Extracting & Learning a Dependency-Enhanced Grammar for Dutch

Konstantinos Kogkalidis
Supervised by: M. Moortgat, T. Deoskar, R. Moot

June 2019

Utrecht University

Goal & Themes

Goal

Design a Syntactic Framework for Semantic Compositionality

Implement Computational Tools for Large-Scale Applicability

Core Themes

Formally Grounded (Typelogical Grammars, Parsing as Deduction, . . .)

Pragmatic (Corpus-Driven, Probabilistic, . . .)

Overview

- Theory
  - Parsing as Deduction
- Practice
  - Grammar Extraction
  - Supertagging
  - Parsing

Parsing as Deduction

Intro: Categorial Grammars

- Lexicon $\mathcal{L}$: words $\rightarrow$ categories
- Categories defined by some inductive scheme
  - Atomic Categories: full phrases $\{\mathrm{NP}, \mathrm{S}, \ldots\}$
  - Complex Categories: fractionals $\{\mathrm{NP}\backslash\mathrm{S}, \mathrm{NP}\backslash(\mathrm{S}/\mathrm{NP}), \ldots\}$
- Category Interactions: Combination Rules
- Parsing: Rule Application

Intro: Typelogical Grammars

- Lexicon $\mathcal{L}$: words $\rightarrow$ types
- Types defined by some inductive scheme
  - Atomic Types: full phrases $\{\mathrm{NP}, \mathrm{S}, \ldots\}$
  - Complex Types: fractionals $\{\mathrm{NP}\backslash\mathrm{S}, \mathrm{NP}\backslash(\mathrm{S}/\mathrm{NP}), \ldots\}$
- Type Interactions: Logical Rules
- Parsing: Proof Search

Blast from the Past: Lambek Calculus

$\mathcal{L} :=$ ducks : NP,

fish : NP,

fly : $\mathrm{NP}\backslash\mathrm{S}$,

eat : $\mathrm{NP}\backslash(\mathrm{S}/\mathrm{NP})$,

majestically : $(\mathrm{NP}\backslash\mathrm{S})\backslash(\mathrm{NP}\backslash\mathrm{S})$

$$\frac{\Gamma \vdash B/A \quad \Delta \vdash A}{\Gamma, \Delta \vdash B}\ (/\mathrm{E}) \qquad \frac{\Delta \vdash A \quad \Gamma \vdash A\backslash B}{\Delta, \Gamma \vdash B}\ (\backslash E)$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash B/A}\ (/\mathrm{I}) \qquad \frac{A, \Gamma \vdash B}{\Gamma \vdash A\backslash B}\ (\backslash I)$$

Flying Ducks

$$\text{ducks fly majestically} \vdash \mathrm{S}$$

Flying Ducks

$$\dfrac{\dfrac{}{\text{ducks} \vdash \text{NP}}\ Ax. \qquad \text{fly majestically} \vdash \text{NP}\backslash\text{S}}{\text{ducks fly majestically} \vdash \text{S}}\ \backslash E$$

Flying Ducks

$$\dfrac{\dfrac{}{\text{ducks} \vdash \text{NP}}\ Ax. \qquad \dfrac{\dfrac{\dfrac{}{\text{fly} \vdash \text{NP}\backslash\text{S}}\ Ax. \qquad \dfrac{}{\text{majestically} \vdash (\text{NP}\backslash\text{S})\backslash(\text{NP}\backslash\text{S})}\ Ax.}{\text{fly majestically} \vdash \text{NP}\backslash\text{S}}\ \backslash E}{\text{ducks fly majestically} \vdash \text{S}}\ \backslash E$$

Ducks Eating Fish

$$\text{ducks eat fish} \vdash \mathrm{s}$$

Ducks Eating Fish

$$\dfrac{\text{ducks eat} \vdash \mathrm{S}/\mathrm{NP} \qquad \dfrac{}{\text{fish} \vdash \mathrm{NP}}\,Ax.}{\text{ducks eat fish} \vdash \mathrm{S}}\,/E$$

Ducks Eating Fish

$$\dfrac{\text{ducks eat} \vdash \mathrm{S}/\mathrm{NP} \qquad \dfrac{}{\text{fish} \vdash \mathrm{NP}}\,Ax.}{\text{ducks eat fish} \vdash \mathrm{S}}\,/E$$

Ducks Eating Fish

$$\dfrac{\dfrac{\dfrac{}{\text{ducks} \vdash \text{NP}}\,Ax. \quad \dfrac{}{\text{eat} \vdash \text{NP}\backslash(\text{S}/\text{NP})}\,Ax.}{\text{ducks eat} \vdash \text{S}/\text{NP}}\,\backslash E \quad \dfrac{}{\text{fish} \vdash \text{NP}}\,Ax.}{\text{ducks eat fish} \vdash \text{S}}\,/E$$

Directionality & Lexical Ambiguity

$\mathcal{L} :=$ eten$_1$: $\mathrm{NP}\backslash(\mathrm{S}/\mathrm{NP})$, eenden eten$_1$ vis (SVO)

eten$_2$: $(\mathrm{S}/\mathrm{NP})/\mathrm{NP}$ eten$_2$ eenden vis? (VSO)

eten$_{3,4}$: $\mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S})$, eenden die vis eten$_{3,4}$ (SOV / OSV)

...

Directionality & Lexical Ambiguity

$\mathcal{L} :=$ eten$_1$: $\mathrm{NP}\backslash(\mathrm{S}/\mathrm{NP})$, eenden eten$_1$ vis (SVO)

eten$_2$: $(\mathrm{S}/\mathrm{NP})/\mathrm{NP}$ eten$_2$ eenden vis? (VSO)

eten$_{3,4}$: $\mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S})$, eenden die vis eten$_{3,4}$ (SOV / OSV)

...

☹

Beyond Directionality: MILL

Can we abstract directionality away?

Beyond Directionality: MILL

Can we abstract directionality away?

Yes: Multiplicative Intuionistic Linear Logic (ACG, LP . . .)

Beyond Directionality: MILL

Can we abstract directionality away?

Yes: Multiplicative Intuionistic Linear Logic (ACG, LP . . .)

Types:

$$\mathrm{T} := \mathrm{A} \mid \mathrm{T}_1 \to \mathrm{T}_2$$

Beyond Directionality: MILL

Can we abstract directionality away?

Yes: Multiplicative Intuionistic Linear Logic (ACG, LP . . .)

Types:

$$\mathrm{T} := \mathrm{A} \mid \mathrm{T}_1 \to \mathrm{T}_2$$

Logical Rules:

$$\frac{\Gamma \vdash A \to B \quad \Delta \vdash A}{\Gamma, \Delta \vdash B} \quad (\to \mathrm{E}) \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \quad (\to I)$$

MILL - The Good (1): Syntax-Semantics Interface

Curry-Howard Correspondence:

$$\text{MILL} \equiv \text{Simply-typed Linear } \lambda\text{-calculus}$$

MILL - The Good (1): Syntax-Semantics Interface

Curry-Howard Correspondence:

$$\text{MILL} \equiv \text{Simply-typed Linear } \lambda\text{-calculus}$$

$$\frac{\Gamma : f \vdash A \to B \quad \Delta : x \vdash A}{\Gamma, \Delta \vdash f(x) : B} \qquad \frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x.u : A \to B}$$

MILL: The good (2): Lesser Lexical Ambiguity

$$\mathcal{L} := \mathsf{eten}_1 : \mathrm{NP}\backslash(\mathrm{S}/\mathrm{NP}),$$
$$\mathsf{eten}_2 : (\mathrm{S}/\mathrm{NP})/\mathrm{NP}$$
$$\mathsf{eten}_{3,4} : \mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S}),$$

MILL: The good (2): Lesser Lexical Ambiguity

$$\mathcal{L} := \mathsf{eten}_1 : \mathrm{NP}\backslash(\mathrm{S}/\mathrm{NP}),$$
$$\mathsf{eten}_2 : (\mathrm{S}/\mathrm{NP})/\mathrm{NP}$$
$$\mathsf{eten}_{3,4} : \mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S}),$$

$$\mathcal{L}' := \mathsf{eten}_{1,2,3,4} : \mathrm{NP} \to \mathrm{NP} \to \mathrm{S}$$

☺

MILL - The Bad: Structural Ambiguity

$$\dfrac{\dfrac{\dfrac{}{\text{eten} \vdash \text{NP} \to \text{NP} \to \text{S}}\to Ax. \quad \dfrac{}{\text{vis} \to \text{NP}}\to Ax.}{\text{eten, vis} \vdash \text{NP} \to \text{S}}\to E \quad \dfrac{}{\text{eenden} \vdash \texttt{eenden} : \text{NP}}Ax.}{\text{eenden, eten, vis} \vdash \text{S}}\to E$$

`(eten vis) eenden`

$$\dfrac{\dfrac{\dfrac{}{\text{eten} \vdash \text{NP} \to \text{NP} \to \text{S}}\to Ax. \quad \dfrac{}{\text{enden} \to \text{NP}}\to Ax.}{\text{eten, eenden} \vdash \text{NP} \to \text{S}}\to E \quad \dfrac{}{\text{vis} \vdash \text{NP}}Ax.}{\text{eenden, eten, vis} \vdash \text{S}}\to E$$

