Lecture 15: Generative Linguistics versus Generative AI
Ma 191c: Mathematical Models of Generative Linguistics

Matilde Marcolli

Caltech, Spring 2024

this part based on

- Matilde Marcolli, Robert C. Berwick, Noam Chomsky, *Syntax-semantics interface: an algebraic model*, arXiv:2311.06189

also included in the book:
Matilde Marcolli, Noam Chomsky, Robert C. Berwick, "Mathematical structure of syntactic Merge", MIT Press.

LLM (large language models): a deep dive backward in time!

Zellig Harris circa **1940s, early 1950s**

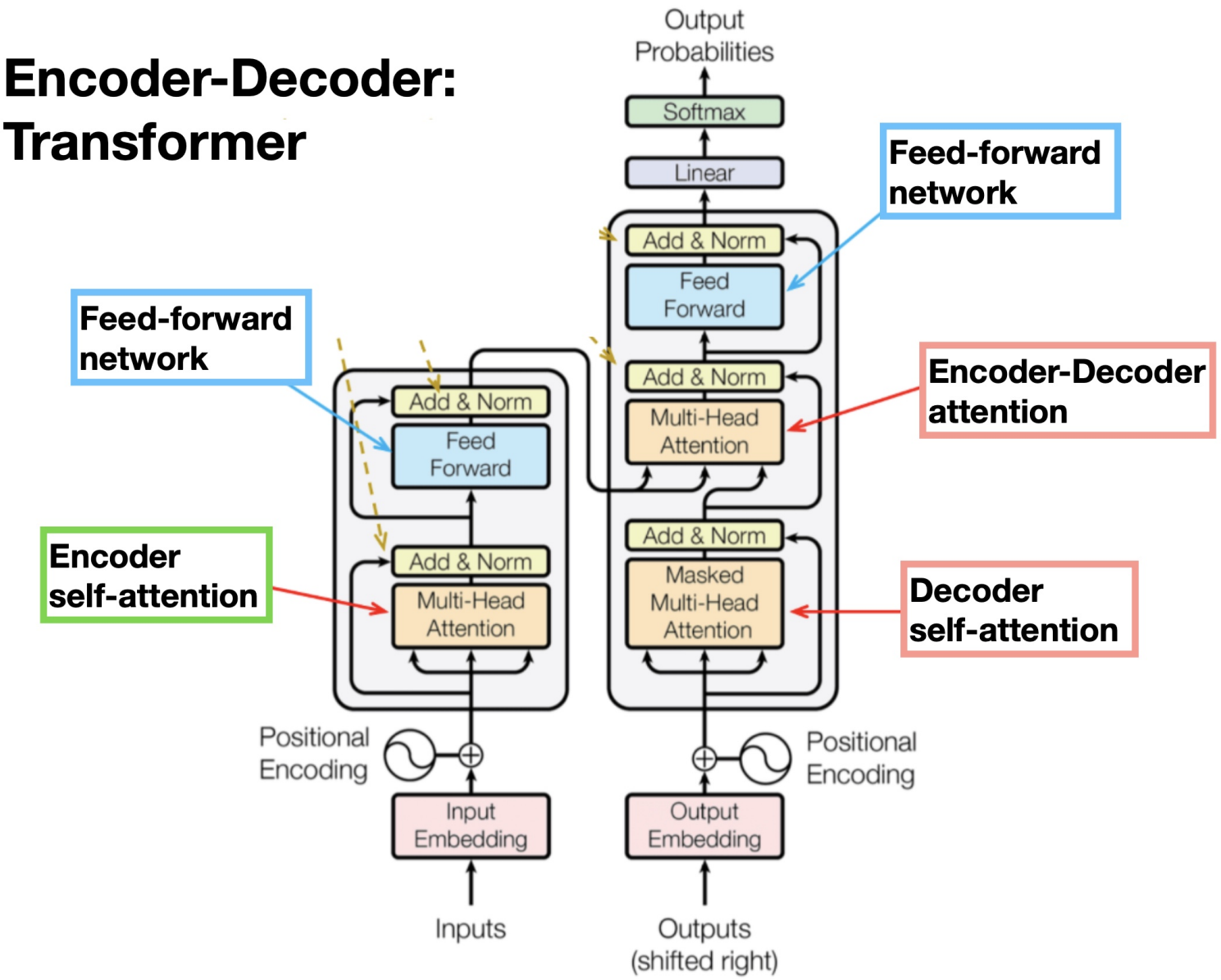
- view language as strings of texts (not as structures)
- primarily seen through their *distributional properties* (texts included as subtexts of other texts)
- syntax is seen primarily in terms of such distributional relations (no internal computational modeling)
- relations between constituents in a syntactic structure are *probabilistic relations*; no abstract structural relations
- rules for modification of sentences are to be extracted "mechanically" from distributional data through algorithms (see below discussion of transformers and circuits: "mechanistic interpretability")
- view of linguistics just prior to the main shift of scientific paradigm to modern linguistics (mid 1950s)
- intuitively simple model, but also known to be inaccurate
- Zellig Harris, *Distributional structure*, 1954

- view influenced by *behaviorism* in psychology: can only statistically observe behavior, no theoretical modeling of mind (contrary to later development of cognitive science and neuroscience)
- change of paradigm in linguistics happened in 1955:
  - Noam Chomsky, *The logical structure of linguistic theory*. Ms., Harvard/MIT 1955. [Published in part, Plenum 1975]
- among main criticisms of Harris' distributional viewpoint: different sentences with *same* syntactic structure but very different probabilities; not capturing intrinsic computational structure of syntax
- scientific method: theoretical hypotheses and models tested on data, different view of role of predictions; difference between "taxonomic" linguistics and theoretical linguistics
- current anti-science, anti-theory stance of part of the LLM and ML community is just revamped old behaviorism on steroids

Behaviorism's new clothes: Trasformers

ok, not quite those transformers...

Behaviorism's new clothes: Trasformers



Attention modules and transformer architectures (first look)

- in self-attention modules one considers three linear transformations: Q (queries), K (keys), and V (values), $Q, K \in \mathrm{Hom}(\mathcal{S}, \mathcal{S}')$ and $V \in \mathrm{Hom}(\mathcal{S}, \mathcal{S}'')$, where $\mathcal{S}'$ and $\mathcal{S}''$ are themselves vector spaces of semantic vectors (in general of dimensions not necessarily equal to that of $\mathcal{S}$)
- these encode (statistically) other words that are structurally related to ("called by" or "calling for") the given word
- fixed identifications $\mathcal{S} \simeq \mathbb{R}^n$, $\mathcal{S}' \simeq \mathbb{R}^m$, $\mathcal{S}'' \simeq \mathbb{R}^d$ with Euclidean vector spaces, with assigned bases, and one works with the corresponding matrix representations of $Q, K \in \mathrm{Hom}(\mathbb{R}^n, \mathbb{R}^m)$ and $V \in \mathrm{Hom}(\mathbb{R}^n, \mathbb{R}^d)$
- target Euclidean space $\mathcal{S}'$ is endowed with an inner product $\langle \cdot, \cdot \rangle$, that can be used to estimate semantic similarity

