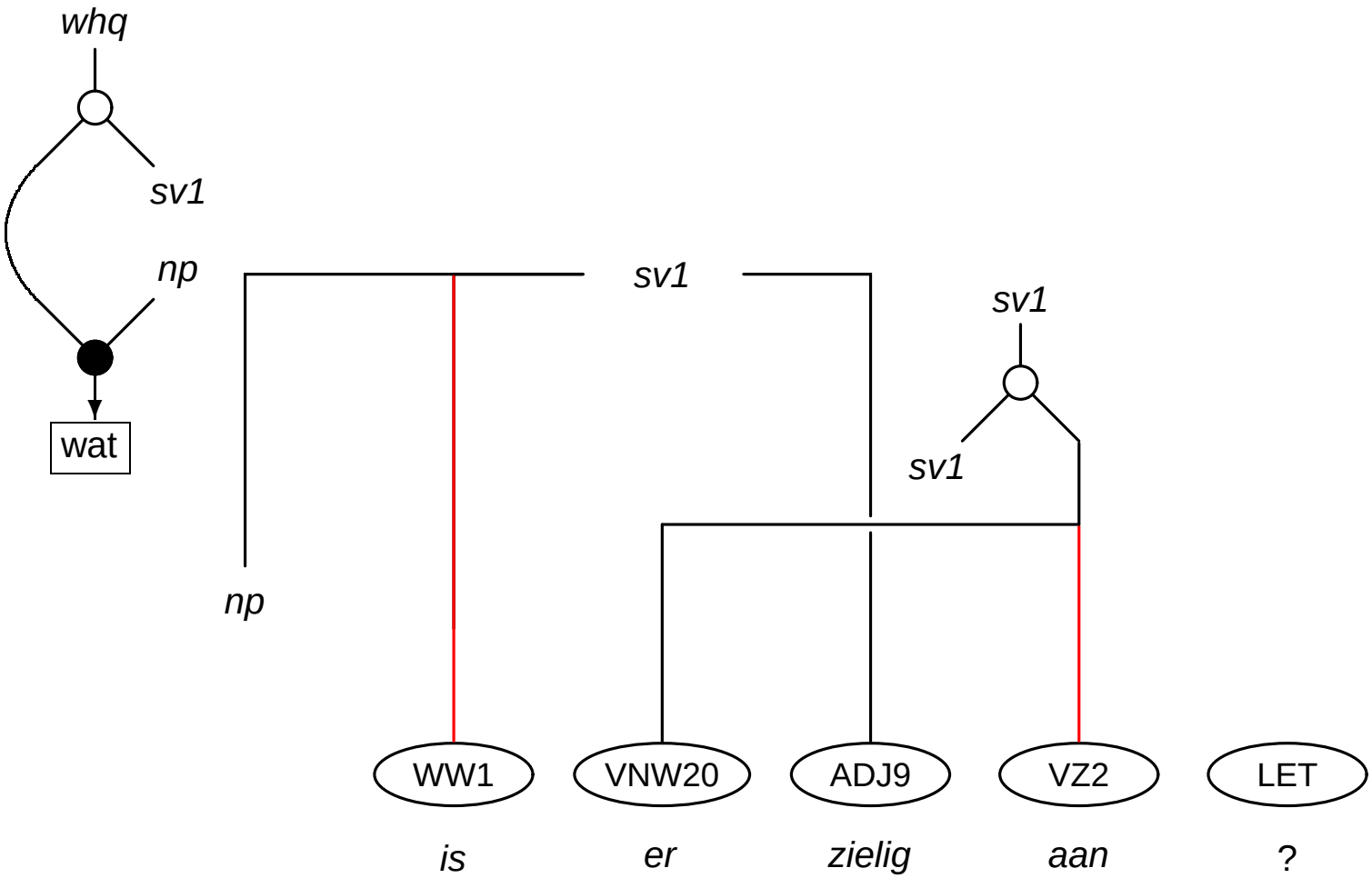
Treebank Extraction



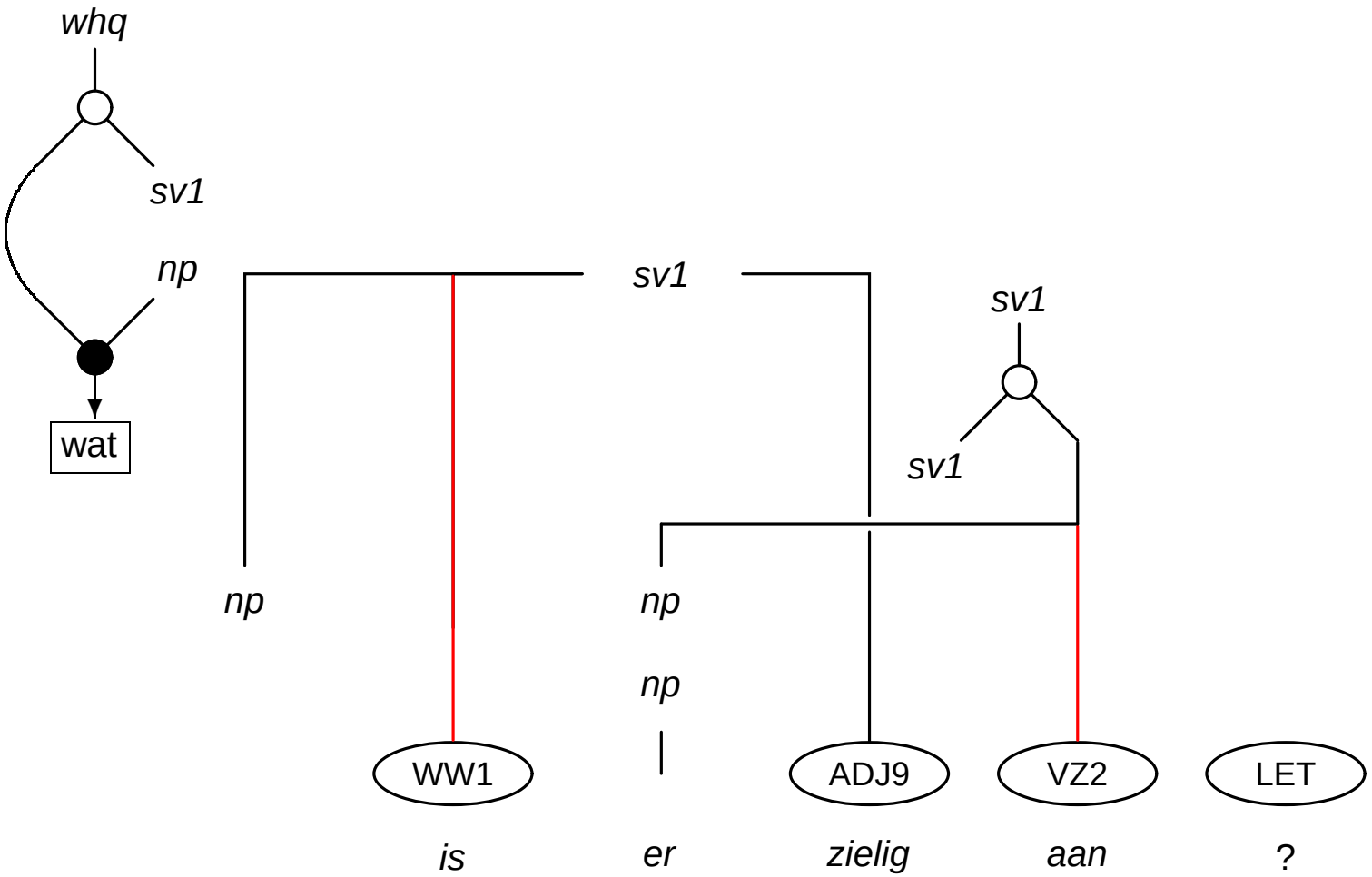
Treebank Extraction



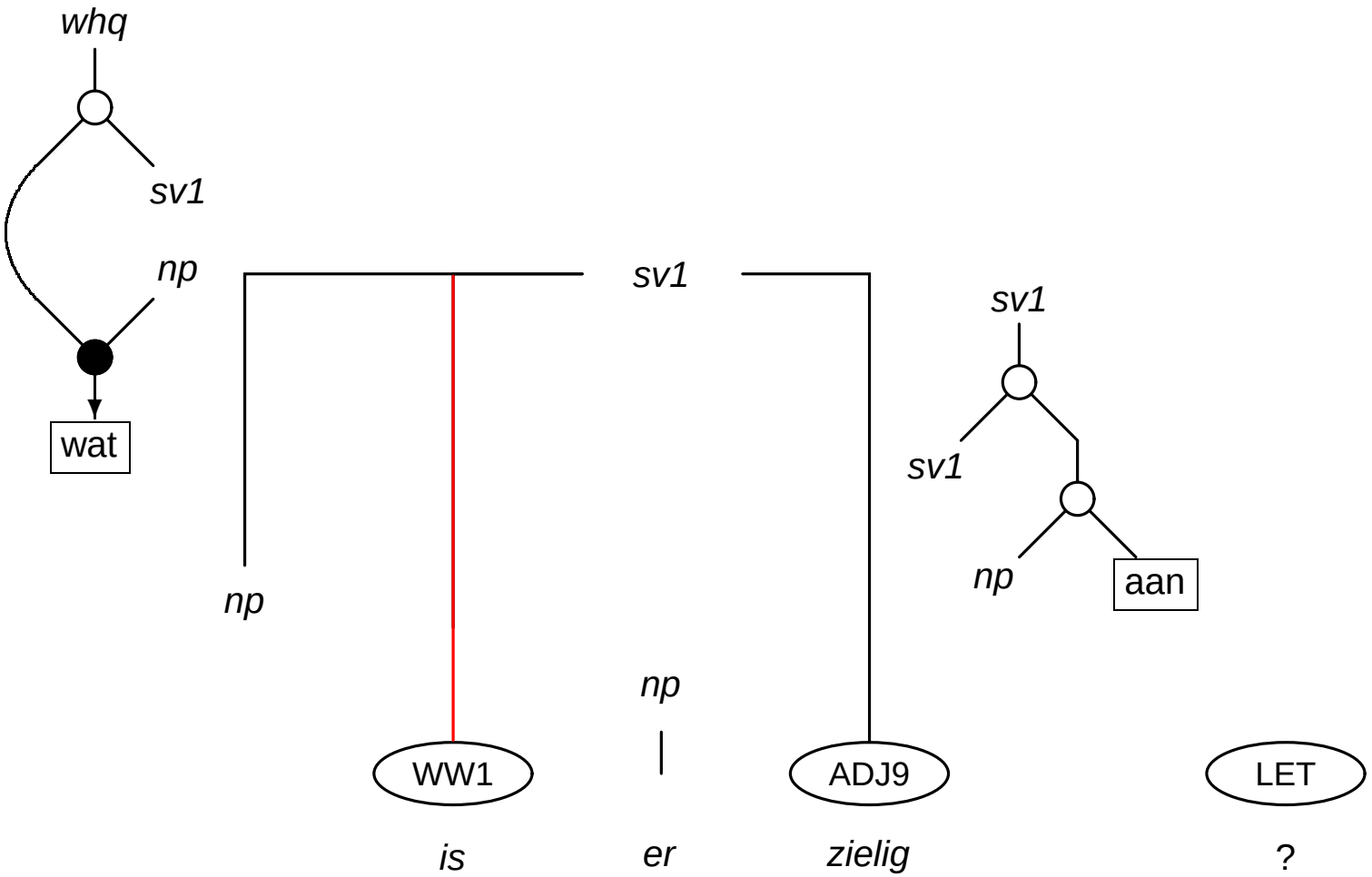
Treebank Extraction



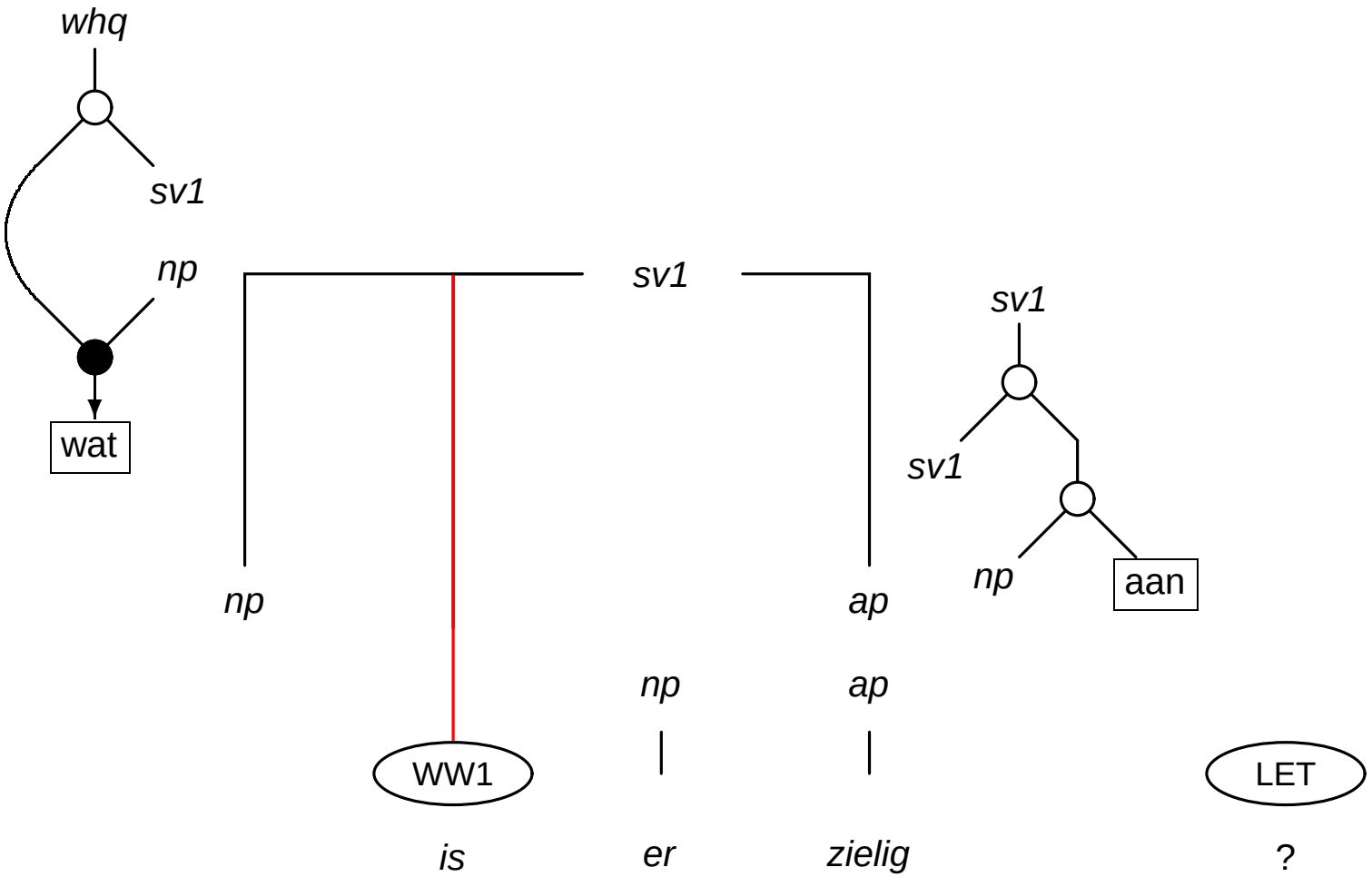
Treebank Extraction



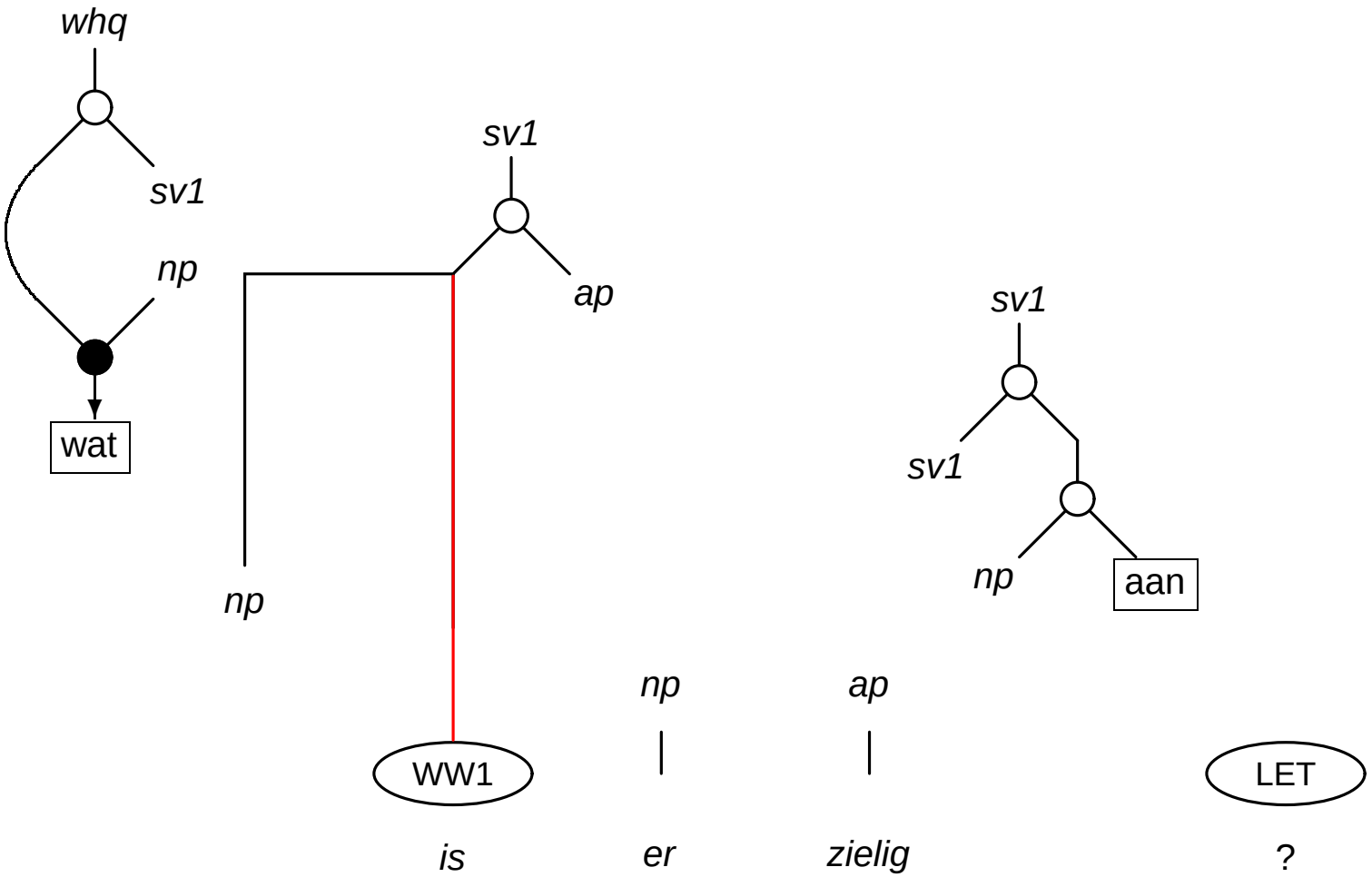
Treebank Extraction



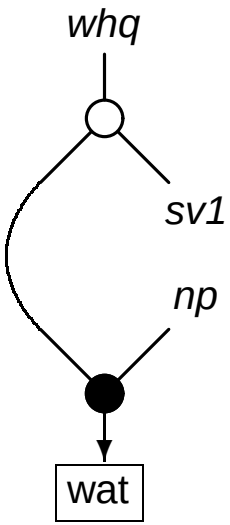
Treebank Extraction

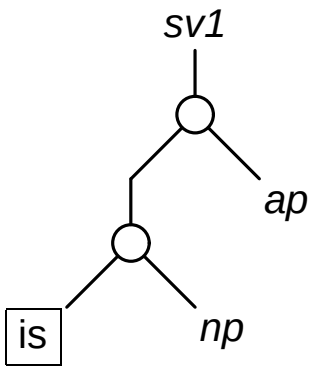


