Discontinuous Constituency and BERT: Case Studies of Dutch

a.k.a. Diamonds are Forever

Gijs Wijnholds (+collaborators)

CLASP Seminar
Göteborgs Universitet
17 February 2023

Project

Context:

2017-2022 (RIP)

"A composition calculus for vector- based semantic modelling with a localization for Dutch" (360-89-070)

Utrecht University

Website: https://compositioncalculus.sites.uu.nl

Team:

Michael Moortgat
PI

Giuseppe Greco
Postdoc

Gijs Wijnholds
Postdoc

Kokos Kogkalidis
PhD

Adriana Correia*
PhD

A bit of background

I want to integrate of the mathematical structures one finds in logic and linguistics, with state-of-the-art machine learning techniques in Natural Language Processing (NLP). The goal is to learn how to characterise natural language structures in a machine-learnable way, grounded in linguistic theory, with explainability at the forefront.

Compositional Distributional Semantics Coecke et al. [2010] Using category theory to unify grammar and meaning

$$\dfrac{\dfrac{\dfrac{\dfrac{\overline{np \longrightarrow np} \quad \overline{s \longrightarrow s}}{np\backslash s \longrightarrow np\backslash s}\,(\backslash) \qquad \overline{np \longrightarrow np}}{(np\backslash s)/np \longrightarrow (np\backslash s)/np}\,(/)}{((np\backslash s)/np) \otimes np \longrightarrow np\backslash s}\,\triangleright^{-1}}{\underset{man}{np} \otimes (\underset{bites}{(np\backslash s)/np} \otimes \underset{dog}{np}) \longrightarrow s}\,\triangleleft^{-1}$$

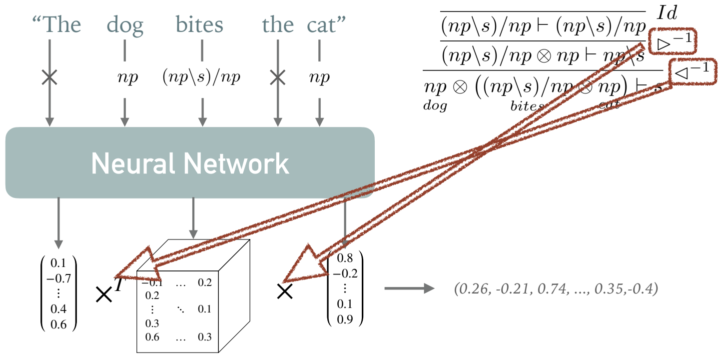
Sentence structure as a proof **Word meaning as a tensor**

Timeline

- 2008-2014: BSc AI, Utrecht University, MSc in Logic, ILLC, University of Amsterdam Proof Theory, Formal Grammar, Category Theory

A bit of background

A shift to the applied



Timeline

- 2016-2019: PhD in CS in the Theory Group, Queen Mary University of London, with prof. dr. Mehrnoosh Sadrzadeh Compositional Distributional Semantics, Machine Learning, Evaluation methods

THE ROBOTS ARE COMING

NL#TIMES

TOP STORIES HEALTH CRIME POLITICS B

ChatGPT on a computer screen - Credit: rokas91 / DepositPhotos - License: DepositPhotos

TECH INNOVATION AI CHATGPT PLAGIARISM EDUCATION » MORE TAGS

SHA

MONDAY, 16 JANUARY 2023 - 09:09

Dutch Students using ChatGPT to finish homework; Teachers aren't noticing

nature

Explore content About the journal Publish with us Subscribe

nature › news › article

NEWS | 18 January 2023

ChatGPT listed as author on research papers: many scientists disapprove

At least four articles credit the AI tool as a co-author, as publishers scramble to regulate its use.

Motivation: the bigger picture

Large-scale language models like GPT-3, BERT and others have attracted much attention in the NLP research community and beyond. But much is unknown about the mechanisms by which these models learn about and understand language, if at all.

Motivation: the bigger picture

Large-scale language models like GPT-3, BERT and others have attracted much attention in the NLP research community and beyond. But much is unknown about the mechanisms by which these models learn about and understand language, if at all.

LMs use *masked language modelling* to learn to complete text:

```
The [MASK] is rising and all life on [MASK] will be [MASK] soon
```

Motivation: the bigger picture

Large-scale language models like GPT-3, BERT and others have attracted much attention in the NLP research community and beyond. But much is unknown about the mechanisms by which these models learn about and understand language, if at all.

LMs use *masked language modelling* to learn to complete text:

```
The [MASK] is rising and all life on [MASK] will be [MASK] soon
```

Stochastic Parrots "Contrary to how it may seem when we observe its output, an LM is a system for haphazardly stitching together sequences of linguistic forms it has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning: a stochastic parrot"

Emily Bender, Timnit Gebru, Angelina McMillan-Major, Shmargaret Shmitchell
On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?

Motivation: the bigger picture

Large-scale language models like GPT-3, BERT and others have attracted much attention in the NLP research community and beyond. But much is unknown about the mechanisms by which these models learn about and understand language, if at all.

LMs use *masked language modelling* to learn to complete text:

```
The [MASK] is rising and all life on [MASK] will be [MASK] soon
```

Stochastic Parrots "Contrary to how it may seem when we observe its output, an LM is a system for haphazardly stitching together sequences of linguistic forms it has observed in its vast training data, according to probabilistic information about how they combine, but without any reference to meaning: a stochastic parrot"

Emily Bender, Timnit Gebru, Angelina McMillan-Major, Shmargaret Shmitchell
On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?

A Central Question to Ask

- Do language models have any linguistic 'understanding'?

Motivation: discontinuities

Probing

- Extracting information from a language model by attaching a small task-specific neural network.
- Has been shown to reveal some syntactic understanding Rogers et al. [2020], Hewitt and Manning [2019]
- A latent bias persists because of focus on English and resources being context free/grammatically simple.

Motivation: discontinuities

Probing

- Extracting information from a language model by attaching a small task-specific neural network.
- Has been shown to reveal some syntactic understanding Rogers et al. [2020], Hewitt and Manning [2019]
- A latent bias persists because of focus on English and resources being context free/grammatically simple.

Discontinuous patterns results about linguistic 'understanding' may not transfer between languages:

... dat	Jan	Marie	de kinderen	ziet	leren	fietsen
... that	*Jan*	*Marie*	*the children*	*see*	*teach*	*cycle*

'...that John sees Mary teach the kids to cycle'

- Can a language model draw the links?

