

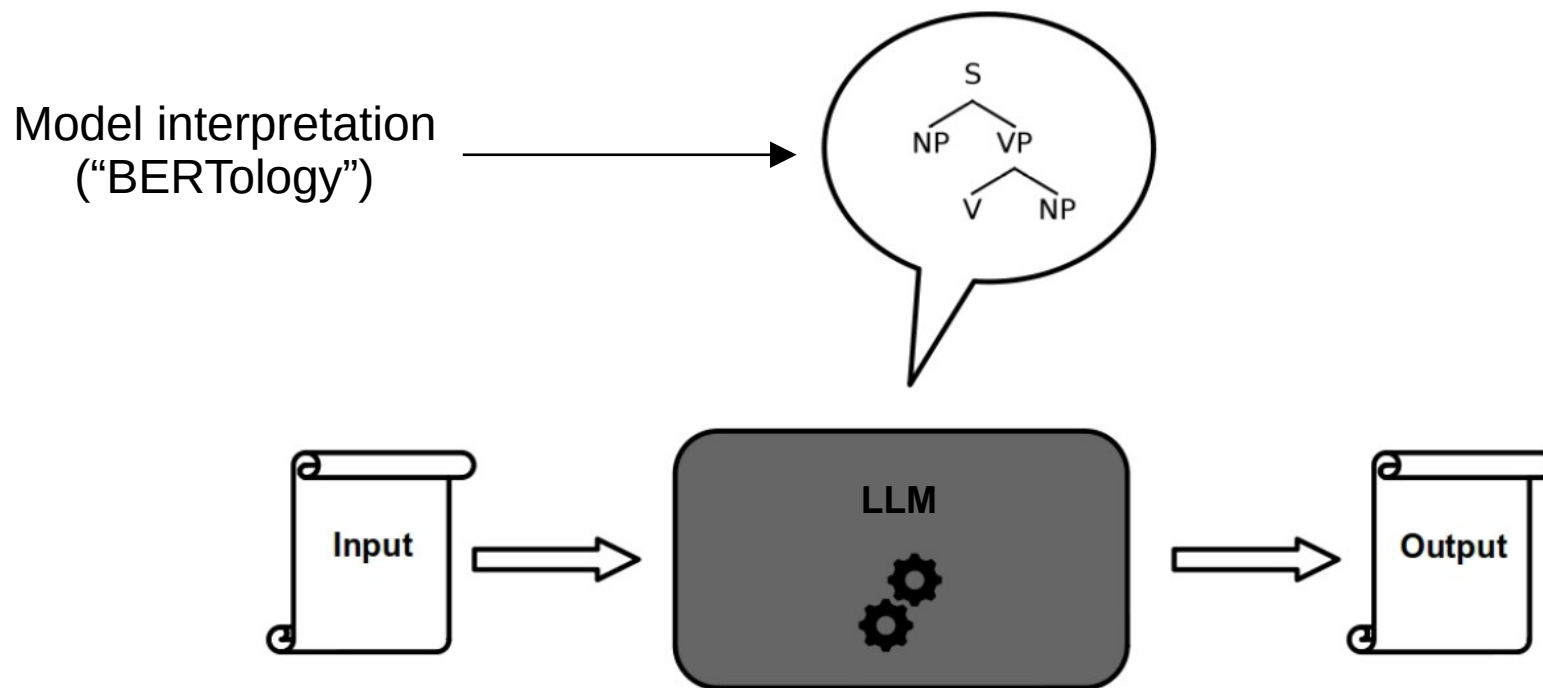


Linguistic Representations in LLMs: Some Foundational Problems

Tommi Buder-Gröndahl

Starting point

- Large language models (LLMs) are “black boxes”; how do they process language?



Starting point

- Common interpretation: LLMs have *internal linguistic representations*

“Our goal is to design a simple method for testing whether **a neural network embeds each sentence’s dependency parse tree** in its contextual word representations – a structural hypothesis.” (Hewitt & Manning, 2019, 4129–4130)

“Investigating how **BERT represents syntax**, we describe evidence that attention matrices **contain grammatical representations.**” (Coenen et al., 2019, 8592)

“In this work, we investigate the **linguistic structure implicitly learned by BERT’s representations.**” (Jawahar et al., 2019, 3652)

“Another theme that emerges in several studies is the **hierarchical nature of the learned representations.**” (Belinkov & Glass, 2019, 52)

“We propose a methodology and offer the first detailed analysis of BERT’s capacity to **capture different kinds of linguistic information by encoding it in its self-attention weights.**” (Kovaleva et al., 2019, 4365)

“We find that **the model represents the steps of the traditional NLP pipeline in an interpretable and localizable way**, and that the regions responsible for each step appear in the expected sequence: POS tagging, parsing, NER, semantic roles, then coreference.” (Tenney, Das, & Pavlick, 2019, 4593)

Starting point

- Common interpretation: LLMs have *internal linguistic representations*

BERT Rediscovered the Classical NLP Pipeline

Ian Tenney¹ Dipanjan Das¹ Ellie Pavlick^{1,2}

¹Google Research ²Brown University

{iftenney, dipanjand, epavlick}@google.com

Abstract

Pre-trained text encoders have rapidly advanced the state of the art on many NLP tasks. We focus on one such model, BERT, and aim to quantify where linguistic information is captured within the network. We find that the model represents the steps of the traditional NLP pipeline in an interpretable and localizable way, and that the regions responsible for each step appear in the expected sequence: POS tagging, parsing, NER, semantic roles, then coreference. Qualitative analysis reveals that the model can and often does adjust this pipeline dynamically, revising lower-level decisions on the basis of disambiguating information from higher-level representations.

of the network directly, to assess whether there exist localizable regions associated with distinct types of linguistic decisions. Such work has produced evidence that deep language models can encode a range of syntactic and semantic information (e.g. Shi et al., 2016; Belinkov, 2018; Tenney et al., 2019), and that more complex structures are represented hierarchically in the higher layers of the model (Peters et al., 2018b; Blevins et al., 2018).

We build on this latter line of work, focusing on the BERT model (Devlin et al., 2019), and use a suite of probing tasks (Tenney et al., 2019) derived from the traditional NLP pipeline to quantify where specific types of linguistic information are

Emergent linguistic structure in artificial neural networks trained by self-supervision

Christopher D. Manning^{a,1}, Kevin Clark^a, John Hewitt^a, Urvashi Khandelwal^a, and Omer Levy^b

^aComputer Science Department, Stanford University, Stanford, CA 94305; and ^bFacebook Artificial Intelligence Research, Facebook Inc., Seattle, WA 98109

Edited by Matan Gavish, Hebrew University of Jerusalem, Jerusalem, Israel, and accepted by Editorial Board Member David L. Donoho April 13, 2020 (received for review June 3, 2019)

This paper explores the knowledge of linguistic structure learned by large artificial neural networks, trained via self-supervision, whereby the model simply tries to predict a masked word in a given context. Human language communication is via sequences of words, but language understanding requires constructing rich hierarchical structures that are never observed explicitly. The mechanisms for this have been a prime mystery of human language acquisition, while engineering work has mainly proceeded by supervised learning on treebanks of sentences hand labeled for this latent structure. However, we demonstrate that modern deep contextual language models learn major aspects of this structure, without any explicit supervision. We develop methods for identifying linguistic hierarchical structure emergent in artificial neural networks and demonstrate that components in these models focus on syntactic grammatical relationships and anaphoric coreference. Indeed, we show that a linear transformation of learned embeddings in these models captures parse tree distances to a surprising degree, allowing approximate reconstruction of the sentence tree structures normally assumed by linguists. These results help explain why these models have brought such large improvements across many language-understanding tasks.

own supervised learning problems by choosing to interpret some of the data as a “label” to be predicted.[†] The canonical case for human language is the language-modeling task of trying to predict the next word in an utterance based on the temporally preceding words (Fig. 2). Variant tasks include the masked language-modeling task of predicting a masked word in a text [a.k.a. the cloze task (11)] and predicting the words likely to occur around a given word (12, 13). Autoencoders (14) can also be thought of as self-supervised learning systems. Since no explicit labeling of the data is required, self-supervised learning is a type of unsupervised learning, but the approach of self-generating supervised learning objectives differentiates it from other unsupervised learning techniques such as clustering.

