From Raw Text to Linear λ-Terms

Konstantinos Kogkalidis

July 2019

Compositionality in formal and distributional models of natural language

Overview

Big Question

Where do λ-terms come from anyway?

Overview

Big Question

Where do λ-terms come from anyway?

A syntactic framework for Semantic Compositionality

1. Type Logic
2. Type Lexicon
3. Type Assignment (Supertagging)
4. Parsing & Surface Form

Logic

Lambek Types, Lexical Ambiguity & Wide Coverage

Phrase	Structure	Verbal Type
eenden eten$_1$ vis *ducks eat fish*	SVO	$(\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}$
eten$_2$ eenden vis? *do ducks eat fish?*	VSO	$(\mathrm{S}/\mathrm{NP})/\mathrm{NP}$
eenden die vis eten$_{3,4}$ *ducks that eat fish*	SOV	$\mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S})$
eenden die vis eten$_{3,4}$ *ducks that fish eat*	OSV	$\mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S})$

...

Lambek Types, Lexical Ambiguity & Wide Coverage

Phrase	Structure	Verbal Type
eenden eten$_1$ vis *ducks eat fish*	SVO	$(\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP}$
eten$_2$ eenden vis? *do ducks eat fish?*	VSO	$(\mathrm{S}/\mathrm{NP})/\mathrm{NP}$
eenden die vis eten$_{3,4}$ *ducks that eat fish*	SOV	$\mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S})$
eenden die vis eten$_{3,4}$ *ducks that fish eat*	OSV	$\mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S})$

...

$$\begin{aligned}\mathcal{L} := \text{eten}_1 &: (\mathrm{NP}\backslash\mathrm{S})/\mathrm{NP},\\ \text{eten}_2 &: (\mathrm{S}/\mathrm{NP})/\mathrm{NP}\\ \text{eten}_{3,4} &: \mathrm{NP}\backslash(\mathrm{NP}\backslash\mathrm{S}),\\ &\ldots\end{aligned}$$

☹

Abstract Syntax with MILL

Inductive Type Scheme

$$\mathcal{T}_A := A \mid T_1 \to T_2$$

Logical Rules & Computational Terms

$$\frac{\Gamma \vdash s : A \to B \quad \Delta \vdash t : A}{\Gamma, \Delta \vdash s\langle t \rangle : B}$$

$$\frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x.u : A \to B}$$

Lexical vs. Structural Ambiguity

Smaller Lexicon

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

Lexical vs. Structural Ambiguity

Smaller Lexicon

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

More Proofs

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

`(eten vis) eenden` ✓

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

`(eten eenden) vis` ✗

Dependency Decorations

Replace $\rightarrow$ with dependency-decorated variants:

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

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

Dependency Decorations

Replace $\rightarrow$ with dependency-decorated variants:

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

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

Lexical Preferences + Decorations $\Longrightarrow$ reduced ambiguity

Dependency Decorations

Replace $\rightarrow$ with dependency-decorated variants:

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

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

Lexical Preferences + Decorations $\Longrightarrow$ reduced ambiguity

Formally

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

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

Lexicon

Grammar Extraction

Goal

From syntactically-annotated corpora to type grammars

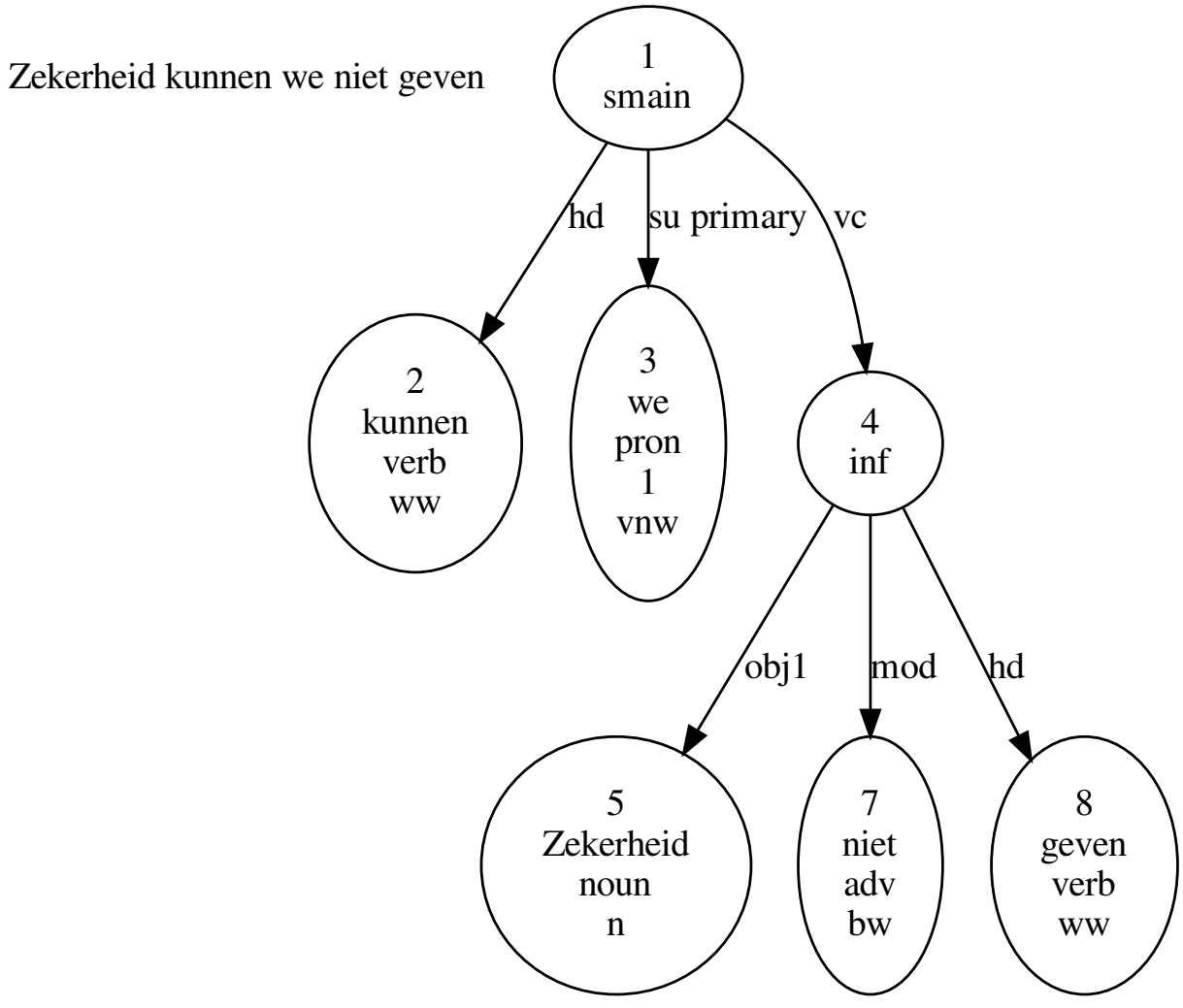
Grammar Extraction

Goal

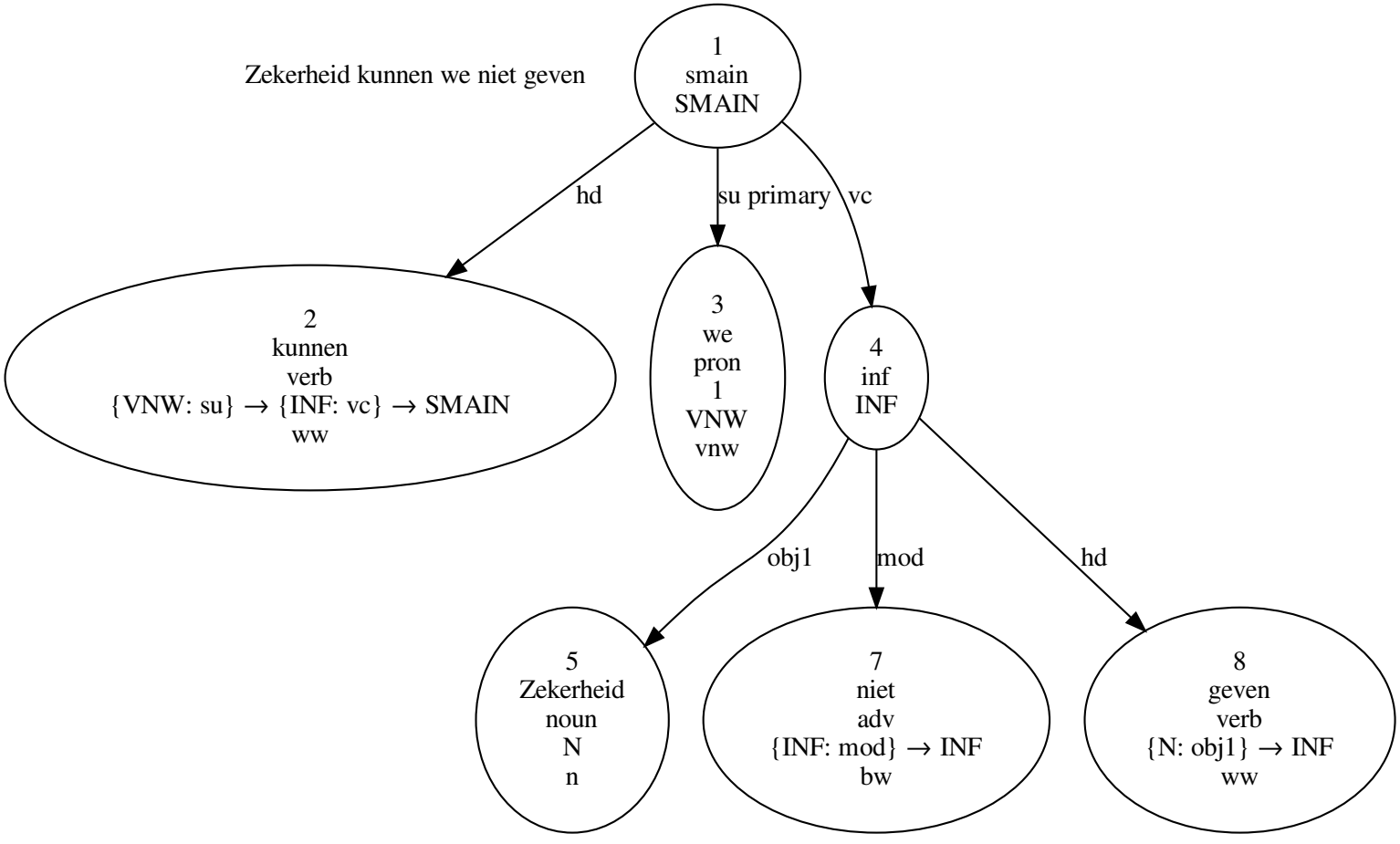
From syntactically-annotated corpora to type grammars

Lassy-Small	$\rightarrow$	**Type Grammar**
$\sim$65 000 sentences	$\rightarrow$	Type Sequences
$\sim$30 POS Tags & Phrasal Categories	$\rightarrow$	Atomic Types
$\sim$30 Dependency Labels	$\rightarrow$	Modal Decorations
$\sim$1 mil words	$\rightarrow$	Type Lexicon $\mathcal{L}$