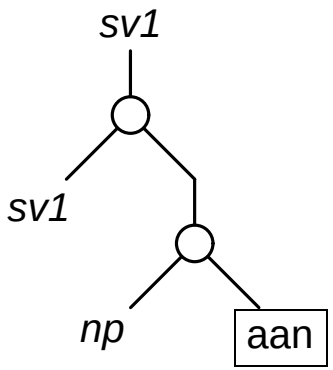
Treebank Extraction

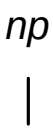


Treebank Extraction

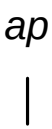








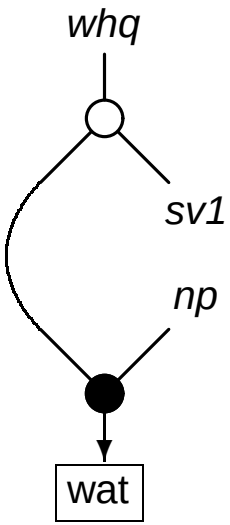
er

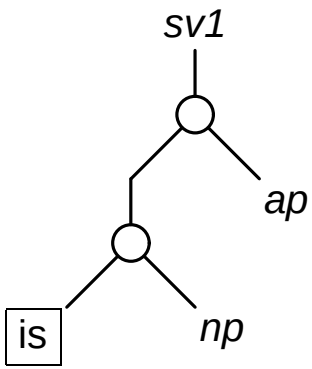


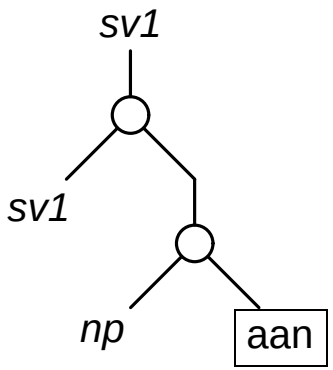
zielig

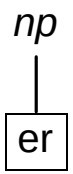
?

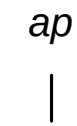
Treebank Extraction