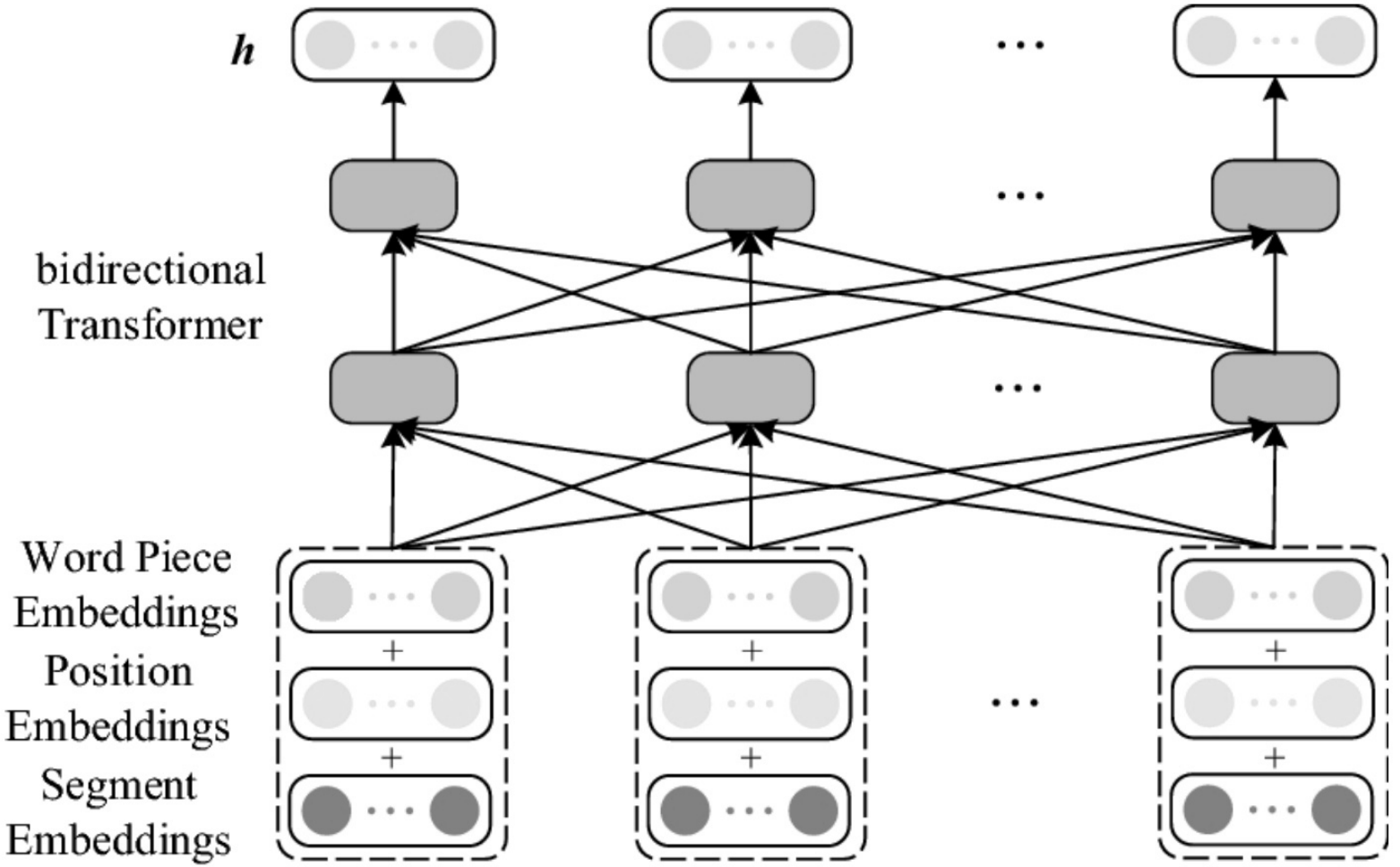
- query vector $Q(s(\ell))$, for $\ell \in \mathcal{SO}_0$, can be thought of performing a role analogous to the *semantic probes* discussed before
- think of queries as elements $q \in \mathcal{S}^\vee$ dual vector space $\mathcal{S}^\vee = \mathrm{Hom}(\mathcal{S}, \mathbb{R})$, so query matrix in

$$\mathcal{S}^\vee \otimes \mathbb{R}^m \simeq \mathcal{S}^\vee \otimes \mathcal{S}' = \mathrm{Hom}(\mathcal{S}, \mathcal{S}')$$

m-fold probe Q evaluated on the given semantic vector $s(\ell)$

- similarly key vector $K(s(\ell))$, for $\ell \in \mathcal{SO}_0$, in $K \in \mathrm{Hom}(\mathcal{S}, \mathcal{S}')$, creating an m-fold probe out of the given vector $s(\ell)$
- dual role of $\mathcal{S}'$: (m-fold) probes to be evaluated on input semantic vector $s(\ell)$, or new probes generated by semantic vector $s(\ell)$ (reflected in terminology "query" and "key")
- values vector $V(s(\ell))$ representation of semantic vectors $s(\ell)$ in a vector space $\mathcal{S}''$ dimension lower than $\mathcal{S}$ ($d = \dim \mathcal{S}''$ embedding dimension)

- a set $L \subset \mathcal{SO}_0$: usually seen as an ordered set, but *in fact it should not be* (can use bi-directional architectures like BERT)



Neural network architecture of BERT. The input word piece, position and segment embeddings are summed

- to an element $\ell \in L$ assign attention operator $A_\ell : L \subset \mathcal{S} \to \mathcal{S}'$ given by

$$A_\ell(s(\ell')) = \sigma(\langle Q(s(\ell)), K(s(\ell'))\rangle)$$

where σ softmax

$$\sigma(x)_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}, \quad \text{for } x = (x_i)$$

- Note: ignoring usual rescaling by $\sqrt{d}$, no influence on algebraic structure
- $A_{\ell,\ell'} := A_\ell(s(\ell'))$ attention matrix

- $A_{\ell,\ell'}$ a probability measure on how attention from position ℓ is distributed towards other positions ℓ' in the set L
- assign an output (in $\mathcal{S}''$) to input $s(L) \subset \mathcal{S}$, as vectors