Grammar Extraction: Example



Extraction: Example

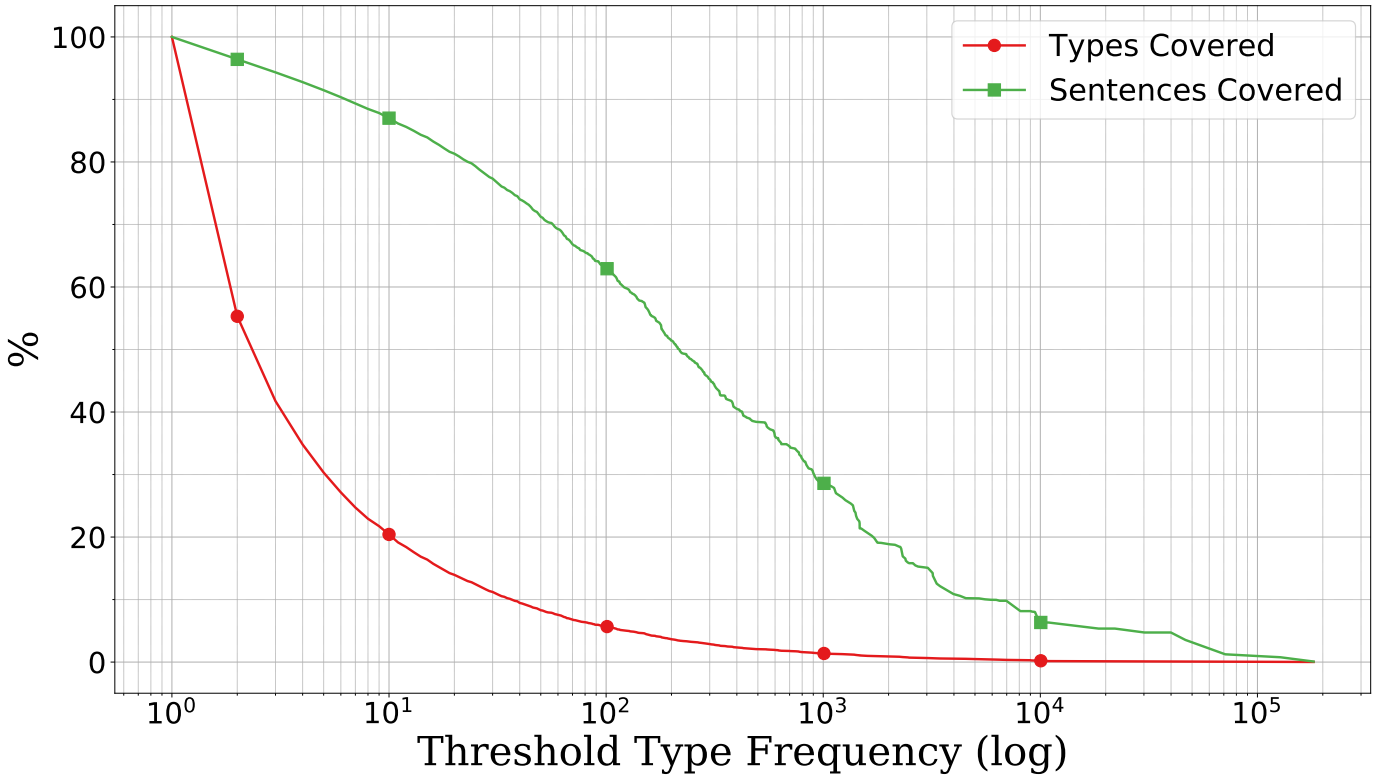


Grammar Extraction: Lexicon

Size

70 000 unique tokens

6 000 unique types (!)



Supertagging

Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

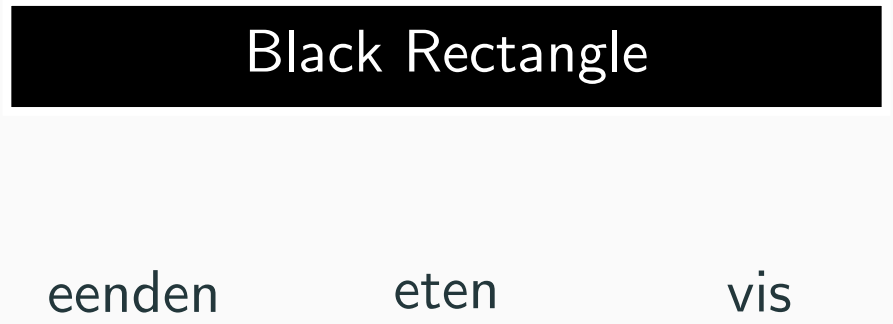
predict a class for each sequence item (types)

Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)

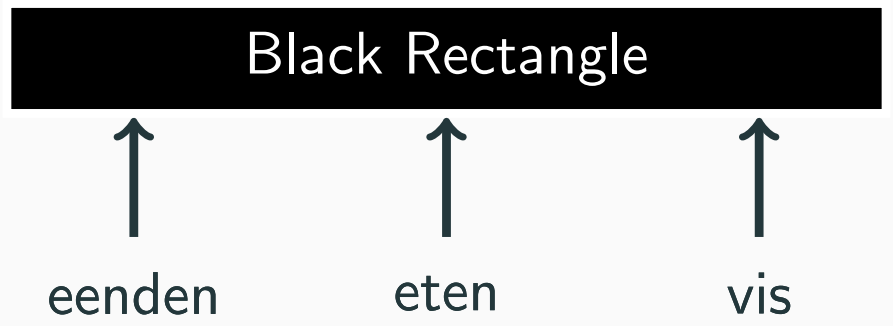


Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)

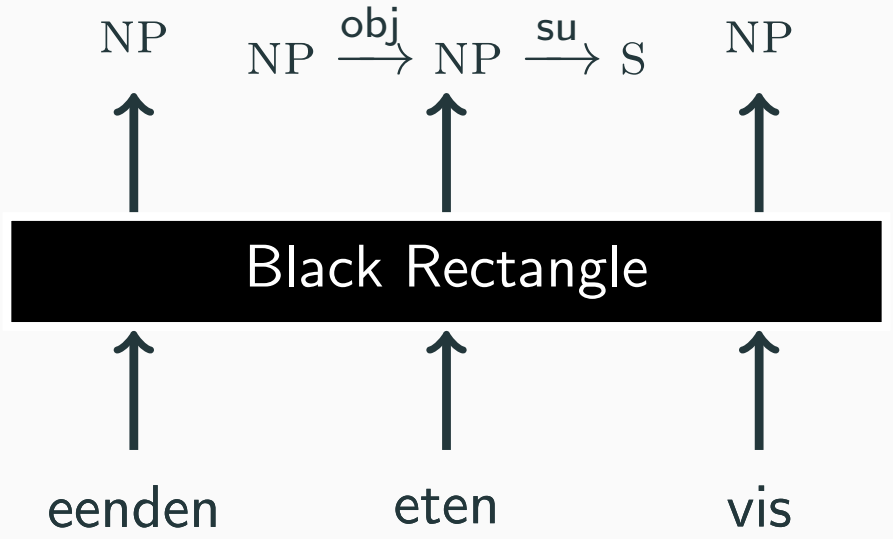


Supertagging: Standard Approach

Sequence Classification

Given input data sequence (word vectors)

predict a class for each sequence item (types)



Supertagging: Standard Approach

The Problem

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

Supertagging: An Alternative

Type Syntax

A CFG of two meta-rules

$$\forall\, A \in \mathcal{A} : S \implies A$$

$$\forall\, d \in \mathcal{D} : S \implies S \xrightarrow{\mathrm{d}} S$$

Supertagging: An Alternative

Type Syntax

A CFG of two meta-rules

$$\forall\, A \in \mathcal{A} : S \implies A$$

$$\forall\, d \in \mathcal{D} : S \implies S \xrightarrow{\mathrm{d}} S$$

CFGs: learnable

Supertagging: learnable

Supertagging: An Alternative

Type Syntax

A CFG of two meta-rules

$$\forall\, A \in \mathcal{A} : S \implies A$$

$$\forall\, d \in \mathcal{D} : S \implies S \xrightarrow{\mathrm{d}} S$$

CFGs: learnable

Supertagging: learnable

CFG + Supertagging $\implies$ Unbounded Co-domain

Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418

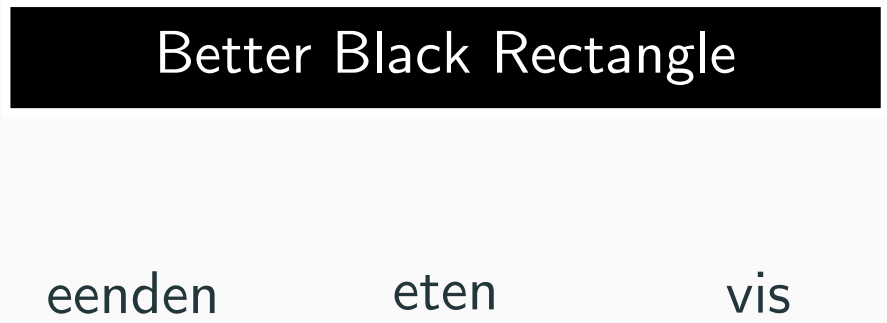
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