`(eten eenden) vis`

☹

Dependency Decorations

Replace $\rightarrow$ with dependency-decorated variants:

$$\{\xrightarrow{\text{su}}, \xrightarrow{\text{obj}}, \xrightarrow{\text{predc}}, \xrightarrow{\text{mod}}, \ldots\}$$

$$\text{eten} : \mathrm{NP} \xrightarrow{\text{su}} \mathrm{NP} \xrightarrow{\text{obj}} \mathrm{S}$$

Dependency Decorations

Replace $\rightarrow$ with dependency-decorated variants:

$$\{\xrightarrow{\text{su}}, \xrightarrow{\text{obj}}, \xrightarrow{\text{predc}}, \xrightarrow{\text{mod}}, \ldots\}$$

$$\text{eten} : \mathrm{NP} \xrightarrow{\text{su}} \mathrm{NP} \xrightarrow{\text{obj}} \mathrm{S}$$

Lexical preferences + decorations $\Longrightarrow$ reduced ambiguity

Dependency Decorations

Replace $\rightarrow$ with dependency-decorated variants:

$$\{\xrightarrow{\text{su}}, \xrightarrow{\text{obj}}, \xrightarrow{\text{predc}}, \xrightarrow{\text{mod}}, \ldots\}$$

$$\text{eten} : \mathrm{NP} \xrightarrow{\text{su}} \mathrm{NP} \xrightarrow{\text{obj}} \mathrm{S}$$

Lexical preferences + decorations $\Longrightarrow$ reduced ambiguity

Formally:

Unary modality $\Diamond^d$ for $d \in \{\text{su}, \text{obj}, \text{predc}, \text{mod}, \ldots\}$

$$\frac{\Gamma \vdash A}{\langle \Gamma \rangle^d \vdash \Diamond^d A} \quad (\Diamond^d I) \qquad \frac{\Delta \vdash \Diamond^d A \quad \Gamma[\langle A \rangle^d] \vdash B}{\Gamma[\Delta] \vdash B} \quad (\Diamond^d E)$$

Extraction

Extraction: Intro

Goal

Syntactically-Annotated Corpus $\rightarrow$ Type Lexicon

Extraction: Intro

Goal

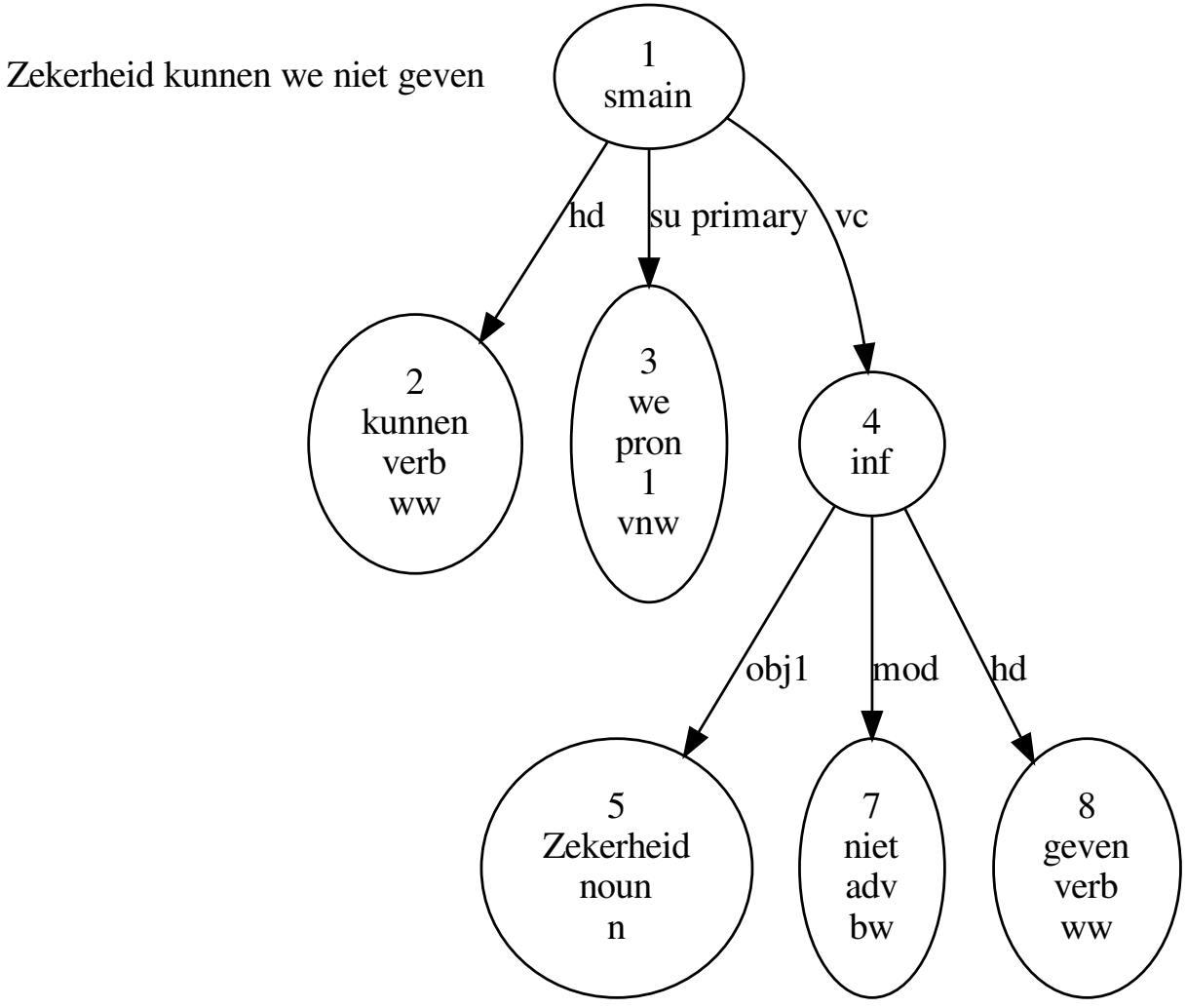
Syntactically-Annotated Corpus $\rightarrow$ Type Lexicon

Corpus

Lassy-Small:

- 65 000 Sentences (in DAG form)
- 1 000 000 Words
- $\sim$ 30 Dependency Labels
- $\sim$ 30 POS & Phrasal Category Tags

Extraction: Example (1)



Extraction: Parameters

Parameters

- Translation Tables
  - Atomic Types: POS & Phrasal Category Tags
    $\{np\colon \mathrm{NP},\ vnw\colon \mathrm{VNW},\ \ldots\}$
  - Implications: Dependency Labels
    $\{su\colon \xrightarrow{\text{su}},\ obj\colon \xrightarrow{\text{obj}},\ \ldots\}$
- Head Dependencies
  $\{hd,\ rhd,\ whd,\ crd,\ cmp\}$
- Modifier Dependencies
  $\{mod,\ app,\ predm\}$