Understanding verb clusters

A Case Study (or two): Dutch verb clusters

Verb clusters arise in Dutch embedded clauses, when verb raisers are stacked, passing their subject/object to the embedded infinitive.

Raising example 1 *He will* $\left[\textit{say something to her}\right]$

$$\underset{su}{\text{Hij}} \quad \underset{vr}{\text{zal}} \quad \left[\underset{iobj}{\text{haar}} \quad \underset{dobj}{\text{iets}} \quad \underset{inf}{\text{zeggen}}\right]_{vc}$$

A Case Study (or two): Dutch verb clusters

Verb clusters arise in Dutch embedded clauses, when verb raisers are stacked, passing their subject/object to the embedded infinitive.

Raising example 1 *He will* $\left[\textit{say something to her}\right]$

$$\underset{su}{\text{Hij}} \quad \underset{vr}{\text{zal}} \quad \left[\underset{iobj}{\text{haar}} \quad \underset{dobj}{\text{iets}} \quad \underset{inf}{\text{zeggen}}\right]_{vc}$$

Raising example 2 *He will* $\left[\textit{want}\ \left[\textit{to say something to her}\right]\right]$

$$\underset{su}{\text{Hij}} \quad \underset{vr}{\text{zal}} \quad \left[\left[\underset{iobj}{\text{haar}} \quad \underset{dobj}{\text{iets}}\right] \quad \underset{vr}{\text{willen}} \quad \left[\underset{inf}{\text{zeggen}}\right]\right]_{vc}$$

- zullen, willen: obligatory verb raiser

A Case Study (or two): subject flipping

Raising example 2 *He will* $\Big[$*want* $\big[$*to say something to her*$\big]\Big]$

$$\underset{su}{\text{Hij}}\quad \underset{vr}{\text{zal}}\quad \Big[\big[\underset{iobj}{\text{haar}}\quad \underset{dobj}{\text{iets}}\big]\quad \underset{vr}{\text{willen}}\quad \big[\underset{inf}{\text{zeggen}}\big]\Big]_{vc}$$

Raising example 3 *He will* $\Big[$*want* $\big[$*to let her [say something]*$\big]\Big]$

$$\underset{su}{\text{Hij}}\quad \underset{vr}{\text{zal}}\quad \Big[\big[\underset{dobj}{\text{haar}}\big]\quad \big[\underset{dobj}{\text{iets}}\big]\quad \underset{vr}{\text{willen}}\quad \big[\underset{vr}{\text{laten}}\big]\quad \big[\underset{inf}{\text{zeggen}}\big]\Big]_{vc}$$

- zullen, willen: obligatory verb raiser
- laten: obligatory verb raiser, subject flipper

A Case Study (or two): raising versus extraposition

Extraposition vs raising *(I see that) he tries* $\left[\textit{to defeat her}\right]$

Extraposition (Ik zie dat) $\underset{su}{\text{hij}}$ $\underset{ext}{\text{probeert}}$ $[\underset{dobj}{\text{haar}}$ $\underset{inf}{\text{te verslaan}}]_{vc}$

Raising (Ik zie dat) $\underset{su}{\text{hij}}$ $\left[\underset{dobj}{\text{haar}}\right]_{vc}$ $\underset{vr}{\text{probeert}}$ $\left[\underset{inf}{\text{te verslaan}}\right]_{vc}$

A Case Study (or two): raising versus extraposition

Extraposition vs raising *(I see that) he tries* $\left[\textit{to defeat her}\right]$

Extraposition (Ik zie dat) $\underset{su}{\text{hij}}$ $\underset{ext}{\text{probeert}}$ $[\underset{dobj}{\text{haar}}$ $\underset{inf}{\text{te verslaan}}]_{vc}$

Raising (Ik zie dat) $\underset{su}{\text{hij}}$ $\left[\underset{dobj}{\text{haar}}\right]_{vc}$ $\underset{vr}{\text{probeert}}$ $\left[\underset{inf}{\text{te verslaan}}\right]_{vc}$

Control verbs *(I see that) he asks/promises us to defeat her*

(Ik zie dat) $\underset{su}{\text{hij}}$ $\underset{iobj}{\text{ons}}$ $\underset{ctrl}{\substack{\text{vraagt}\\ \text{belooft}}}$ $\left[\underset{dobj}{\text{haar}} \quad \underset{inf}{\text{te verslaan}}\right]_{vc}$

A Case Study (or two): classifying verbal categories

Raising, Extraposition, Infinitives

Description	Examples
intransitive infinitive	vertrekken, stemmen, verliezen, ...
transitive infinitive with inanimate object	zeggen, begrijpen, merken, ...
transitive infinitive, animate object	ontmoeten, bedanken, kennen, ...
obligatory verb raiser	willen, zullen, moeten, ...
obligatory verb raiser, subject flipper	laten, doen
non-obligatory verb raiser	proberen, weigeren, trachten, ...
extraposition	proberen, weigeren, trachten, ...
extraposition, object control	verzoeken, dwingen, verplichten, ...
extraposition, subject control	beloven, verzekeren, zweren, ...

Sources

- Verbs sampled from Algemene Nederlandse Spraakkunst (ans.ruhosting.nl)

Probing pt. 1

Probing Discontinuity

Goal Setting up a general probing model that recognizes verb-subject dependencies, to evaluate whether Dutch language models contain lexical knowledge about control verbs, and whether they are invariant under word order permutations in the case of verb raising.

The setup

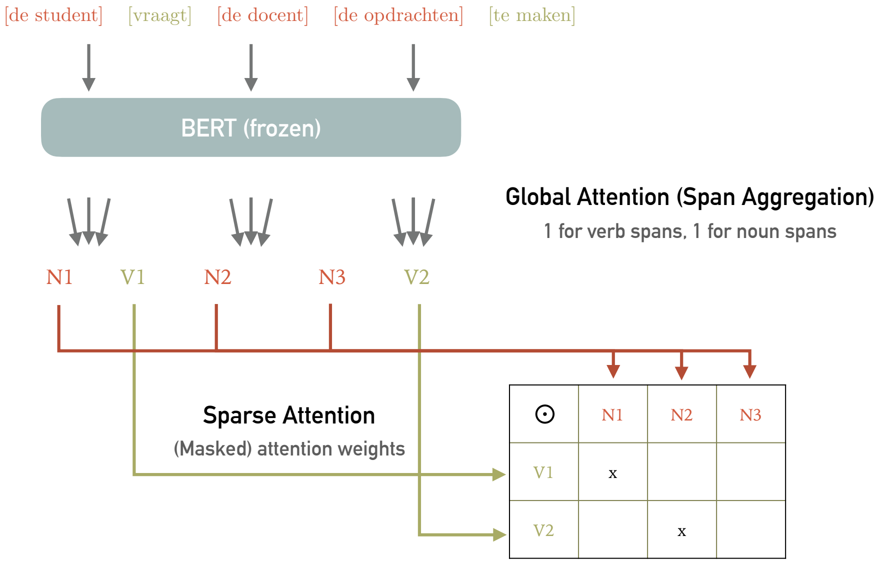
1. Design a probing model that can recognise verb-subject dependencies,
2. Gather appropriate training data,
3. Generate test data in a controlled/naturalistic way and test.

