Deductive Parsing

WITH AN UNBOUNDED TYPE LEXICON

K. Kogkalidis, M. Moortgat, R. Moot, G. Tziafas

August 2019

SemSpace 2019

Why Parsing?

Compositionality

Meaning of complex expression derived by constituent expressions and their means of interaction.

Why Parsing?

Compositionality

Meaning of complex expression derived by constituent expressions and their means of interaction.

Syntax

Algebra of sentence structure

Base for linguistically informed compositional semantics

Why Deductive?

Syntax

Why Deductive?

Type-Logical Grammars

Words $\rightarrow$ Logical Formulas

Well-Formedness $\equiv$ Provability

$$\text{Syntax} \xleftrightarrow{\text{TLG}} \text{Logic}$$

Why Deductive?

Type-Logical Grammars

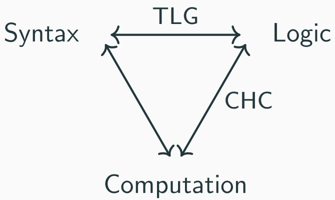
Words $\rightarrow$ Logical Formulas

Well-Formedness $\equiv$ Provability

Curry-Howard Correspondence

Logical Formulas $\leftrightarrow$ Typed Variables

Proofs $\equiv$ Functional Programs



Why Deductive?

Type-Logical Grammars

Words $\rightarrow$ Logical Formulas

Well-Formedness $\equiv$ Provability

Curry-Howard Correspondence

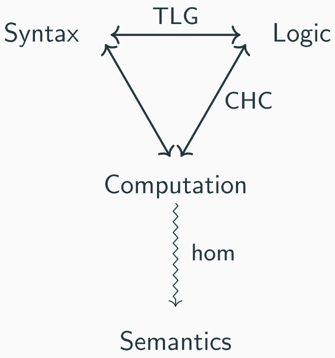
Logical Formulas $\leftrightarrow$ Typed Variables

Proofs $\equiv$ Functional Programs

Syntax-Semantics Interface

Syntactic Types $\rightarrow$ Semantic Spaces

Derivations $\rightarrow$ Semantic Programs



A Dependency-Decorated TLG

Lexicon: Words $\rightarrow$ dependency-decorated MILL types (*à la ACG*)

Constants: $\{$ NP, S, PRON $\ldots\}$

Functions: $\{ \diamond^{\mathsf{su}}\text{NP} \rightarrow \text{S}, \diamond^{\mathsf{su}}\text{NP} \rightarrow \left(\diamond^{\mathsf{obj}}\text{NP} \rightarrow \text{S}\right), \ldots\}$

$$\mathcal{T} := A \mid \diamond^{d}\ T_0 \mid T_1 \rightarrow T_2$$

A Dependency-Decorated TLG

Lexicon: Words $\rightarrow$ dependency-decorated MILL types (*à la ACG*)

Constants: $\{\ \textsc{np}, \textsc{s}, \textsc{pron}\ \ldots\}$

Functions: $\{\ \diamond^{\mathsf{su}}\textsc{np} \rightarrow \textsc{s},\ \diamond^{\mathsf{su}}\textsc{np} \rightarrow \left(\diamond^{\mathsf{obj}}\textsc{np} \rightarrow \textsc{s}\right),\ \ldots\}$

$$\mathcal{T} := A \mid\ \diamond^{d}\ T_0 \mid T_1 \rightarrow T_2$$

Parsing: Proof Search

$$\frac{\Gamma \vdash s : A \rightarrow B \quad \Delta \vdash t : A}{\Gamma, \Delta \vdash s\langle t \rangle : B} \rightarrow E \qquad\qquad \frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x.u : A \rightarrow B} \rightarrow I$$

Parsing Framework

Parse State

- A logical judgement (premises & conclusion)
- Word associations for (some) premise formulas
- A single element stack

Parsing Framework

Parse State

- A logical judgement (premises & conclusion)
- Word associations for (some) premise formulas
- A single element stack

Framework

Given a parse state

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

Ambiguities (the bad kind)