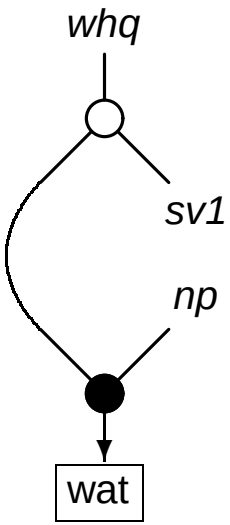


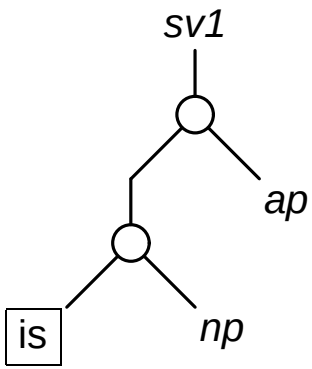


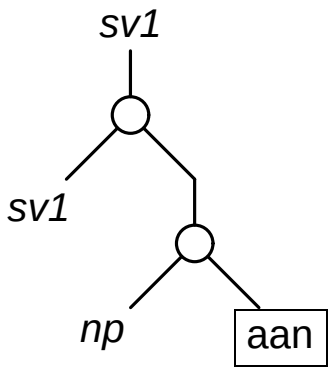
zielig

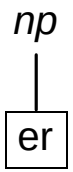
?

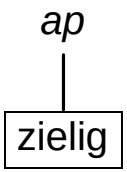
Treebank Extraction





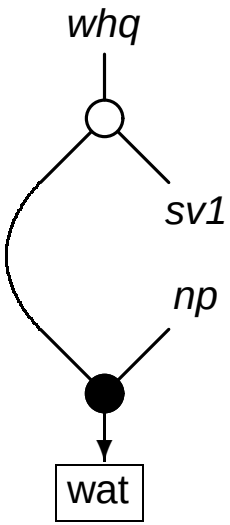


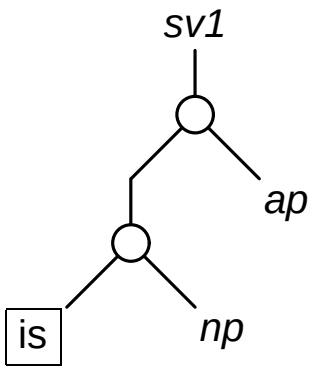


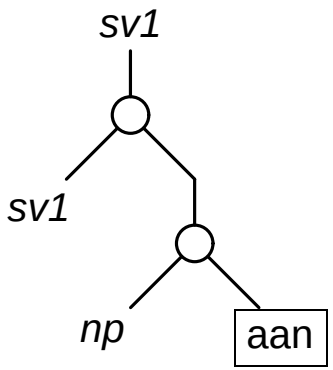


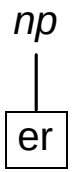
?