References

- Konstantinos Kogkalidis and Gijs Wijnholds. Discontinuous Constituency and BERT: A Case Study of Dutch. *Findings of ACL 2022*.
- DYI: https://github.com/gijswijnholds/discontinuous-probing

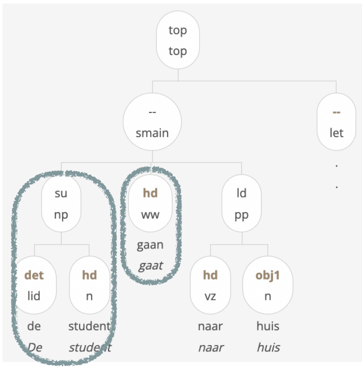
Probe design

The student asks the teacher to do the exercises

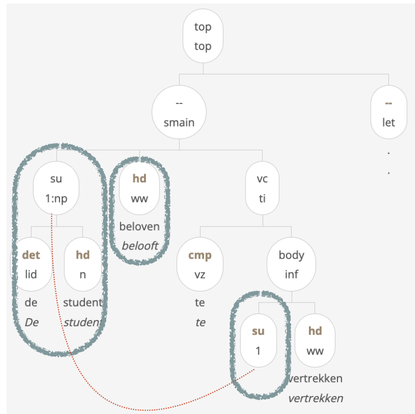


Training the probe

Lassy-Small a gold standard dataset of written Dutch with ca. 65k sentences, both continuous and discontinuous verb-subject dependencies Van Noord et al. [2013]



The student goes home



The student promises to leave

Models BERTje de Vries et al. [2019], RobBERT Delobelle et al. [2020]

Probing attempt #1

Modelling discontinuities We use a *mildly context sensitive* grammar formalism, Multiple Context Free Grammar, to generate test samples.

Syntax vs. lexicon One grammar for verb raising constructions, a separate one for control verbs:

(a)	de docent ziet [de student] [de collega] [de professor] de oefeningen [helpen] [leren] [maken]
(EN)	*the teacher sees [the student] [help] [the colleague] [teach] [the professor] [to do] the exercises*

Validation vs test results While the prober performs very well, the test sets are challenging:

Model	Lassy	Control	Raising
BERTje	97.6	48	43.1
RobBERT	92.5	40.6	29.2

A downside because the grammar is rule-based, we need to write complex specifications of how subjects are inherited by verbal complements.

Learn you a categorial grammar for great good!

A user-friendly format: Natural Deduction

Structures, sequents Judgements $\Gamma \vdash A$ with A a formula, Γ a structure:

$$\Gamma, \Delta \quad ::= \quad A \mid \Gamma \cdot \Delta$$

Axiom, logical rules For the base logic, we have the *axiom* $A \vdash A$ and as logical inference rules, for each connective an *elimination* rule and an *introduction* rule, e.g.

$$\frac{\Gamma \vdash A \quad \Delta \vdash A\backslash B}{\Gamma \cdot \Delta \vdash B}\ \backslash E \qquad \frac{A \cdot \Gamma \vdash B}{\Gamma \vdash A\backslash B}\ \backslash I$$

A user-friendly format: Natural Deduction

Structures, sequents Judgements $\Gamma \vdash A$ with A a formula, Γ a structure:

$$\Gamma, \Delta \quad ::= \quad A \mid \Gamma \cdot \Delta$$

Axiom, logical rules For the base logic, we have the *axiom* $A \vdash A$ and as logical inference rules, for each connective an *elimination* rule and an *introduction* rule, e.g.

$$\frac{\Gamma \vdash A \quad \Delta \vdash A\backslash B}{\Gamma \cdot \Delta \vdash B}\ \backslash E \qquad \frac{A \cdot \Gamma \vdash B}{\Gamma \vdash A\backslash B}\ \backslash I$$

Example in steno format

$$\dfrac{\dfrac{\dfrac{\text{the}}{np/n} \quad \dfrac{\text{temperature}}{n}}{\text{the} \cdot \text{temperature} \vdash np}\ /E \quad \dfrac{\text{rises}}{np\backslash s}}{(\text{the} \cdot \text{temperature}) \cdot \text{rises} \vdash s}\ \backslash E$$

Notation: $\Gamma[\Delta]$ for a structure Γ containing a substructure Δ

Control operators

The need for control languages exhibit phenomena that seem to require a form of

reordering, restructuring, copying

The logical answer Structures $\Gamma, \Delta ::= A \mid \langle\Gamma\rangle \mid \Gamma \cdot \Delta$

$$\frac{\langle\Gamma\rangle \vdash A}{\Gamma \vdash \Box A}\ \Box I \qquad\qquad \frac{\Gamma \vdash \Box A}{\langle\Gamma\rangle \vdash A}\ \Box E$$

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

Control operators

The need for control languages exhibit phenomena that seem to require a form of

reordering, restructuring, copying

The logical answer Structures $\Gamma, \Delta ::= A \mid \langle\Gamma\rangle \mid \Gamma \cdot \Delta$

$$\frac{\langle\Gamma\rangle \vdash A}{\Gamma \vdash \Box A}\ \Box I \qquad\qquad \frac{\Gamma \vdash \Box A}{\langle\Gamma\rangle \vdash A}\ \Box E$$

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

Structure global rules $\rightsquigarrow$ $\Diamond$ controlled restricted versions, e.g.

$$A^\diamond : \quad (A \bullet B) \bullet \Diamond C \longrightarrow A \bullet (B \bullet \Diamond C)$$
$$C^\diamond : \quad (A \bullet B) \bullet \Diamond C \longrightarrow (A \bullet \Diamond C) \bullet B$$

Control operators

The need for control languages exhibit phenomena that seem to require a form of

reordering, restructuring, copying

The logical answer Structures $\Gamma, \Delta ::= A \mid \langle\Gamma\rangle \mid \Gamma \cdot \Delta$

$$\frac{\langle\Gamma\rangle \vdash A}{\Gamma \vdash \Box A}\ \Box I \qquad\qquad \frac{\Gamma \vdash \Box A}{\langle\Gamma\rangle \vdash A}\ \Box E$$