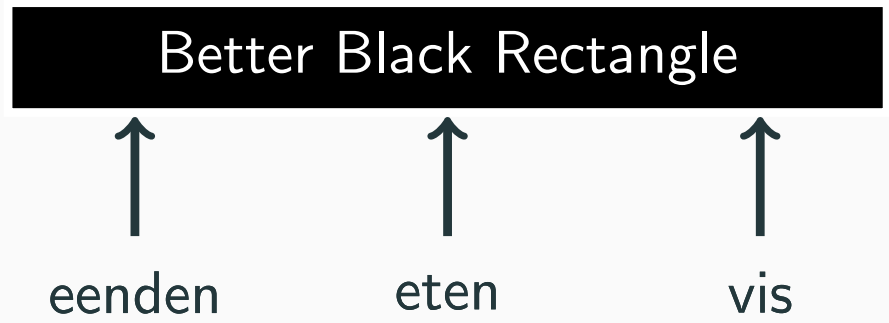
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



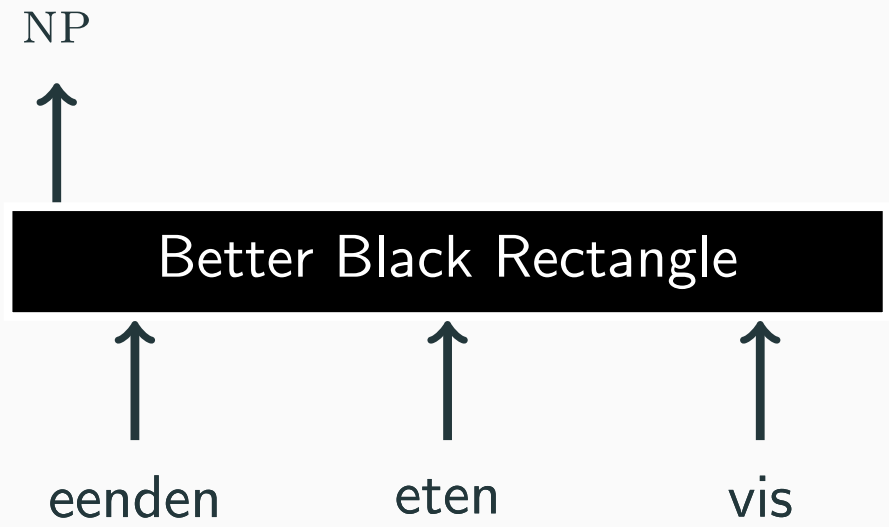
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



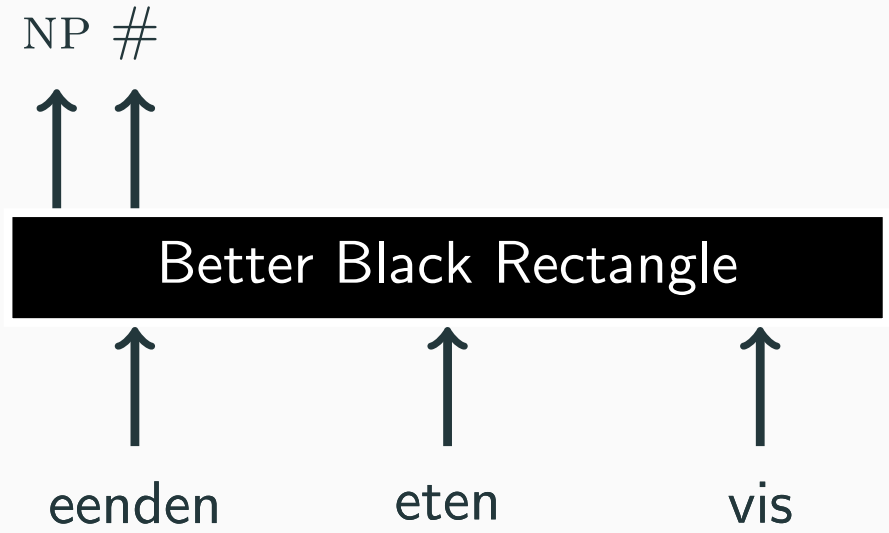
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



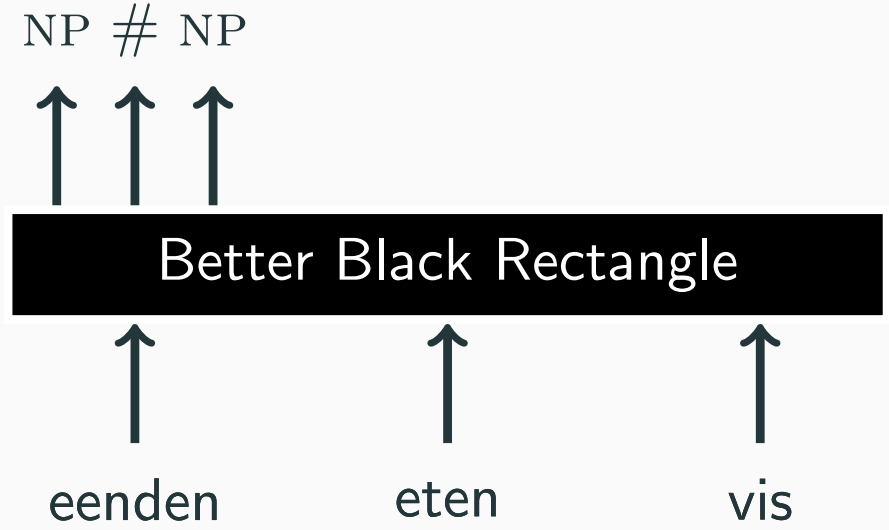
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



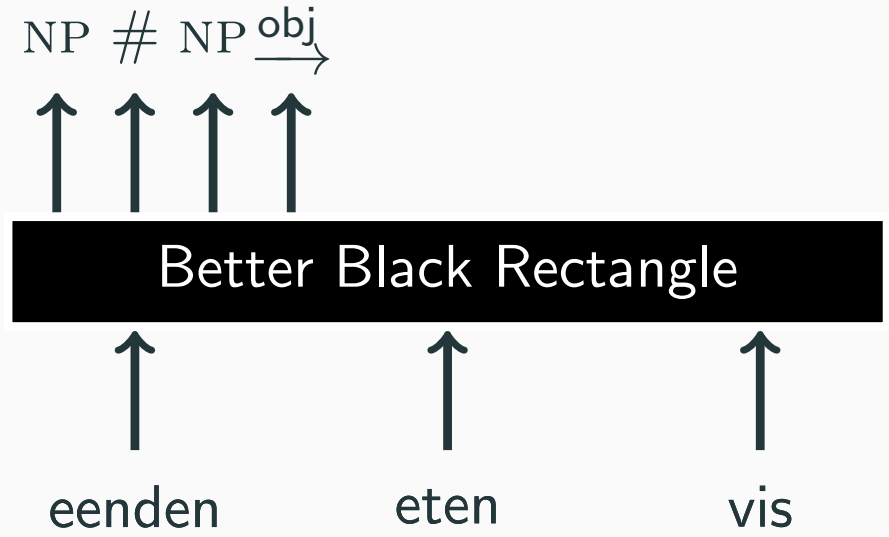
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