One might expect that a machine-learning model trained to predict the next word in a text will just be a giant associational learning machine, with lots of statistics on how often the word restaurant is followed by kitchen and perhaps some basic abstracted sequence knowledge such as knowing that adjectives are commonly followed by nouns in English. It is not at all clear that such a system can develop interesting knowledge of the linguistic structure of whatever human language the system is trained on. Indeed, this has been the dominant perspective in linguis-

Challenge

- Common interpretation: LLMs have *internal linguistic representations*
- Problems:
 1. What does this really mean?
 2. How to determine what kinds of representations (if any) LLMs have?

Challenge

- Common interpretation: LLMs have *internal linguistic representations*
- Problems:
 1. What does this really mean?
 2. How to determine what kinds of representations (if any) LLMs have?
- Historically, this is *not* how connectionist language models have usually been interpreted.

Eliminative vs. implementational connectionism

- Eliminative connectionism:

“(...) a reasonable account of the acquisition of past tense can be provided without recourse (...) to the notion of a ‘rule’ as anything more than a description of the language. (...) The child need not figure out what the rules are, nor even that there are rules.”
(Rumelhart and McClelland 1986: 267)

Eliminative vs. implementational connectionism

- Eliminative connectionism:

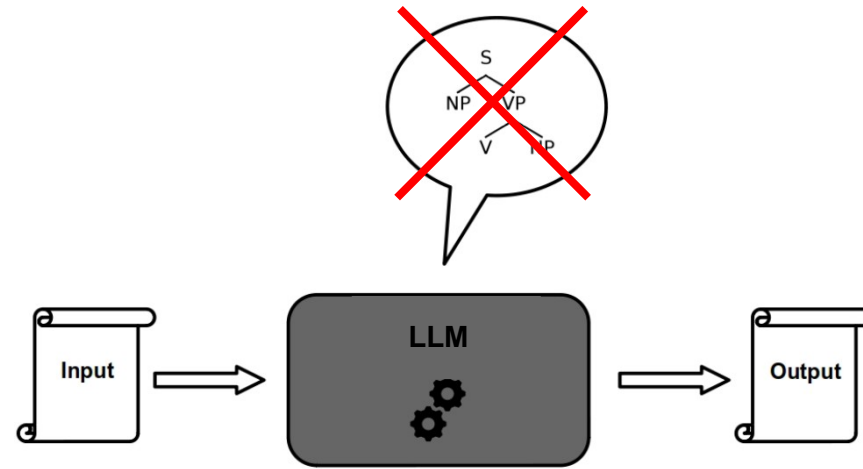
“(...) a reasonable account of the acquisition of past tense can be provided without recourse (...) to the notion of a ‘rule’ as anything more than a description of the language. (...) The child need not figure out what the rules are, nor even that there are rules.”
(Rumelhart and McClelland 1986: 267)

- Implementational connectionism:

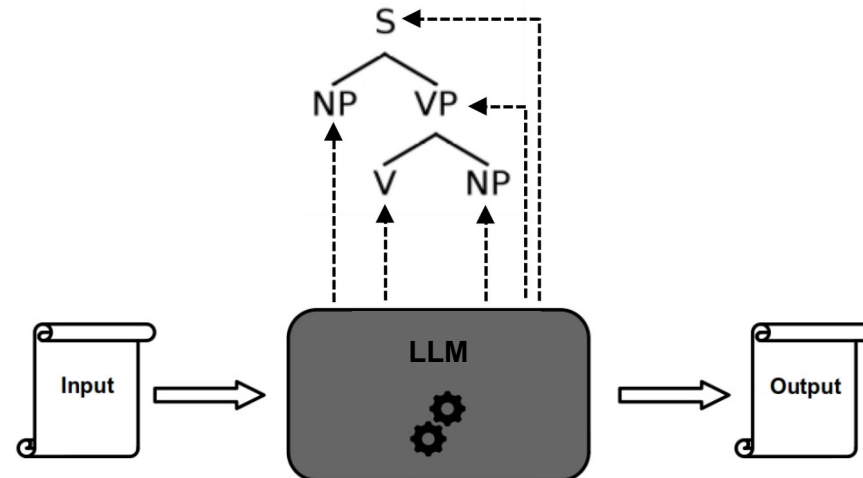
“(...) the way the overall output of one network feeds into the input of another would be isomorphic to the structure of the symbol manipulations captured in the statements of rules.”
(Pinker and Price 1988: 76)

Eliminative vs. implementational connectionism

- Eliminative connectionism:



- Implementational connectionism:



Eliminative vs. implementational connectionism

“Whenever I fire a linguist our system performance improves” (attributed to Jelinek 1988)

Language Resources and Evaluation (2005) 39: 25–34
DOI 10.1007/s10579-005-2693-4

© Springer 2005

Some of my Best Friends are Linguists

FREDERICK JELINEK

*Department of electrical and Computer Engineering, Johns Hopkins University, Barton Hall
320, Baltimore, MD 21218, USA
E-mail: jelinek@jhu.edu*

1. Introduction

This article concerns the relationship between linguistics and the work carried out during 1972–1993 at IBM Research in automatic speech recognition (ASR) and natural language processing (NLP). Many statements I will make will be incomplete: I am not that conversant with the literature. I apologize to those whom I may offend. Conceivably it would have been much better to leave things alone, stay silent. Hopefully this journal will be willing to devote some of its

Eliminative vs. implementational connectionism

“Whenever I fire a linguist our system performance improves” (attributed to Jelinek 1988)

“Due to the otherwise opaque, black-box nature of [LLMs], researchers have employed aspects of linguistic theory in order to characterize their behavior. Questions central to syntax — the study of the hierarchical structure of language — have factored heavily into such work.” (Kulmizev and Nivre 2022: 02)

Language Resources and Evaluation (2005) 39: 25–34
DOI 10.1007/s10579-005-2693-4

© Springer 2005

 frontiers | Frontiers in Artificial Intelligence

TYPE Conceptual Analysis
PUBLISHED 17 October 2022
DOI 10.3389/frai.2022.796788

Some of my Best Friends are Linguists

FREDERICK JELINEK

*Department of electrical and Computer Engineering, Johns Hopkins University, Barton Hall
320, Baltimore, MD 21218, USA
E-mail: jelinek@jhu.edu*

1. Introduction

This article concerns the relationship between linguistics and the work carried out during 1972–1993 at IBM Research in automatic speech recognition (ASR) and natural language processing (NLP). Many statements I will make will be incomplete: I am not that conversant with the literature. I apologize to those whom I may offend. Conceivably it would have been much better to leave things alone, stay silent. Hopefully this journal will be willing to devote some of its

 Check for updates

OPEN ACCESS

EDITED BY
Sebastian Padó,
University of Stuttgart, Germany

REVIEWED BY
William Schuler,
The Ohio State University,
United States
Felice Dell'Orletta,
National Research Council (CNR), Italy

*CORRESPONDENCE
Artur Kulmizev
artur.kulmizev@lingfil.uu.se

SPECIALTY SECTION
This article was submitted to
Natural Language Processing,
a section of the journal
Frontiers in Artificial Intelligence

RECEIVED 17 October 2021
ACCEPTED 02 September 2022
PUBLISHED 17 October 2022

Schrödinger's tree—On syntax and neural language models

Artur Kulmizev^{1*} and Joakim Nivre^{1,2}

¹Computational Linguistics Group, Department of Linguistics and Philology, Uppsala University, Uppsala, Sweden, ²RISE Research Institutes of Sweden, Kista, Sweden

In the last half-decade, the field of natural language processing (NLP) has undergone two major transitions: the switch to neural networks as the primary modeling paradigm and the homogenization of the training regime (pre-train, then fine-tune). Amidst this process, language models have emerged as NLP's workhorse, displaying increasingly fluent generation capabilities and proving to be an indispensable means of knowledge transfer downstream. Due to the otherwise opaque, black-box nature of such models, researchers have employed aspects of linguistic theory in order to characterize their behavior. Questions central to syntax—the study of the hierarchical structure of language—have factored heavily into such work, shedding invaluable

Eliminative vs. implementational connectionism

- Ambiguity in interpreting claims concerning linguistic competence of LLMs:
 1. LLMs have linguistic competence without internal linguistic representations
 2. LLMs develop linguistic representations without innate linguistic competence

Modern language models refute Chomsky's approach to language

Steven T. Piantadosi^{a,b}

^aUC Berkeley, Psychology ^bHelen Wills Neuroscience Institute

The rise and success of large language models undermines virtually every strong claim for the innateness of language that has been proposed by generative linguistics. Modern machine learning has subverted and bypassed the entire theoretical framework of Chomsky's approach, including its core claims to particular insights, principles, structures, and processes. I describe the sense in which modern language models implement genuine *theories* of language, including representations of syntactic and semantic structure. I highlight the relationship between contemporary models and prior approaches in linguistics, namely those based on gradient computations and memorized constructions. I also respond to several critiques of large language models, including claims that they can't answer "why" questions, and skepticism that they are informative about real life acquisition. Most notably, large language models have attained remarkable success at discovering grammar without using any of the methods that some in linguistics insisted were necessary for a science of language to progress.