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

Structure global rules $\rightsquigarrow$ $\Diamond$ controlled restricted versions, e.g.

$$\begin{array}{ll} A^\diamond: & (A \bullet B) \bullet \Diamond C \longrightarrow A \bullet (B \bullet \Diamond C) \\ C^\diamond: & (A \bullet B) \bullet \Diamond C \longrightarrow (A \bullet \Diamond C) \bullet B \end{array}$$

Multimodal generalization families $\{\Diamond_i, \Box_i\}_{i \in I}$ for particular structural choices

Encoding dependency structure

Heads vs dependents

Dependency roles articulate the linguistic material on the basis of two oppositions:

- head - complement relations
  - verbal domain: subj, (in)direct object, ...
  - nominal domain: prepositional object, ...
- adjunct - head relations
  - verbal domain: (time, manner, ...) adverbial
  - nominal domain: adjectival, numeral, determiner, ...

Compare: fa-structure: function vs argument

Heads vs dependents

Dependency roles articulate the linguistic material on the basis of two oppositions:

- head - complement relations
  - verbal domain: subj, (in)direct object, ...
  - nominal domain: prepositional object, ...
- adjunct - head relations
  - verbal domain: (time, manner, ...) adverbial
  - nominal domain: adjectival, numeral, determiner, ...

Compare: fa-structure: function vs argument

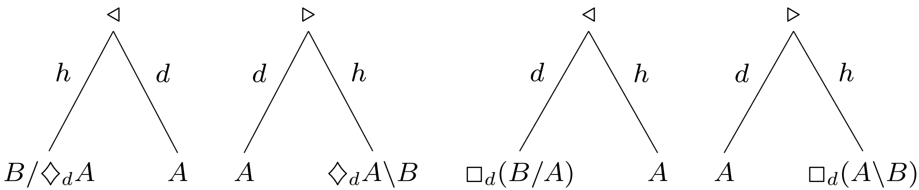
Orthogonality The fa and the dependency articulation are in general not aligned. This asks for a multidimensional type logic.

E.g. Determiner. Semantically, characteristic function of $([\![N]\!], [\![VP]\!])$ relation; morphologically, dependent on head noun.

Defining a headed product

Multimodal generalization families $\{\Diamond_d, \Box_d\}_{d \in DepLabel}$

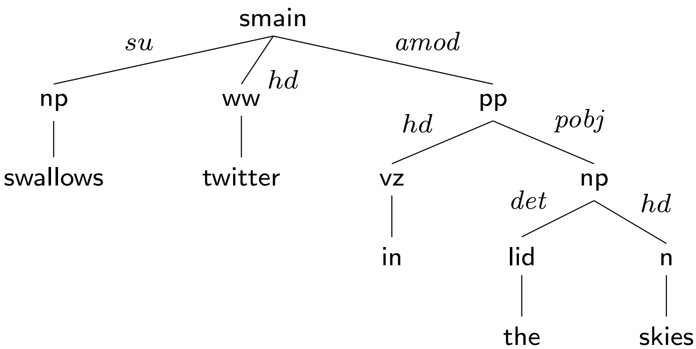
- $\Diamond_d A \backslash C$, $C / \Diamond_d B$ head functor assigning dependency role d to its complement
- $\Box_d (A \backslash C)$, $\Box_d (C / B)$ dependent functor projecting adjunct role d



Example Determiner: $\Box det(np/n)$, after projecting its determiner dependency role it can act as a function of its argument noun.

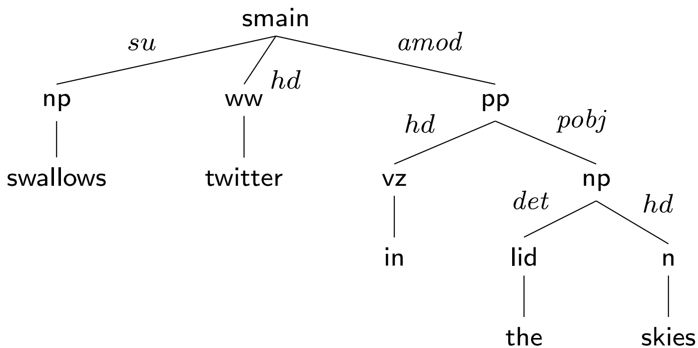
Extracting types from structured data

Dutch treebank LASSY Annotation DAGs, nodes: synt categories, edges: dependency relations. Re-entrancy: higher-order types.



Extracting types from structured data

Dutch treebank LASSY Annotation DAGs, nodes: synt categories, edges: dependency relations. Re-entrancy: higher-order types.



Extracted types:

swallows : np twitter : $\Diamond_{su} np \backslash s$ in : $\Box_{amod}(s \backslash s)/\Diamond_{pobj} np$ the : $\Box_{det}(np/n)$ skies : n

Dependency structure

Derivation, N.D. style:

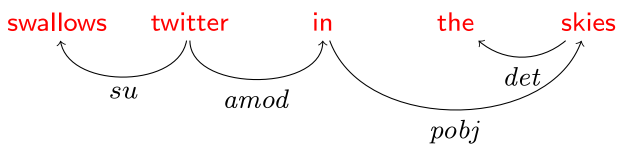
$$
\dfrac{
\dfrac{
\dfrac{\dfrac{\mathsf{swallows}}{np}}{\langle\mathsf{swallows}\rangle^{su} \vdash \Diamond_{su} np}\,\Diamond I
\qquad
\dfrac{\mathsf{twitter}}{\Diamond_{su} np\backslash s}
}{\langle\mathsf{swallows}\rangle^{su}\cdot\mathsf{twitter}\vdash s}\,\backslash E
\qquad
\dfrac{
\dfrac{
\dfrac{\mathsf{in}}{\Box_{amod}(s\backslash s)/\Diamond_{pobj}np}
\qquad
\dfrac{
\dfrac{
\dfrac{\dfrac{\mathsf{the}}{\Box_{det}(np/n)}}{\langle\mathsf{the}\rangle^{det}\vdash np/n}\,\Box E
\qquad
\dfrac{\mathsf{skies}}{n}
}{\langle\mathsf{the}\rangle^{det}\cdot\mathsf{skies}\vdash np}\,/E
}{\langle\langle\mathsf{the}\rangle^{det}\cdot\mathsf{skies}\rangle^{pobj}\vdash\Diamond_{pobj}np}\,\Diamond I
}{\mathsf{in}\cdot\langle\langle\mathsf{the}\rangle^{det}\cdot\mathsf{skies}\rangle^{pobj}\vdash\Box_{amod}(s\backslash s)}\,/E
}{\langle\mathsf{in}\cdot\langle\langle\mathsf{the}\rangle^{det}\cdot\mathsf{skies}\rangle^{pobj}\rangle^{amod}\vdash s\backslash s}\,\Box E
}{(\langle\mathsf{swallows}\rangle^{su}\cdot\mathsf{twitter})\cdot\langle\mathsf{in}\cdot\langle\langle\mathsf{the}\rangle^{det}\cdot\mathsf{skies}\rangle^{pobj}\rangle^{amod}\vdash s}\,\backslash E
$$