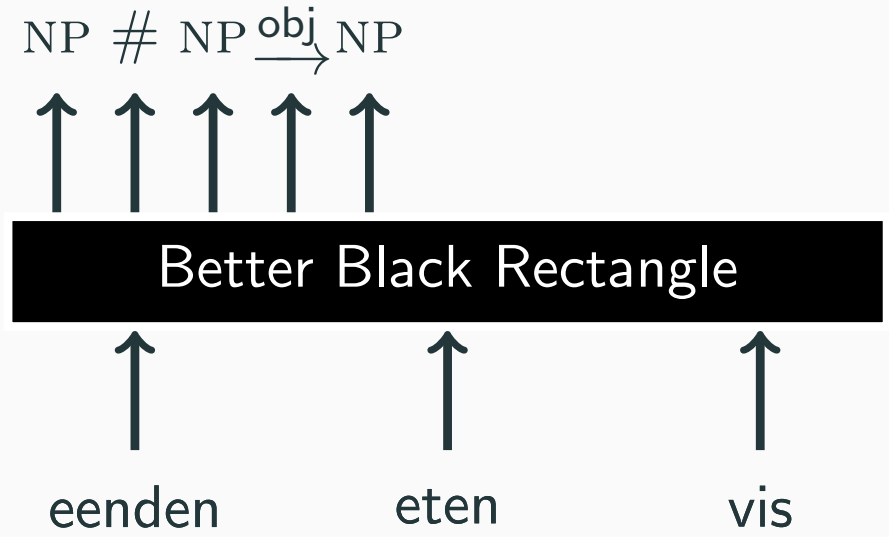
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



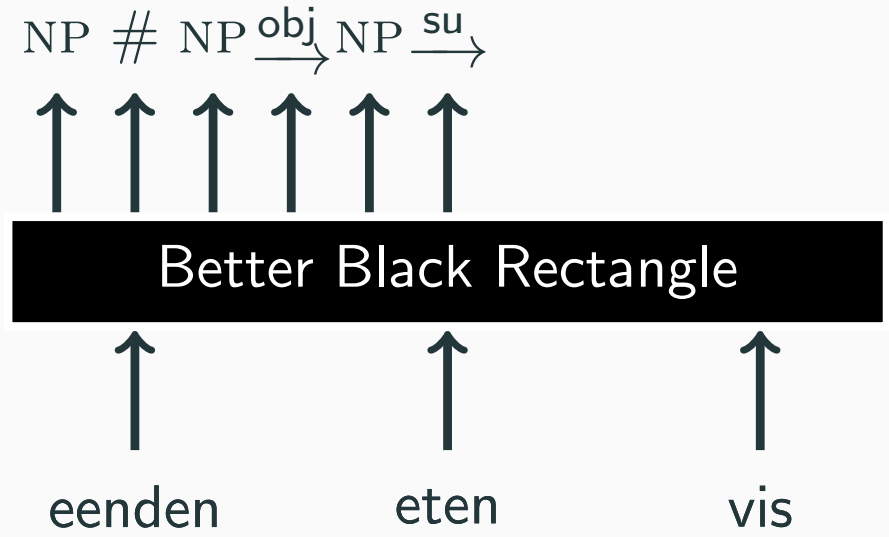
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



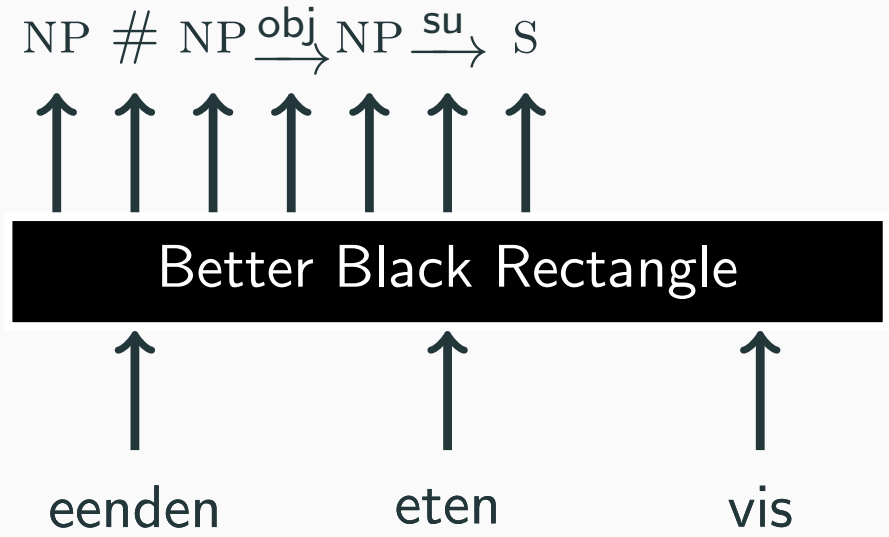
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



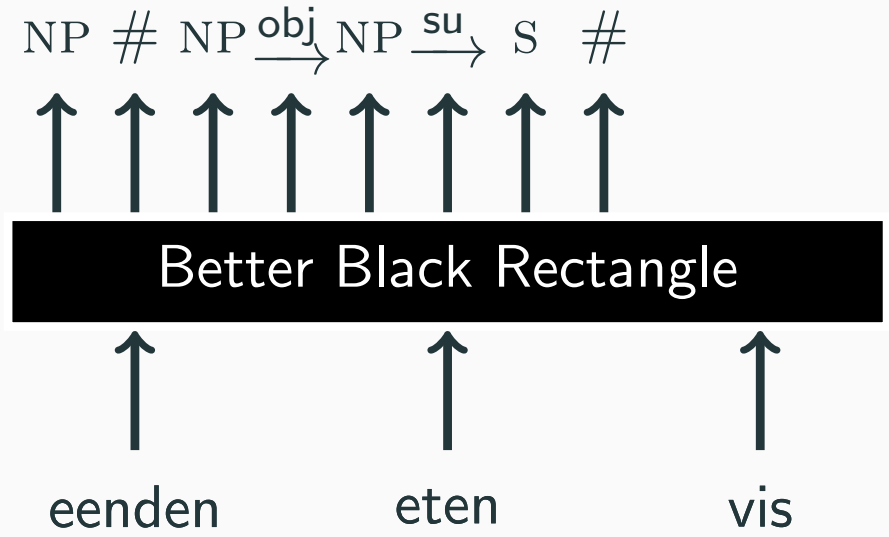
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