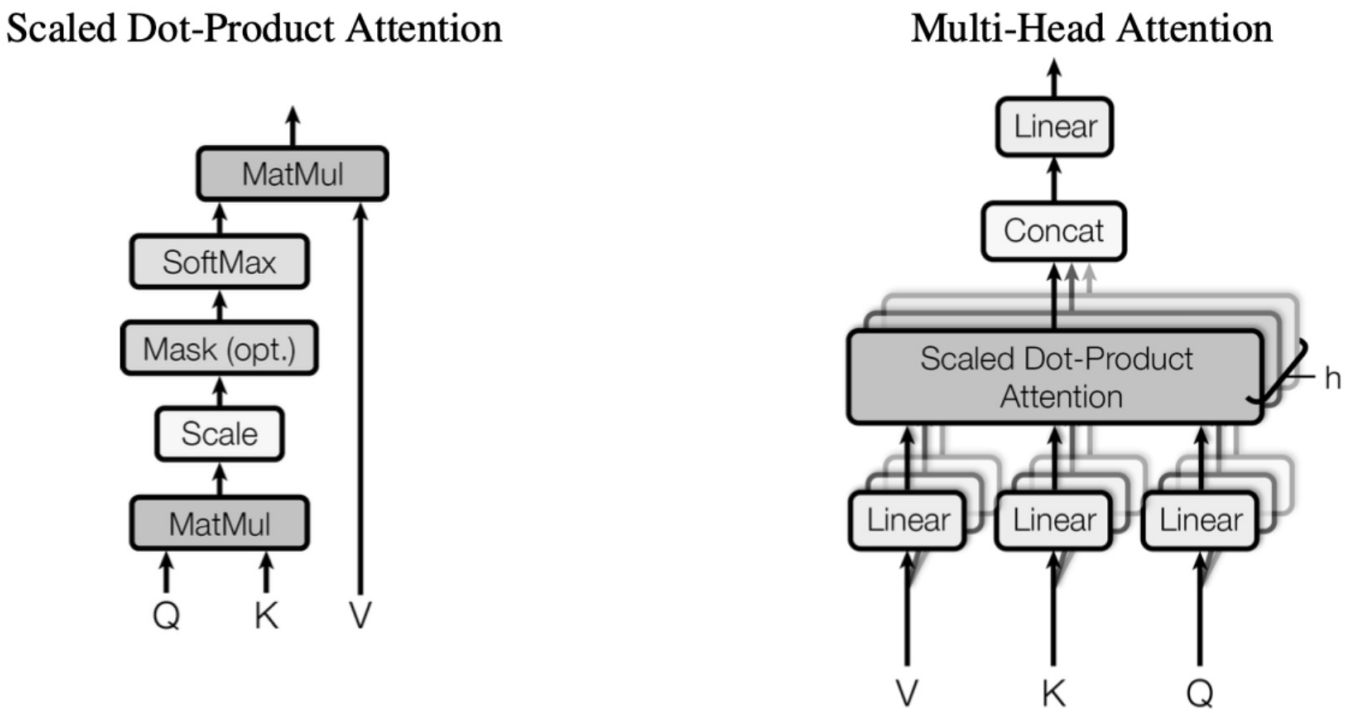
$$y_\ell = \sum_{\ell'} A_{\ell,\ell'} V(s(\ell'))$$

for each $\ell \in L$, have

$$y_\ell = (y_\ell)_{i=1}^d \in \mathcal{S}'' \simeq \mathbb{R}^d$$

- matrix representation $A_{\ell,\ell'}$ uses ordering of $\ell \in L$ but underlying linear operator does not; resulting y_ℓ also symmetric in ordering

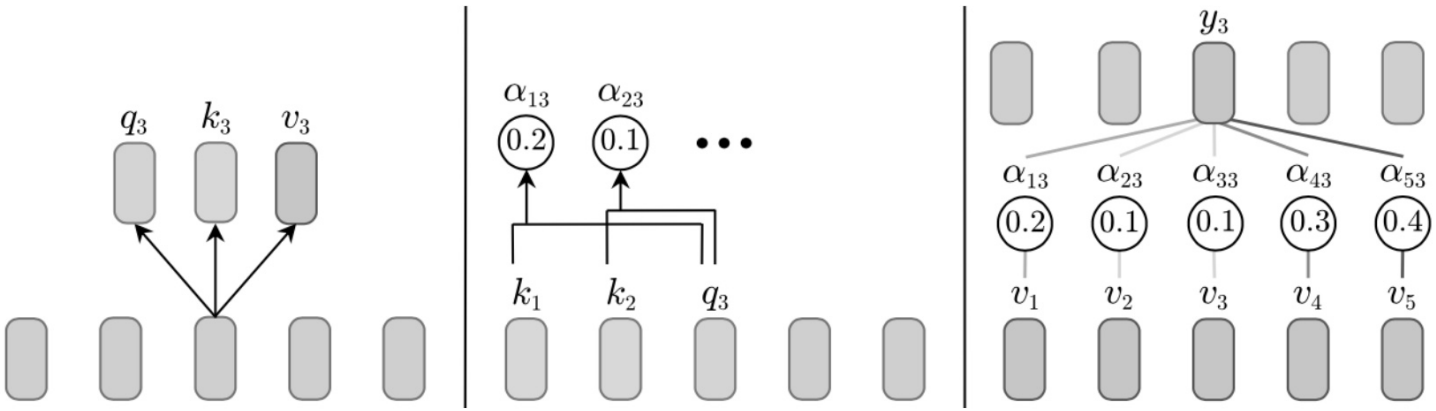
- usually several such attention modules running in parallel: multi-head attention



- vectors $Q(s(\ell)) = \oplus_i Q(s(\ell))_i$, $K(s(\ell)) = \oplus_i K(s(\ell))_i$, and $V(s(\ell)) = \oplus_j V(s(\ell))_j$ are split into blocks of decomposition $\mathcal{S}' = \oplus_{i=1}^{N} \mathcal{S}'_i$
- attention matrices, for $i = 1, \ldots, N$,

$$A^{(i)}_{\ell,\ell'} = \sigma(\langle Q(s(\ell))_i, K(s(\ell))_i \rangle_{\mathcal{S}'_i})$$

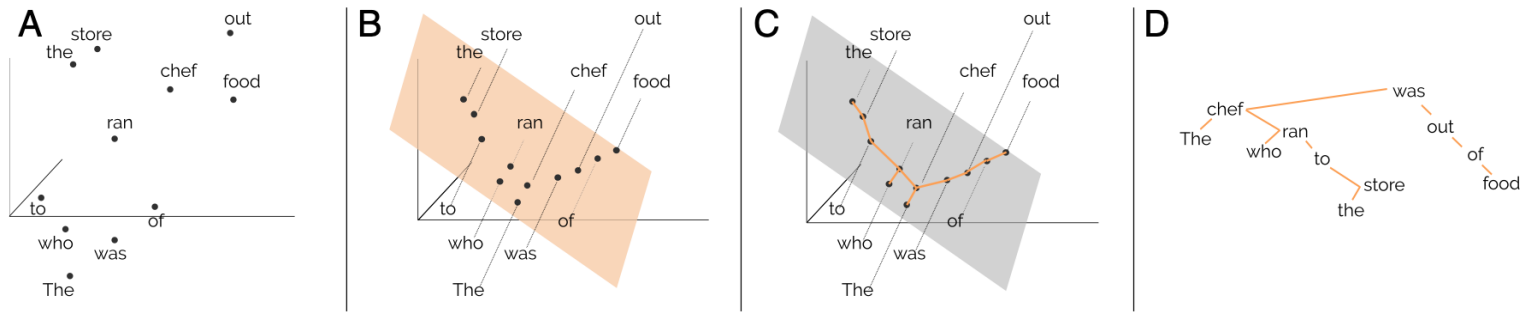
  attention distribution with *attention head i*
- not consider here multiple attention heads: enough to use a single one to see the conceptual structure



queries, keys, values from input semantic vectors, attention matrices, probabilities, and weighted output

What is actually happening?