Dependency structure

Derivation, N.D. style:

$$
\begin{prooftree}
\AxiomC{swallows}
\UnaryInfC{np}
\RightLabel{$\Diamond I$}
\UnaryInfC{$\langle \text{swallows} \rangle^{su} \vdash \Diamond_{su} np$}
\AxiomC{twitter}
\UnaryInfC{$\Diamond_{su} np \backslash s$}
\RightLabel{$\backslash E$}
\BinaryInfC{$\langle \text{swallows} \rangle^{su} \cdot \text{twitter} \vdash s$}
\AxiomC{in}
\UnaryInfC{$\Box_{amod}(s \backslash s) / \Diamond_{pobj} np$}
\AxiomC{the}
\UnaryInfC{$\Box_{det}(np/n)$}
\RightLabel{$\Box E$}
\UnaryInfC{$\langle \text{the} \rangle^{det} \vdash np/n$}
\AxiomC{skies}
\UnaryInfC{n}
\RightLabel{$/E$}
\BinaryInfC{$\langle \text{the} \rangle^{det} \cdot \text{skies} \vdash np$}
\RightLabel{$\Diamond I$}
\UnaryInfC{$\langle \langle \text{the} \rangle^{det} \cdot \text{skies} \rangle^{pobj} \vdash \Diamond_{pobj} np$}
\RightLabel{$/E$}
\BinaryInfC{$\text{in} \cdot \langle \langle \text{the} \rangle^{det} \cdot \text{skies} \rangle^{pobj} \vdash \Box_{amod}(s \backslash s)$}
\RightLabel{$\Box E$}
\UnaryInfC{$\langle \text{in} \cdot \langle \langle \text{the} \rangle^{det} \cdot \text{skies} \rangle^{pobj} \rangle^{amod} \vdash s \backslash s$}
\RightLabel{$\backslash E$}
\BinaryInfC{$(\langle \text{swallows} \rangle^{su} \cdot \text{twitter}) \cdot \langle \text{in} \cdot \langle \langle \text{the} \rangle^{det} \cdot \text{skies} \rangle^{pobj} \rangle^{amod} \vdash s$}
\end{prooftree}
$$

Induced dependency structure:



$\leadsto$ within dependency domain, outgoing arcs from head to (head of) dependents

Benefitting from a multidimensional setup

Kokos Kogkalidis worked on resources and neural tools for parsing Dutch in the multimodal setup:

- Kogkalidis et al 2020a, Æthel: Automatically extracted typelogical derivations for Dutch. LREC.
- Kogkalidis et al 2020b, Neural proof nets. CoNLL
- Kogkalidis et al 2022, Geometry-Aware Supertagging with Heterogeneous Dynamic Convolutions, arXiv

If you want to try things out, see the readme on

https://github.com/konstantinosKokos/lassy-tlg-extraction

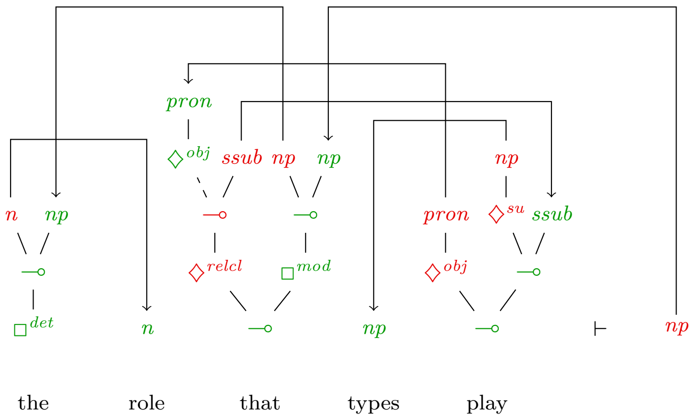
for the extracted proofbank

https://github.com/konstantinosKokos/dynamic-proof-nets

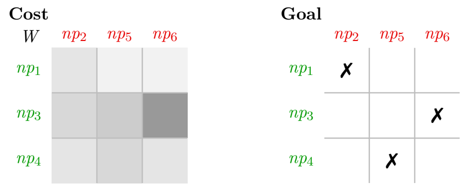
for the parser

Parser explained in one slide

Proof net



Sinkhorn



$$\overset{\eta}{\rightsquigarrow}\ \blacktriangledown^{mod}(\text{that}\ \ \triangle^{relcl}\,\lambda x.(\text{play}\ x\ \ \triangle^{su}\,\text{types}))\ \ (\blacktriangledown^{det}\text{the role})$$

Probing pt. 2

Handling verb clusters: the ACG approach

- the Abstract Categorial Grammar method

  abstract syntax, divergent compositional translations:

  $\lceil\cdot\rceil^{string}$ string semantics

  $\lceil\cdot\rceil^{sem}$ meaning assembly

The ACG method is easily adapted to our **NL** source: words as abstract constants.

Simple combinatorics, inflated type homomorphism String semantics: higher-order modelling of tuples

$$\lceil INFP \rceil = (\sigma \multimap \sigma \multimap \sigma) \multimap \sigma \qquad \triangleq \sigma^{(2)}$$

References

- Michael Moortgat, Konstantinos Kogkalidis and Gijs Wijnholds. Diamonds are Forever: Theoretical and Empirical Support for a Dependency-Enhanced Type Logic. To appear in: *Logic and Algorithms in Computational Linguistics 2021*.
- DYI: https://github.com/gijswijnholds/malin_2022

ACG method (cont'd)

Abstract syntax The syntax types don't yield the surface string, but the closest you can get using logical rules only.