Large Language Models Demonstrate the Potential of Statistical Learning in Language

Pablo Contreras Kallens, Ross Deans Kristensen-McLachlan, Morten H. Christiansen✉

First published: 25 February 2023 | <https://doi.org/10.1111/cogs.13256> | Citations: 2

This article is part of the "Progress & Puzzles of Cognitive Science" letter series.

[Read the full text >](#)

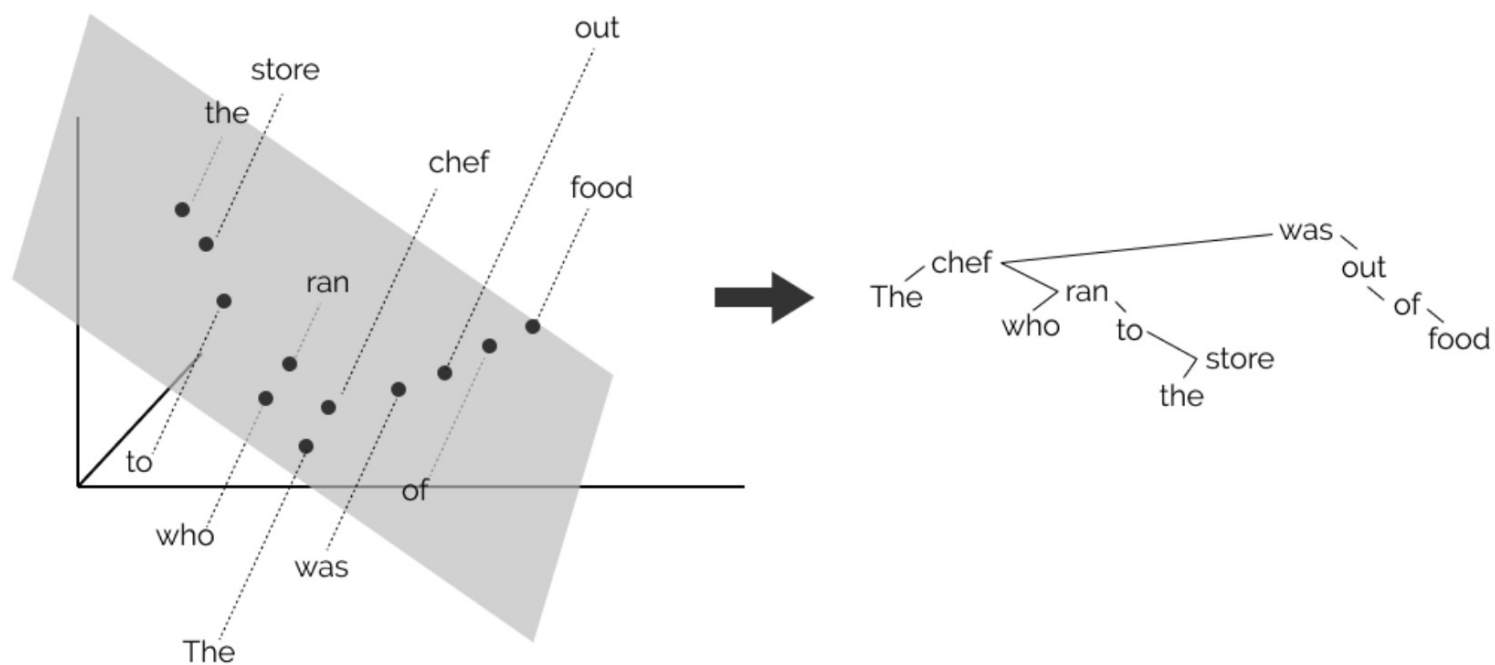
 PDF  TOOLS  SHARE

Abstract

To what degree can language be acquired from linguistic input alone? This question has vexed scholars for millennia and is still a major focus of debate in the cognitive science of language. The complexity of human language has hampered progress because studies of language—especially those involving computational modeling—have only been able to deal with small fragments of our linguistic skills. We suggest that the most recent generation of Large Language Models (LLMs) might finally provide the computational tools to determine empirically how much of the human language ability can be acquired from linguistic experience. LLMs are sophisticated deep learning architectures trained on vast amounts of natural language data, enabling them to perform an impressive range of linguistic tasks. We argue that, despite their clear semantic and pragmatic limitations, LLMs have already demonstrated that human-like grammatical language can be acquired without the need for a built-in grammar. Thus, while there is still much to learn about how humans acquire and use language, LLMs provide full-fledged computational models for cognitive scientists to empirically evaluate just how far statistical learning might take us in explaining the full complexity of human language.

Probing

- Mapping embeddings of pre-trained LLMs to linguistic labels



<https://nlp.stanford.edu/~johnhew/structural-probe.html>

Probing

- Mapping embeddings of pre-trained LLMs to linguistic labels
- Typically *supervised*: labels obtained from human-made (or rule-based) annotations

$$d_B(\mathbf{h}_i, \mathbf{h}_j) = (B(\mathbf{h}_i - \mathbf{h}_j))^T (B(\mathbf{h}_i - \mathbf{h}_j))$$

d_B : parse tree distance between tokens

\mathbf{h}_i : encoding of i :th token

\mathbf{h}_j : encoding of j :th token

B = probe parameter matrix

(Hewitt & Manning 2019)

Probing

- Mapping embeddings of pre-trained LLMs to linguistic labels
- Typically *supervised*: labels obtained from human-made (or rule-based) annotations
- *Parameter-free probing*: unsupervised “bottom-up” alternative

$$f(x_i, x_j) = d(H_\theta(\mathbf{x} \setminus \{x_i\})_i, H_\theta(\mathbf{x} \setminus \{x_i, x_j\})_i)$$

f : impact between two tokens \rightarrow syntactic relation (dependency/phrase)

d = Euclidean distance

$H_\theta(\mathbf{x})_i$: encoding of i :th token of input \mathbf{x} (model parameters θ)

$\mathbf{x} \setminus \{x_i\}$: input \mathbf{x} with i :th token masked

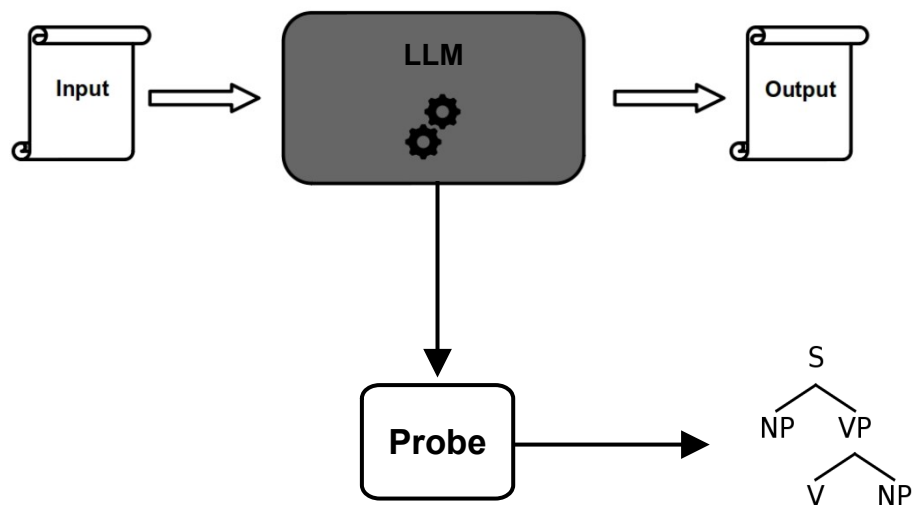
$\mathbf{x} \setminus \{x_i, x_j\}$: input \mathbf{x} with i :th token masked

(Wu et al. 2020)