- conflicting results on handling of syntax by LLMs when syntactic structures become complex
- syntactic trees can be "seen" from the weights of attention modules (Mannings et al.2020):



- "poverty of the stimuls" for human learning versus "overwhelming richness of the stimulus" for LLM training

Some References: LLMs detection/handling of syntax

- C.D.Manninga, K.Clarka, J.Hewitta, U.Khandelwala, O.Levy, *Emergent linguistic structure in artificial neural networks trained by self-supervision*, PNAS, 117 (2020) N.48, 30046–30054.
- V.Dentella, F.Güntherb, E.Leivada, *Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias*, PNAS, 120 (2023) N.51
- J.Sprousea, C.T. Schütze, D. Almeida, *A comparison of informal and formal acceptability judgments using a random sample from Linguistic Inquiry 2001–2010*, Lingua 134 (2013) 219—248
- H. Vazquez, A. Heuser, C. Yang, J. Kodner. *Evaluating neural language models as cognitive models of language acquisition*, GenBench23 (2023)

LLMs perform a (partial) solution of the inverse problem

- the keys and queries are a statistical proxy for the Generative Linguistics notion of syntactic relationship (c-command) and the corresponding positions (in terms of structural relations) in a syntactic tree
- very large parallel computing searching through huge corpora for an image of syntax projected upon semantics (a difficult and imperfect inverse problem)
- syntactic trees are imperfectly encoded in the weights of the attention modules and can be read from them

encoding of syntactic trees in the attention structure can be seen as another instance of mapping from syntax to a semantic space

Attention modules and Hopf algebra characters

- the attention modules of transformer architectures of LLMs fit as another example of the Hopf algebra characters of our syntax-semantics interface model
- given map function $s : \mathcal{SO}_0 \to \mathcal{S}$ of lexical items to a (vector space) model of semantics
- focus here on attention modules, in the case of self-attention in transformer architectures

attention as character

- Hopf algebra character

$$\phi_A(T) = \max_{\ell \in L(T)} A_{h(T),\ell}$$

if $T \in \mathrm{Dom}(h)$ and zero otherwise

- *syntactic relation*: collection $\rho = \rho_T$ of relations

$$\rho_T \subset L(T) \times L(T)$$

equiv with $\rho_T(\ell, \ell') = 1$ is $\ell, \ell' \in L(T)$ in the relation and $\rho_T(\ell, \ell') = 0$ otherwise

- ρ *exactly attention-detectable* if $\exists$ query/key linear maps $Q_\rho, K_\rho \in \mathrm{Hom}(\mathcal{S}, \mathcal{S}')$ and *head function* h_ρ

$$\rho_T(h_\rho(T), \ell_{\max,h_\rho}) = 1$$

for $T \in \mathrm{Dom}(h_\rho)$ with

$$\ell_{\max,h_\rho} = \mathrm{argmax}_{\ell \in L(T)} A_{h_\rho(T),\ell}$$

$A =$ attention matrix built from Q_ρ, K_ρ

- syntactic relation ρ is *approximately attention-detectable* if $\exists$ query/key linear maps $Q_\rho, K_\rho \in \mathrm{Hom}(\mathcal{S}, \mathcal{S}')$ and *head function* h_ρ

$$\frac{1}{\#\mathcal{D}} \sum_{T \in \mathcal{D}} \rho(h_\rho(T), \ell_{\max, h_\rho}) \sim 1$$

  for some sufficiently large set $\mathcal{D} \subset \mathrm{Dom}(h_\rho)$ of trees
- existence of query/key linear maps Q_ρ, K_ρ is relative to specified context (a corpus, a dataset, etc)

- threshold Rota-Baxter operator c_λ

$$\phi_{A,-}(T) = c_\lambda(\max\{\phi_A(T), c_\lambda(\phi_A(F_{\underline{v}}))\cdot\phi_A(T/F_{\underline{v}}),\dots, c_\lambda(\phi_A(F_{\underline{v_N}}))\cdot\phi_A(F_{\underline{v}_{N-1}}/F_{\underline{v_N}})\cdots\phi_A(T/F_{\underline{v}_1})\}).$$

- for simplicity focusing on the case of chains of subtrees $T_{v_N} \subset T_{v_{N-1}} \subset \cdots \subset T_{v_1} \subset T$

- use quotient given by contraction $h(T/T_v) = h(T)$ so that

$$\max_{\ell\in L(T/T_v)} A_{h(T),\ell} \leq \max_{\ell\in L(T)} A_{h(T),\ell}$$

attention along syntactic substructures

- $\phi_{-}(T)$ identifies chains of accessible terms of T for which
  1. all values
  $$\phi_A(T_{v_i}) = \max_{\ell \in L(T_{v_i})} A_{h(T_{v_i}),\ell}$$
  are above threshold λ
  2. all the quotients $T_{v_{i-1}}/T_{v_i}$ have
  $$\phi_A(T_{v_{i-1}}/T_{v_i}) = \max_{\ell \in L(T_{v_{i-1}}/T_{v_i})} A_{h(T_{v_{i-1}}),\ell} =$$
  $$\max_{\ell \in L(T_{v_{i-1}})} A_{h(T_{v_{i-1}}),\ell} = \phi_A(T_{v_{i-1}})$$
- tracking where attention concentrates over substructures: first condition max attention from the *head* of each subtree sufficiently large; second guarantees that when considering next nested subtree trying to maximize its attention value, one does not spoil optimizations achieved at previous steps for larger subtrees

incorporating syntactic relations as character

- syntactic relation ρ Boolean valued $\mathcal{B} = (\{0,1\}, \max, \cdot)$

$$\phi_\rho(T) = \max_{\ell \in L(T)} \rho(h(T), \ell)$$

detects whether ρ is realized in T or not

- can combine characters, values in Viterbi $\mathcal{P} = ([0,1], \max, \cdot)$
(commonly used in NLP for probabilistic values)

$$\phi_{A,\rho}(T) = \max_{\ell \in L(T)} \rho(h(T), \ell) \cdot A_{h(T),\ell}$$

- here one maximizes attention from syntactc *head* over set of $\ell \in L(T)$ that *already satisfy* syntactic relation with the *head*
- Birkhoff factorization as before but subtrees with $\phi_\rho(T_v) = 0$ do not contribute even if their $\max_\ell A_{h(T),\ell}$ is large
- comparison between ϕ_A and $\phi_{\rho,A}$ identifies attention-detectability of ρ
- if detectability fails, identifies where in substructures attention matrix maximum happens outside of where the syntactic relation holds

To what extent do LLMs solve this inverse problem?
reconstruction of the computational mechanism of syntax from its (probabilistically smeared) image inside semantics

Empirical tests of different kinds: some examples

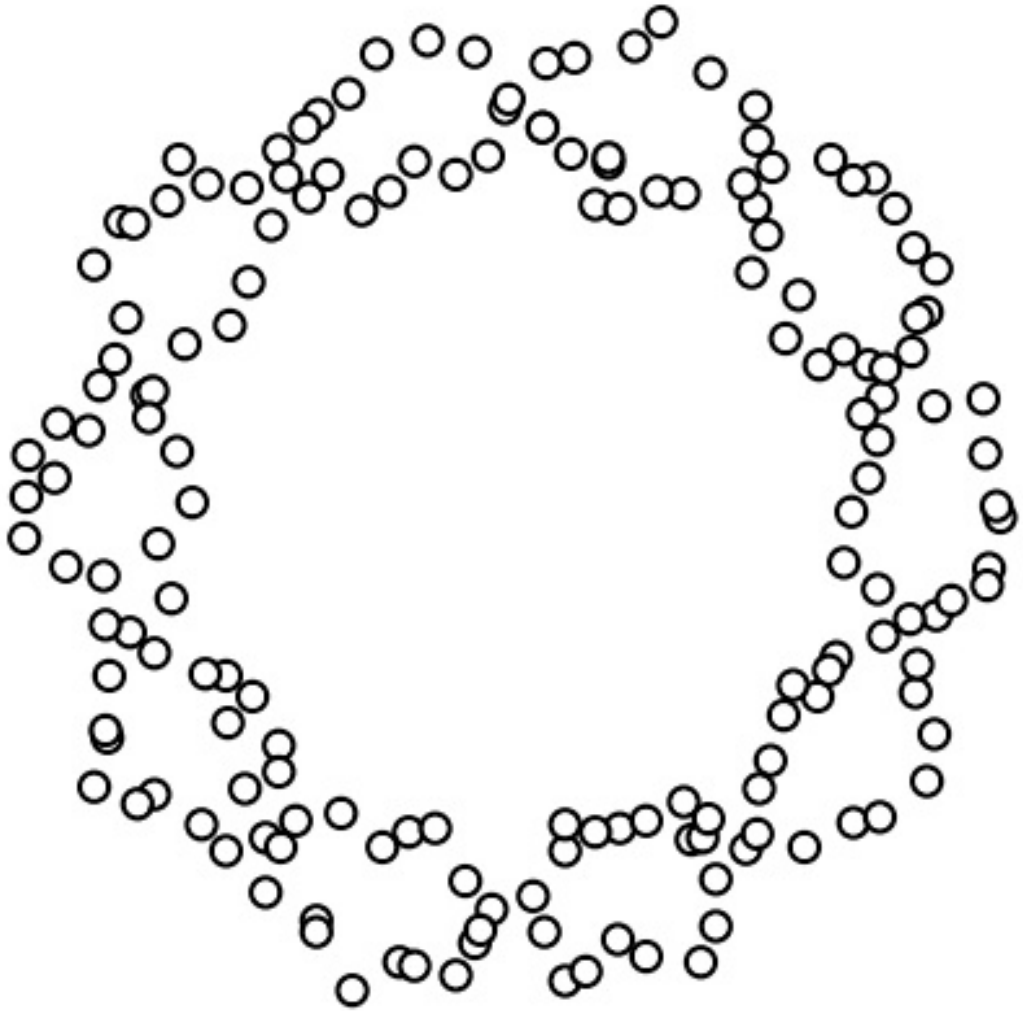
1. systematic detectable differences between text generated by AIs and by humans
2. limitations in the handling of nontrivial syntactic constructions
3. non-linguistic dependence on prompts (control theory)