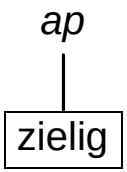
Treebank Extraction

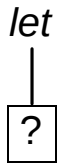




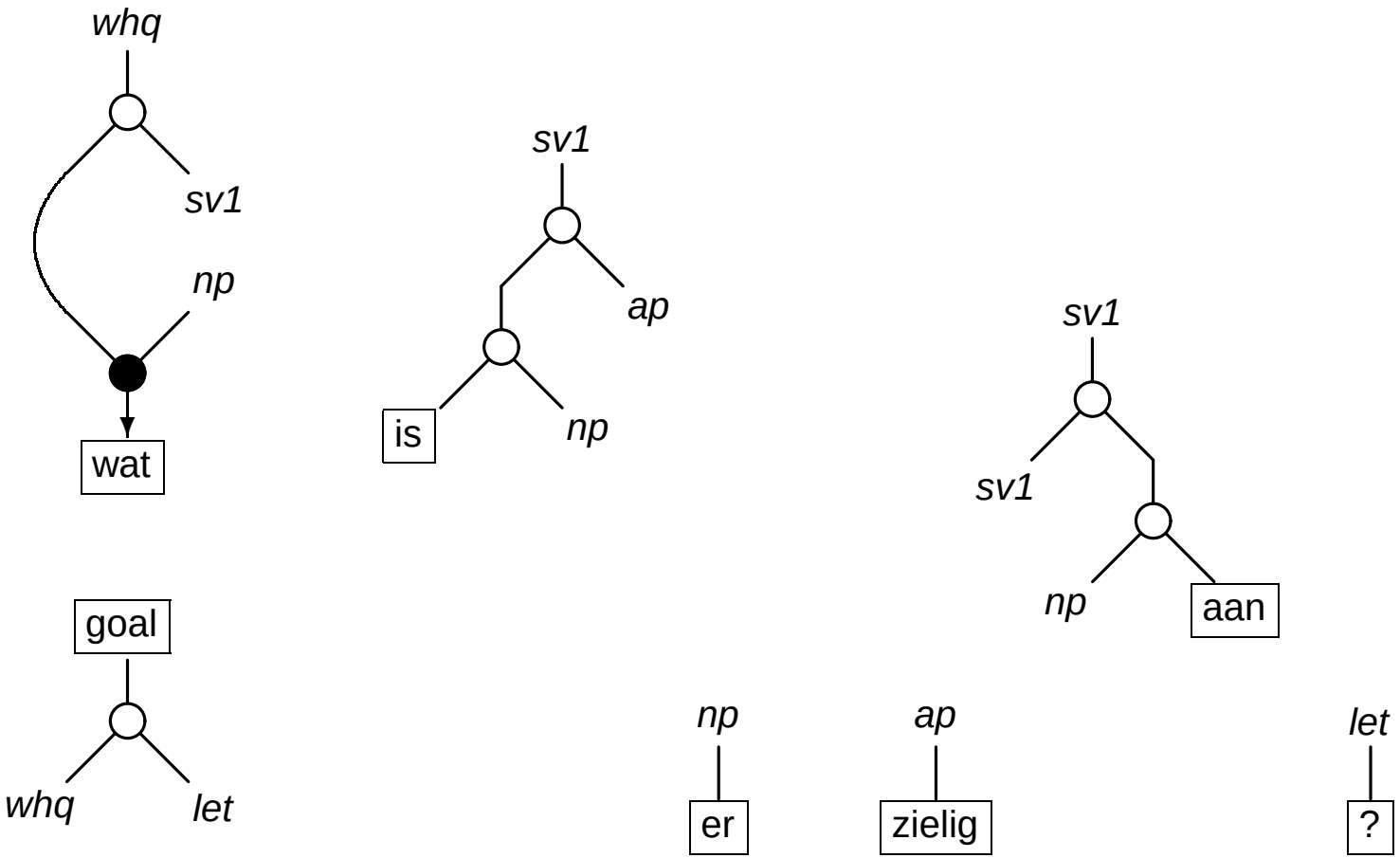








Treebank Extraction

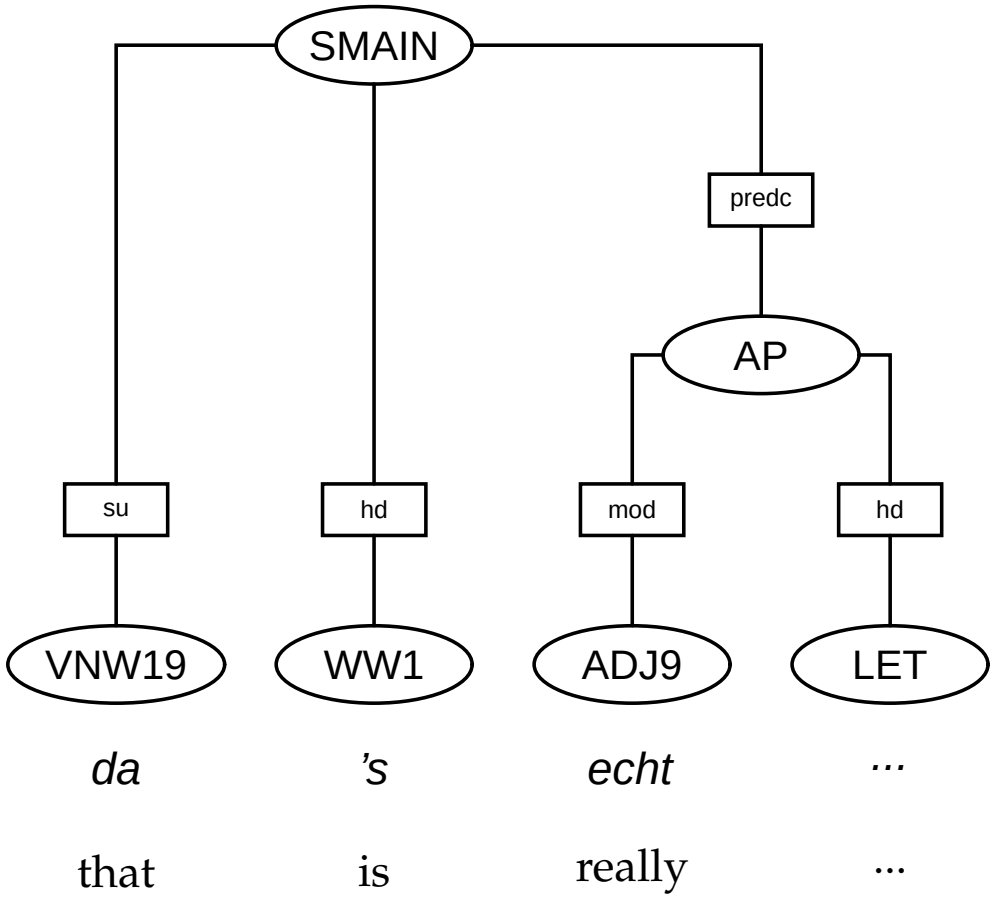


Preprocessing

- ‘isolated’ words like interpunction symbols, but also hesitation marks like ‘uh’ are almost always given the same tags.
- filtering out the isolated vertices gives us a corpus containing 87.404 sentences (out of 114.801) and 794.872 words (out of 1.002.098).
- we use the filtered corpus for the treebank extraction but we will see that it is possible to do this filtration automatically.

Preprocessing

Interpunction With Grammatical Role



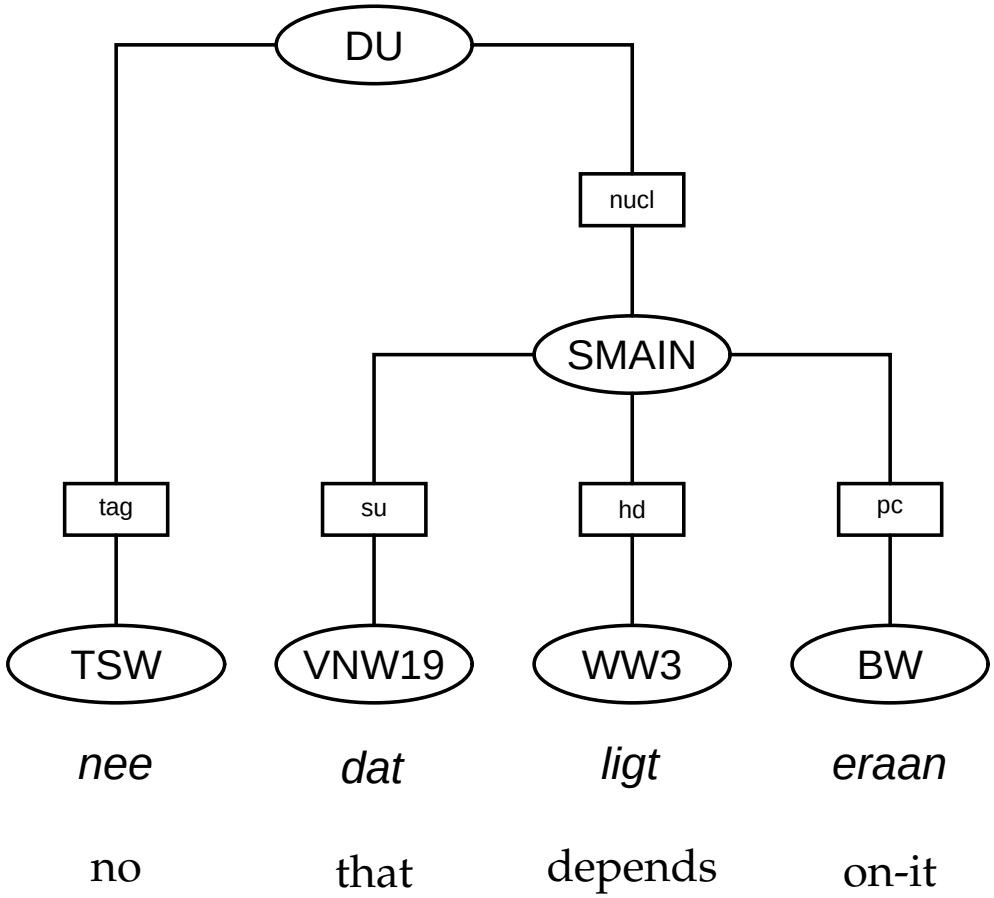
Refinement

Some of the vertex labels of the Spoken Dutch Corpus, like

- DU (discourse unit),
- CONJ (conjunction),
- LIST and
- MWU (merged word unit, a category assigned to multi-word names and fixed expressions)

are not really grammatical categories.

Refinement


DU
nucl
SMAIN
tag
su
hd
pc
TSW
VNW19
WW3
BW
nee
dat
ligt
eraan
no
that
depends
on-it

Refinement

(1)

wij	*hadden*	*nog*	[	*meetkunde*	*en*	*algebra*	$]_{CONJ}$
we	had	still	[	geometry	and	algebra	$]_{CONJ}$

'we still had geometry and algebra'

(2)

koffie	[*geen*	*melk*	*geen*	*suiker*	$]_{LIST}$
coffee	[no	milk	no	sugar	$]_{LIST}$

'coffee, no milk, no sugar'

(3)

[*boulimia*	*nervosa*	$]_{MWU}$	*heet*	*'t*
[boulimia	nervosa	$]_{MWU}$	called	it

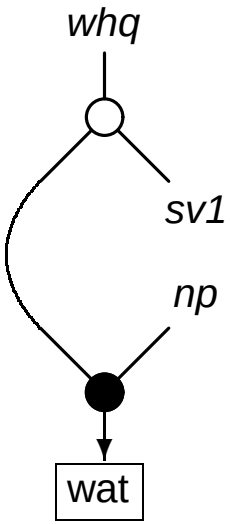
'it is called boulimia nervosa'

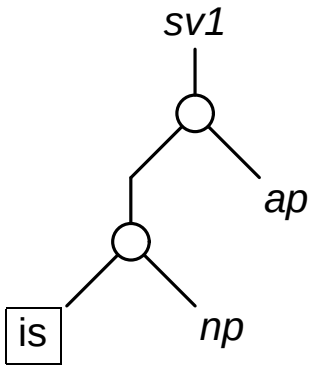
Reducing the Lexicon Size

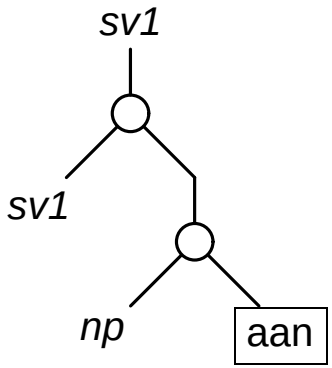
- Simply keeping all syntactic categories of the Spoken Dutch Corpus gives us a very large treebank, containing 6.817 different lexical types.
- As a first reduction, we map all different sentence types to *s* and AP to noun modifier; this reduces the size of the treebank to 3.539 lexical types.