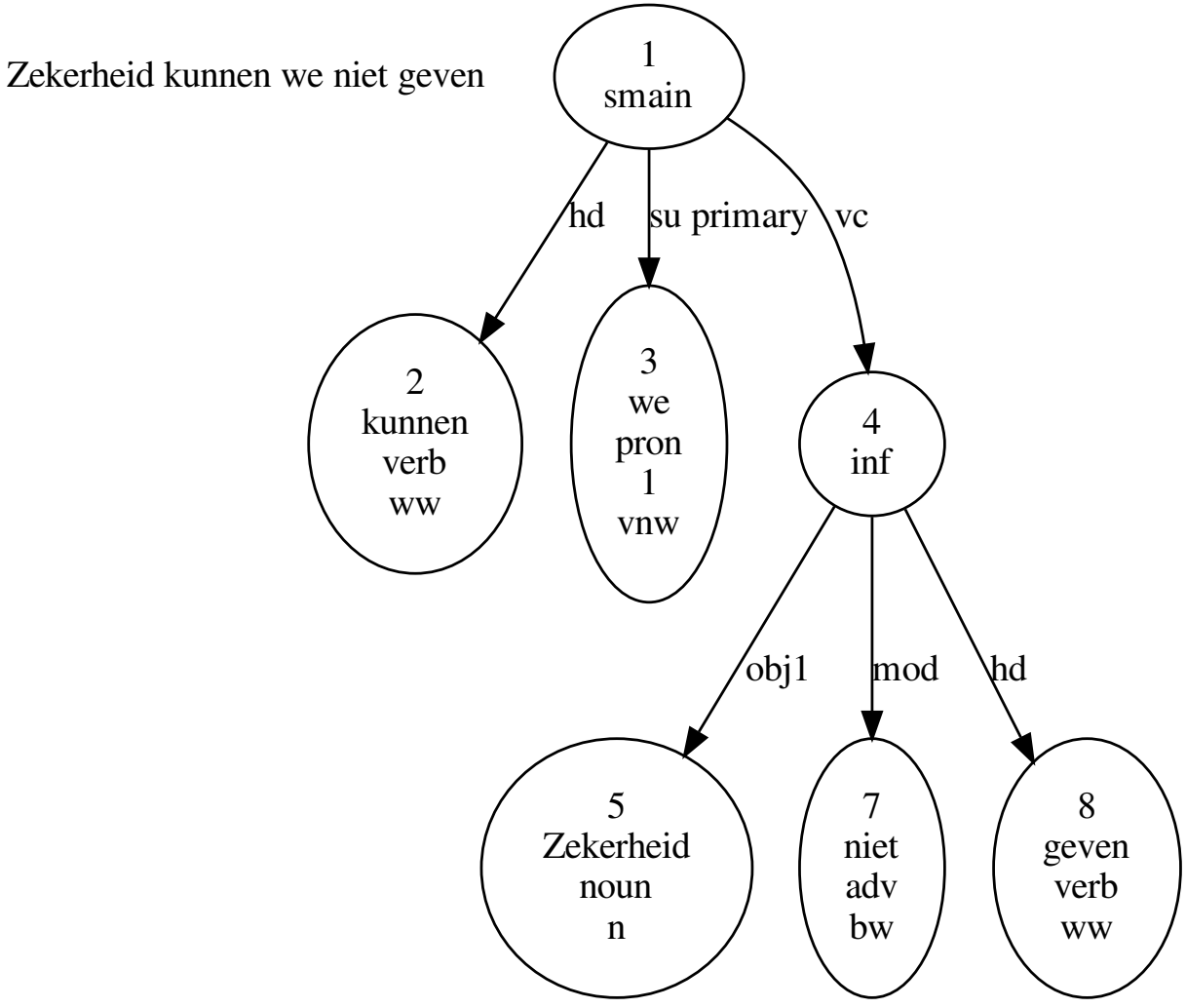
Extraction: Algorithm

General Idea

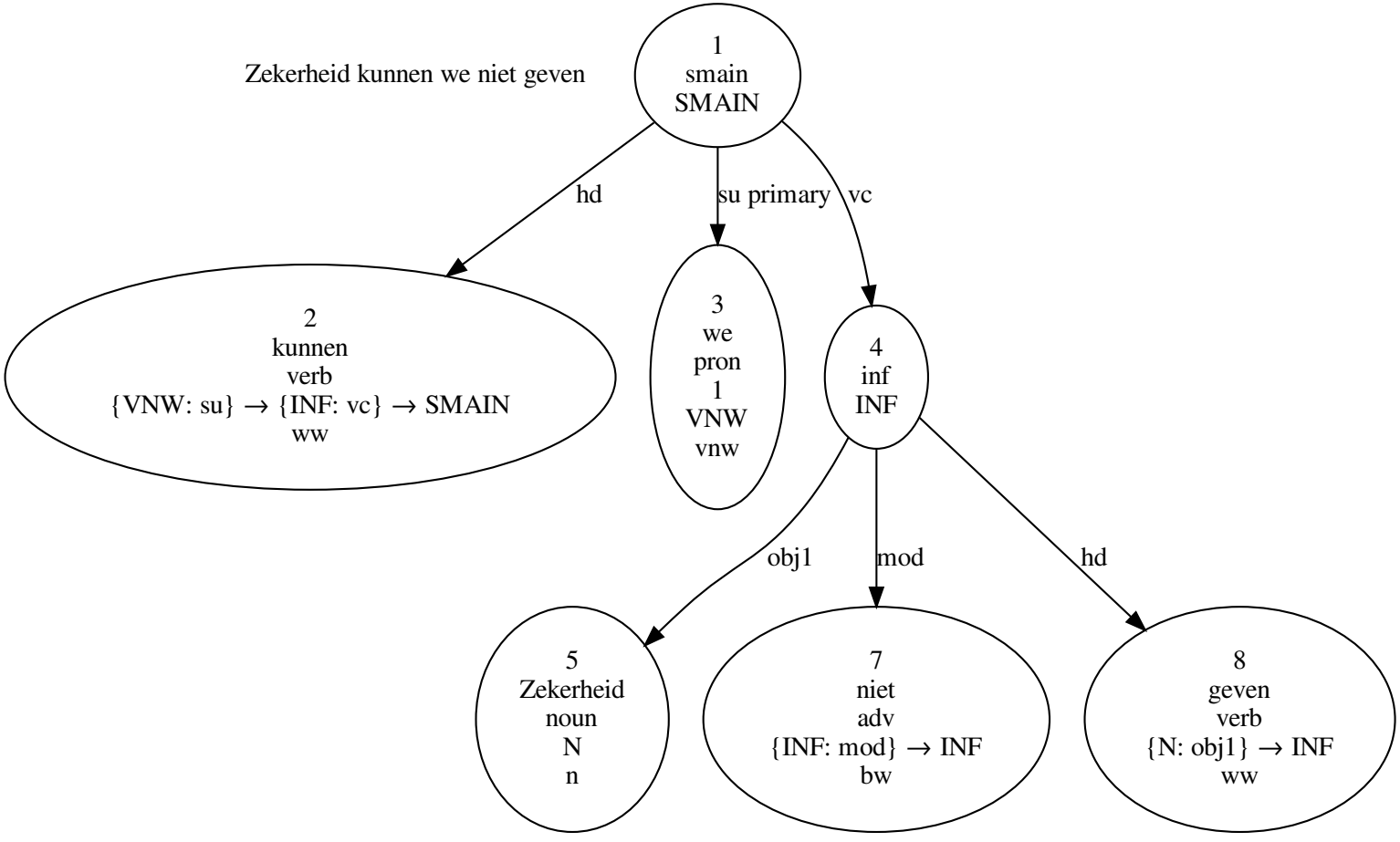
For each branch

- Find phrasal head, dependants, modifiers
- Collect & arrange dependant types
- Type head as a functor from argument types to parent type
- Type modifiers as morphisms from parent type to itself

Extraction: Example (1)



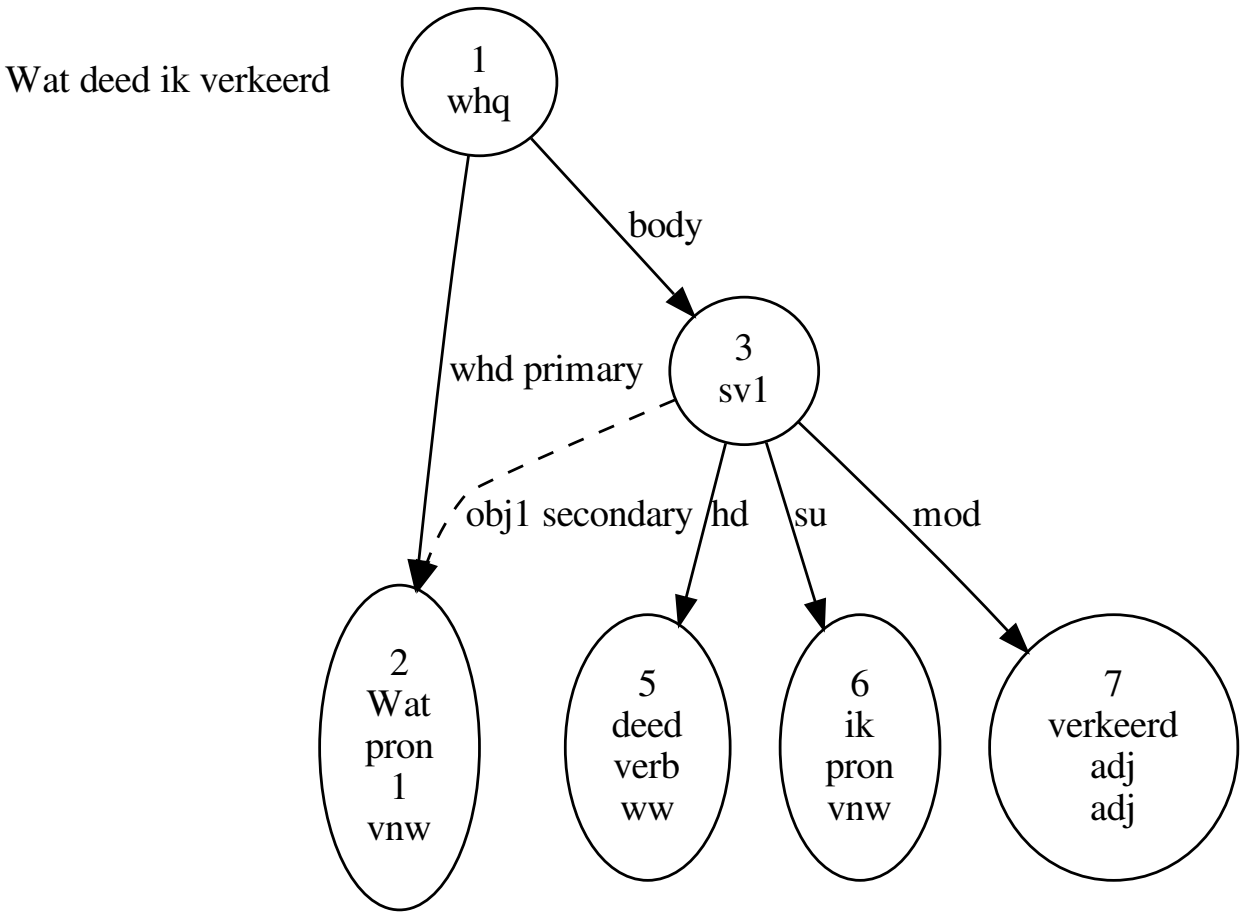
Extraction: Example (1)



Extraction: Example (2)