Detectable difference in dimensionality between human and AI linguistic text

- Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey Nikolenko, Evgeny Burnaev, Serguei Barannikov, Irina Piontkovskaya, *Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts*, 37th Conference on Neural Information Processing Systems (NeurIPS 2023)

- testing the difference between AI and human generated linguistic texts in terms of *dimensionality*, using
- method based on *persistent homology dimension theory*

Persistent Topology of Data Sets

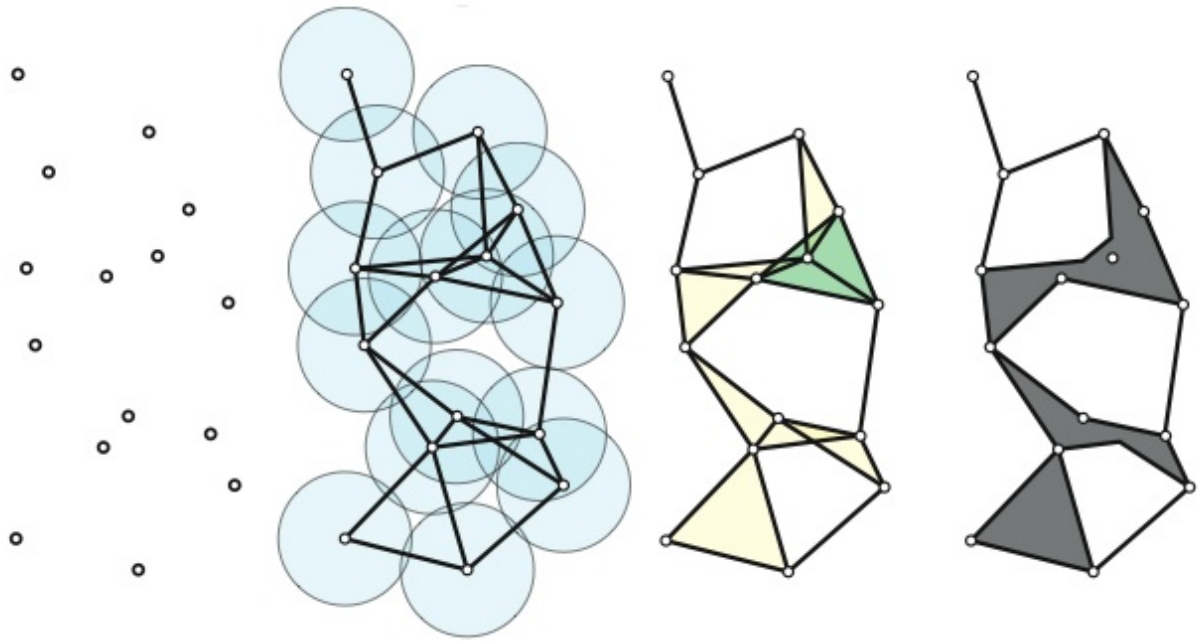


how data cluster around topological shapes at different scales

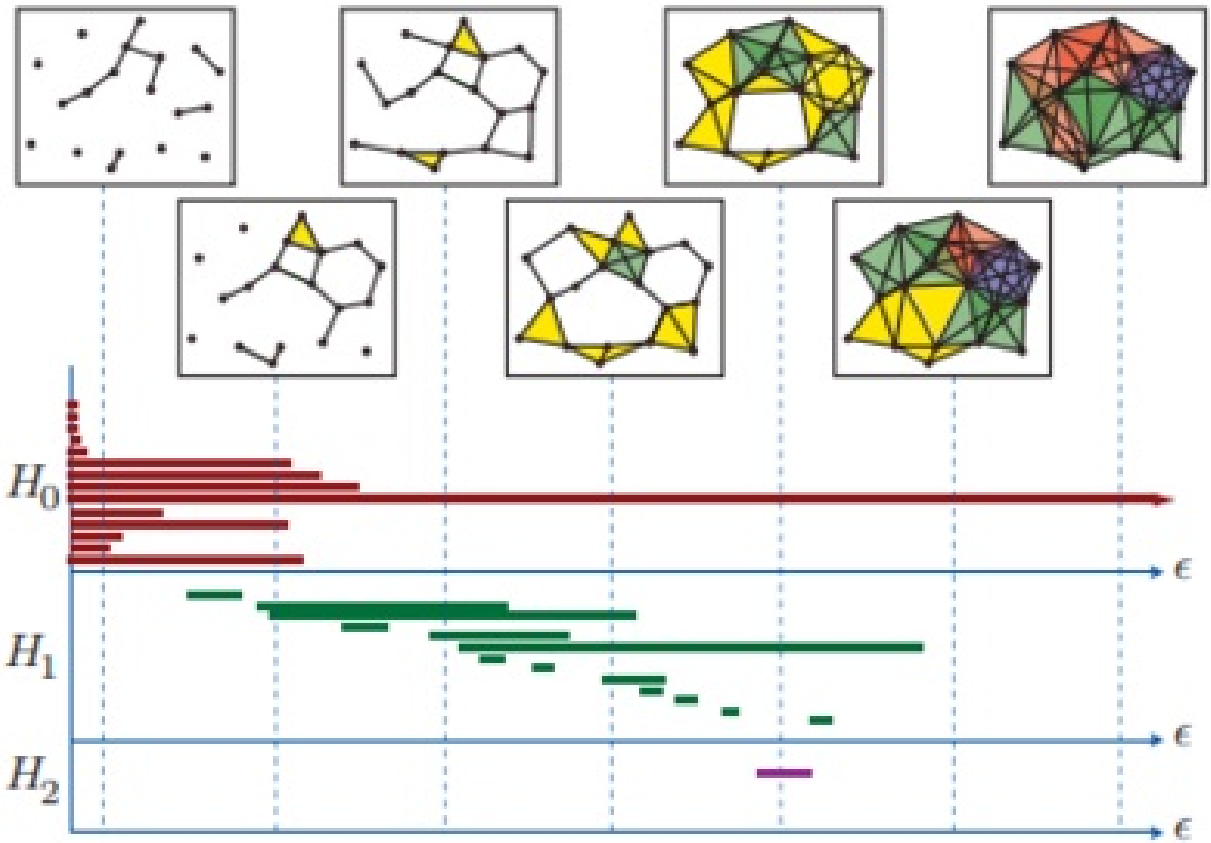
Vietoris–Rips complexes

- set $X = \{x_\alpha\}$ of points in Euclidean space $\mathbb{E}^N$, distance $d(x, y) = \|x - y\| = (\sum_{j=1}^{N}(x_j - y_j)^2)^{1/2}$
- Vietoris-Rips complex $R(X, \epsilon)$ of scale ϵ over field $\mathbb{K}$:

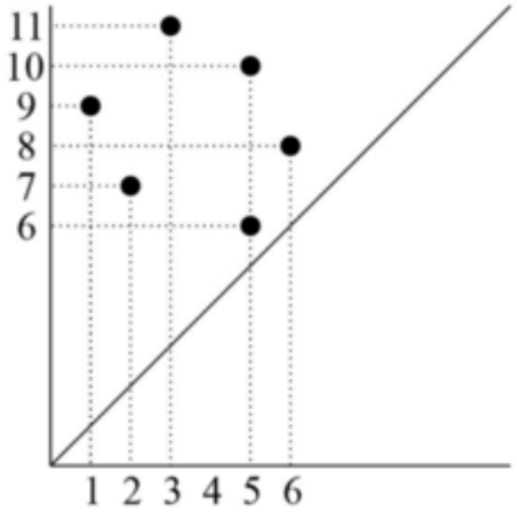
$R_n(X, \epsilon)$ is $\mathbb{K}$-vector space spanned by all unordered $(n+1)$-tuples of points $\{x_{\alpha_0}, x_{\alpha_1}, \ldots, x_{\alpha_n}\}$ in X where all pairs have distances $d(x_{\alpha_i}, x_{\alpha_j}) \leq \epsilon$

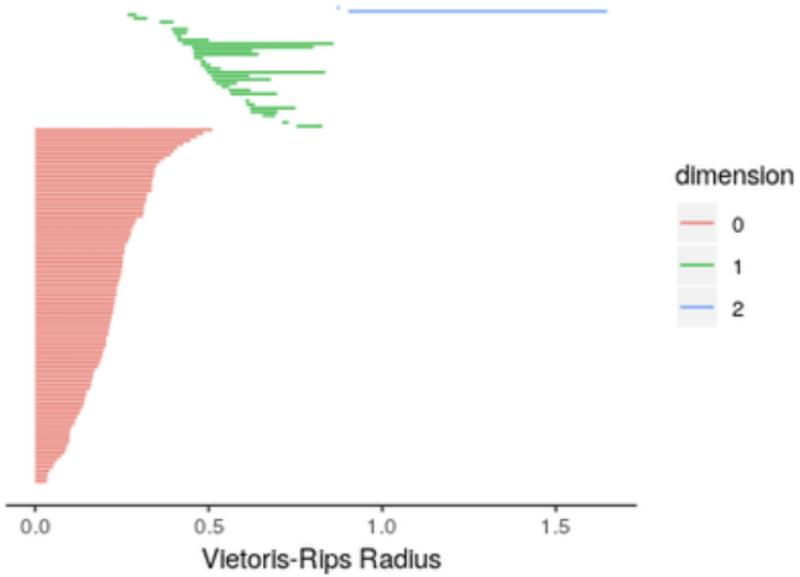


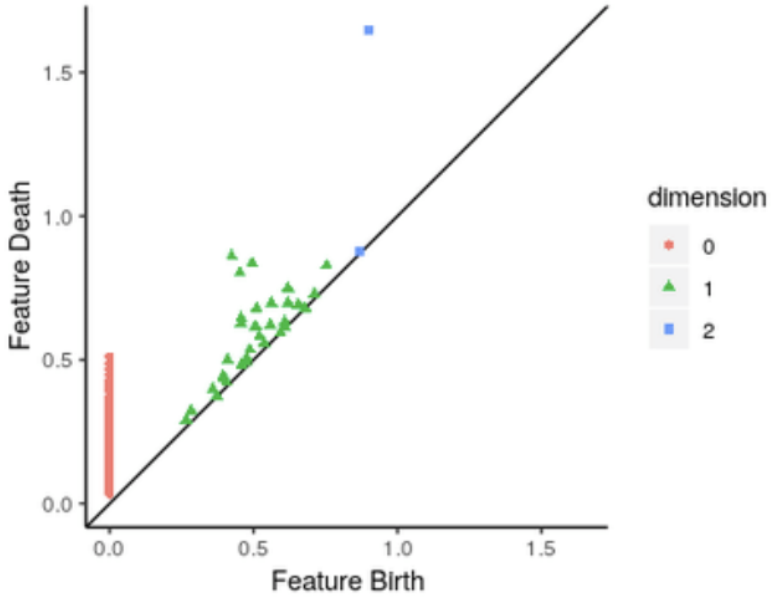
• inclusion maps $R(X, \epsilon_1) \hookrightarrow R(X, \epsilon_2)$ for $\epsilon_1 < \epsilon_2$ induce maps in homology by functoriality $H_n(X, \epsilon_1) \to H_n(X, \epsilon_2)$



barcode diagrams: births and deaths of persistent generators







other equivalent way of writing persistent homology diagrams

persistent homology dimension estimator (PHD)

- d-dimensional Riemannian manifold M, ball of radius r in M, volume grows like r^d
- points uniformly sampled from M (volume form distribution): number of points on ball also grows like r^d
- PHD combines local and global properties of dataset, also stable under noise in data
- set of points $X = \{x_1, \ldots, x_N\} \subset \mathbb{R}^n$, parameter $\alpha > 0$: weighted sum

$$E^i_\alpha(X) = \sum_{\gamma \in PH_i(X)} |\mathcal{I}(\gamma)|^\alpha$$

$\mathcal{I}(\gamma) = t_{\text{death}}(\gamma) - t_{\text{birth}}(\gamma)$ lifespan of persistence of the persistent homology generator γ

- for $i = 0$ persistent connected components (minimal spanning tree:$|e|$ = length of edges of tree)

