Typed Supertags and Semantic Parses for Dutch

Konstantinos Kogkalidis, Michael Moortgat, Richard Moot

January 2020

UiL-OTS Utrecht University, Université de Montpellier, CNRS

Overview

- λ **Introduction** *(or: why types?)*
  Type-Logical Grammars
- λ **Framework** *(or: types, how?)*
  Type System
- λ **Resources**
  Type Lexicon
  Semantic Parses
- λ **Use Cases**

Type-Logical Grammars

TL;DR

Words assigned *formulas*, parsing a process of formal *deduction*.

Syntax

Structural Well-Formedness $\equiv$ Derivability

Type-Logical Grammars

TL;DR

Words assigned *formulas*, parsing a process of formal *deduction*.

Syntax

Structural Well-Formedness $\equiv$ Derivability

Curry-Howard Isomorphism

Propositions $\equiv$ Types

Proofs $\equiv$ Functional Programs

Syntax-Semantics Interface

Parse $\equiv$ Proof $\equiv$ Computational Terms

Type System (1/2)

IILL

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

$A \in \mathcal{A}$ :: Atoms denoting complete phrases

N, NP, PRON, S ...

$T_1 \multimap T_2$ :: Linear functor from T_1 to T_2

NP $\multimap$ S, NP $\multimap$ NP, NP $\multimap$ NP $\multimap$ S, (NP $\multimap$ NP) $\multimap$ (NP $\multimap$ NP)

Type System (1/2)

IILL

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

$A \in \mathcal{A}$ :: Atoms denoting complete phrases

N, NP, PRON, S ...

$T_1 \multimap T_2$ :: Linear functor from T_1 to T_2

NP $\multimap$ S, NP $\multimap$ NP, NP $\multimap$ NP $\multimap$ S, (NP $\multimap$ NP) $\multimap$ (NP $\multimap$ NP)

$$\frac{\Gamma \vdash M : A \multimap B \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash (M\ N) : B}\ E$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \multimap B}\ I$$

Type System (2/2)

Modal Decoration Refinement

$$\mathcal{T} := A \mid T_1 \multimap T_2 \mid \Diamond^d T_1 \multimap T_2 \mid \Box^d(T_1 \multimap T_2)$$

$d \in \mathcal{D}$ :: Dependency relations

subj, dobj, body, mod, app, det ...

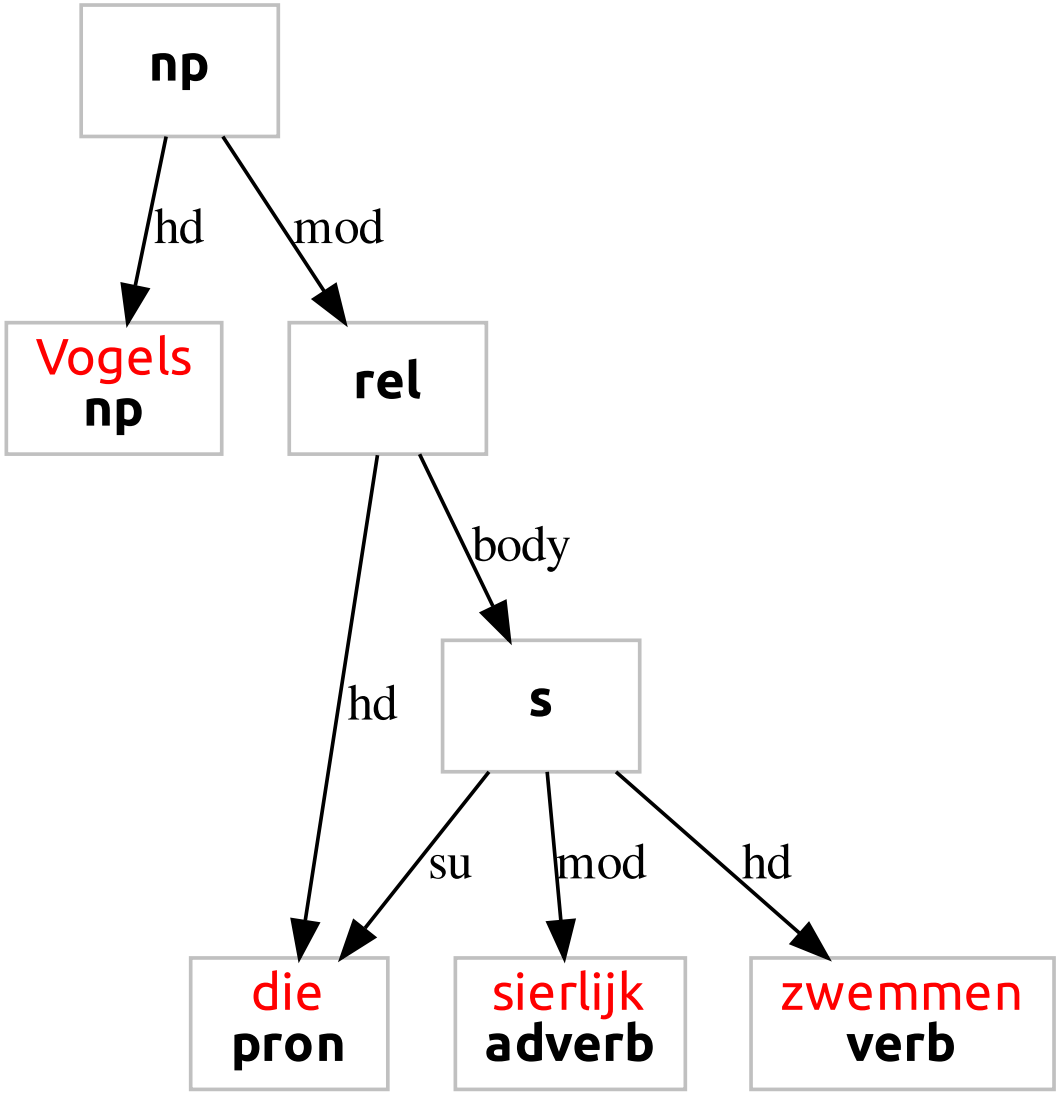
$\Diamond^d T_1 \multimap T_2$ :: .. head fn selecting complement with dependency role d