$$
\dfrac{\dfrac{\textsf{haar}}{NP} \qquad \dfrac{\dfrac{\dfrac{\textsf{iets}}{NP} \quad \dfrac{\textsf{zeggen}}{NP\backslash INFP}}{\textsf{iets}\cdot\textsf{zeggen}\vdash INFP}\,\backslash E \qquad \dfrac{\textsf{laten}}{INFP\backslash(NP\backslash INFP)}}{(\textsf{iets}\cdot\textsf{zeggen})\cdot\textsf{laten}\vdash NP\backslash INFP}\,\backslash E}{\textsf{haar}\cdot((\textsf{iets}\cdot\textsf{zeggen})\cdot\textsf{laten})\vdash INFP}\,\backslash E \qquad \dfrac{\textsf{willen}}{INFP\backslash INFP}
$$

$$
\dfrac{\textsf{haar}\cdot((\textsf{iets}\cdot\textsf{zeggen})\cdot\textsf{laten})\vdash INFP \qquad INFP\backslash INFP}{\dagger \qquad (\textsf{haar}\cdot((\textsf{iets}\cdot\textsf{zeggen})\cdot\textsf{laten}))\cdot\textsf{willen}\vdash INFP}\,\backslash E
$$

ACG method (cont'd)

Abstract syntax The syntax types don't yield the surface string, but the closest you can get using logical rules only.

$$
\dfrac{\dfrac{\text{haar}}{NP} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\text{iets}}{NP} \qquad \dfrac{\text{zeggen}}{NP\backslash INFP}}{\text{iets}\cdot\text{zeggen} \vdash INFP}\backslash E \qquad \dfrac{\text{laten}}{INFP\backslash(NP\backslash INFP)}}{(\text{iets}\cdot\text{zeggen})\cdot\text{laten} \vdash NP\backslash INFP}\backslash E}{}}{\dfrac{\text{haar}\cdot((\text{iets}\cdot\text{zeggen})\cdot\text{laten}) \vdash INFP \qquad \dfrac{\text{willen}}{INFP\backslash INFP}}{\dagger \qquad (\text{haar}\cdot((\text{iets}\cdot\text{zeggen})\cdot\text{laten}))\cdot\text{willen} \vdash INFP}\backslash E}\backslash E
$$

$$
\begin{array}{rcllcl}
\lceil \text{zeggen} \rceil^{string} & = & \lambda x \lambda f.(f\ x\ \texttt{zeggen}) & & :: & \sigma \multimap \sigma^{(2)} \\
\lceil \text{willen} \rceil^{string} & = & \lambda q \lambda f.(q\ \lambda y \lambda z.(f\ y\ \texttt{willen}\cdot z)) & & :: & \sigma^{(2)} \multimap \sigma^{(2)} \\
\lceil \text{laten} \rceil^{string} & = & \lambda q \lambda x \lambda f.(q\ \lambda z \lambda w.(f\ x\cdot z\ \texttt{laten}\cdot w)) & & :: & \sigma^{(2)} \multimap \sigma \multimap \sigma^{(2)}
\end{array}
$$

$$
\begin{array}{lrcl}
 & \lceil \dagger \rceil^{string} & = & \lambda f.(f\ \texttt{haar}\cdot\texttt{iets}\ \ \texttt{willen}\cdot\texttt{laten}\cdot\texttt{zeggen}) \\
\text{compare} & \lceil \dagger \rceil^{sem} & = & \textsc{want}\ (\textsc{let}\ (\textsc{say}\ \textsc{something})\ \textsc{her})
\end{array}
$$

Dependency enhancement

function types $A\backslash B \leadsto \Diamond_d A\backslash B$

vc: verbal complement

$$
\begin{prooftree}
\AxiomC{haar}
\UnaryInfC{np}
\RightLabel{$\Diamond I$}
\UnaryInfC{$\langle \textsf{haar} \rangle^{obj} \vdash \Diamond_{obj} np$}
\AxiomC{iets}
\UnaryInfC{np}
\RightLabel{$\Diamond I$}
\UnaryInfC{$\langle \textsf{iets} \rangle^{obj} \vdash \Diamond_{obj} np$}
\AxiomC{zeggen}
\UnaryInfC{$\Diamond_{obj} np\backslash inf$}
\RightLabel{$\backslash E$}
\BinaryInfC{$\langle \textsf{iets} \rangle^{obj} \cdot \textsf{zeggen} \vdash inf$}
\RightLabel{$\Diamond I$}
\UnaryInfC{$\langle \langle \textsf{iets} \rangle^{obj} \cdot \textsf{zeggen} \rangle^{vc} \vdash \Diamond_{vc} inf$}
\AxiomC{laten}
\UnaryInfC{$\Diamond_{vc} inf\backslash(\Diamond_{obj} np\backslash inf)$}
\RightLabel{$\backslash E$}
\BinaryInfC{$\langle \langle \textsf{iets} \rangle^{obj} \cdot \textsf{zeggen} \rangle^{vc} \cdot \textsf{laten} \vdash \Diamond_{obj} np\backslash inf$}
\RightLabel{$\backslash E$}
\BinaryInfC{$\langle \textsf{haar} \rangle^{obj} \cdot (\langle \langle \textsf{iets} \rangle^{obj} \cdot \textsf{zeggen} \rangle^{vc} \cdot \textsf{laten}) \vdash inf$}
\RightLabel{$\Diamond I$}
\UnaryInfC{$\langle \langle \textsf{haar} \rangle^{obj} \cdot (\langle \langle \textsf{iets} \rangle^{obj} \cdot \textsf{zeggen} \rangle^{vc} \cdot \textsf{laten}) \rangle^{vc} \vdash \Diamond_{vc} inf$}
\AxiomC{willen}
\UnaryInfC{$\Diamond_{vc} inf\backslash inf$}
\RightLabel{$\backslash E$}
\BinaryInfC{$\langle \langle \textsf{haar} \rangle^{obj} \cdot (\langle \langle \textsf{iets} \rangle^{obj} \cdot \textsf{zeggen} \rangle^{vc} \cdot \textsf{laten}) \rangle^{vc} \cdot \textsf{willen} \vdash inf$}
\end{prooftree}
$$

$$(\textsc{want}\ \ \triangle^{vc}((\textsc{let}\ \ \triangle^{vc}(\textsc{say}\ \ \triangle^{obj}\ \textsc{something}))\ \ \triangle^{obj}\ \textsc{her}))$$

Diamonds are forever

$$\frac{\textsf{hij}}{np} \quad \frac{\textsf{haar}}{np} \quad \frac{\textsf{zal}}{(\Diamond_{su} np \backslash s)/\Diamond_{vc} inf} \quad \frac{\textsf{iets}}{np} \quad \frac{\textsf{zeggen}}{\Diamond_{obj} np \backslash inf} \quad \frac{\textsf{laten}}{\Diamond_{vc} inf \backslash (\Diamond_{obj} np \backslash inf)} \quad \frac{\textsf{willen}}{\Diamond_{vc} inf \backslash inf}$$