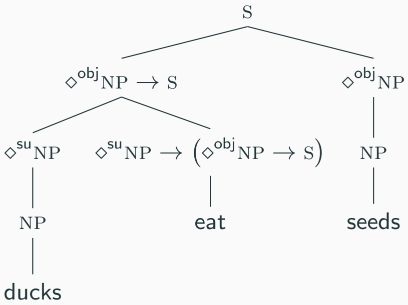
$$\mathcal{L} := \{\text{"}\textit{ducks}\text{"}: \textsc{np},\ \text{"}\textit{eat}\text{"}: \diamond^{\mathsf{su}}\textsc{np} \rightarrow \left(\diamond^{\mathsf{obj}}\textsc{np} \rightarrow \textsc{s}\right),\ \text{"}\textit{seeds}\text{"}: \textsc{np}\}$$

$$\text{"}\textit{ducks eat seeds}\text{"} \vdash^{?} \textsc{s}$$

Ambiguities (the bad kind)

$$\mathcal{L} := \{\text{“}\textit{ducks}\text{”}: \mathrm{NP},\ \text{“}\textit{eat}\text{”}: \diamond^{\mathsf{su}}\mathrm{NP} \to (\diamond^{\mathsf{obj}}\mathrm{NP} \to \mathrm{S}),\ \text{“}\textit{seeds}\text{”}: \mathrm{NP}\}$$

$$\text{“}\textit{ducks eat seeds}\text{”} \vdash^{?} \mathrm{S}$$

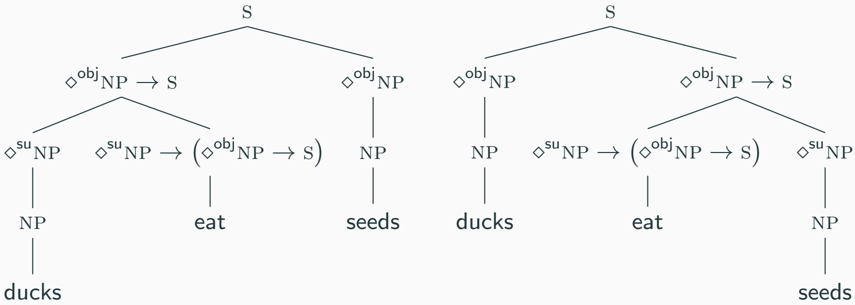


`(eat seeds) ducks` ✓

Ambiguities (the bad kind)

$$\mathcal{L} := \{\text{“}\textit{ducks}\text{”}: \mathrm{NP},\ \text{“}\textit{eat}\text{”}: \diamond^{\mathsf{su}}\mathrm{NP} \rightarrow (\diamond^{\mathsf{obj}}\mathrm{NP} \rightarrow \mathrm{S}),\ \text{“}\textit{seeds}\text{”}: \mathrm{NP}\}$$

$$\text{“}\textit{ducks eat seeds}\text{”} \vdash^{?} \mathrm{S}$$



`(eat seeds) ducks` ✓

`(eat ducks) seeds` ✗

Resolving Ambiguities

Key insight

Structure can be disambiguated by utilizing word and position information on top of types.

Resolving Ambiguities

Key insight

Structure can be disambiguated by utilizing word and position information on top of types.

Words (& Position)

Contextualized embeddings from some LM

Resolving Ambiguities

Key insight

Structure can be disambiguated by utilizing word and position information on top of types.

Words (& Position)

Contextualized embeddings from some LM

Types

Type-level recursive GRU

$$\lceil A \rceil = \overrightarrow{A}$$

$$\lceil \diamond^d X \to Y \rceil = GRU\left(\left[\overrightarrow{d}, \lceil X \rceil, \lceil Y \rceil\right]\right)$$

Elimination $\sim$?

Problem

Given a *judgement*, decide between possible *branchings*..

Elimination $\sim$?

Problem

Given a *judgement*, decide between possible *branchings*..

.. given a sequence of *word & type pairs*, assign each item a *binary label*

Elimination ∼ Binary Classification

Problem

Given a *judgement*, decide between possible *branchings*..

.. given a sequence of *word & type pairs*, assign each item a *binary label*

Binary Sequence classification (Deep bi-GRU)

- **Input**: Sequence of word & type vectors (conc.)
- **Output**: Sequence of binary labels

Proof Traversal

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

Deep bi-GRU

$$\left(\overrightarrow{\text{ducks}}; \lceil \mathrm{NP} \rceil\right), \quad \left(\overrightarrow{\text{eat}}; \lceil \mathrm{NP} \to \mathrm{NP} \to \mathrm{S} \rceil\right), \quad \left(\overrightarrow{\text{seeds}}; \lceil \mathrm{NP} \rceil\right) \vdash \lceil \mathrm{S} \rceil$$

Proof Traversal

$$\dfrac{\dfrac{\dfrac{}{\text{eat} \vdash \text{NP} \to \text{NP} \to \text{S}}\ Ax. \quad \dfrac{}{\text{seeds} \vdash \text{NP}}\ Ax.}{\text{eat}, \text{seeds} \vdash \text{NP} \to \text{S}}\ \to E \quad \dfrac{}{\text{ducks} \vdash \text{NP}}\ Ax.}{\text{ducks}, \text{eat}, \text{seeds} \vdash \text{S}}$$