$\Diamond^{subj}$ NP $\multimap$ S, $\Diamond^{dobj}$ NP $\multimap \Diamond^{subj}$ NP $\multimap$ S, ...

$\Box^d(T_1 \multimap T_2)$ :: .. non-head fn projecting dependency d

$\Box^{det}$ N $\multimap$ NP, $\Box^{mod}$ ($\Box^{mod}$ NP $\multimap$ NP) $\multimap \Box^{mod}$ NP $\multimap$ NP

Ducks, in Lassy



Ducks, Proven

$$\dfrac{\dfrac{vogels}{NP}\,\mathcal{L} \qquad \dfrac{\dfrac{die}{RELPRO^{1}}\,\mathcal{L} \qquad \dfrac{\dfrac{\dfrac{}{\diamond^{subj} PRON}\,Ax \qquad \dfrac{zwemmen}{\diamond^{subj} PRON \multimap S}\,\mathcal{L}}{\langle PRON\rangle^{subj}\,\text{zwemmen} \vdash S}\,E \qquad \dfrac{sierlijk}{\square^{mod}(S \multimap S)}\,\mathcal{L}}{\langle PRON\rangle^{subj}\,\langle \text{sierlijk}\rangle^{mod}\,\text{zwemmen} \vdash S}\,E}{\langle \text{sierlijk}\rangle^{mod}\,\text{zwemmen} \vdash \diamond^{subj} PRON \multimap S}\,I}{\text{die}\left\langle \langle \text{sierlijk}\rangle^{mod}\,\text{zwemmen}\right\rangle^{body} \vdash \square^{mod}(NP \multimap NP)}\,E}{vogels\left\langle die\left\langle \langle sierlijk\rangle^{mod}\ zwemmen\right\rangle^{body}\right\rangle^{mod} \vdash NP}\,E$$

$$\texttt{die}\,(\lambda x.\,(\texttt{sierlijk}\ (\texttt{zwemmen}\ x)))\ \texttt{vogels}$$

1: $RELPRO := \diamond^{body}(\diamond^{su} PRON \multimap S) \multimap \square^{mod}(NP \multimap NP)$