Derivation

$$\overline{\langle \textsf{hij} \rangle^{su} \cdot (\textsf{zal} \cdot \langle \langle \langle \textsf{haar} \rangle^{obj} \cdot (\langle \langle \textsf{iets} \rangle^{obj} \cdot \textsf{zeggen} \rangle^{vc} \cdot \textsf{laten}) \rangle^{vc} \cdot \textsf{willen} \rangle^{vc}) \vdash s}$$

273 abstract samples Each word is a unique instance of a word category, used to generate many more samples

AST	$r_0\ (g_0\ (g_1\ d_4\ (f_1\ d_5\ d_1)\ d_3)\ d_2)$
Surface	`hij zal haar iets willen laten zeggen`
Semantics	$(\texttt{zal}\ (\Diamond_{vc}(\texttt{willen}\ \Diamond_{vc}((\texttt{laten}\ \Diamond_{vc}(\texttt{zeggen}\ (\Diamond_{obj1}\texttt{iets})))\ (\Diamond_{obj1}\texttt{haar})))))\ (\Diamond_{su}\texttt{hij})$
Pairing	$[(\texttt{zal}, \texttt{hij}), (\texttt{willen}, \texttt{hij}), (\texttt{laten}, \texttt{hij}), (\texttt{zeggen}, \texttt{haar})]$

Populating the lexicon

The lexicon

Category	Description	Examples
INF0	intransitive infinitive	vertrekken, stemmen, verliezen, …
INF1	transitive infinitive with inanimate object	zeggen, begrijpen, merken, …
INF1A	transitive infinitive, animate object	ontmoeten, bedanken, kennen, …
IVR0	obligatory verb raiser	willen, zullen, moeten, …
IVR1	obligatory verb raiser, subject flipper	laten, doen
IVR2	non-obligatory verb raiser	proberen, weigeren, trachten, …
INF2	extraposition	proberen, weigeren, trachten, …
INF3	extraposition, object control	verzoeken, dwingen, verplichten, …
INF4	extraposition, subject control	beloven, verzekeren, zweren, …
OBJ1A	animate direct object	Karin, Wouter, …
OBJ1I	inanimate direct object	iets, veel, een ding, …
OBJ2	indirect object	Karin, Wouter, …

Sources

- Verbs sampled from Algemene Nederlandse Spraakkunst (ans.ruhosting.nl)
- Names samples from the Nederlandse Voornamenbank (www.meertens.knaw.nl/nvb)

Results (1/3)

Validation vs test results The probe again does not perform well on the generated data:

	Validation set (Lassy)	**Test set (generated)**
Accuracy	97.60	79.47
Random Baseline	13.24	39.24

Results (1/3)

Validation vs test results The probe again does not perform well on the generated data:

	Validation set (Lassy)	Test set (generated)
Accuracy	97.60	79.47
Random Baseline	13.24	39.24

By number of nouns Just a check that the results follow the expected pattern:

Number of nouns	2	3	4
Accuracy	86.87	75.66	68.76
Random Baseline	50.00	33.33	25.00

Results (1/3)

Validation vs test results The probe again does not perform well on the generated data:

	Validation set (Lassy)	Test set (generated)
Accuracy	97.60	79.47
Random Baseline	13.24	39.24

By number of nouns Just a check that the results follow the expected pattern:

Number of nouns	2	3	4
Accuracy	86.87	75.66	68.76
Random Baseline	50.00	33.33	25.00

By verbal type Extraposition easier than raisers, as there is no cluster. Infinitives even worse, most likely because they appear at long distance.

Verbal type	Raising	Extraposition	Infinitive
Accuracy	81.00	87.03	68.77
Random	39.86	38.27	39.24

Results (2/3)

Dominance

		$\mathtt{OBJ}_2$	$\mathtt{OBJ}_1^I$	$\mathtt{IVR}_0$	$\mathtt{IVR}_1$	$\mathtt{INF}_1$
hij	zal	haar	iets	willen	laten	zeggen
he	*will*	*her*	*something*	*want*	*let*	*say*

		$\mathtt{INF}_2$	$\mathtt{OBJ}_2$	$\mathtt{OBJ}_1^I$	TE	$\mathtt{IVR}_1$	$\mathtt{INF}_1$
hij	zal	proberen	haar	iets	te	laten	zeggen
he	*will*	*try*	*her*	*something*	*to*	*let*	*say*

Governed verbs

		Dominating verb, by subcategory		
Dominated by raising	**Overall**	**IVR0**	**IVR1**	**IVR2**
Accuracy	76.18	78.54	71.41	77.95
Random Baseline	39.86	41.06	37.09	41.05
Dominated by extraposition	**Overall**	**INF2**	**INF3**	**INF4**
Accuracy	66.70	86.74	57.12	47.12
Random Baseline	38.27	42.58	35.13	35.13

- Accuracy declines for verbs governed by a subject flipping verb raiser (IVR1)
- Under an extraposition verb, control verbs (INF3/INF4) are the challenging ones.

Results (3/3)

Semantic equivalence comparing samples with a different AST and surface realization, but identical semantics:

a. *hij zal haar proberen*[IVR2] *te willen ontmoeten*
he will her try to want meet

b. *hij zal proberen*[INF2] *haar te willen ontmoeten*
he will try her to want meet
'he will try to want to meet her'

Results Extraposition easier construction to handle, minor difference on the surrounding context.

	Context in the sentence		
Raising construction	**Above**	**Verb**	**Below**
Accuracy	95.09	86.22	78.15
Random Baseline	42.54	41.47	41.44
Extraposition construction			
Accuracy	96.49	93.04	78.50
Random Baseline	42.54	41.48	41.44

Summary, Discussion

- Dutch BERT does not seem to inherently capture verb-subject dependencies very well (in verb clusters),
- Specific verb categories introduce their own complexity to the model (extraposition vs raising vs control),

Summary, Discussion

- Dutch BERT does not seem to inherently capture verb-subject dependencies very well (in verb clusters),
- Specific verb categories introduce their own complexity to the model (extraposition vs raising vs control),

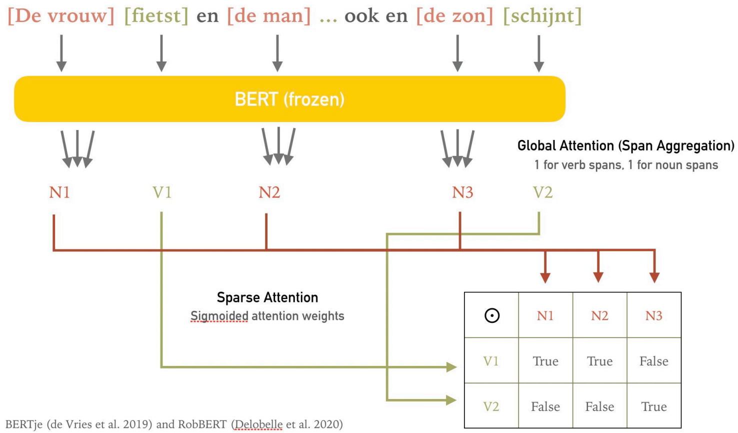
What's next

- Going multilingual, using the same abstract syntax to generate surface forms in several languages, e.g.