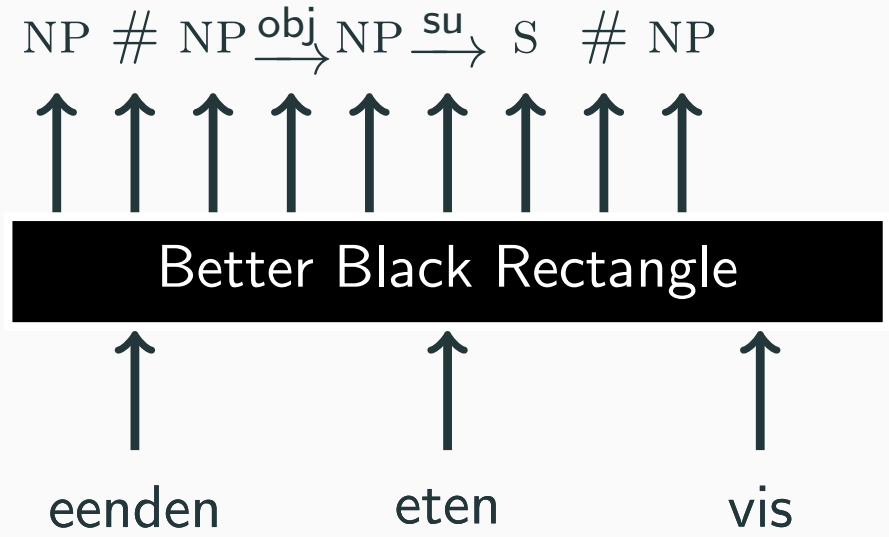
Supertagging: Unbounded co-domain

Reformulation

Given input data sequence (word vectors)

generate an output sequence (atomic types & binary connectives)

arxiv: 1905.13418



Parsing

Parsing: Overview

Parse $\equiv$ Proof

- Simulate the logical rules
- Navigate the proof space

Parsing: Overview

Parse $\equiv$ Proof

- Simulate the logical rules
- Navigate the proof space

ACG Perspective: $\mathcal{S} \xrightarrow{hom} \mathcal{T}$

From Abstract Structure to Surface Form

Parsing: Framework

Parse State

- A logical judgement
- Word associations for (some of) the premise formulas
- 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 (non-contiguous) sequences..

Parsing: Elimination

Given a sequence of word & type pairs

Split into two disjoint (non-contiguous) sequences..

..by assigning each item one of two labels

..binary sequence classification (!)

Parsing: Example

$$\dfrac{\dfrac{}{\text{eten} \vdash \text{NP} \xrightarrow{\text{obj}} \text{NP} \xrightarrow{\text{su}} \text{S}}\,Ax. \qquad \dfrac{}{\text{vis} \vdash \text{NP}}\,Ax.}{\text{eten, vis} \vdash \text{NP} \xrightarrow{\text{su}} \text{S}}\,{\to}E \qquad \dfrac{}{\text{eenden} \vdash \text{NP}}\,Ax.}{\text{eenden, eten, vis} \vdash \text{S}}$$

Black Rectangle of Parsing

$$\langle \text{eenden}, \text{NP} \rangle,\ \langle \text{eten}, \text{NP} \xrightarrow{\text{obj}} \text{NP} \xrightarrow{\text{su}} \text{S} \rangle,\ \langle \text{vis}, \text{NP} \rangle \vdash \text{S}$$

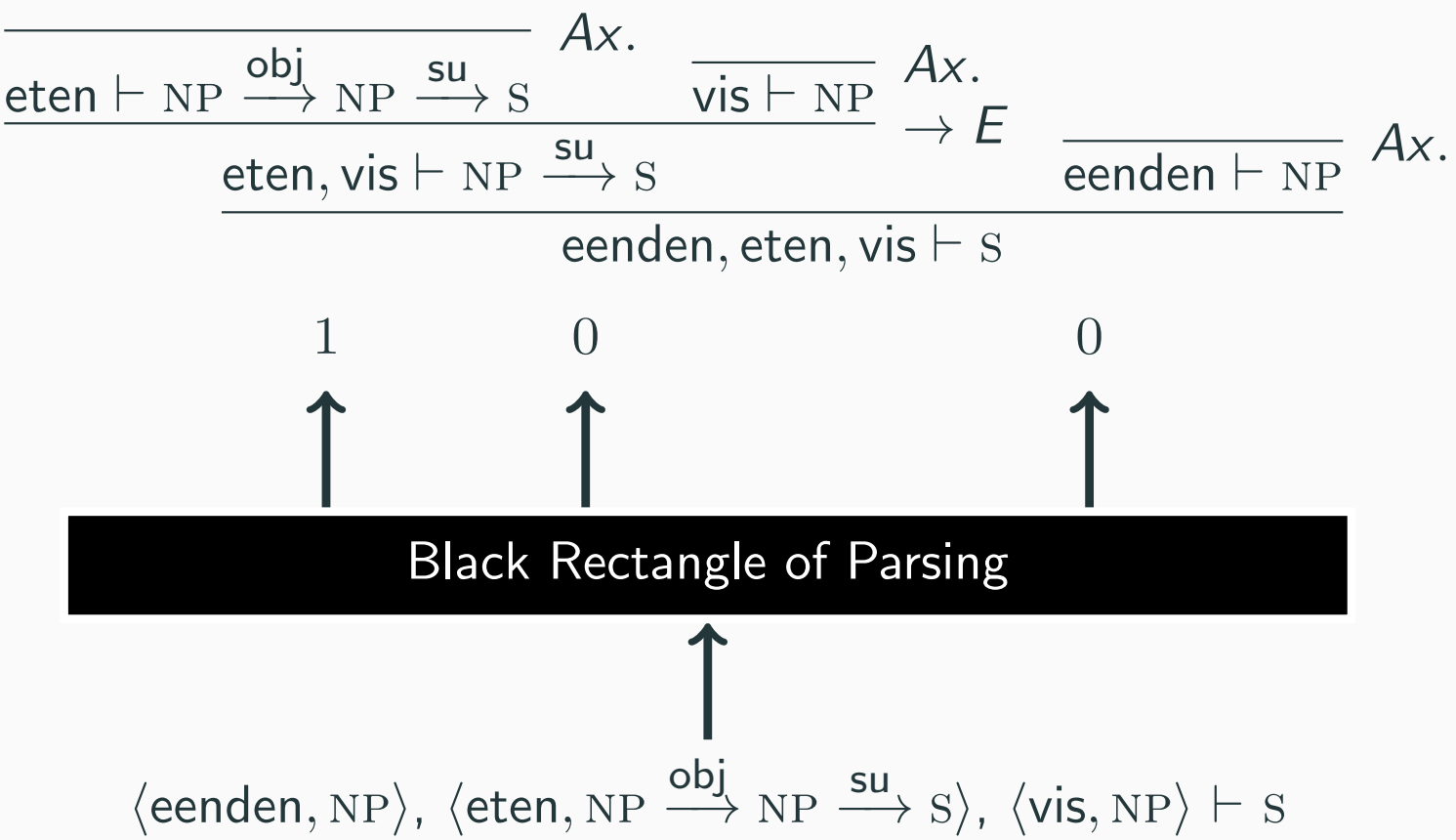
Parsing: Example

$$\dfrac{\dfrac{}{\text{eten} \vdash \text{NP} \xrightarrow{\text{obj}} \text{NP} \xrightarrow{\text{su}} \text{S}}\,Ax. \qquad \dfrac{}{\text{vis} \vdash \text{NP}}\,Ax.}{\text{eten, vis} \vdash \text{NP} \xrightarrow{\text{su}} \text{S}}\,{\to}E \qquad \dfrac{}{\text{eenden} \vdash \text{NP}}\,Ax.}{\text{eenden, eten, vis} \vdash \text{S}}$$