Extraction

From Lassy Parses to IILL Types & Theorems

arxiv: `abs/1912.12635`

Resources

1. Type Lexicon: Word $\rightarrow$ Type Distribution
2. Proofs: Lassy DAG $\rightarrow$ IILL Proof

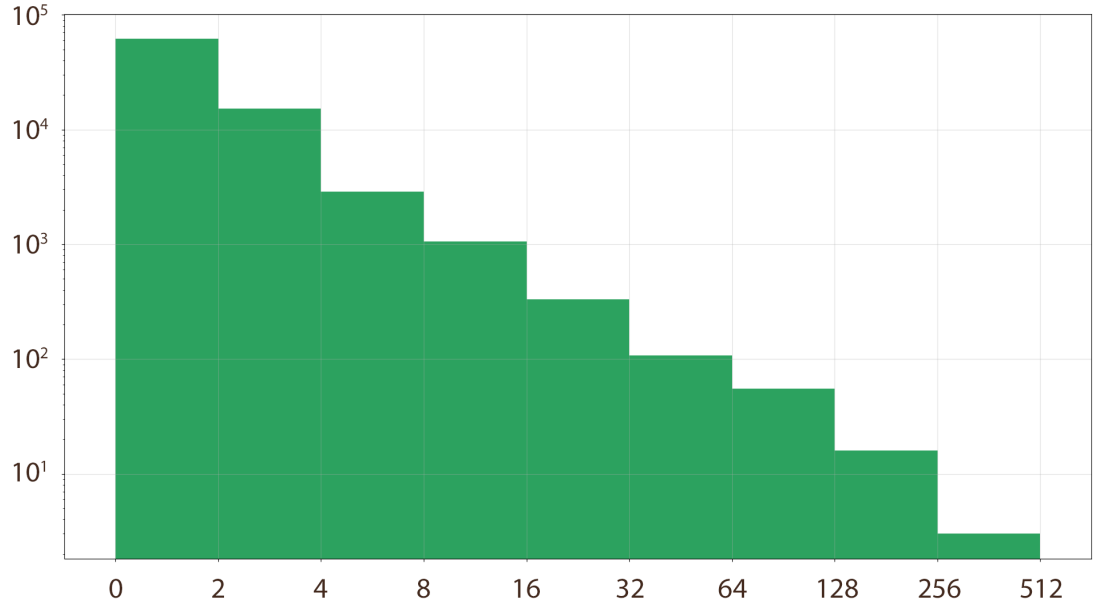
$\sim$ 97% *coverage*

Wikipedia subset publicly available at
`github.com/konstantinosKokos/aethel-public`

ÆTHEL: Lexicon

Stats

- ∼900 000 word & type pairs
- 81 730 unique words
- 5 771 unique semantic types



Lexical Type Ambiguity Histogram (*log10-log2*)

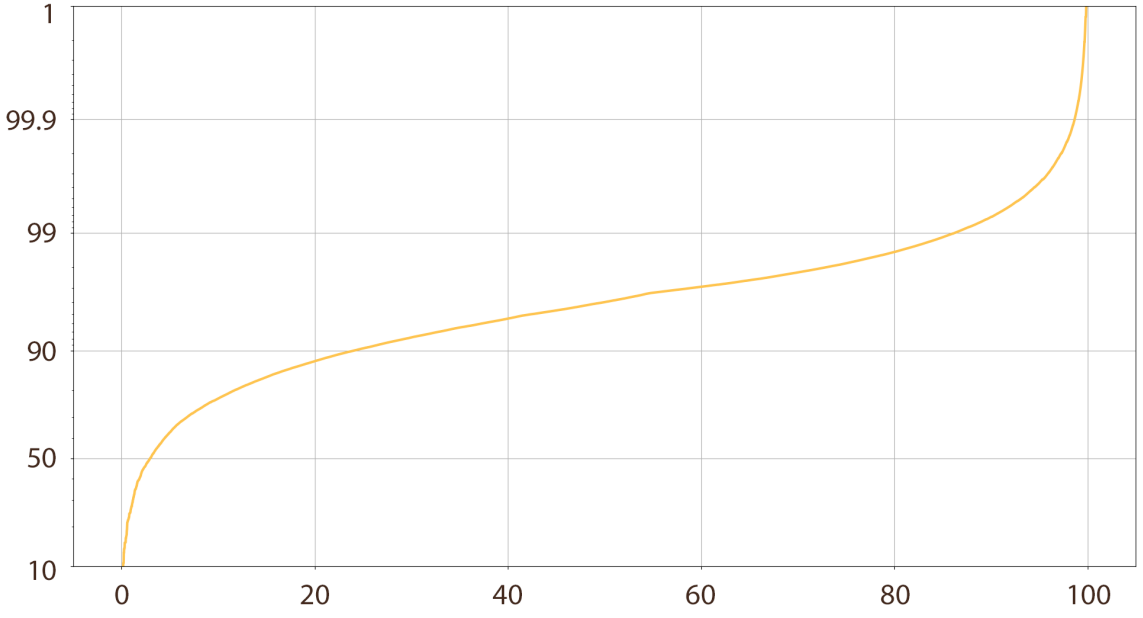
ÆTHEL: Proofs

Formats

- N.D. Proofs
- S.S. Proofs
- Linear Proofnets
- λ-terms

Stats

- 65 020 Lassy DAGs
- 72 263 IILL Proofs



Proof coverage w.r.t. most frequent types (*logit-linear*)

Use Cases & Applications

- Supertagging with no type lexicon

arxiv: `abs/1905.13418`

- Parsing with type hints

hal-lirmm: `lirmm-02313572`

- Type-aware language modeling
- Text to λ-term translation
- Semantic Compositionality
- ...?