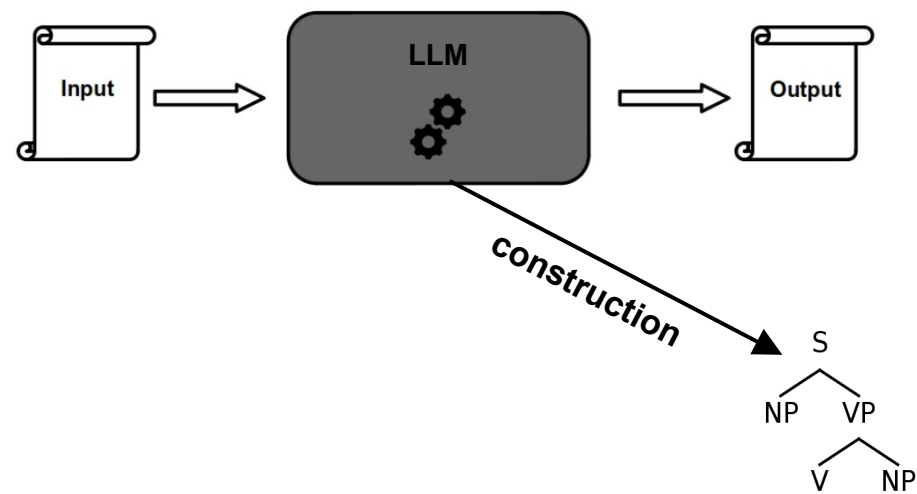
Probing

- Mapping embeddings of pre-trained LLMs to linguistic labels
- Typically *supervised*: labels obtained from human-made (or rule-based) annotations
- *Parameter-free probing*: unsupervised “bottom-up” alternative

Supervised:



Parameter-free:



Papers

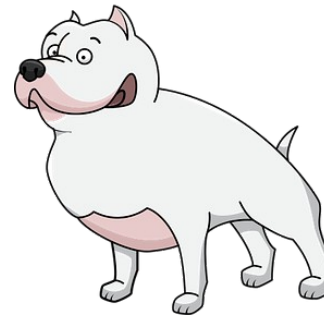
Buder-Gröndahl (2023): “The ambiguity of BERTology: What do large language models represent?”
(published in *Synthese*)

Buder-Gröndahl (in submission): “What does parameter-free probing really uncover?”
(submitted to *ACL 2024*)

Ambiguity of BERTology

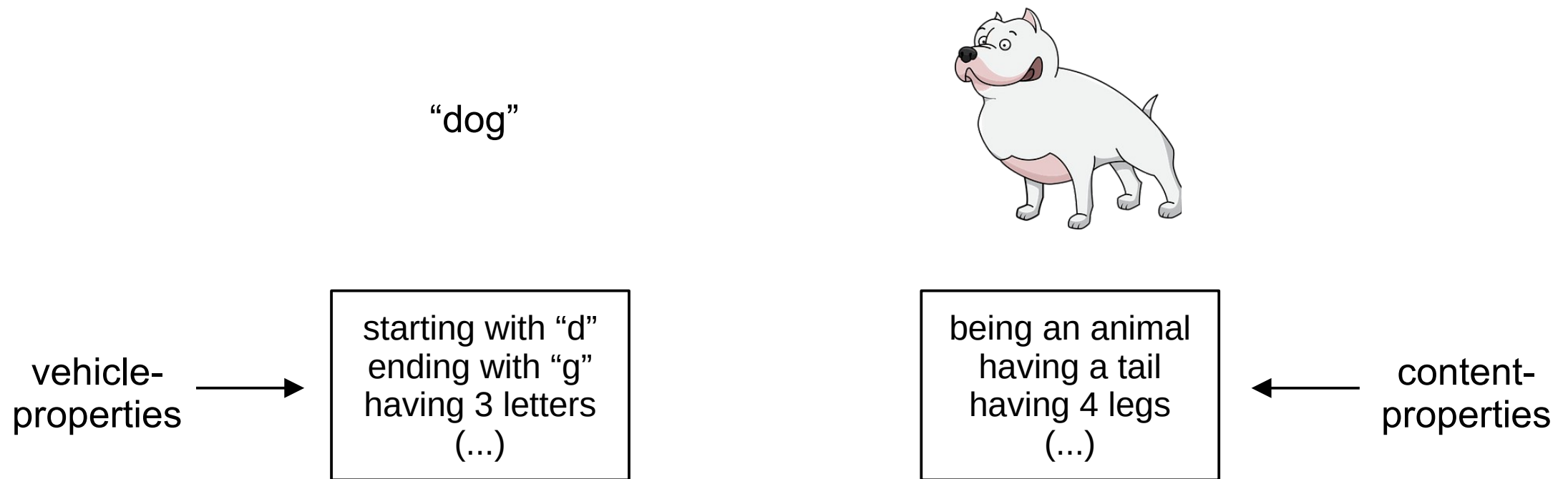
- *Vehicle* = concrete unit operated on in a computational system
- *Content* = interpretation of a vehicle

“dog”



Ambiguity of BERTology

- *Vehicle* = concrete unit operated on in a computational system
- *Content* = interpretation of a vehicle



Ambiguity of BERTology

- *Vehicle* = concrete unit operated on in a computational system
- *Content* = interpretation of a vehicle
- Individuation of vehicles
 1. by vehicle-properties: word that begins with “d”, ends with “g”, (...)
 2. by content-properties: word that means a certain animal that has a tail, 4 legs (...)

Ambiguity of BERTology

- *Vehicle* = concrete unit operated on in a computational system
- *Content* = interpretation of a vehicle
- Individuation of vehicles
 1. by vehicle-properties: word that begins with “d”, ends with “g”, (...)
 2. by content-properties: word that means a certain animal that has a tail, 4 legs (...)
- *Linguistic representation* = vehicle individuated by some linguistic property
 - BERTology: vehicles are LLM-states
 - Linguistic properties: categories (e.g. part-of-speech), relations (e.g. dependency), structures (e.g. phrases)

Ambiguity of BERTology

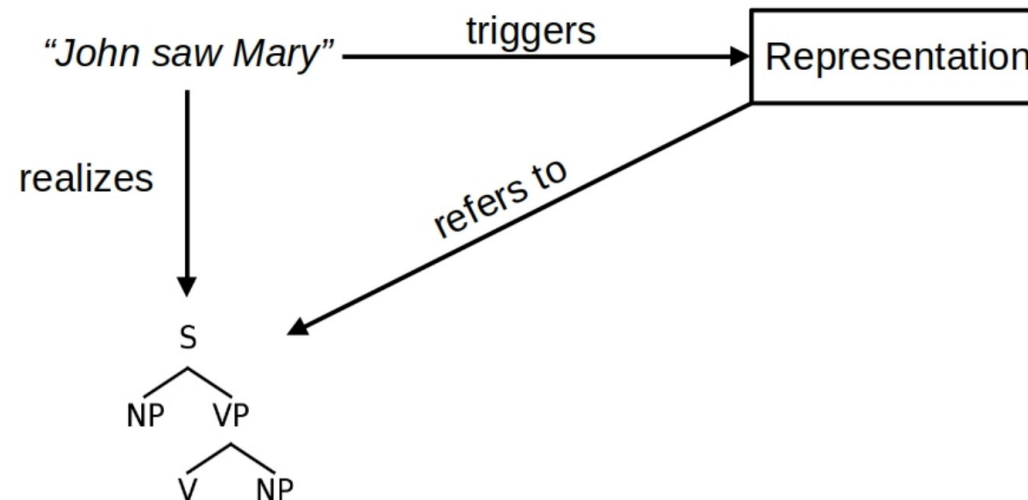
- *Vehicle* = concrete unit operated on in a computational system
- *Content* = interpretation of a vehicle
- Individuation of vehicles
 1. by vehicle-properties: word that begins with “d”, ends with “g”, (...)
 2. by content-properties: word that means a certain animal that has a tail, 4 legs (...)
- *Linguistic representation* = vehicle individuated by some linguistic property
 - BERTology: vehicles are LLM-states
 - Linguistic properties: categories (e.g. part-of-speech), relations (e.g. dependency), structures (e.g. phrases)
- Are linguistic representations individuated by vehicle-properties or content-properties?

Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- *Content reading*: linguistic representations are individuated by linguistic contents

Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- *Content reading*: linguistic representations are individuated by linguistic contents
- Content reading is intuitive, but contrasts ubiquitous assumptions of formal linguistic theory
 - Typical idea: representational content is based on information picked up from the data



Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- *Content reading*: linguistic representations are individuated by linguistic contents
- Content reading is intuitive, but contrasts ubiquitous assumptions of formal linguistic theory
 - Typical idea: representational content is based on information picked up from the data
 - Such information must be *in the data* to begin with
 - But abstract syntax does *not* reduce to properties of linear strings

Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- *Content reading*: linguistic representations are individuated by linguistic contents
- Content reading is intuitive, but contrasts ubiquitous assumptions of formal linguistic theory
 - Typical idea: representational content is based on information picked up from the data
 - Such information must be *in the data* to begin with
 - But abstract syntax does *not* reduce to properties of linear strings

“syntax determines units of combined lexical items that are not identifiable or individuated in terms of linear order or any other perceptible property associated with morphophonemic form.”

(Collins 2023, 7)

“the perspective in [Chomsky (1975)] is top-down rather than bottom up. (...) the ‘representations’ are not derived from the utterance.”

(Adger 2022, 251)

Ambiguity of BERTology

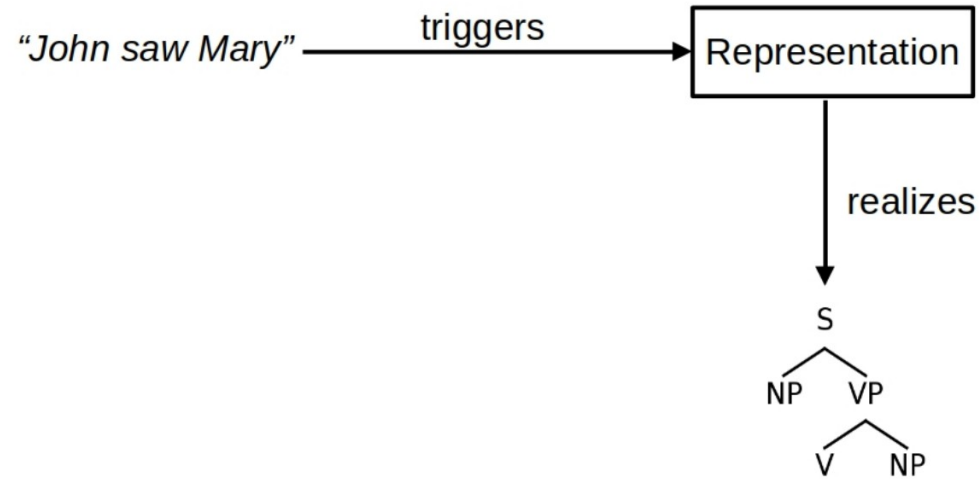
- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- *Content reading*: linguistic representations are individuated by linguistic contents
- Content reading is intuitive, but contrasts ubiquitous assumptions of formal linguistic theory
 - Typical idea: representational content is based on information picked up from the data
 - Such information must be *in the data* to begin with
 - But abstract syntax does *not* reduce to properties of linear strings
 - Accepting abstract syntax makes claims of its representation by LLMs *trivially false*
 - ...But rejecting abstract syntax *also* makes such claims trivially false (eliminativism)

Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- ~~*Content reading*: linguistic representations are individuated by linguistic contents~~
- Content reading is intuitive, but contrasts ubiquitous assumptions of formal linguistic theory
 - Typical idea: representational content is based on information picked up from the data
 - Such information must be *in the data* to begin with
 - But abstract syntax does *not* reduce to properties of linear strings
 - Accepting abstract syntax makes claims of its representation by LLMs *trivially false*
 - ...But rejecting abstract syntax *also* makes such claims trivially false (eliminativism)

Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- Vehicle-reading succumbs to a different triviality problem
 - Basic idea: LLM-state somehow realizes abstract linguistic structure



Ambiguity of BERTology

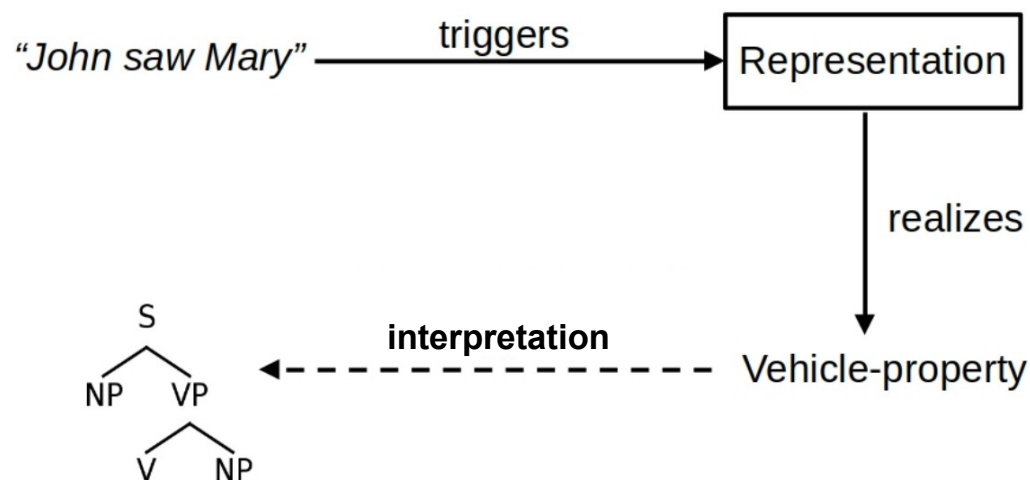
- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- Vehicle-reading succumbs to a different triviality problem
 - Basic idea: LLM-state somehow realizes abstract linguistic structure
 - But abstract structures are not literally “in” concrete systems → more indirect relation needed

“We don’t have sets in our heads. So you have to know that when we develop a theory about our thinking, about our computation, internal processing and so on in terms of sets, that it’s going have to be translated into some terms that are neurologically realizable.”

(Chomsky 2012, 91)

Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- Vehicle-reading succumbs to a different triviality problem
 - Basic idea: LLM-state somehow realizes abstract linguistic structure
 - But abstract structures are not literally “in” concrete systems → more indirect relation needed
 - *Mapping* concrete vehicles to abstract structures



Ambiguity of BERTology

- *Vehicle reading*: linguistic representations are individuated by their linguistic vehicle-properties
- Vehicle-reading succumbs to a different triviality problem
 - Basic idea: LLM-state somehow realizes abstract linguistic structure
 - But abstract structures are not literally “in” concrete systems → more indirect relation needed
 - *Mapping* concrete vehicles to abstract structures
 - But mapping theories of computational implementation have well-known triviality problems
 - Any sufficiently complex system can be mapped to abstract structures ([Searle 1992](#))
 - Anything can be mapped to finite-storage computation ([Putnam 1988](#), [Sprevak 2018](#))

Ambiguity of BERTology

- The mapping account could be salvaged by considering *explanatory virtues* of different mappings
- Abstract formalisms are used for *surrogate reasoning* about concrete systems ([Swoyer 1991](#))

Example – An Adder



([Egan 2010: 115](#))

Ambiguity of BERTology

- The mapping account could be salvaged by considering *explanatory virtues* of different mappings
- Abstract formalisms are used for *surrogate reasoning* about concrete systems ([Swoyer 1991](#))
- Some formalisms yield better surrogate reasoning than others
- BERTology: find formalism that yields *best surrogate reasoning about LLMs*

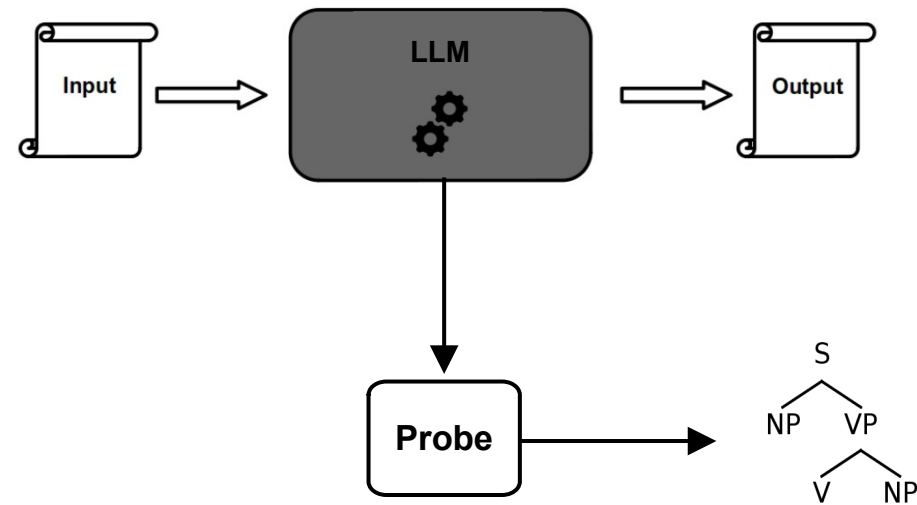
Example – An Adder



([Egan 2010: 115](#))