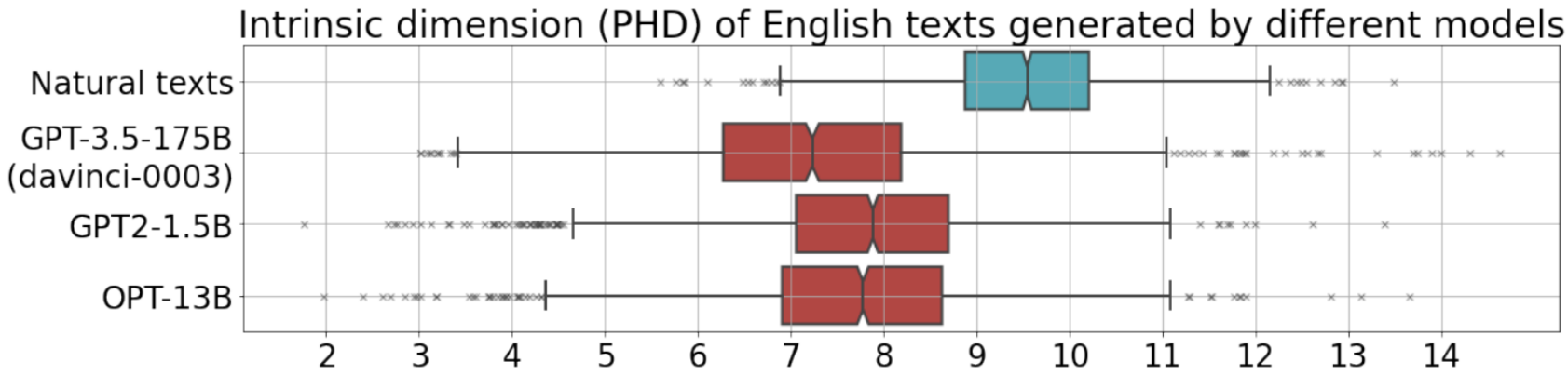
$$E^0_\alpha(X) = \sum_{\gamma \in MST(X)} |e|^\alpha$$

- *growth rate*: for $E^0_\alpha(X) \sim N^{\frac{d-\alpha}{d}}$ for $N \to \infty$ (if x_i indep random var with distrib w/ comp support)

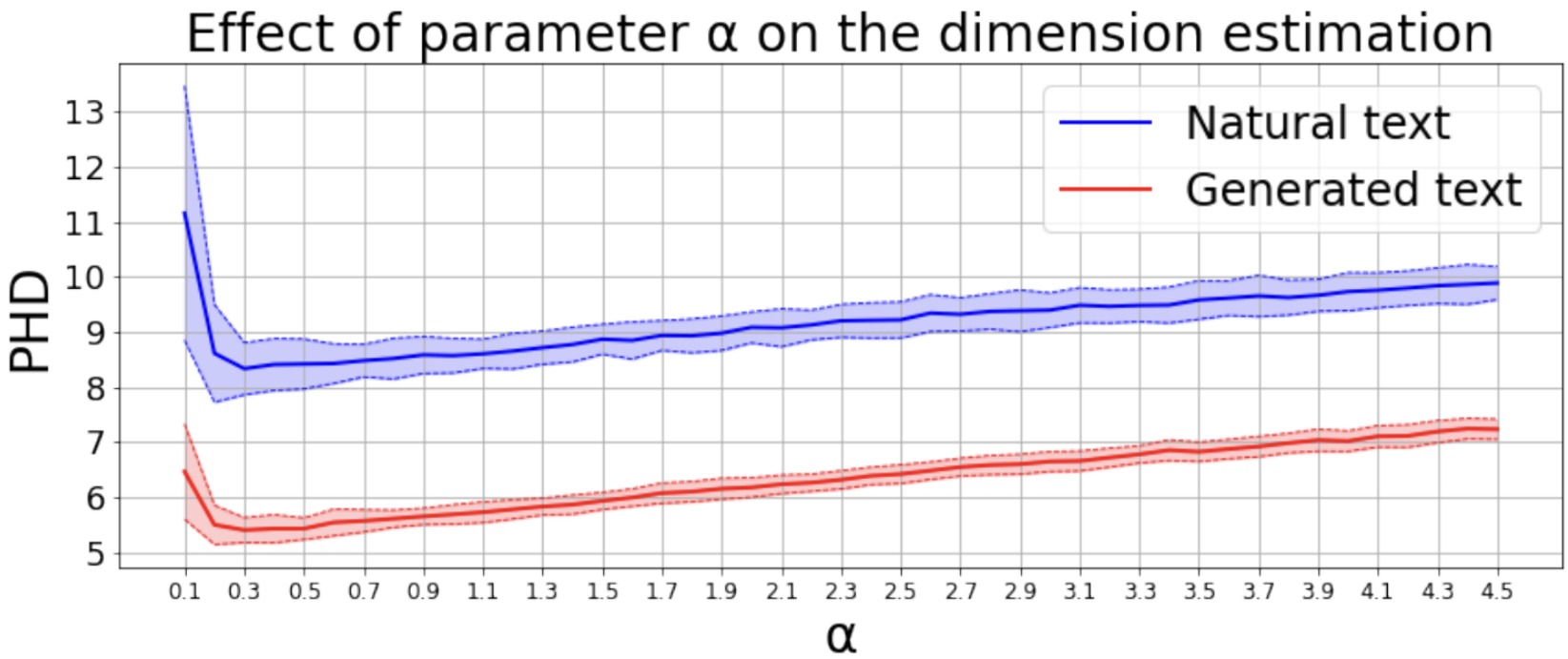
$$\dim_{MST}(M) = \inf\{d \,|\, \exists C > 0 \,:\, E^0_d(X) \leq C\,, \forall X \subset M\}$$

$$\log E^0_\alpha(X) \sim (1 - \frac{\alpha}{d}) \log \#X + \tilde{C} \qquad \text{for } \#X \to \infty$$

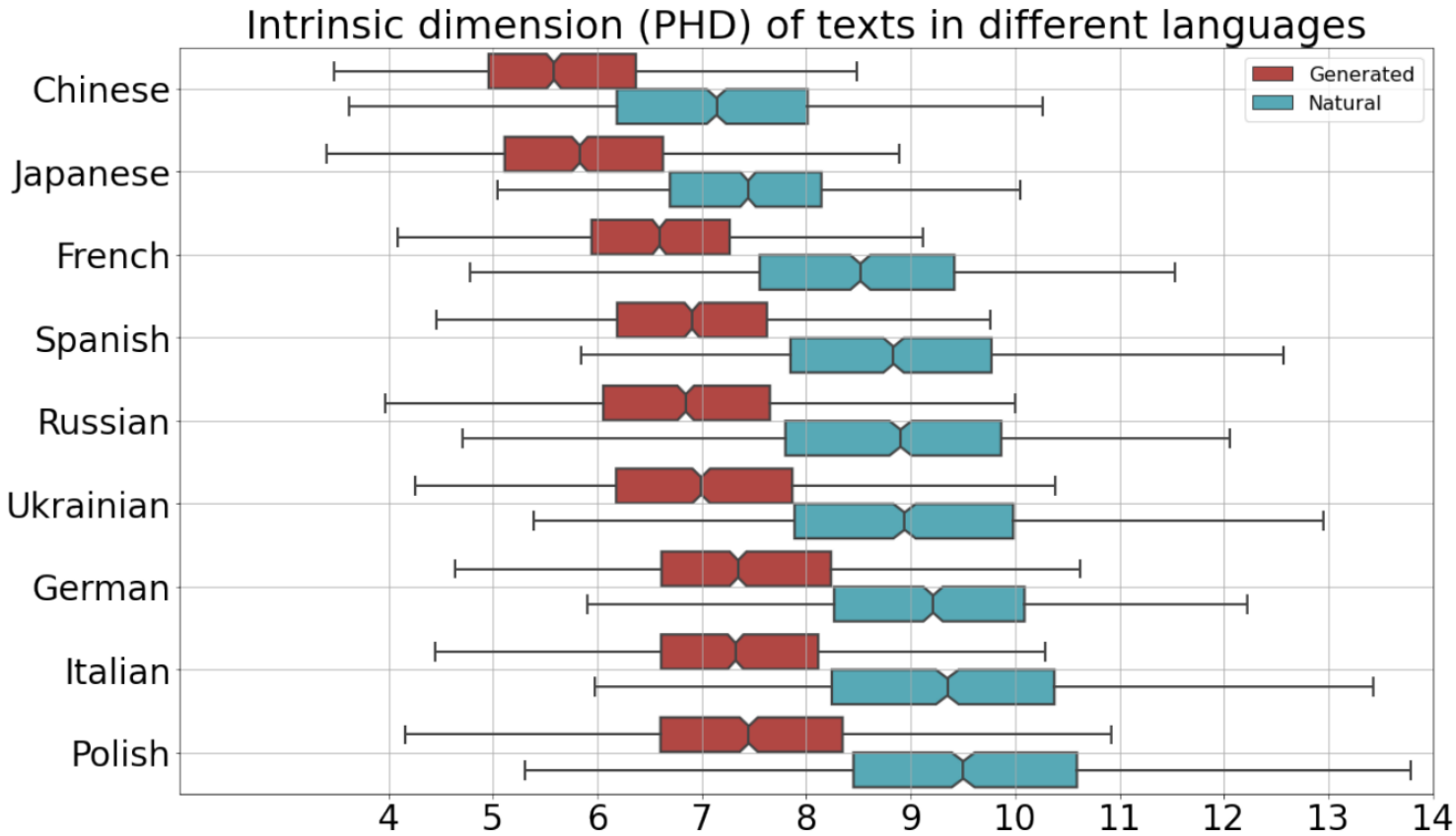
- on text of medium size $\sim$ 300 tokens; contextualized embeddings for every token in a pretrained transformer encoder; view resulting vectors as points of ambient Euclidean space; persistent homology of resulting point cloud
- significant and systemic gap in dimension between human text and AI generated text
- misclassified cases (either way) tend to happen on short texts
- What does it capture? it is related to the positions of embedding vectors: does it reflect hierarchical structures of syntax relating them? (note: tree of persistent components likely correlates to parse tree)