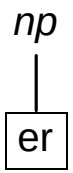
Reducing the Lexicon Size

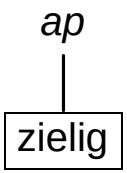
Previous Result





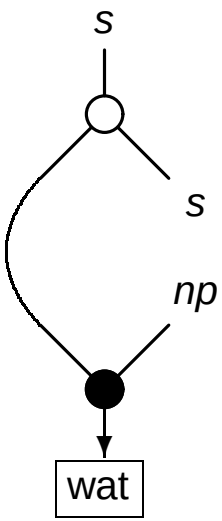


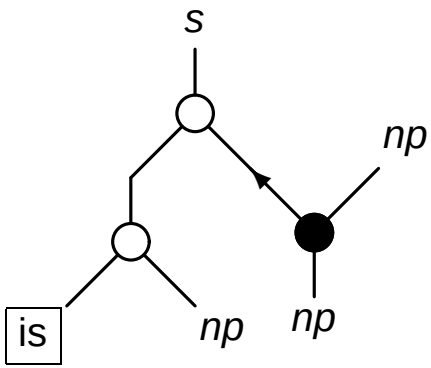


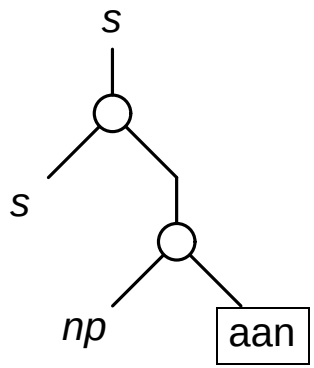


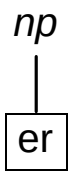
Reducing the Lexicon Size

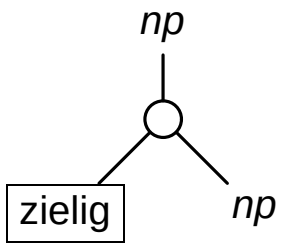
New Result





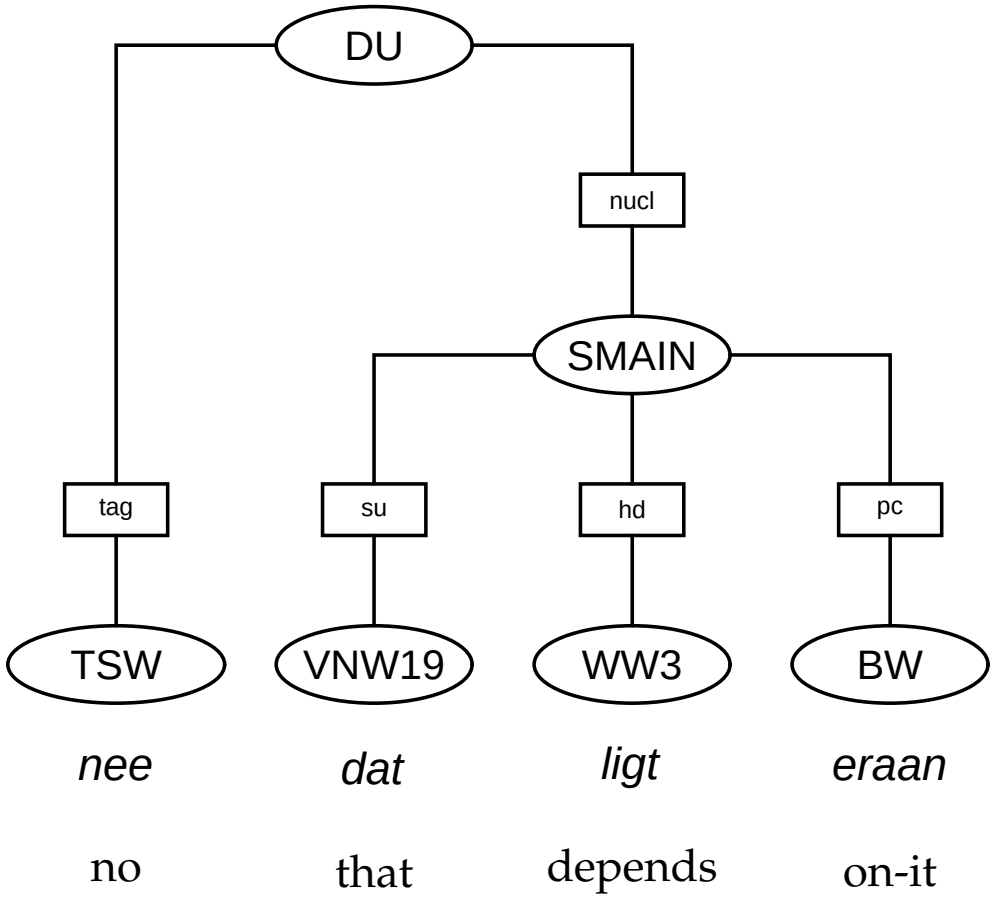






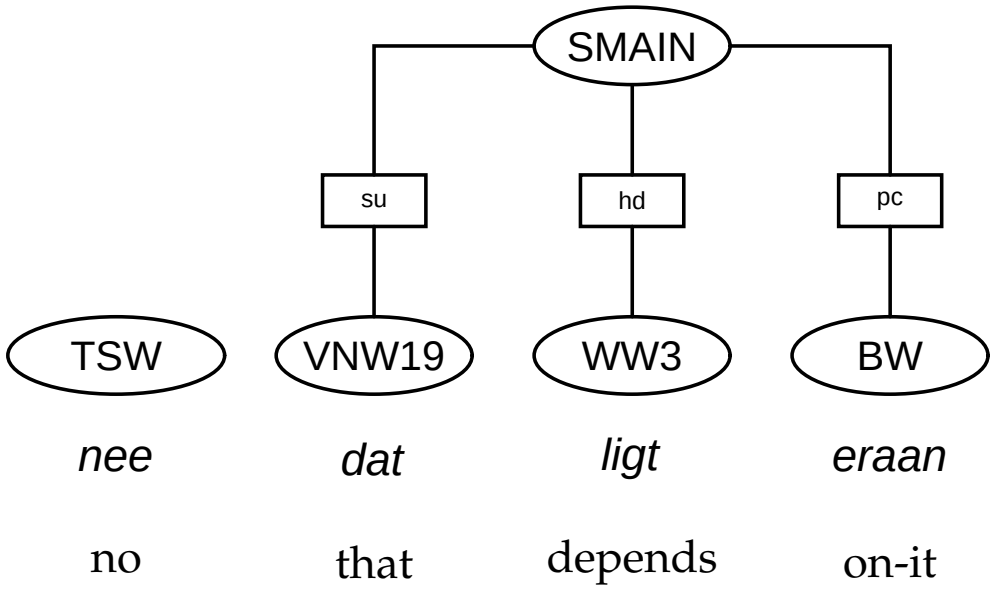
Reducing the Lexicon Size

Splitting Discourse Units



Reducing the Lexicon Size

Splitting Discourse Units



Splitting Discourse Units

(4)

deze	*onderaan*	*hier*
this	at the bottom	here

‘this one at the bottom here’

(5)

mama	*dronken*
mother	drunk

‘mother (is) drunk’

(6)

positief	*tenzij*
positive	unless

‘I am of positive opinion, unless ...’

Further Reductions

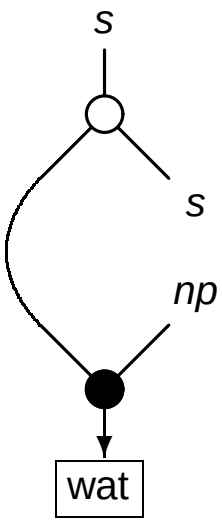
- Splitting discourse units $\rightsquigarrow$ 2.201 lexical formulas
- Identifying even more atomic formulas $\rightsquigarrow$ 1.962 lexical formulas
- **AB** lexicon $\rightsquigarrow$ 1.761 lexical formulas
- **LP** lexicon $\rightsquigarrow$ 1.137 lexical formulas

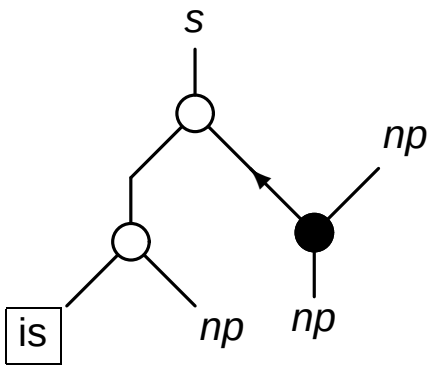
Keeping track of discontinuity

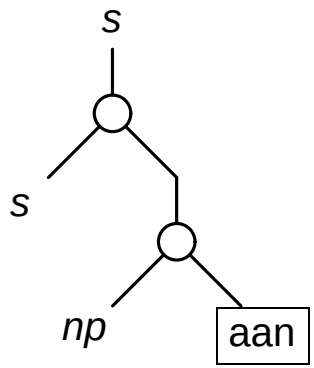
- Note that for the moment, our lexical entries only indicate whether their arguments and modifiers are generally to the left or generally to the right.
- For parsing, it would be useful to distinguish between *directly* to the left (right) and *at a distance* to the left (right).
- Adding this information increases the 2.201 lexical formulas of the split lexicon to 4.744 lexical formulas