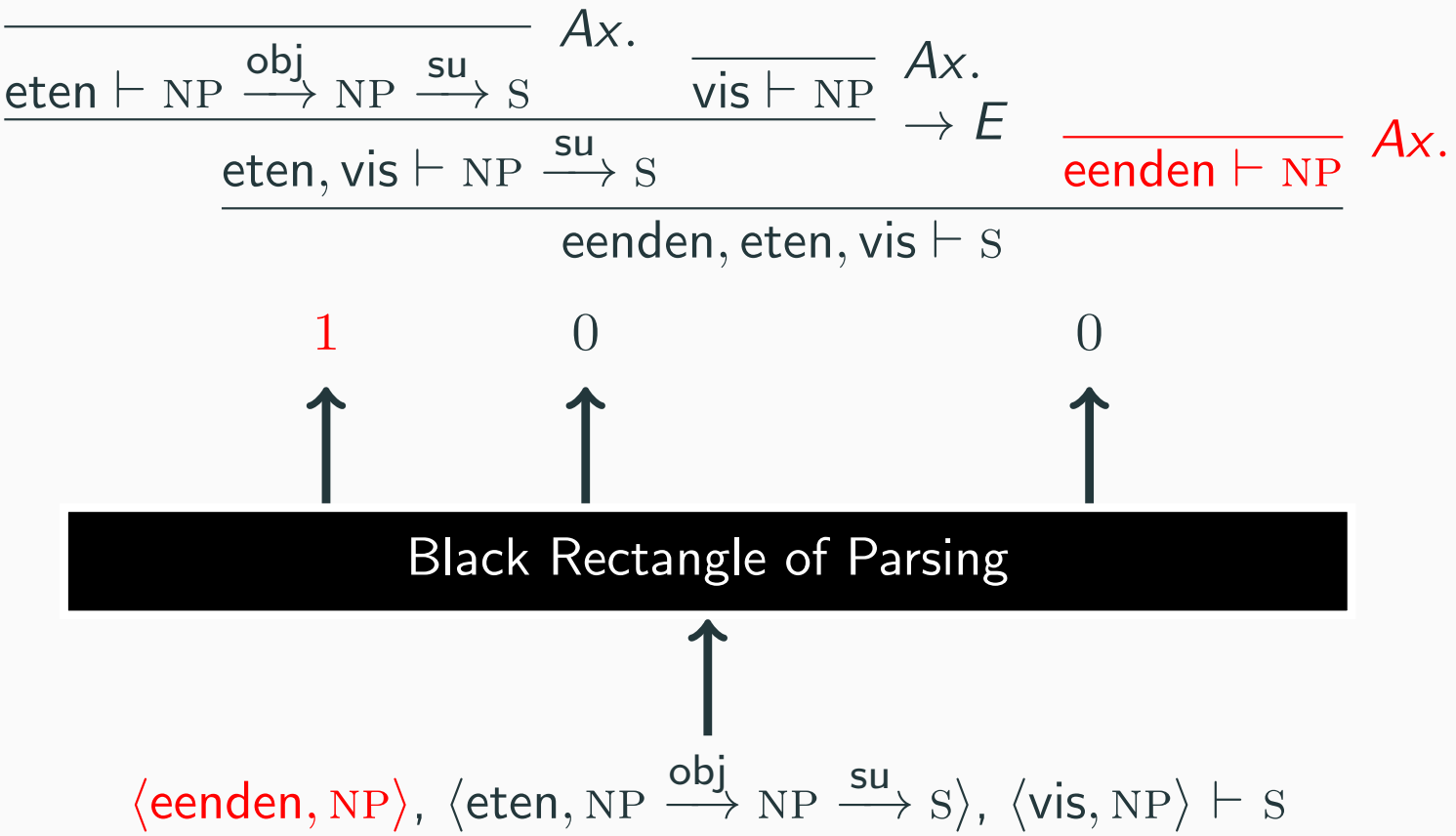
Black Rectangle of Parsing

$$\langle \text{eenden}, \text{NP} \rangle,\ \langle \text{eten}, \text{NP} \xrightarrow{\text{obj}} \text{NP} \xrightarrow{\text{su}} \text{S} \rangle,\ \langle \text{vis}, \text{NP} \rangle \vdash \text{S}$$