Hypothetical Reasoning

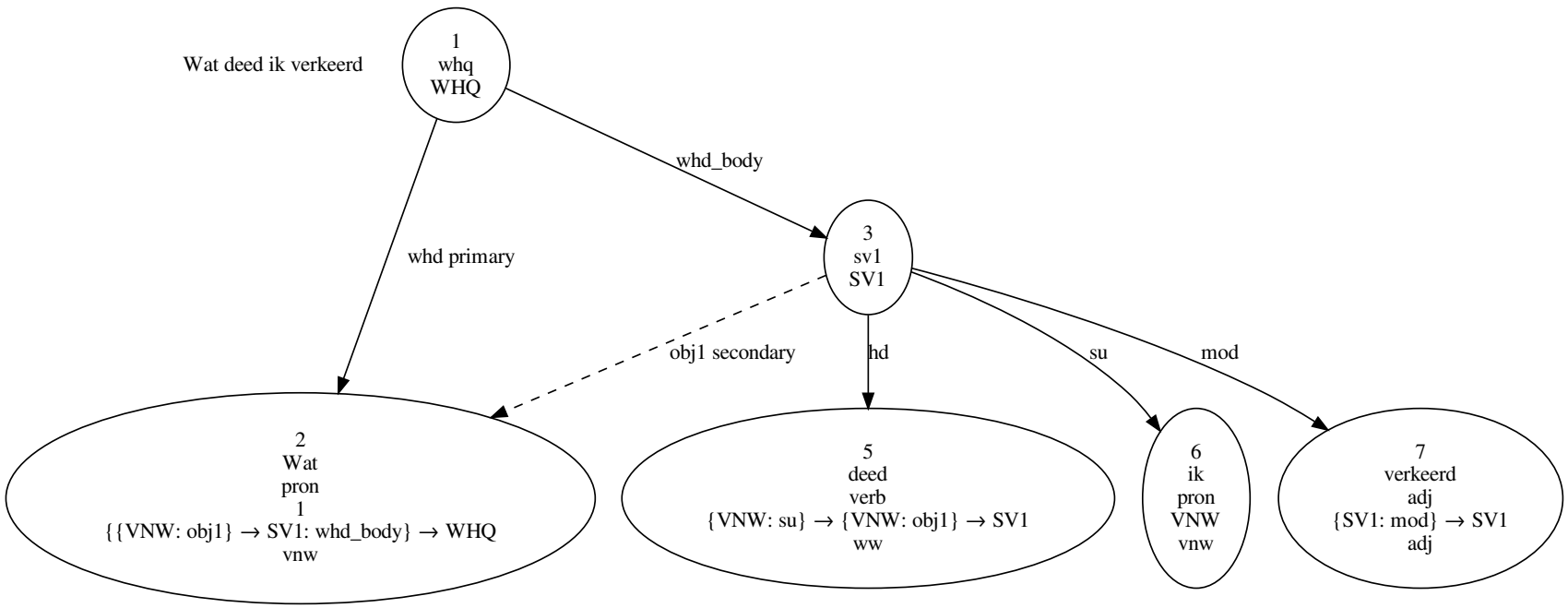
When type assigning arguments, consider internal "gaps"



Extraction: Example (2)

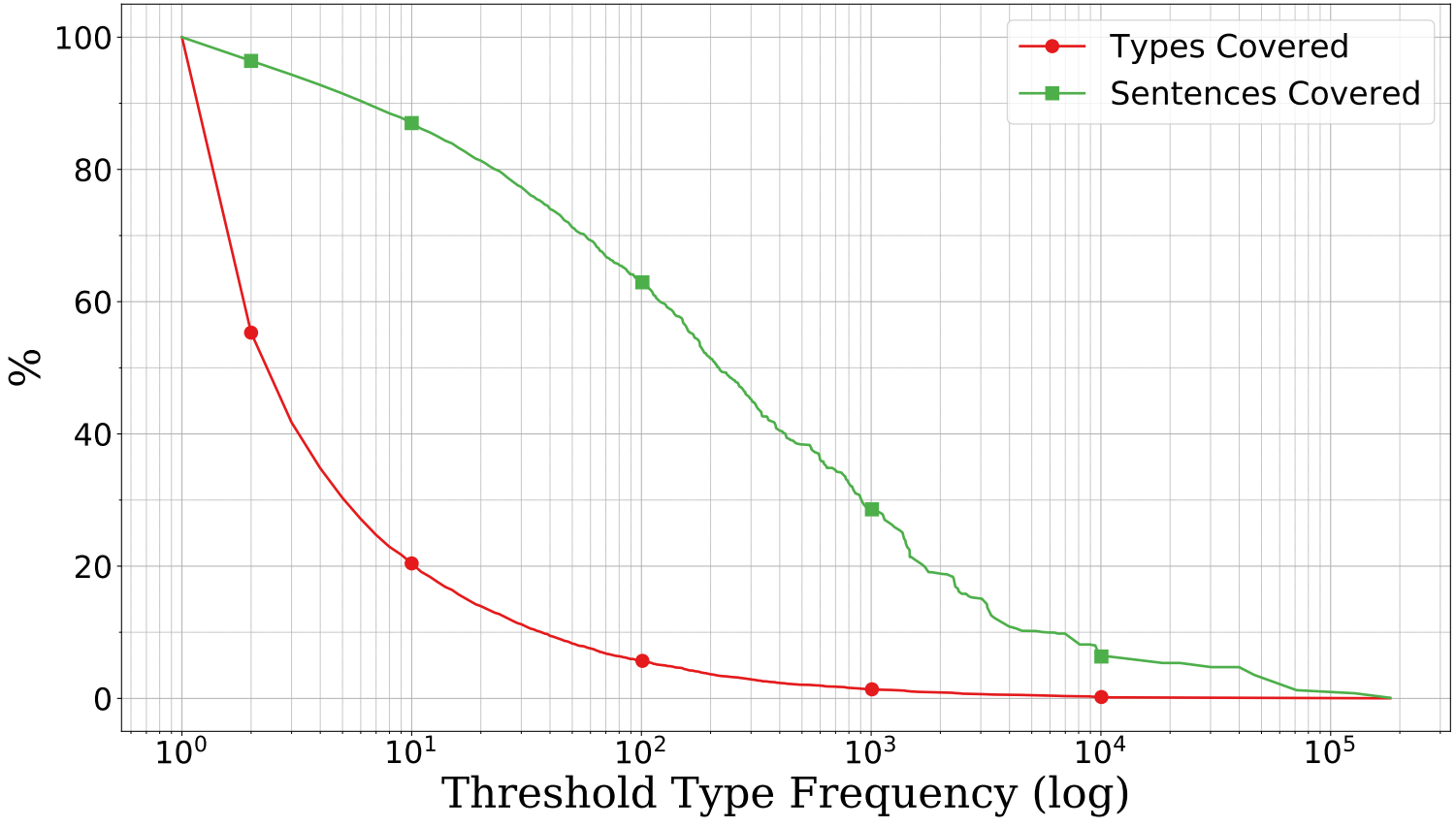
Hypothetical Reasoning

When type assigning arguments, consider internal "gaps"

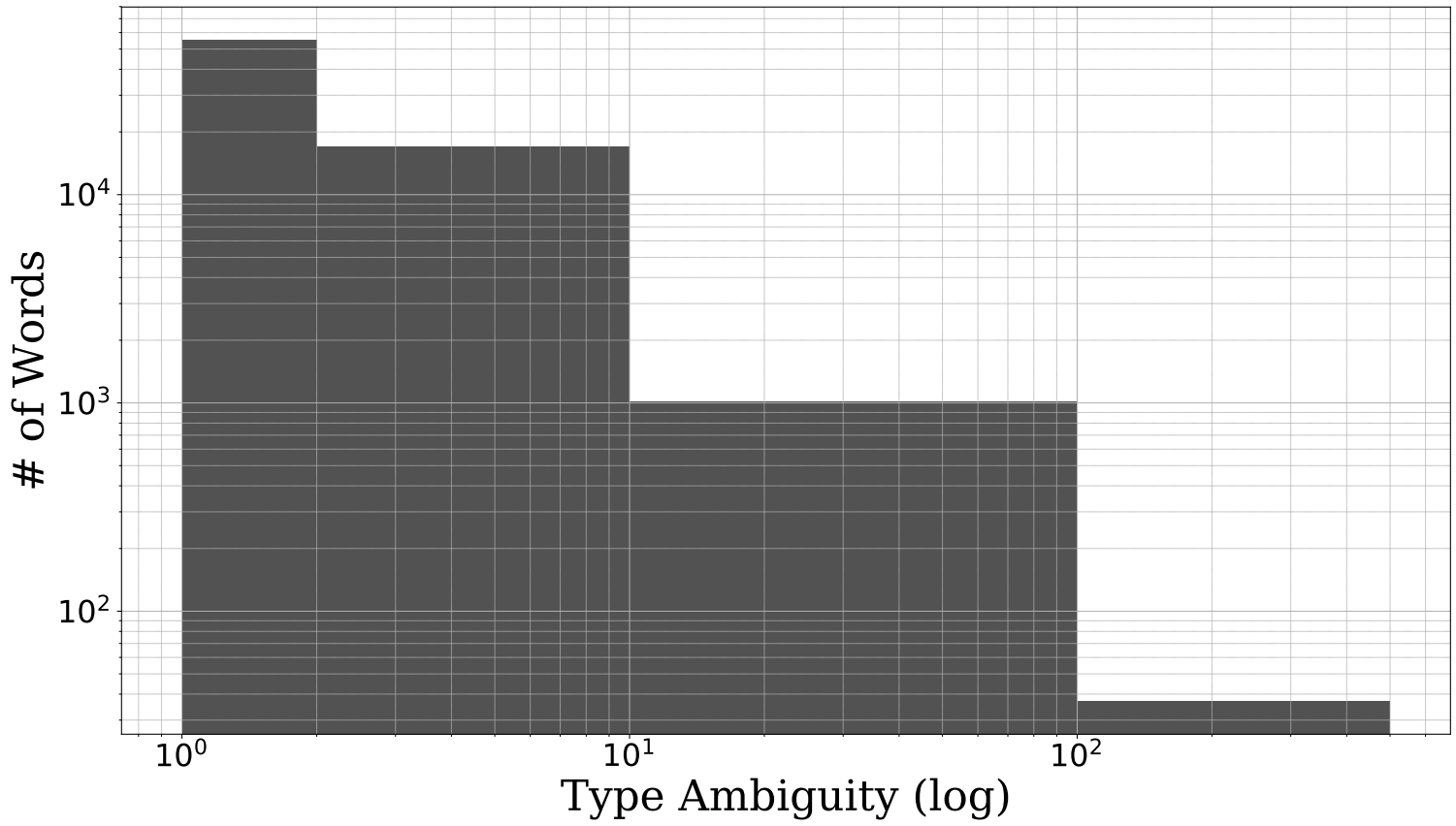


Extraction: Results (1)