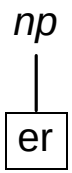
Keeping Track of Discontinuity

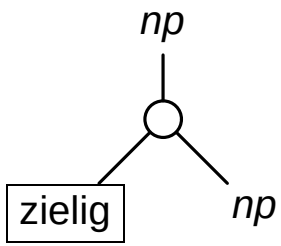
Previous Lexical Trees





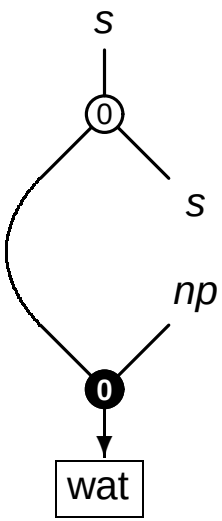


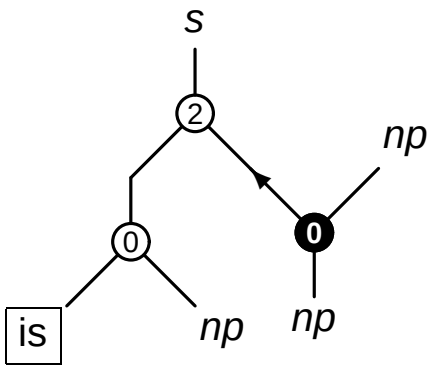


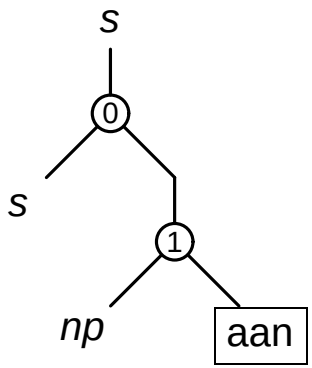


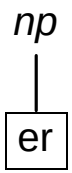
Keeping Track of Discontinuity

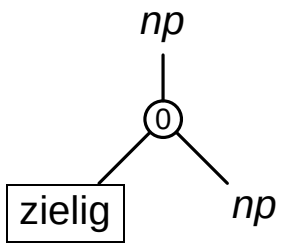
Lexical Trees With Discontinuity Information











Lexical Lookup

- We have seven treebanks, with between 6.817 and 1.137 different lexical trees and with many hundreds of trees possible for several frequent words.
- How are we going to find the correct sequence of trees?

Maximum Entropy Models

- We look at the information provided by the surrounding words.
- During the training phase, the model determines which information is most useful in predicting the correct supertag.
- During the evaluation phase, we predict the best sequence according to our model.

Features

zullen	we	ze	paginagewijs	afhandelen
ww2	vnw1	vnw3	adj9	ww4
$(s/(np\backslash s))/np$	np	?		

Results

	Experiment	Formulas	Result
1	Basic	6.817	70.61%
2	Discontinuous	4.744	75.24%
3	Compact	3.539	72.06%
4	Split	2.201	77.13%
5	Very compact	1.962	77.83%
6	AB	1.761	77.52%
7	LP	1.137	80.50%

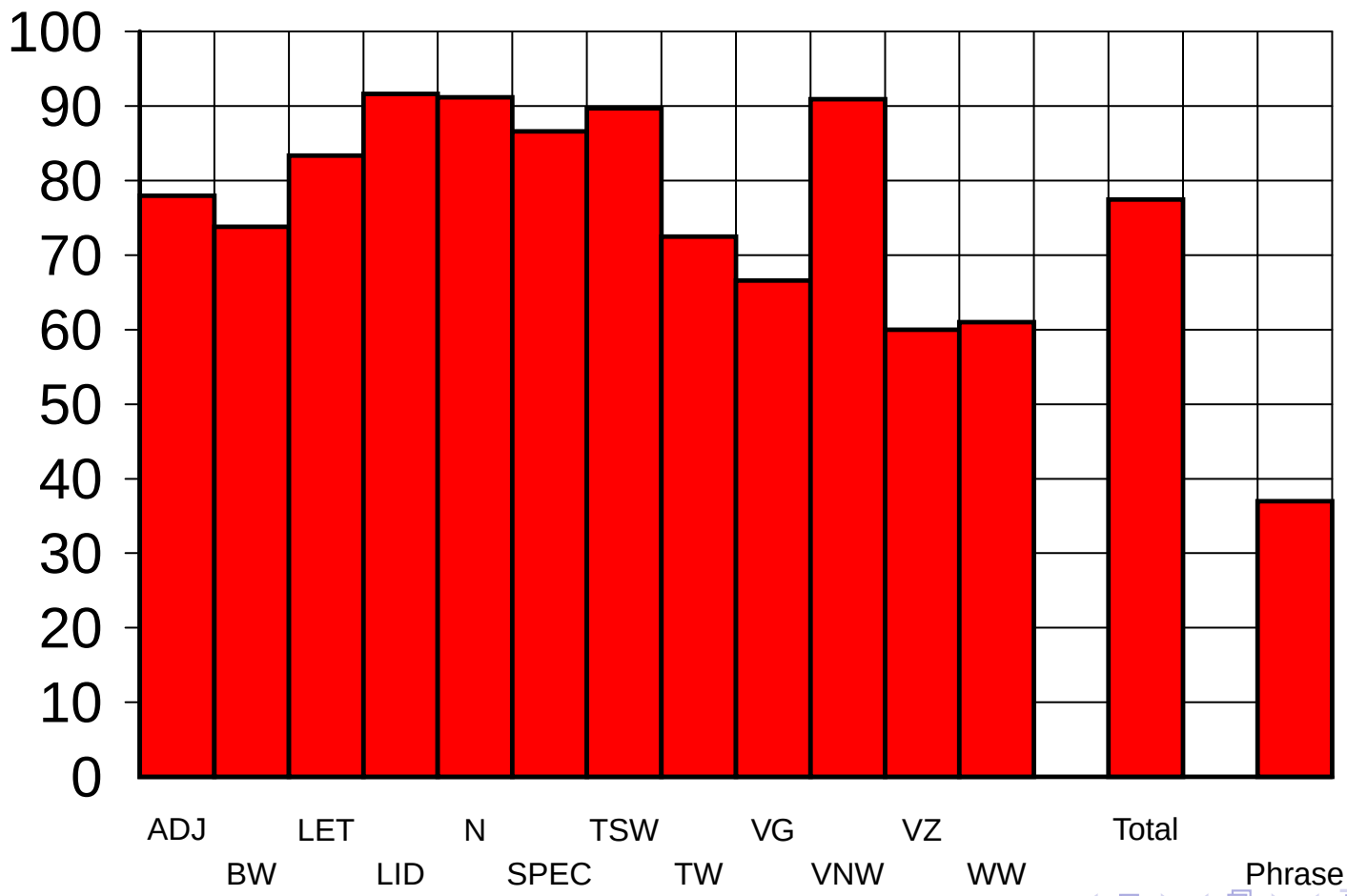
Detecting Isolated Vertices

- a separate model was trained to detect isolated vertices automatically, using simply POS tag of the current and surrounding words
- this received a 98.35% success rate
- remaining errors are due to inconsistencies in the annotation or difficult to detect self-corrections

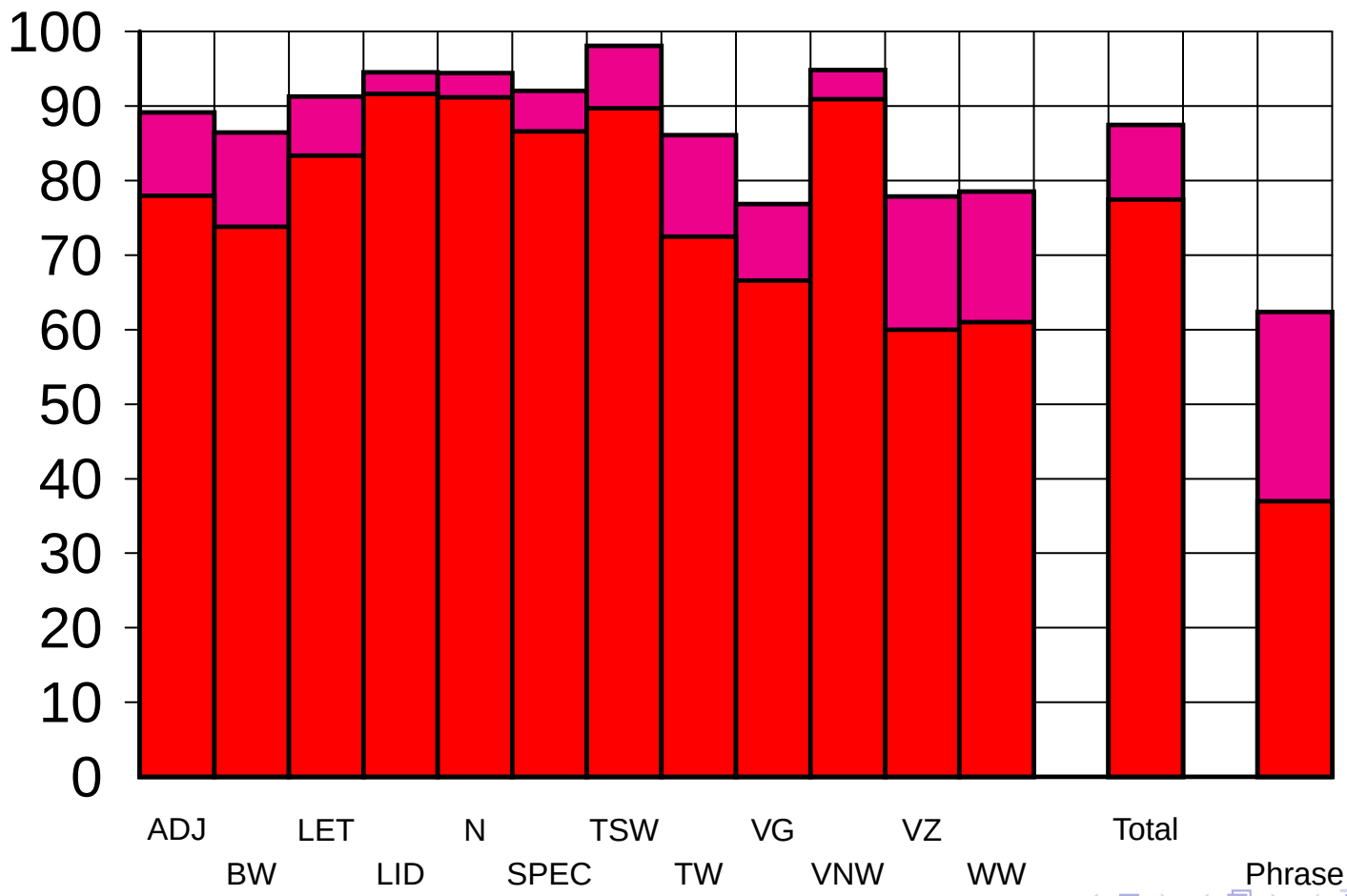
Results

Experiment	Result
Non-filtered, with interpunction	81.26%
Non-filtered, without interpunction	78.85%
Combined, with interpunction	81.50%
Combined, without interpunction	79.11%