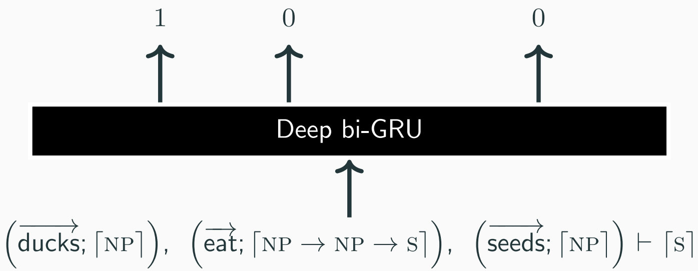
Deep bi-GRU

↑

$$\left(\overrightarrow{\text{ducks}}; \lceil \text{NP} \rceil\right), \quad \left(\overrightarrow{\text{eat}}; \lceil \text{NP} \to \text{NP} \to \text{S} \rceil\right), \quad \left(\overrightarrow{\text{seeds}}; \lceil \text{NP} \rceil\right) \vdash \lceil \text{S} \rceil$$

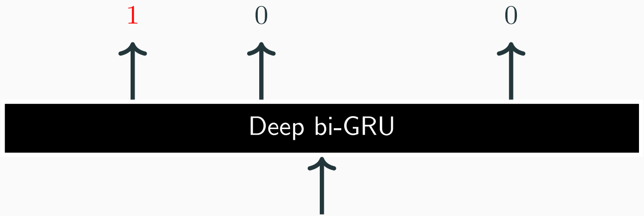
Proof Traversal

$$\dfrac{\dfrac{\dfrac{}{\text{eat} \vdash \mathrm{NP} \to \mathrm{NP} \to \mathrm{S}}\ Ax. \quad \dfrac{}{\text{seeds} \vdash \mathrm{NP}}\ Ax.}{\text{eat}, \text{seeds} \vdash \mathrm{NP} \to \mathrm{S}} \to E \quad \dfrac{}{\text{ducks} \vdash \mathrm{NP}}\ Ax.}{\text{ducks}, \text{eat}, \text{seeds} \vdash \mathrm{S}}$$



Proof Traversal

$$\dfrac{\dfrac{\dfrac{}{\text{eat} \vdash \mathrm{NP} \to \mathrm{NP} \to \mathrm{S}}\ \textit{Ax.} \quad \dfrac{}{\text{seeds} \vdash \mathrm{NP}}\ \textit{Ax.}}{\text{eat}, \text{seeds} \vdash \mathrm{NP} \to \mathrm{S}} \to E \quad \dfrac{}{\color{red}{\text{ducks} \vdash \mathrm{NP}}}\ \color{red}{\textit{Ax.}}}{\text{ducks}, \text{eat}, \text{seeds} \vdash \mathrm{S}}$$



$$\color{red}{\left(\overrightarrow{\text{ducks}}; \lceil \mathrm{NP} \rceil\right)}, \quad \left(\overrightarrow{\text{eat}}; \lceil \mathrm{NP} \to \mathrm{NP} \to \mathrm{S} \rceil\right), \quad \left(\overrightarrow{\text{seeds}}; \lceil \mathrm{NP} \rceil\right) \vdash \lceil \mathrm{S} \rceil$$

Proof Traversal

$$\dfrac{\dfrac{\dfrac{}{\text{eat} \vdash \mathrm{NP} \to \mathrm{NP} \to \mathrm{S}}\,\mathit{Ax.} \qquad \dfrac{}{\text{seeds} \vdash \mathrm{NP}}\,\mathit{Ax.}}{\text{eat}, \text{seeds} \vdash \mathrm{NP} \to \mathrm{S}}\,\to E \qquad \dfrac{}{\text{ducks} \vdash \mathrm{NP}}\,\mathit{Ax.}}{\text{ducks}, \text{eat}, \text{seeds} \vdash \mathrm{S}}$$

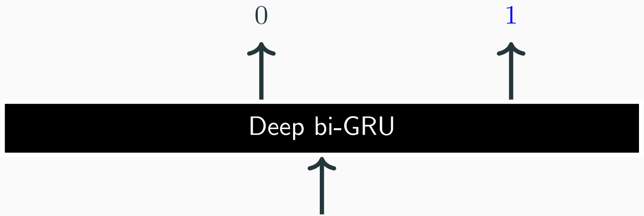
Deep bi-GRU

↑

$$\left(\overrightarrow{\text{eat}}; \lceil \mathrm{NP} \to \mathrm{NP} \to \mathrm{S} \rceil\right), \ \left(\overrightarrow{\text{seeds}}; \lceil \mathrm{NP} \rceil\right) \vdash \lceil \mathrm{NP} \to \mathrm{S} \rceil$$