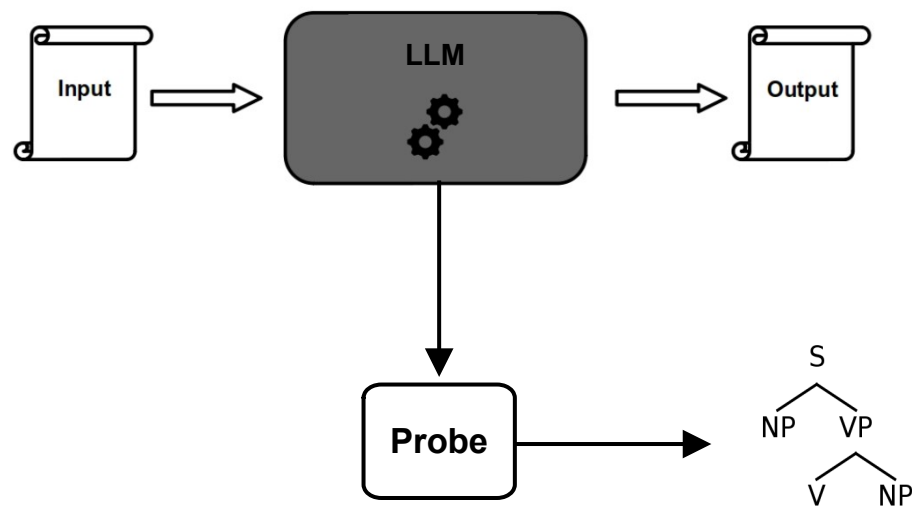
Ambiguity of BERTology

- Supervised probing is insufficient for this:
 - What are the LLM-internal states that best predict formalism F?



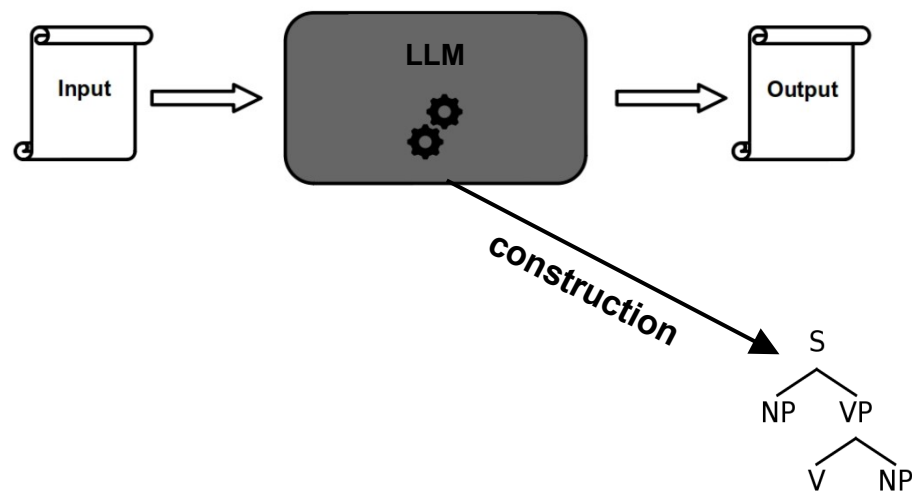
Ambiguity of BERTology

- Supervised probing is insufficient for this:
 - What are the LLM-internal states that best predict formalism F ?
- vs.
What is the formalism F^* that best predicts the LLM-internal pipeline?



Parameter-free probing

- More “bottom-up” than supervised probing: has potential to mitigate some of the issues
- What kind of a grammatical representation is generated from the LLM?



Parameter-free probing

- More “bottom-up” than supervised probing: has potential to mitigate some of the issues
- What kind of a grammatical representation is generated from the LLM?
- Replicated *perturbed masking* results ([Wu et al. 2020](#)), compared BERT-derived parses to Universal Dependencies (UD) in the English Parallel Universal Dependencies (PUD) treebank

Parameter-free probing

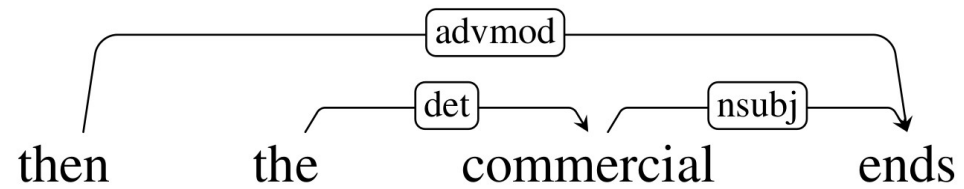
- More “bottom-up” than supervised probing: has potential to mitigate some of the issues
- What kind of a grammatical representation is generated from the LLM?
- Replicated *perturbed masking* results ([Wu et al. 2020](#)), compared BERT-derived parses to Universal Dependencies (UD) in the English Parallel Universal Dependencies (PUD) treebank

“In fact, there is actually no guarantee that our probe will find a strong correlation with human-designed syntax, since we do not introduce the human-designed syntax as supervision. What we found is the ‘natural’ syntax inherent in BERT, which is acquired from self-supervised learning on plain text.”

([Wu et al 2020, 4173](#))

Parameter-free probing

UD:

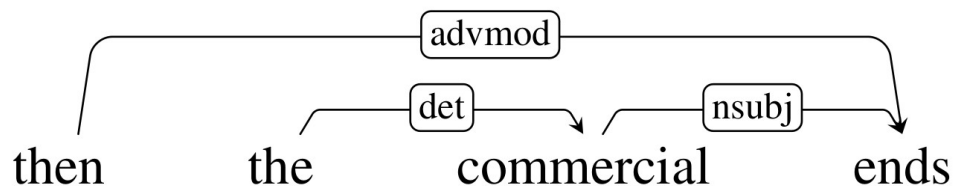


BERT:

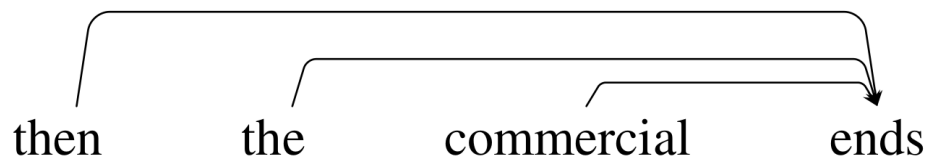


Parameter-free probing

UD:



BERT:



$Dep(x)$: deprel assigned to x by UD

$Head_{UD}(x)$: head assigned to x by UD

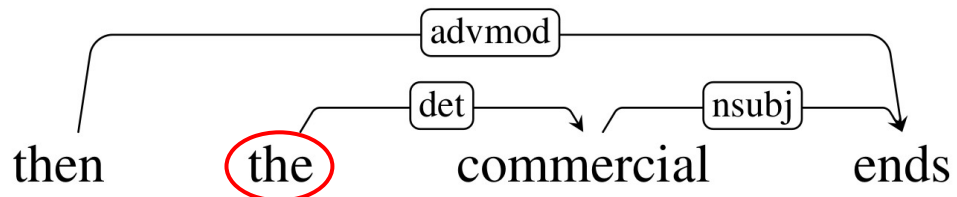
$Head_{BERT}(x)$: head assigned to x by BERT

$H_U(x) = Dep(Head_{UD}(x))$

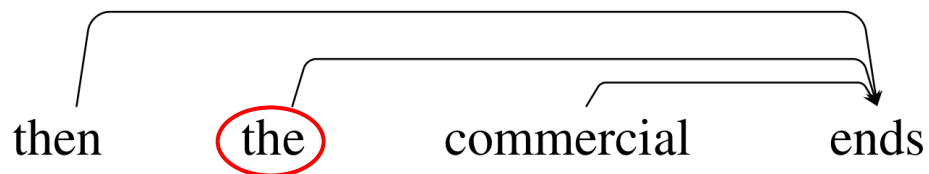
$H_B(x) = Dep(Head_{BERT}(x))$

Parameter-free probing

UD:



BERT:



$Dep(x)$: deprel assigned to x by UD

$Head_{UD}(x)$: head assigned to x by UD

$Head_{BERT}(x)$: head assigned to x by BERT

$H_U(x) = Dep(Head_{UD}(x))$

$H_B(x) = Dep(Head_{BERT}(x))$

$Dep(the) = det,$

$H_U(the) = nsubj,$

$H_B(the) = root$

Parameter-free probing

- Analyzed all dependent-head shifts between UD and BERT, focusing on:
 - verbal argument structure
 - noun phrase structure
 - adjective/adverb modifiers
 - prepositional phrases

Parameter-free probing

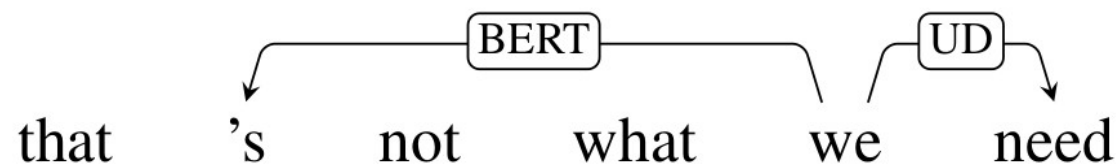
- Analyzed all dependent-head shifts between UD and BERT, focusing on:
 - verbal argument structure
 - noun phrase structure
 - adjective/adverb modifiers
 - prepositional phrases
- General results
 - Shift ratio: 58%
 - 80% of *Dep*-types had a shift rate over 50%
 - Most common H_B : *root* (35% of all shifts)

Parameter-free probing

- BERT systematically over-assigned the root verb as a head

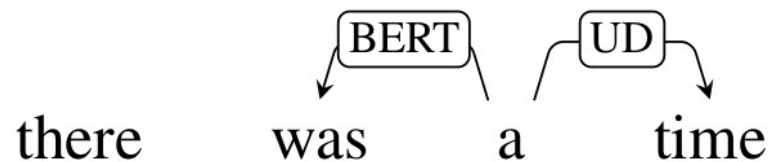
Parameter-free probing

- BERT systematically over-assigned the root verb as a head
 - of embedded clause arguments



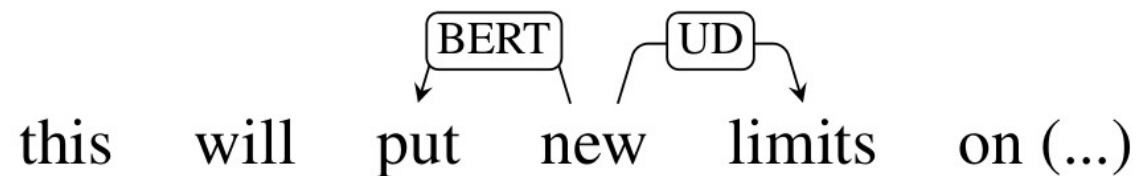
Parameter-free probing

- BERT systematically over-assigned the root verb as a head
 - of embedded clause arguments
 - of determiners



Parameter-free probing

- BERT systematically over-assigned the root verb as a head
 - of embedded clause arguments
 - of determiners
 - of adjective/adverb modifiers



Parameter-free probing

- BERT systematically over-assigned the root verb as a head
 - of embedded clause arguments
 - of determiners
 - of adjective/adverb modifiers
- This behavior is:
 - *non-recursive*: no proper embedding
 - *linguistically incoherent* in any prominent theoretical framework

Summary

- BERTology has committed to representational realism (implementational connectionism)
 - This is ambiguous between vehicle- and content-readings of “linguistic representation”
 - Content-reading makes representation-claims of abstract syntax trivially false
 - Vehicle-reading succumbs to the triviality-problem in mapping accounts of implementation
- Vehicle-reading could be salvaged: which formalism has optimal theoretical virtues for LLMs?
 - Supervised probing is too weak to establish this: linguistic analysis is presupposed
 - Parameter-free probing is more “bottom-up”, but yields linguistically incoherent results

Summary

- BERTology has committed to representational realism (implementational connectionism)
 - This is ambiguous between vehicle- and content-readings of “linguistic representation”
 - Content-reading makes representation-claims of abstract syntax trivially false
 - Vehicle-reading succumbs to the triviality-problem in mapping accounts of implementation
- Vehicle-reading could be salvaged: which formalism has optimal theoretical virtues for LLMs?
 - Supervised probing is too weak to establish this: linguistic analysis is presupposed
 - Parameter-free probing is more “bottom-up”, but yields linguistically incoherent results

Is human grammar an appropriate analogy for LLMs after all?

References

- Adger, D. (2022). What are linguistic representations? *Mind & Language*, 37 (2), 248–260.
- Belinkov, Y., & Glass, J. (2019). Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7 , 49–72.
- Chomsky, N. (2012). *The science of language*. Cambridge: Cambridge University Press.
- Coenen, A., Reif, E., Yuan, A., Kim, B., Pearce, A., Viégas, F., Wattenberg, M. (2019). Visualizing and measuring the geometry of BERT. *Proceedings of the 33rd Conference on Neural Information Processing Systems* (p. 8592-8600).
- Collins, J. (2023). Internalist priorities in a philosophy of words. *Synthese*, 201 (3), 110.
- Egan, F. (2010). Computation models: a modest role for content. *Studies in History and Philosophy of Science*, 41 (3), 253–259.
- Hewitt, J., & Manning, C.D. (2019). A structural probe for finding syntax in word representations. *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)* (pp. 4129–4138).
- Jawahar, G., Sagot, B., Seddah, D. (2019). What does BERT learn about the structure of language? *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (pp. 3651–3657).
- Kovaleva, O., Romanov, A., Rogers, A., Rumshisky, A. (2019). Revealing the dark secrets of BERT. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing* (pp. 4365–4374).
- Kulmizev, A., & Nivre, J. (2022). Schrödinger’s tree—on syntax and neural language models. *Frontiers in Artificial Intelligence*, 5.

References

Kulmizev, A., Ravishankar, V., Abdou, M., Nivre, J. (2020). Do neural language models show preferences for syntactic formalisms? Proceedings of the 58th annual meeting of the association for computational linguistics (pp. 4077–4091).

Pinker, S., & Price, A. (1988). On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28 (1–2), 73–193.

Putnam, H. (1988). *Representation and reality*. Cambridge: MIT Press.

Rumelhart, D.E., & McClelland, J.L. (1986). On learning the past tenses of English verbs. J.L. McClelland, D.E. Rumelhart, & T.P.R. Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition: Vol. 2. psychological and biological models* (pp. 216–271). Cambridge: MIT Press.

Searle, J. (1992). *The Rediscovery of the Mind*. Cambridge: MIT Press.

Sprevak, M. (2018). Triviality arguments about computational implementation. M. Sprevak & M. Colombo (Eds.), *Routledge handbook of the computational mind* (pp. 175–191). London: Routledge.

Swoyer, C. (1991). Structural representation and surrogate reasoning. *Synthese*, 87 (3), 449–508,

Tenney, I., Das, D., Pavlick, E. (2019). BERT rediscovers the classical NLP pipeline. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (pp. 4593–4601).

Wu, Z., Chen, Y, Kao, B. and Liu, Q. (2020). Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4166–4176.

Extra slides: result tables

Dep	H _U	Ratio	Count
nsubj	root	0.24	198
	acl:relcl	0.81	140
	ccomp	0.92	101
	advcl	0.79	80
	conj	0.83	68
	parataxis	0.64	46
nsubj:pass	root	0.29	38
	acl:relcl	0.94	32
	advcl	0.91	21
obj	advcl	0.66	86
	xcomp	0.75	82
	acl:relcl	0.78	58
	conj	0.66	58
	acl	0.73	52
	root	0.15	47
	ccomp	0.73	29

Table 1: Verbal argument structure: subjects and objects.

Dep	H _U	Ratio	Count
det	obl	0.52	261
	obj	0.67	253
	nsubj	0.54	208
	nmod	0.49	191
	conj	0.57	44
	nsubj:pass	0.54	43
	nmod:poss	0.64	23
	appos	0.68	21
nmod:poss	obj	0.70	56
	nmod	0.72	55
	obl	0.58	54
	nsubj	0.70	53
nummod	obl	0.69	55
	nmod	0.71	25

Table 2: Determiners, possessors, and numerals.