$\sim$ 5 700 unique types



Extraction: Results (2)



Supertagging

Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)

Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

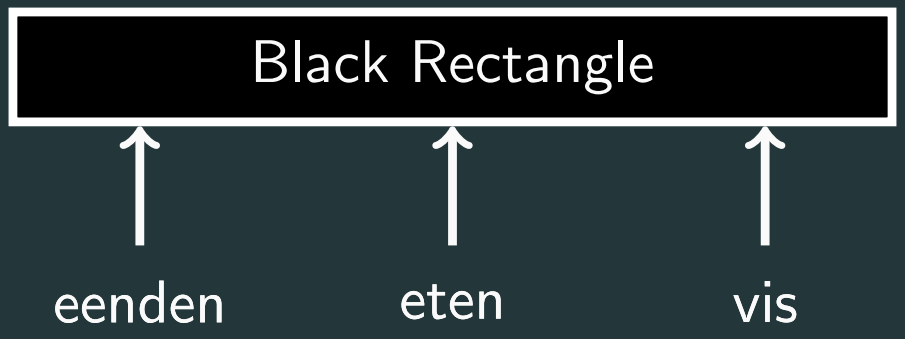
predict a class for each sequence item (types)

Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)

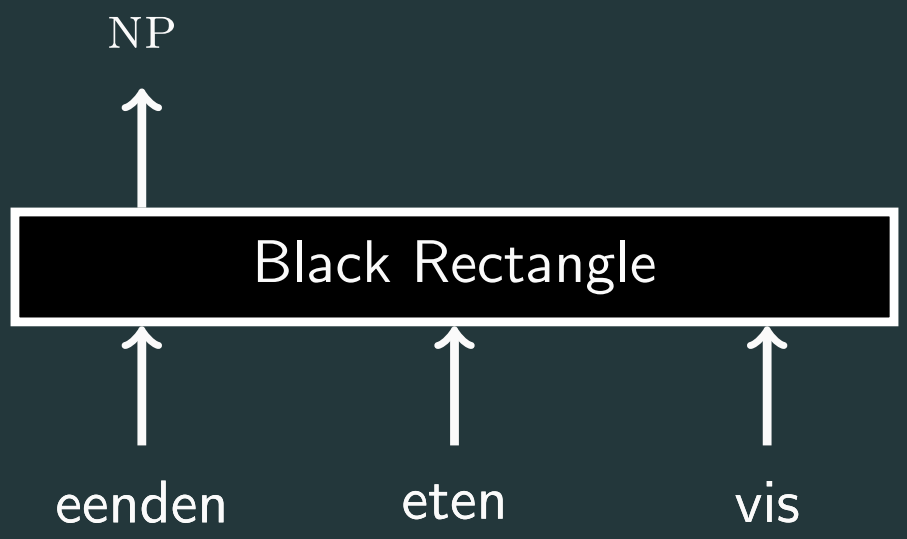


Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)

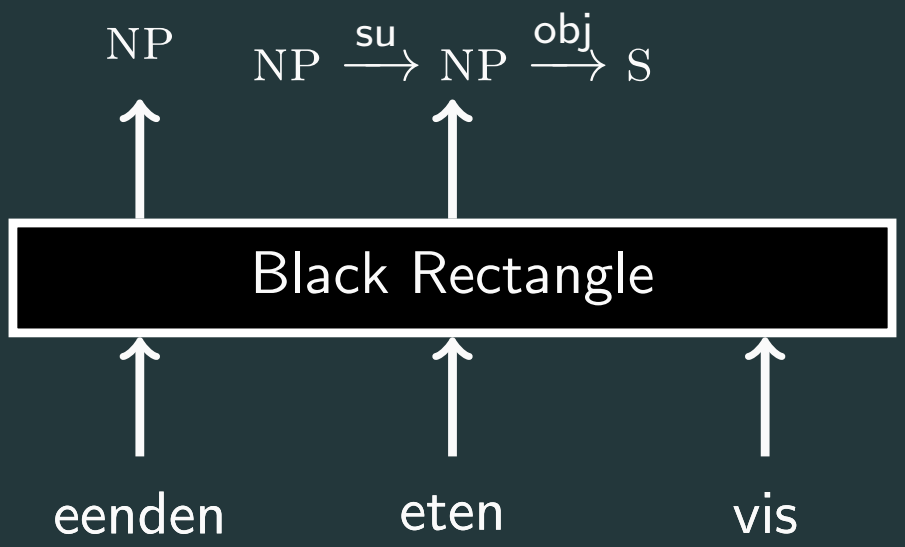


Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)

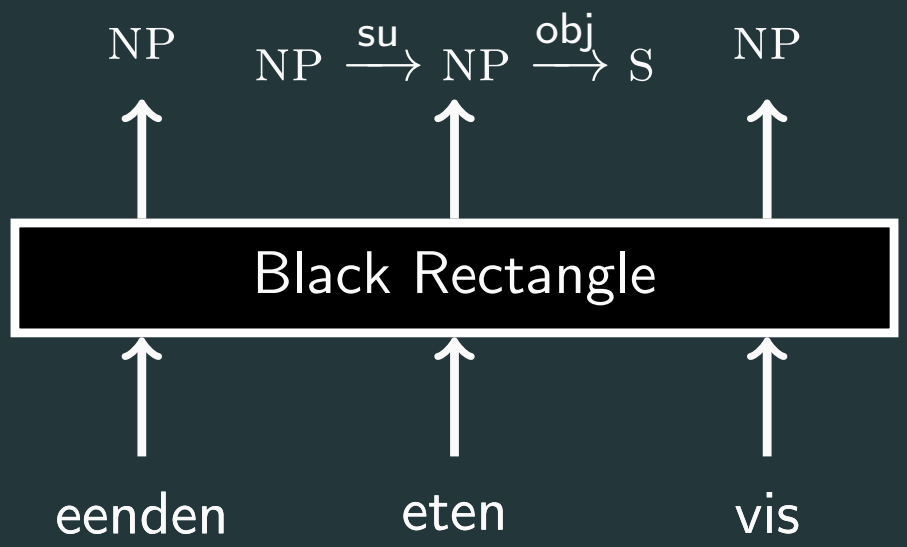


Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)



Supertagging: Standard Approach

The Problem

- Can't predict unseen types
- Bad at predicting rare types

Supertagging: An Alternative

Type Syntax

A CFG of two meta-rules

$$\{(S \implies A) \forall A \in \mathcal{A}\}$$

$$\{(S \implies S \xrightarrow{\mathrm{d}} S) \forall d \in \mathcal{D}\}$$

Supertagging: An Alternative

Type Syntax

A CFG of two meta-rules

$$\{(S \implies A) \forall A \in \mathcal{A}\}$$

$$\{(S \implies S \xrightarrow{\mathsf{d}} S) \forall d \in \mathcal{D}\}$$

CFGs: learnable

Supertagging: learnable

Supertagging: An Alternative

Type Syntax

A CFG of two meta-rules

$$\{(S \implies A) \forall A \in \mathcal{A}\}$$

$$\{(S \implies S \xrightarrow{\mathrm{d}} S) \forall d \in \mathcal{D}\}$$

CFGs: learnable

Supertagging: learnable

CFG embedded within supertagging: learnable..?