Parsing: Example



Parsing: Example



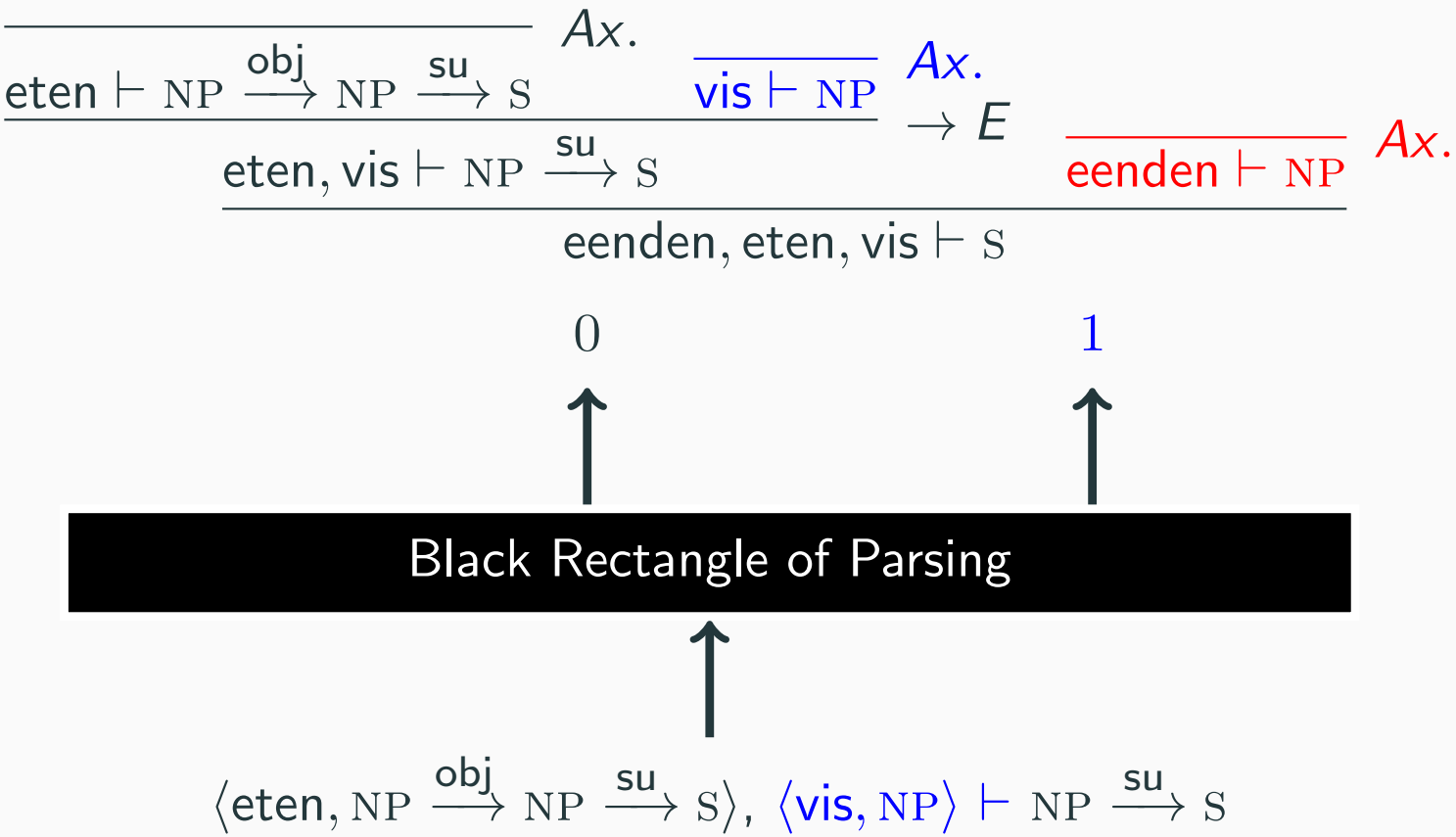
Parsing: Example

$$\dfrac{\dfrac{}{\text{eten} \vdash \text{NP} \xrightarrow{\text{obj}} \text{NP} \xrightarrow{\text{su}} \text{S}}\,Ax. \qquad \dfrac{}{\text{vis} \vdash \text{NP}}\,Ax.}{\text{eten, vis} \vdash \text{NP} \xrightarrow{\text{su}} \text{S}}\,{\to}E \qquad \dfrac{}{\text{eenden} \vdash \text{NP}}\,Ax.}{\text{eenden, eten, vis} \vdash \text{S}}$$

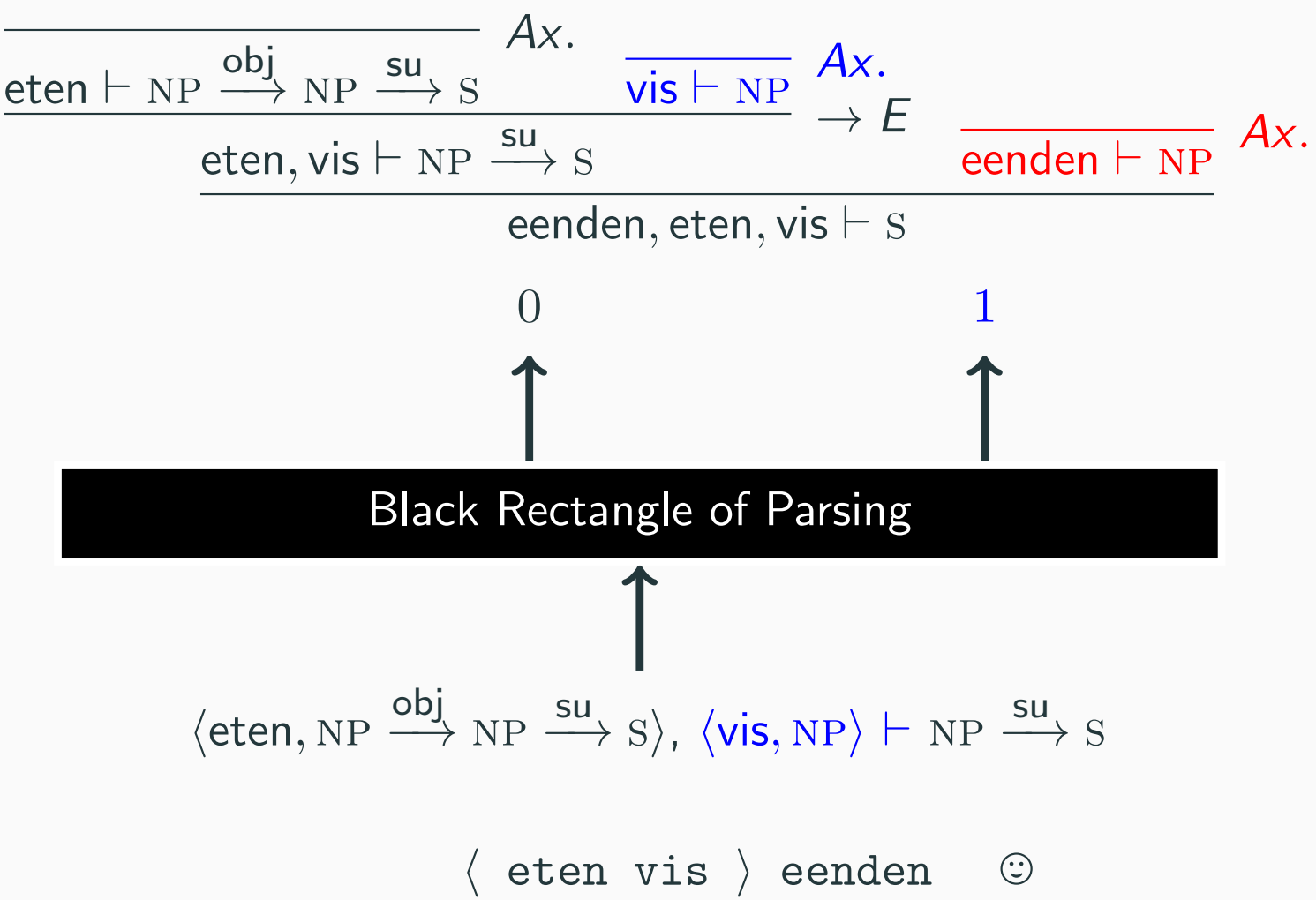
Black Rectangle of Parsing

$$\langle \text{eten}, \text{NP} \xrightarrow{\text{obj}} \text{NP} \xrightarrow{\text{su}} \text{S} \rangle,\ \langle \text{vis}, \text{NP} \rangle \vdash \text{NP} \xrightarrow{\text{su}} \text{S}$$

Parsing: Example



..Semantics, finally (computational)



..Semantics, finally (your own)

Semantic Interpretation

From Abstract Syntax to Concrete Semantics

$\mathcal{S} \xrightarrow{\text{hom}} \mathcal{O}$

- Relate MILL types to semantic counterparts
- Provide lexical meaning formulas for constants

$\lceil \langle \text{eten vis} \rangle \text{ eenden} \rceil = \langle \lceil \text{eten} \rceil \ \lceil \text{vis} \rceil \rangle \ \lceil \text{eenden} \rceil$

Compositional Thanks

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