Extra slides: result tables

Dep	H _U	Ratio	Count
amod	obj	0.62	151
	obl	0.52	151
	nmod	0.53	132
	nsubj	0.53	118
	conj	0.63	56
	nsubj:pass	0.52	29
	compound	0.57	21
advmod	root	0.18	57
	conj	0.62	53
	advcl	0.72	51
	acl:relcl	0.73	40
	amod	0.73	36
	advmod	0.71	32
	nummod	0.75	27
	ccomp	0.68	27
	obl	0.72	21
	xcomp	0.72	21
nmod	obl	0.88	243
	obj	0.89	202
	nsubj	0.87	163
	nmod	0.84	127
	conj	0.88	59
	nsubj:pass	0.83	34
	appos	0.85	23
	root	0.38	20

Table 3: Adjectival, adverbial, and nominal modifiers.

Dep	H _U	Ratio	Count
case	obl	0.72	877
	nmod	0.73	783
	nmod:poss	0.83	85
obl	root	0.47	283
	acl:relcl	0.97	117
	advcl	0.95	92
	acl	0.93	88
	conj	0.91	90
	xcomp	0.95	89
	ccomp	0.96	50
	parataxis	0.96	25

Table 4: Prepositional phrases.

Extra slides: result tables

Dep	Ratio	Count
case	0.7251	1799
punct	0.5135	1252
det	0.5433	1105
nmod	0.8500	912
obl	0.7082	869
amod	0.5402	719
nsubj	0.4683	650
compound	0.6675	538
conj	0.8176	511
mark	0.7964	442
obj	0.5011	438
cc	0.7615	431
advmod	0.5035	426
nmod:poss	0.6703	244
advcl	0.7158	209
aux	0.4474	183
acl:relcl	0.8483	179
xcomp	0.5815	157
nummod	0.6071	153
nsubj:pass	0.5720	135
acl	0.6895	131
appos	0.8310	118
flat	0.4978	114
cop	0.3270	103
ccomp	0.7259	98
aux:pass	0.2915	79
parataxis	0.5979	58
fixed	0.5243	54
root	0.0363	36
compound:prt	0.4714	33
nmod:tmod	0.6667	26
csubj	0.5926	16
expl	0.2459	15
obl:npm	0.7000	14
obl:tmod	0.6111	11
nmod:npm	0.5263	10
det:predet	0.8889	8
cc:preconj	0.5455	6
csubj:pass	1.0000	3
dislocated	1.0000	2
reparandum	1.0000	1
discourse	1.0000	1
iobj	0.1000	1

Table 6: All dependency-head shifts ordered by Dep
 (“Ratio”: ratio of shifts from all tokens with the Dep).

H_U	Ratio	Count
obl	0.6802	2048
root	0.2664	1694
nmod	0.6788	1655
conj	0.7654	1292
obj	0.7283	946
nsubj	0.6651	872
advcl	0.7791	663
acl:relcl	0.8109	579
xcomp	0.8168	495
ccomp	0.8327	458
acl	0.7762	281
appos	0.7301	238
parataxis	0.7409	223
nsubj:pass	0.6494	176
amod	0.7368	140
nmod:poss	0.7707	121
compound	0.6289	100
advmod	0.7810	82
csubj	0.7703	57
nummod	0.8036	45
flat	0.8276	24
cc	0.8750	14
obl:npm	0.6667	14
obl:tmod	0.5833	14
csubj:pass	0.8667	13
mark	0.6000	9
nmod:tmod	0.2857	8
case	0.1591	7
dislocated	1.0000	6
nmod:npm	0.8571	6
iobj	0.8333	5
dep	1.0000	2
det	0.6667	2
cc:preconj	1.0000	1

Table 7: All dependency-head shifts ordered by H_U
 (“Ratio”: ratio of shifts from all tokens with the H_U).

H_D	Ratio	Count
root	0.4763	4244
case	0.9684	1135
amod	0.9386	764
compound	0.9107	602
nsubj	0.5525	542
obl	0.3431	503
nmod	0.3771	474
det	0.9978	453
punct	1.0000	404
obj	0.5306	399
advmod	0.9425	377
cc	0.9936	310
conj	0.4107	276
mark	0.9636	159
nummod	0.9341	156
advcl	0.4519	155
cop	1.0000	122
nsubj:pass	0.5622	122
nmod:poss	0.7707	121
aux	1.0000	119
xcomp	0.5174	119
acl	0.5622	104
flat	0.9533	102
aux:pass	1.0000	92
acl:relcl	0.3571	75
parataxis	0.4621	67
ccomp	0.3907	59
appos	0.3931	57
fixed	1.0000	55
compound:prt	1.0000	33
nmod:tmod	0.5455	24
expl	1.0000	14
obl:npm	0.6316	12
det:predet	1.0000	9
nmod:npm	0.9000	9
csubj	0.3462	9
cc:preconj	1.0000	4
obl:tmod	0.2308	3
reparandum	0.6667	2
dislocated	1.0000	1
discourse	1.0000	1
vocative	1.0000	1
csubj:pass	0.3333	1

Table 8: All dependency-head shifts ordered by H_D
 (“Ratio”: ratio of shifts from all tokens with the H_D).

Extra slides: result tables

Dep-H _U -H _B shift (count)		
case-obl-root (521)	case-nmod-root (231)	cc-conj-root (191)
det-obj-root (141)	det-nsubj-root (134)	case-nmod-obl (122)
punct-root-obl (117)	nmod-obl-root (107)	det-obl-case (101)
det-nmod-case (100)	case-nmod-obj (99)	obl-root-case (97)
mark-xcomp-root (87)	nmod-nsubj-root (85)	mark-advcl-root (84)
nmod-obj-root (83)	punct-root-nsubj (79)	case-nmod-nsubj (79)
case-nmod-nmod (73)	det-obl-amod (66)	nsubj-ccomp-root (66)
amod-obj-root (64)	det-obl-root (62)	amod-obl-root (61)
case-nmod:poss-root (56)	nmod-nmod-root (54)	punct-root-advmod (53)
case-obl-acl (52)	nsubj-acl:relcl-root (52)	amod-nsubj-root (49)
punct-root-punct (45)	compound-nsubj-root (45)	mark-ccomp-root (44)
compound-obl-root (44)	compound-nmod-root (43)	obl-xcomp-root (43)
obl-acl-root (43)	obl-acl:relcl-root (43)	punct-conj-cc (41)
obl-conj-root (41)	amod-obj-det (40)	obl-root-amod (40)
punct-root-nmod (38)	amod-nmod-root (38)	obl-advcl-root (38)
obl-root-compound (38)	nsubj-advcl-root (37)	obj-advcl-root (36)
nummod-obl-root (36)	punct-root-parataxis (35)	nsubj-root-amod (35)
obj-xcomp-root (35)	punct-conj-conj (35)	nmod-obl-case (34)
case-obl-advcl (33)	case-obl-conj (33)	punct-conj-root (32)
nmod-obj-case (32)	det-nmod-amod (31)	amod-nmod-case (31)
nmod-nmod-case (31)	nsubj-root-compound (31)	nmod:poss-obl-case (31)
punct-appos-root (30)	case-obl-acl:relcl (30)	conj-nmod-root (30)
case-nmod-det (29)	det-nsubj-amod (28)	nmod-obj-amod (28)
cc-conj-obl (27)	punct-conj-nmod (26)	case-nmod-conj (26)
det-nmod-root (26)	det-obj-advcl (26)	nmod-obl-compound (26)
det-nmod-compound (25)	nmod-conj-root (25)	compound-obj-root (25)
nsubj-conj-root (25)	obj-acl-root (25)	det-nsubj:pass-root (24)
obl-root-nmod (24)	conj-nsubj-root (24)	amod-obl-det (23)
nmod:poss-nmod-case (23)	nmod:poss-nsubj-root (23)	punct-conj-obl (22)
det-obj-amod (22)	obl-acl:relcl-case (22)	nsubj-root-case (22)
cc-conj-nmod (22)	advmod-advcl-root (22)	conj-nmod-cc (22)
nmod-nsubj-case (21)	obl-root-nummod (21)	flat-nsubj-root (21)
obj-acl:relcl-root (21)	acl-obj-root (21)	punct-root-det (20)
case-obl-xcomp (20)	nmod-obl-amod (20)	compound-obl-det (20)
compound-nmod-case (20)	obl-ccomp-root (20)	

Table 9: Dep-H_U-H_B shifts and their counts (minimum count: 20).