..if so, unbounded co-domain supertagging

Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

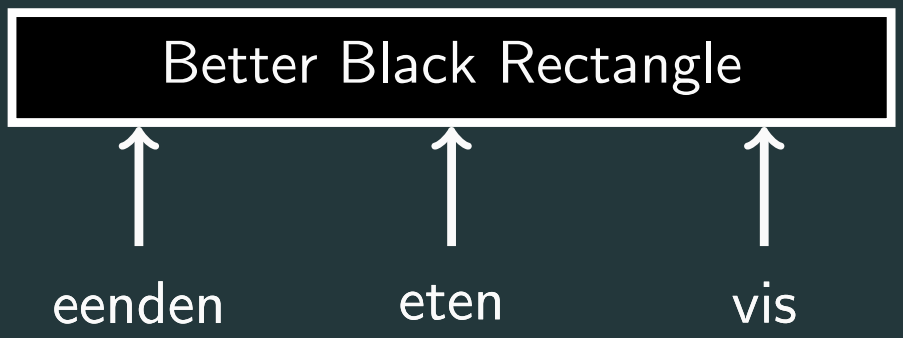
eenden eten vis

Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

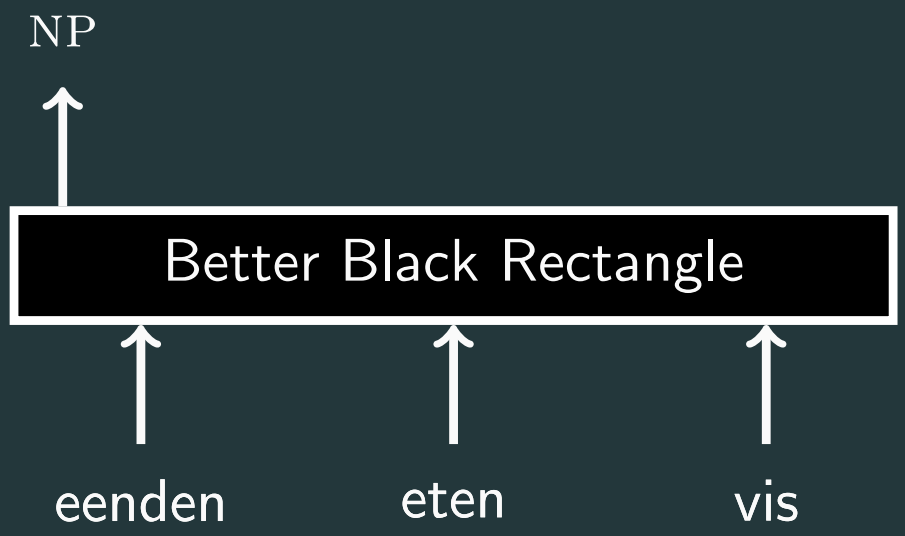


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

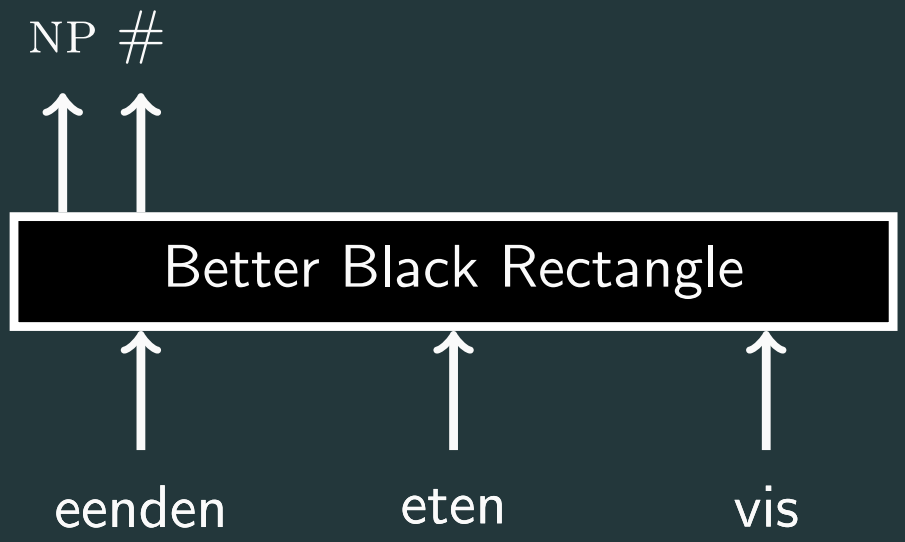


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

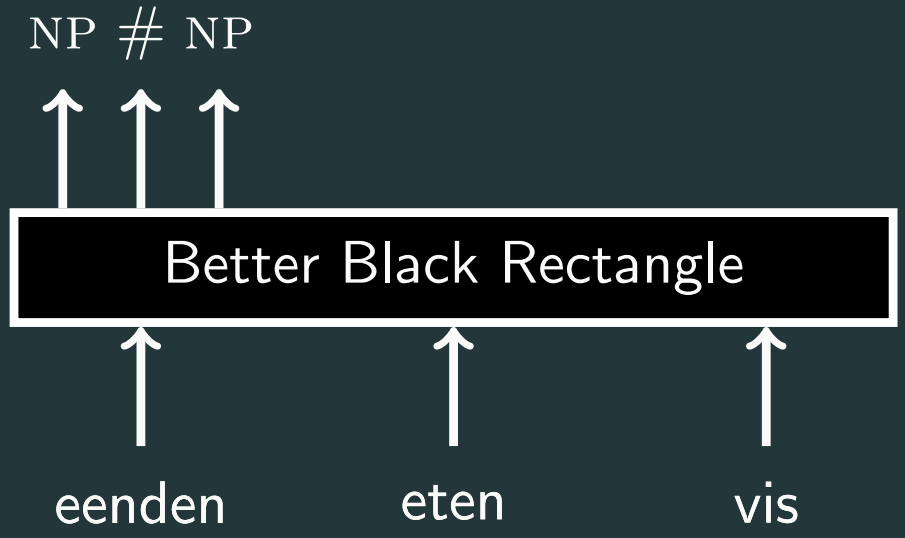


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

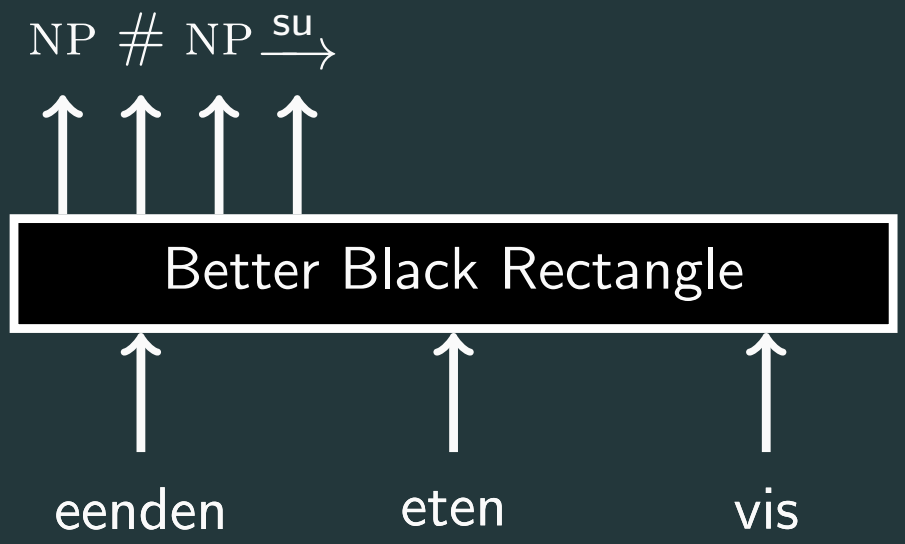


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

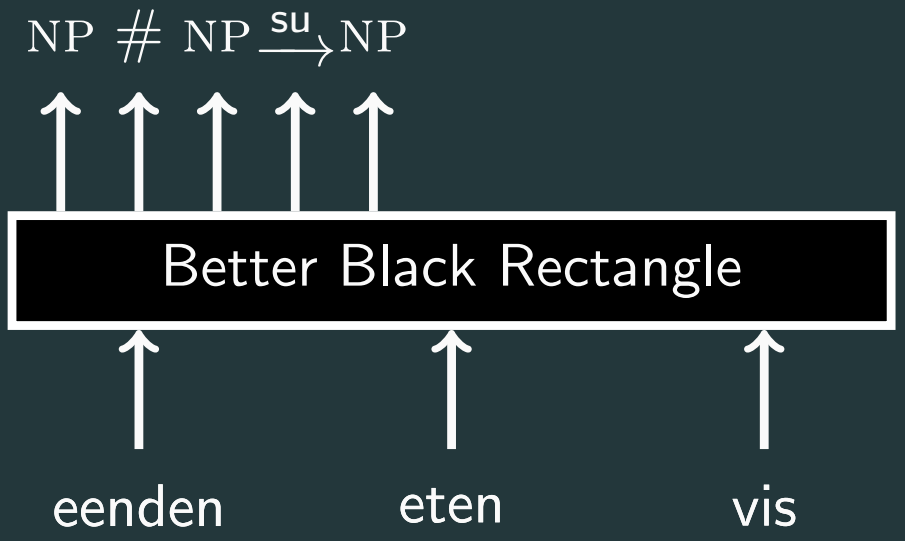


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

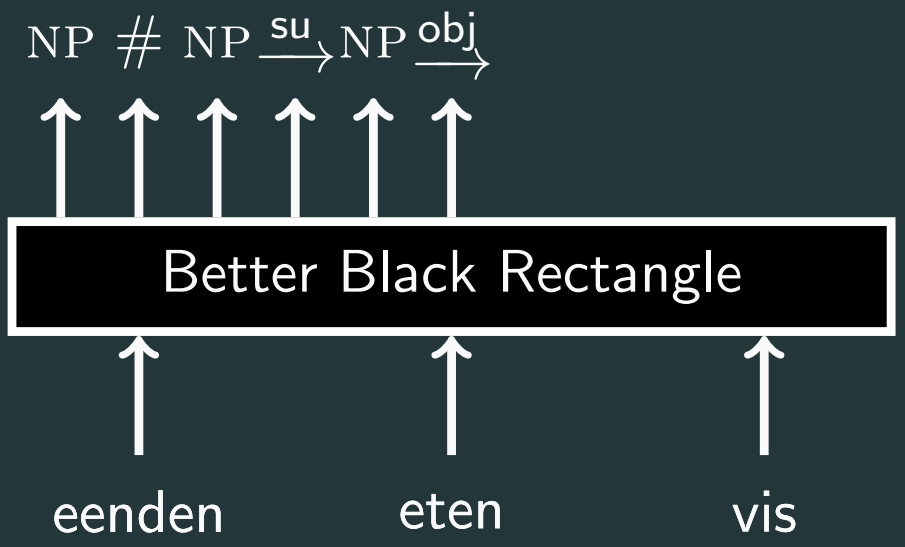


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

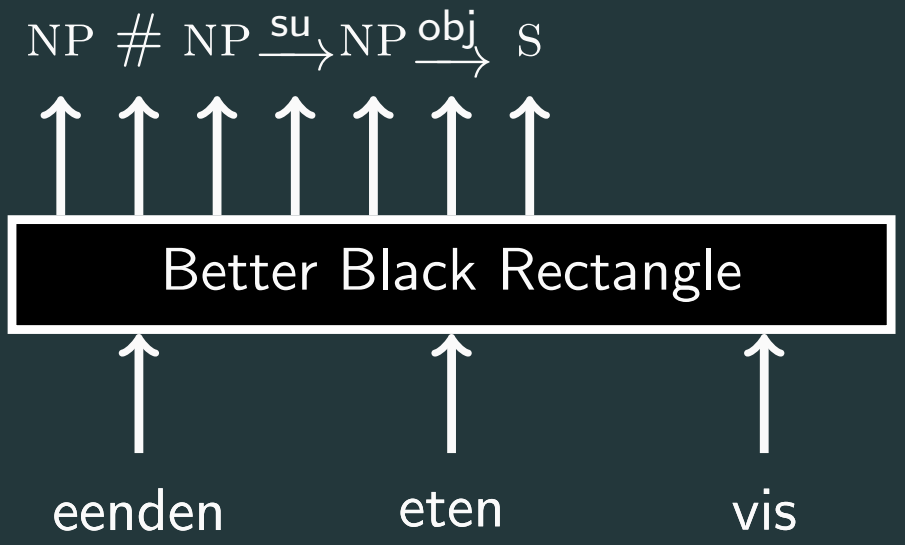


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

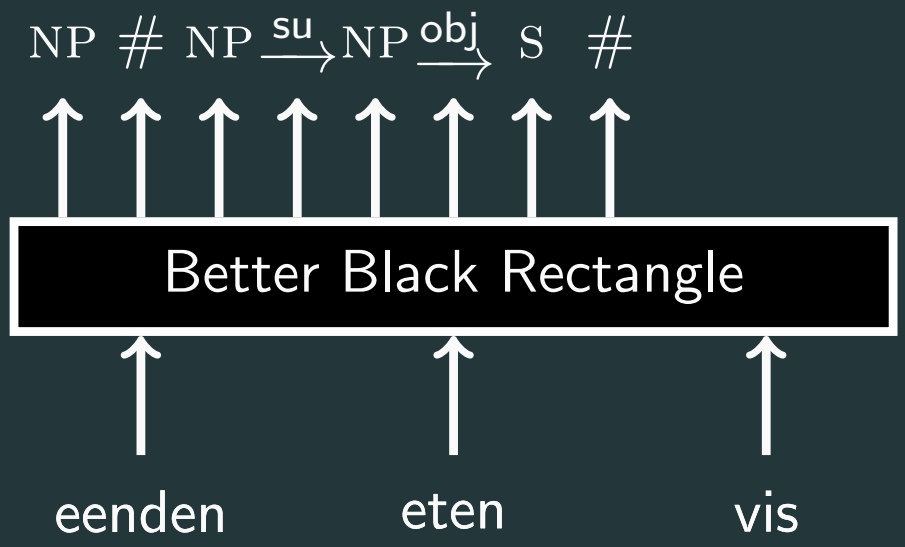


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

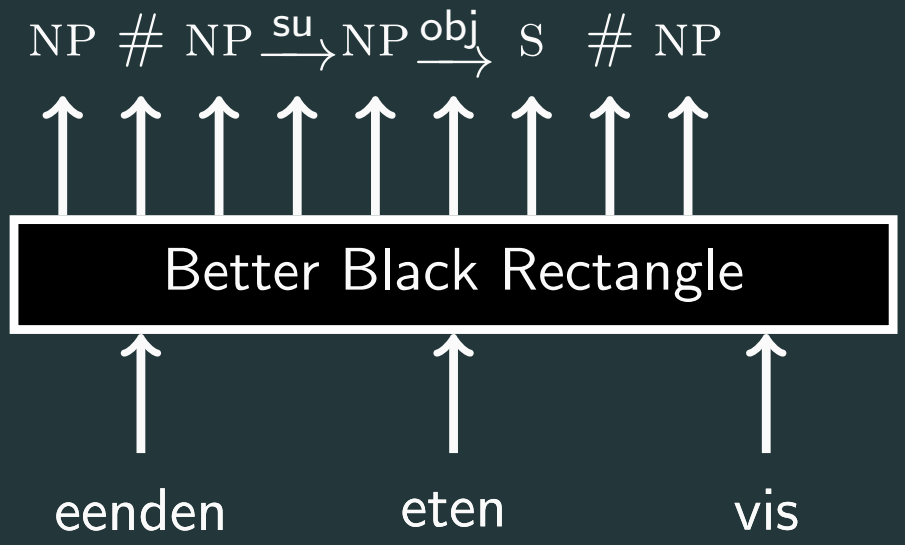


Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)