Ik weet dat	Jan	Marie	de kinderen	ziet	leren	fietsen
Ich weiß dass	Jan	Marie	die Kinder	fahren	lernen	sieht
I know that	John	sees	Mary	teach	the children	to cycle

  - Challenge: finding the appropriate (aligned) training data

Probing ellipsis



Reference

- Tessel Haagen, Loïs Dona, Sarah Bosscha, Beatriz Zamith, Richard Koetschruyter, Gijs Wijnholds. Noun Phrase and Verb Phrase Ellipsis in Dutch: Identifying Subject-Verb Dependencies with BERTje. *Computational Linguistics in the Netherlands 2022*.

Back to the bigger picture

Take away

- Large-scale Language Models are a staple in the NLP toolkit, despite limited knowledge about their ‘intelligence’.
- Using techniques from formal grammar, we can show that they have limited capabilities for linguistic ‘understanding’.

Back to the bigger picture

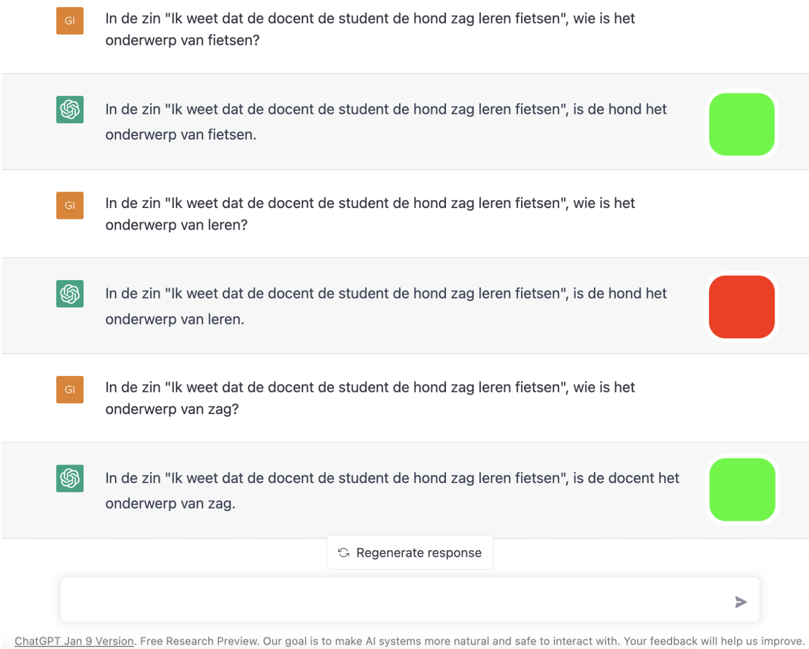
Take away

- Large-scale Language Models are a staple in the NLP toolkit, despite limited knowledge about their ‘intelligence’.
- Using techniques from formal grammar, we can show that they have limited capabilities for linguistic ‘understanding’.

Going further

- *Compositional generalization*: how can LMs generalize, e.g. on objectives like negation, monotonicity? Yanaka et al. [2019], Wijnholds [2023]
- *Data/scale*: what can we do about data bias, pretraining objectives, to create more efficient models?
- *Hybrid models*: in what way can we harness the power of LMs and combine them formal representations?

The robots are not coming (yet)



Thank you!

References to our work

- Konstantinos Kogkalidis and Gijs Wijnholds. Discontinuous Constituency and BERT: A Case Study of Dutch. *Findings of ACL 2022*.
- Michael Moortgat, Konstantinos Kogkalidis and Gijs Wijnholds. Diamonds are Forever: Theoretical and Empirical Support for a Dependency-Enhanced Type Logic. To appear in: *Logic and Algorithms in Computational Linguistics 2021*.
- Tessel Haagen, Loïs Dona, Sarah Bosscha, Beatriz Zamith, Richard Koetschruyter, Gijs Wijnholds. Noun Phrase and Verb Phrase Ellipsis in Dutch: Identifying Subject-Verb Dependencies with BERTje. *Computational Linguistics in the Netherlands 2022*.

Github references

`https://github.com/gijswijnholds/discontinuous-probing`

for the MCFG generation and prober evaluation

`https://github.com/gijswijnholds/malin_2022`

for the ACG style generation and prober evaluation

References

B Coecke, M Sadrzadeh, and S Clark. Mathematical foundations for a compositional distributed model of meaning. *Lambek Festschrift, Linguistic Analysis*, 36, 2010.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. BERTje: A Dutch BERT model. *arXiv preprint arXiv:1912.09582*, 2019.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. RobBERT: a Dutch RoBERTa-based language model. *arXiv preprint arXiv:2001.06286*, 2020. URL https://arxiv.org/pdf/2001.06286.pdf.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866, 2020. doi: 10.1162/tacl_a_00349. URL https://aclanthology.org/2020.tacl-1.54.

Gertjan Van Noord, Gosse Bouma, Frank Van Eynde, Daniel De Kok, Jelmer Van der Linde, Ineke Schuurman, Erik Tjong Kim Sang, and Vincent Vandeghinste. Large scale syntactic annotation of written Dutch: Lassy. In *Essential speech and language technology for Dutch*, pages 147–164. Springer, Berlin, Heidelberg, 2013. URL https://doi.org/10.1007/978-3-642-30910-6_9.

Gijs Wijnholds. Assessing monotonicity reasoning in Dutch through natural language inference. In *EACL 2023 Findings*, 2023.

Hitomi Yanaka, Koji Mineshima, Daisuke Bekki, Kentaro Inui, Satoshi Sekine, Lasha Abzianidze, and Johan Bos. Can neural networks understand monotonicity reasoning? In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 31–40, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4804. URL https://aclanthology.org/W19-4804.