Proof Traversal

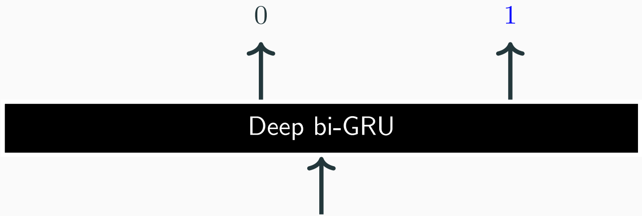
$$\dfrac{\dfrac{}{\text{eat} \vdash \mathrm{NP} \rightarrow \mathrm{NP} \rightarrow \mathrm{S}}\,Ax. \quad \dfrac{}{\text{seeds} \vdash \mathrm{NP}}\,Ax.}{\text{eat}, \text{seeds} \vdash \mathrm{NP} \rightarrow \mathrm{S}}\,\rightarrow E \quad \dfrac{}{\text{ducks} \vdash \mathrm{NP}}\,Ax.}{\text{ducks}, \text{eat}, \text{seeds} \vdash \mathrm{S}}$$



$$\left(\overrightarrow{\text{eat}}; \lceil \mathrm{NP} \rightarrow \mathrm{NP} \rightarrow \mathrm{S} \rceil\right), \quad \left(\overrightarrow{\text{seeds}}; \lceil \mathrm{NP} \rceil\right) \vdash \lceil \mathrm{NP} \rightarrow \mathrm{S} \rceil$$

Proof Traversal

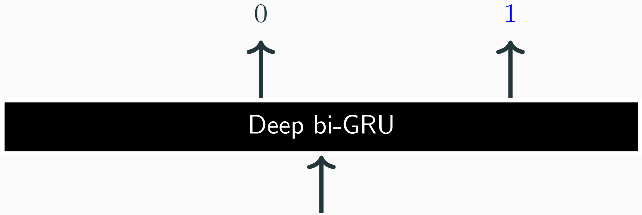
$$\dfrac{\dfrac{\dfrac{}{\text{eat} \vdash \mathrm{NP} \to \mathrm{NP} \to \mathrm{S}}\ Ax. \quad \dfrac{}{\text{seeds} \vdash \mathrm{NP}}\ Ax.}{\text{eat}, \text{seeds} \vdash \mathrm{NP} \to \mathrm{S}}\ \to E \quad \dfrac{}{\text{ducks} \vdash \mathrm{NP}}\ Ax.}{\text{ducks}, \text{eat}, \text{seeds} \vdash \mathrm{S}}$$



$$\left(\overrightarrow{\text{eat}}; \lceil \mathrm{NP} \to \mathrm{NP} \to \mathrm{S} \rceil\right), \quad \left(\overrightarrow{\text{seeds}}; \lceil \mathrm{NP} \rceil\right) \vdash \lceil \mathrm{NP} \to \mathrm{S} \rceil$$

Proof Traversal

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



$$\left(\overrightarrow{\text{eat}}; \lceil \mathrm{NP} \to \mathrm{NP} \to \mathrm{S} \rceil\right), \quad \left(\overrightarrow{\text{seeds}}; \lceil \mathrm{NP} \rceil\right) \vdash \lceil \mathrm{NP} \to \mathrm{S} \rceil$$

Training sample	:	Junction point	Massive Parallelism
Sentence	:	N independent samples	

..but is it working?

Some Concessions

- Up to 2nd order types
- No conjunctions
- Gold types as input

..but is it working?

Some Concessions

Up to 2nd order types

No conjunctions

Gold types as input

Table with Numbers

Input	*Accuracy*
Types & Words & Goal	**97.2**
Types & Words	95.3
Types only	94.2
Words only	87.7

Conclusion & Future

Neural TLG Parsing

- Fast & Efficient
- Accurate
- Formally grounded
- Ideal for semantic tasks

 # todo

- End-to-end integration & evaluation
- Higher-order structures
- Other approaches (.. Shift-Reduce, ProofNets?)

Conclusion & Future

Neural TLG Parsing

- Fast & Efficient
- Accurate
- Formally grounded
- Ideal for semantic tasks

 # todo

- End-to-end integration & evaluation
- Higher-order structures
- Other approaches (.. Shift-Reduce, ProofNets?)
- Thank audience