Supertagging: Unbounded co-domain

Requirements

- Global Receptive Field (long-distance dependencies)
- Auto-Regressive (output conditional on prior output)

Supertagging: Unbounded co-domain

Requirements

- Global Receptive Field (long-distance dependencies)
- Auto-Regressive (output conditional on prior output)

Options

- RNN encoder-decoder

Supertagging: Unbounded co-domain

Requirements

- Global Receptive Field (long-distance dependencies)
- Auto-Regressive (output conditional on prior output)

Options

- RNN encoder-decoder Fixed-length compression
- RNN encoder-separable decoder

Supertagging: Unbounded co-domain

Requirements

- Global Receptive Field (long-distance dependencies)
- Auto-Regressive (output conditional on prior output)

Options

- RNN encoder-decoder Fixed-length compression
- RNN encoder-separable decoder Only locally auto-regressive
- Self-Attentive encoder-decoder

Supertagging: Unbounded co-domain

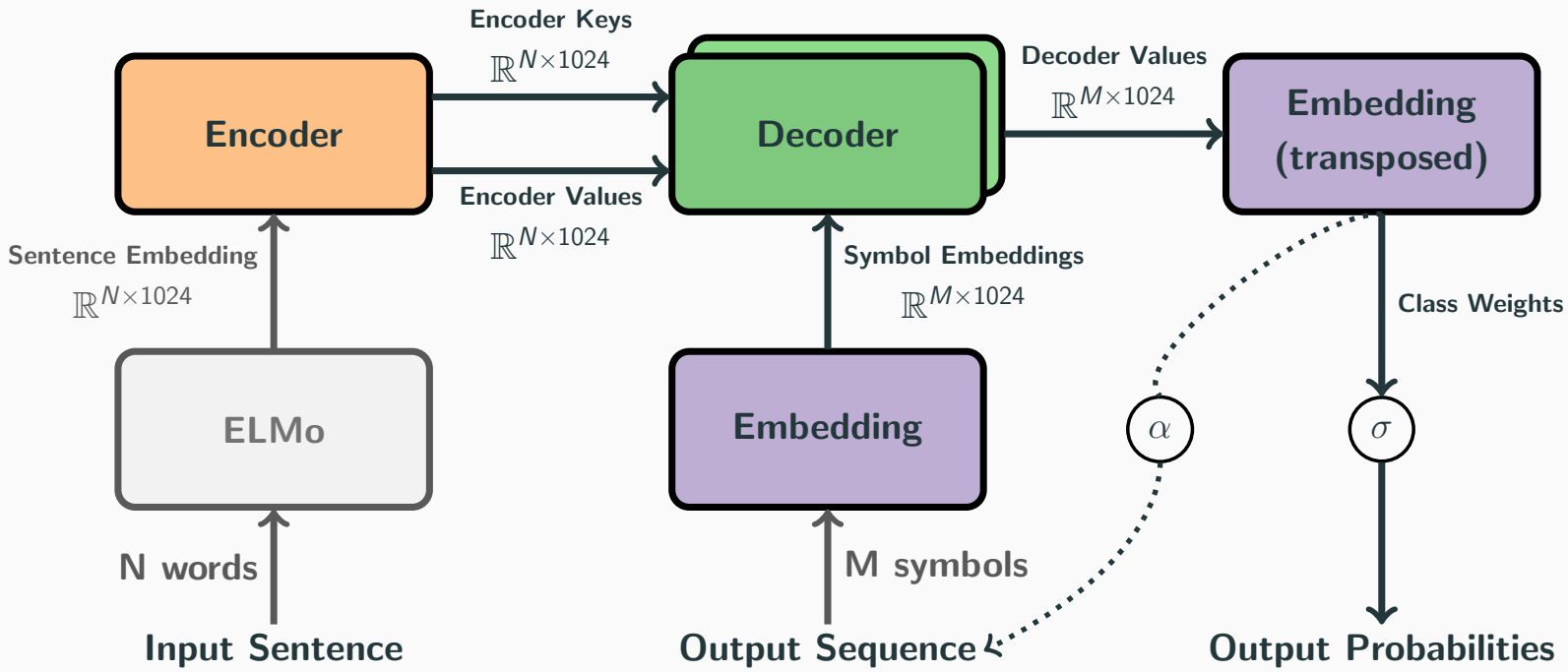
Requirements

- Global Receptive Field (long-distance dependencies)
- Auto-Regressive (output conditional on prior output)