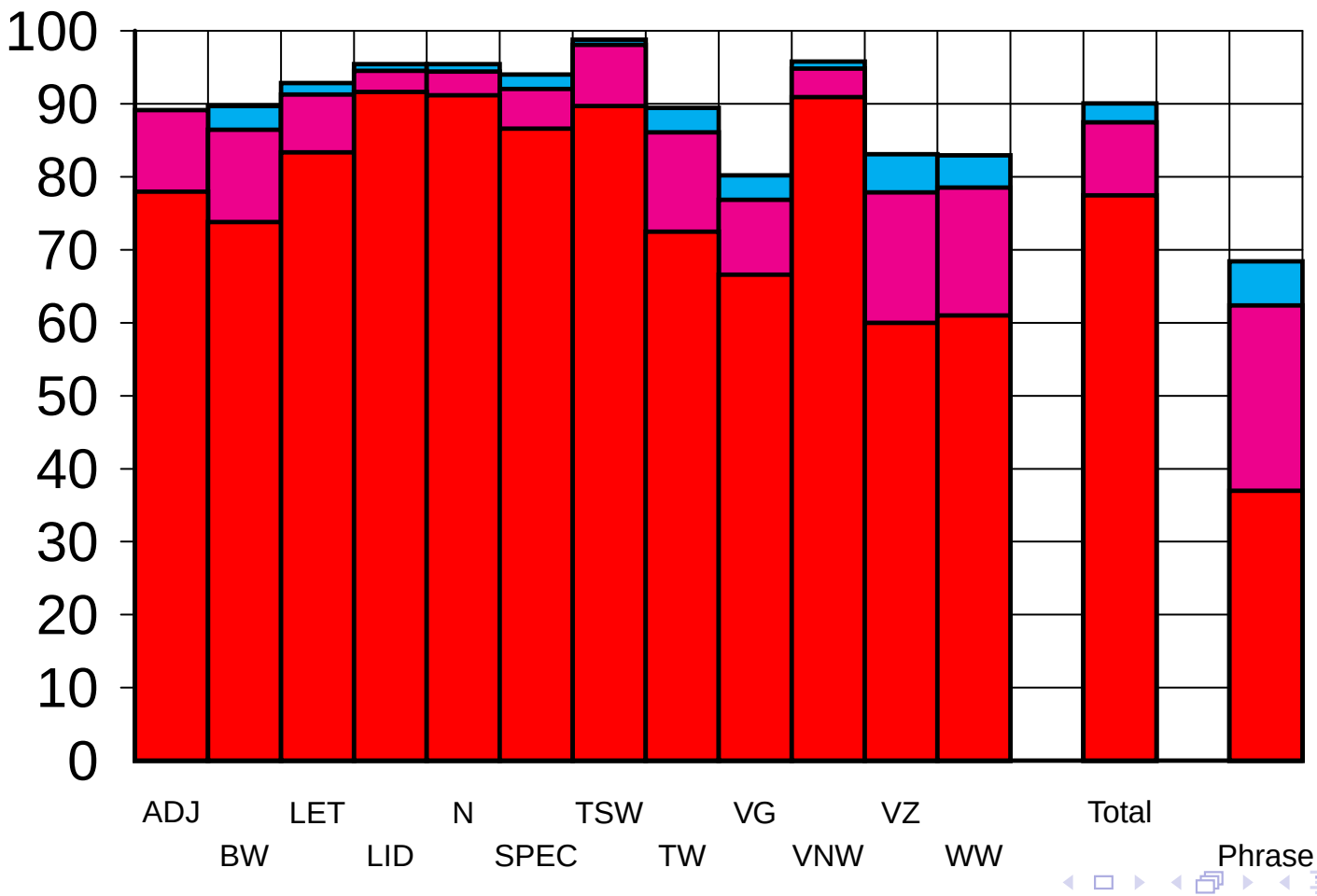
Results: Multiple Solutions 1



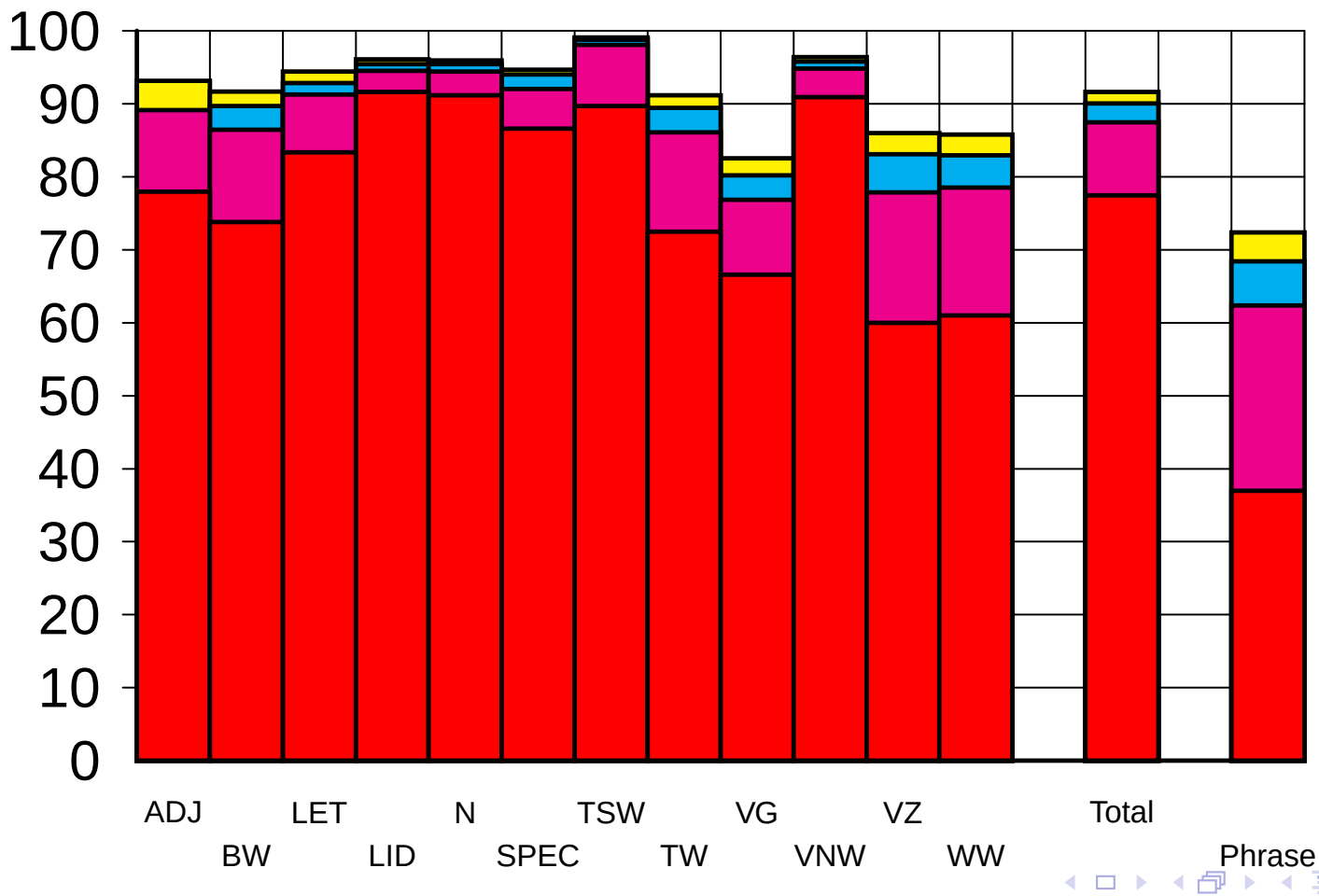
Results: Multiple Solutions 10



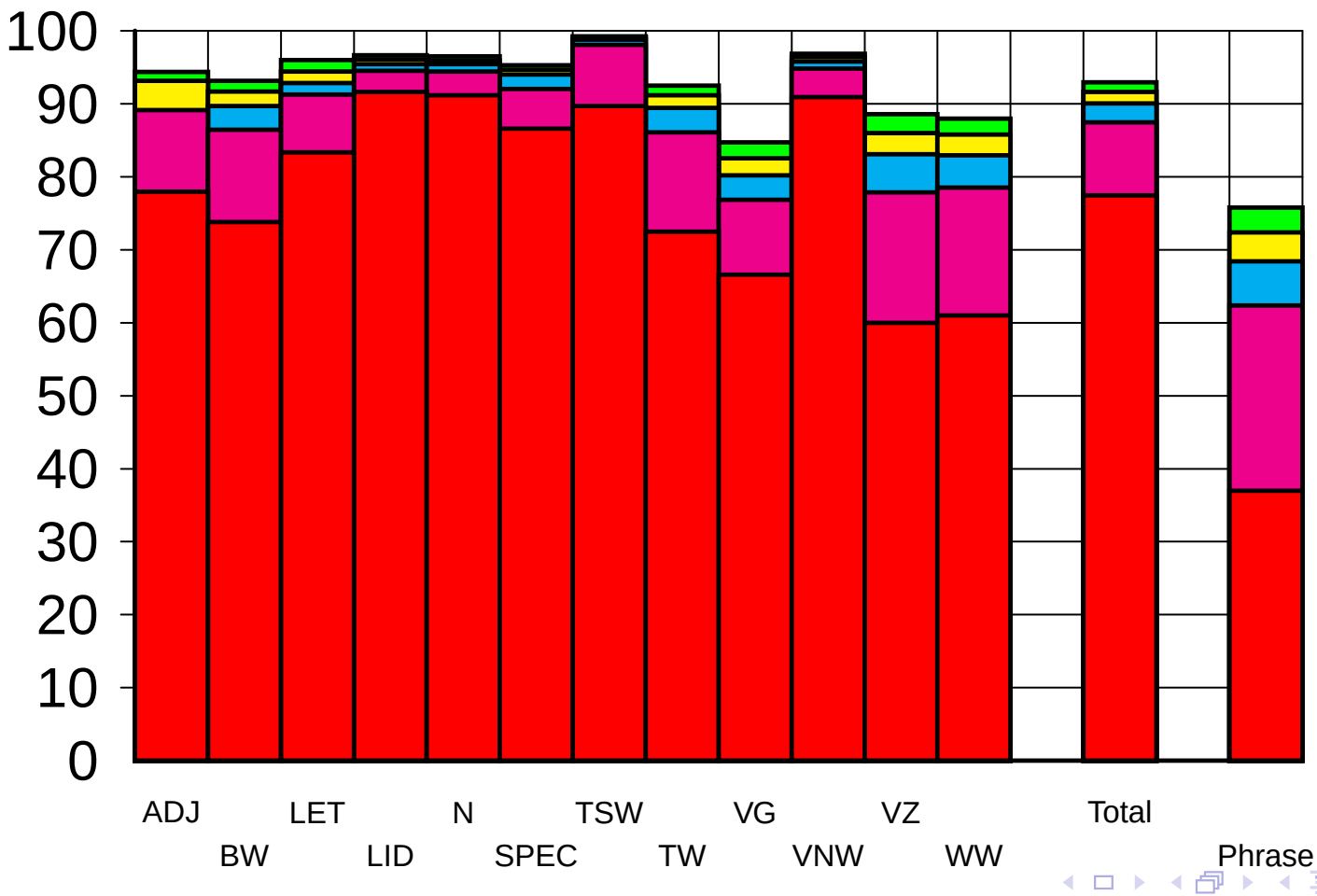
Results: Multiple Solutions 25



Results: Multiple Solutions 50



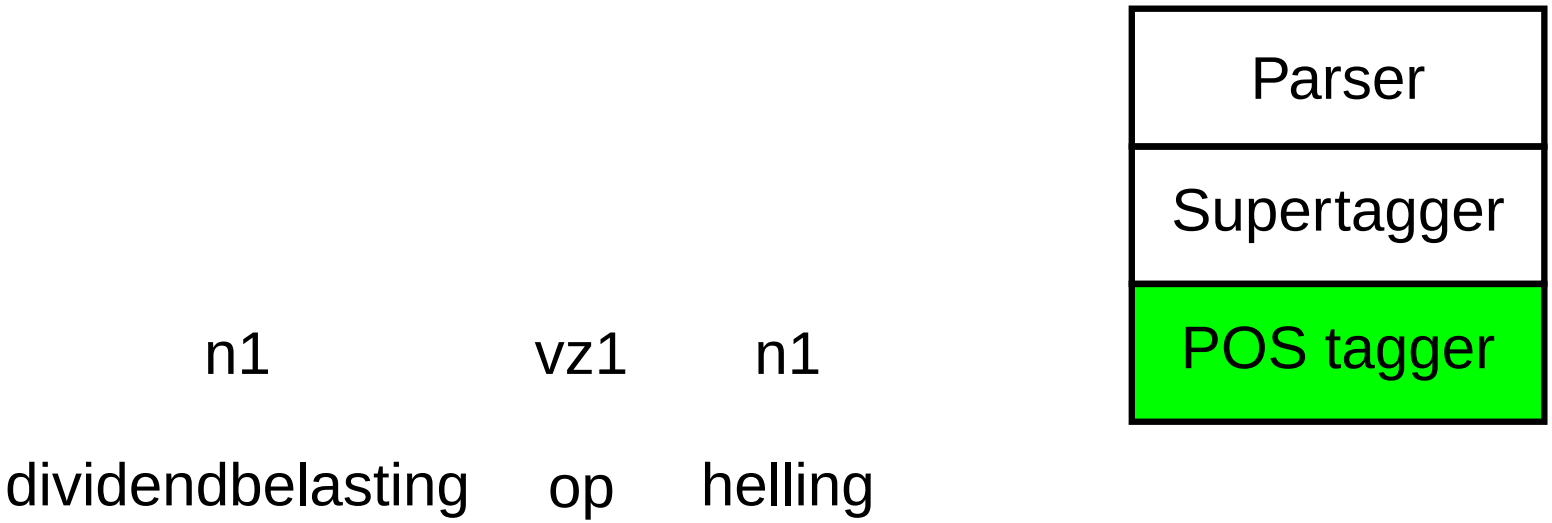
Results: Multiple Solutions 100



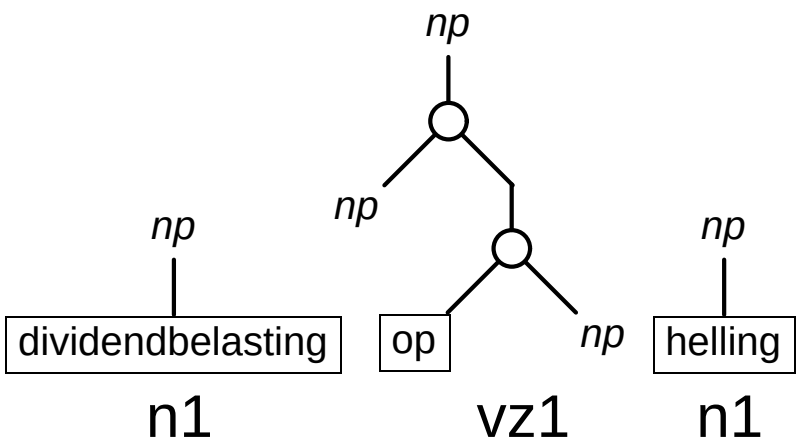
Parsing
Architecture

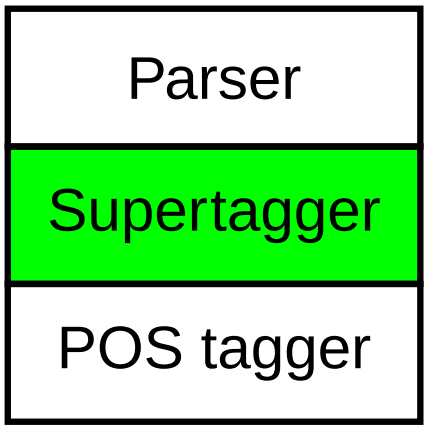
dividendbelasting op helling

Parsing
Architecture

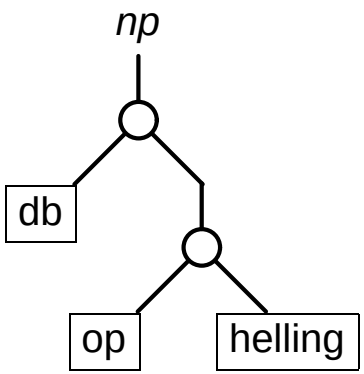


Parsing
Architecture



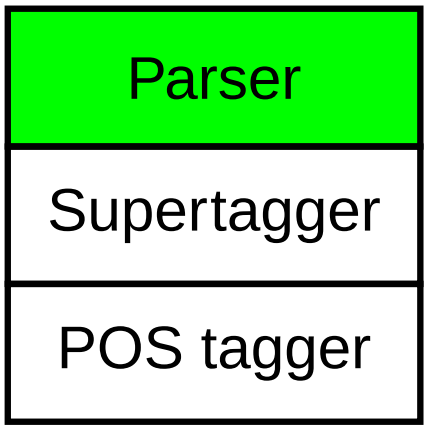


Parsing
Architecture



n1 vz1 n1

dividendbelasting op helling



Parsing
Complexity

- even if we have the correct supertag sequence, parsing it will be NP complete
- however, adding any kind of weights to the different possibilities for the axiom links will let us find the minimum (or maximum) weigth total linking in $O(n^3)$ time or the best k total linkings in $O(kn^3)$ time.
- as a baseline weight function we have implemented the distance between the two atomic formulas

Parsing
Demo

Example

I think it's time for a demo!

Conclusions
And Future Work

- parsing with an automatically extracted grammar presents new challenges to our parsing algorithms
- supertagging helps us deal with massive lexical ambiguity
- weighted axiom links give us a polynomial approximation of parsing

The Future...

- improve the lexicon extraction, especially in the case of ellipsis
- give the supertagger more long-distance information, for example, by using head trigrams
- evaluate the k-best parsing strategy for different grammars and for different weights
- evaluate the combined supertagger/parser