Options

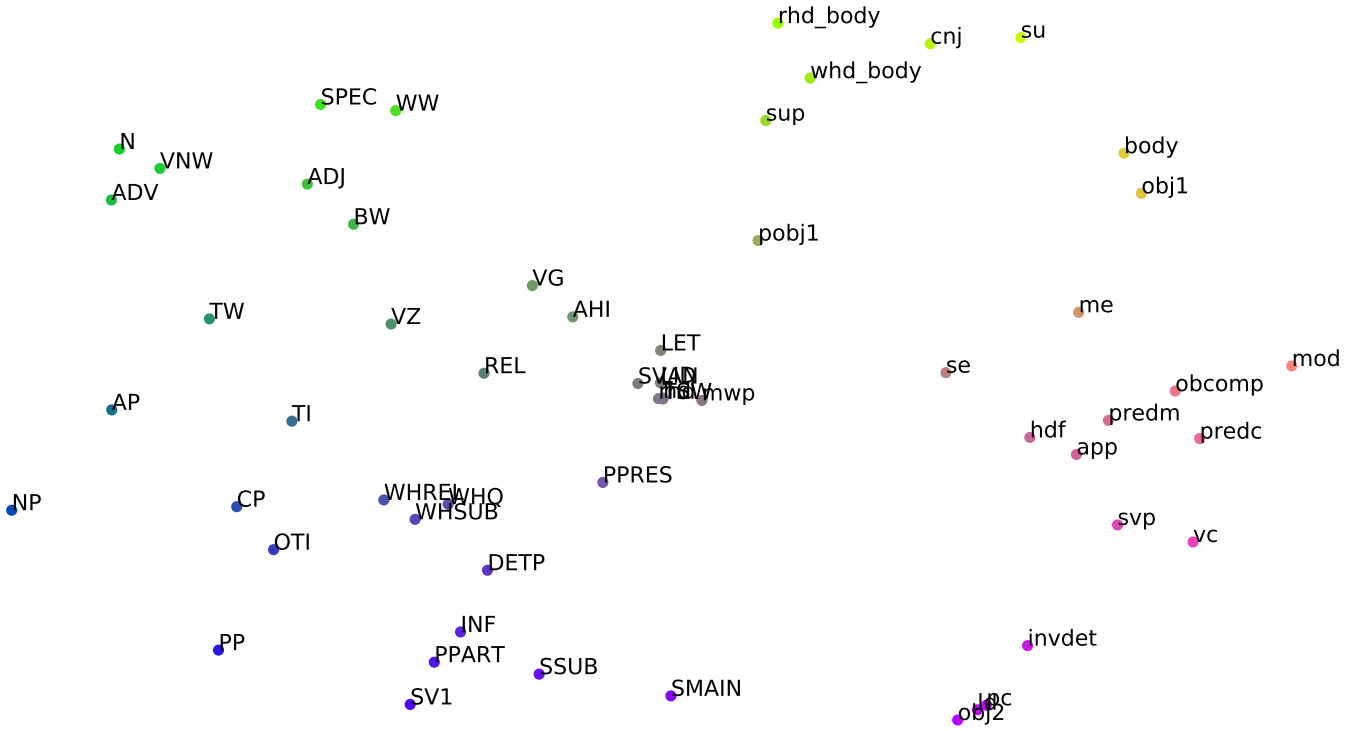
- RNN encoder-decoder Fixed-length compression
- RNN encoder-separable decoder Only locally auto-regressive
- Self-Attentive encoder-decoder ✓

Supertagging: Model



Supertagging: Evaluation

	Frequency				
	Overall	Unseen	Rare	Mid	High
Accuracy	**88.05**	**19.2**	**45.68**	**65.62**	**89.93**



Parsing

Parsing: Intro

Goal

From abstract syntax to surface syntax

Parsing: Intro

Goal

From abstract syntax to surface syntax

Parse $\equiv$ Proof

How can we efficiently navigate the proof space?

Parsing: General Framework

Parse State

- A logical judgement
- Word associations for (some) premises
- A lookahead containing last rule applied

Algorithm

Given a parse state

1. Decide between introduction and elimination
2. Perform either
3. Update state(s)
4. Repeat

Parsing: Elimination

Given a sequence of word & type pairs

Split into two disjoint sequences..

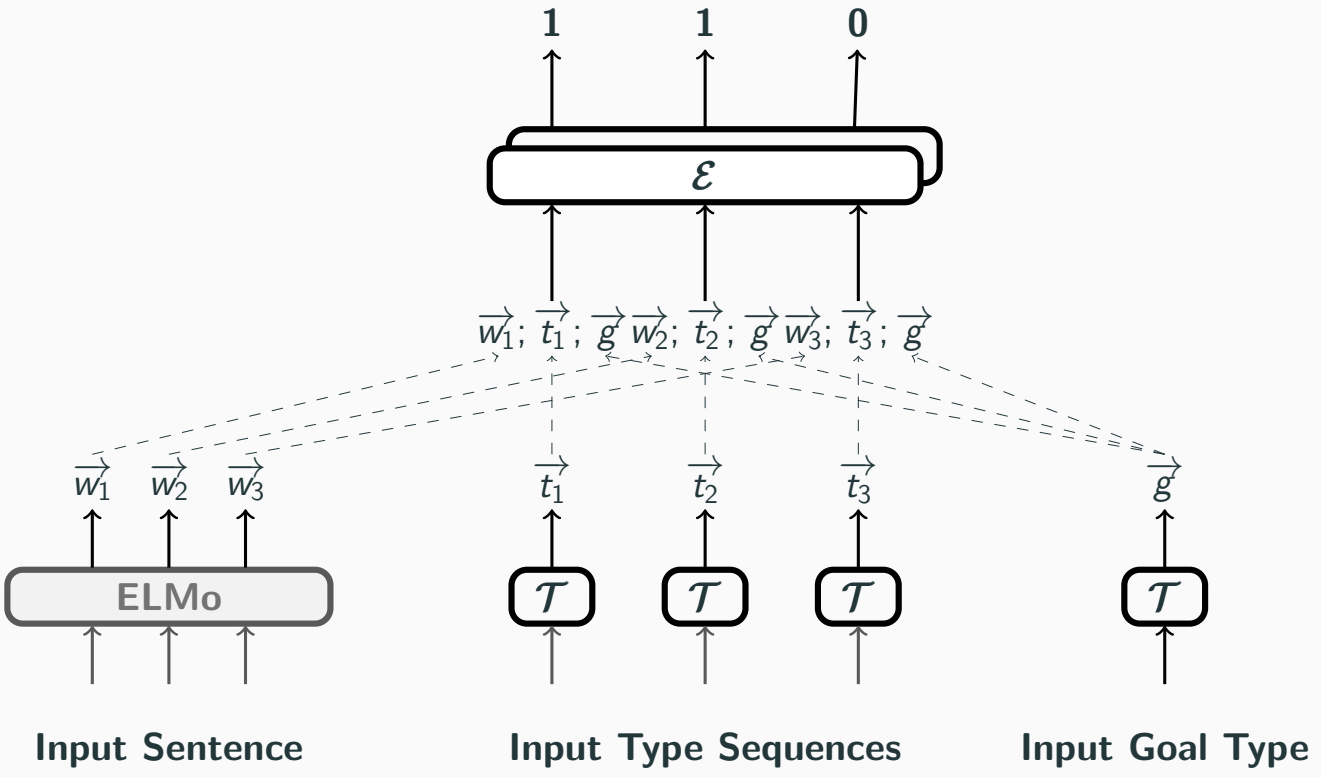
Parsing: Elimination

Given a sequence of word & type pairs

Split into two disjoint sequences..

..by assigning each item one of two labels

Parsing: Model & Results



Model	Full	Full-g	Full-g-t	Full-g-w
Accuracy (%)	97.15	95.3	87.77	94.2

Conclusion

Present & Future

Summary

- Fine-tuning between lexical & structural ambiguitiy
- Corpus-driven lexicon
- Constructive supertagging
- Preliminary parsing experiments

Next Steps

- Parsing higher-order types & coordination
- Supertagging & Parsing Integration
- Semantic Interpretations . . .

El Fin

$$\dfrac{\dfrac{}{\mathsf{thank} \vdash \Diamond^{obj}\,\mathrm{VNW} \to \mathrm{S}}\,Ax. \quad \dfrac{\dfrac{}{\mathsf{you} \vdash \mathrm{VNW}}\,Ax.}{\langle \mathsf{you} \rangle^{obj} \vdash \Diamond^{obj}\,\mathrm{VNW}}\,\Diamond^{obj} I}{\mathsf{thank}, \langle \mathsf{you} \rangle^{obj} \vdash \mathrm{S}}\,\to E$$