comparative test of persistent homology dimension for human and AI generated text, E.Tulchinskii, et al. *Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts*, NeurIPS 2023



effect of parameter α in persistent homology dimension estimation, E.Tulchinskii, et al. *Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts*, NeurIPS 2023



comparative effect across languages: persistent homology dimension gap, E.Tulchinskii, et al. *Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts*, NeurIPS 2023

Handling of nontrivial syntactic constructions

- LI-Adger database of syntactic examples collected from the theoretical linguistics journal *Linguistic Inquiry* 2001–2010
- covering broad range of syntactic phenomena
- two different tests: acceptability and grammaticality

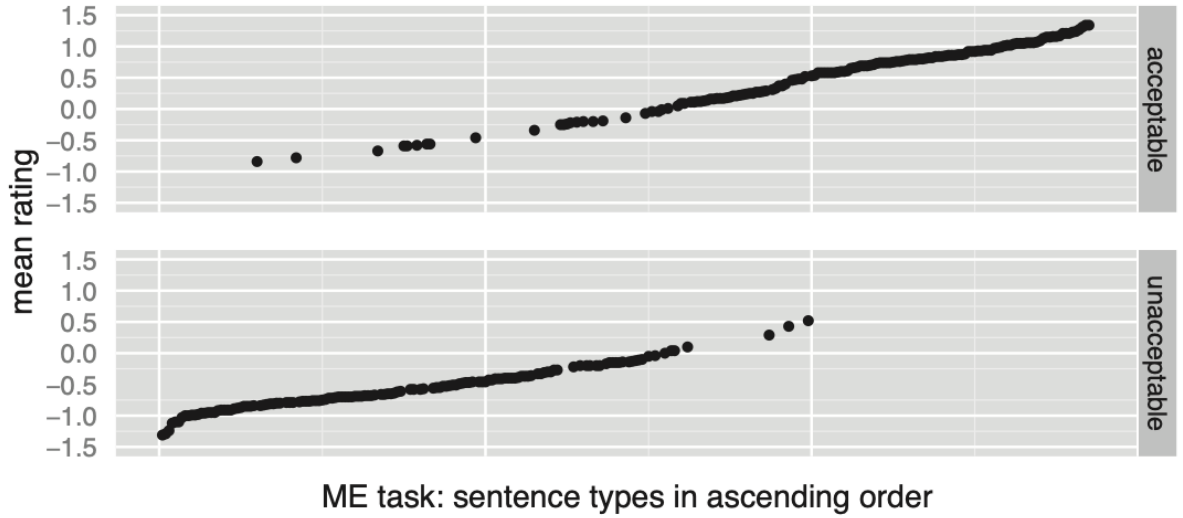
- J. Sprousea, C.T. Schütze, D. Almeida, *A comparison of informal and formal acceptability judgments using a random sample from Linguistic Inquiry 2001–2010*, Lingua, 134 (2013) 219-248
- V.Dentella, F.Gunther, E.Leivada, "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", PNAS 2023
- H.J.Vazquez Martinez, "The Acceptability Delta Criterion: Testing Knowledge of Language using the Gradience of Sentence Acceptability", Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, pages 479-495, 2021

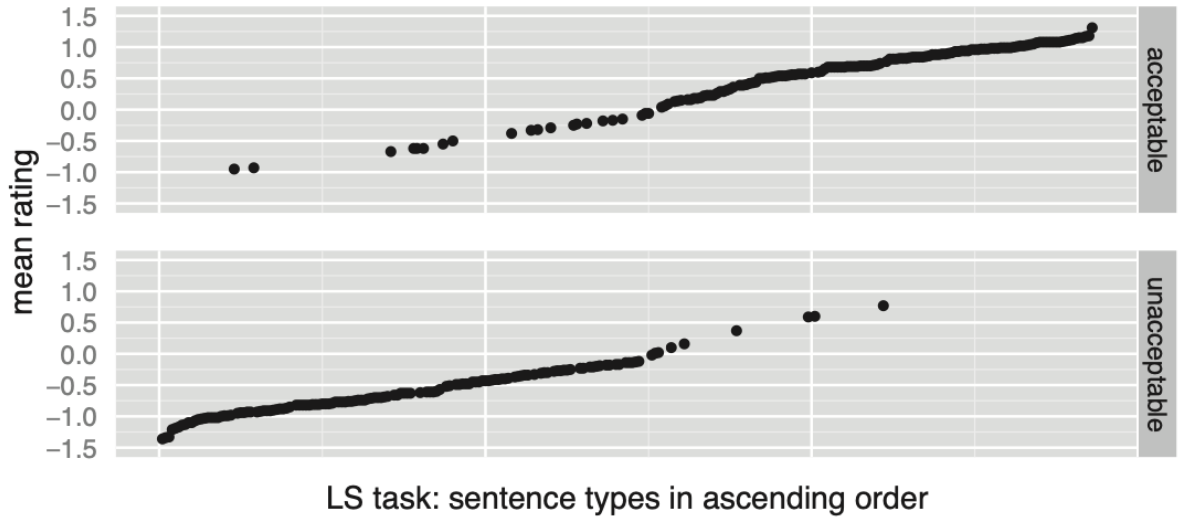
different types of acceptability judgments in the syntax literature

Standard acceptability judgments: These require only that the participant be presented with a sentence and asked to judge its acceptability on an arbitrary scale or in reference to another sentence.
Coreference judgments: These are primarily used to probe binding relationships. Participants must be presented with a sentence that includes two or more noun phrases that are identified in some way. They are then asked to indicate whether the two noun phrases can or must refer to the same entity.
Interpretation judgments: These are judgments based on the meaning of sentences, such as whether a sentence is ambiguous or unambiguous, or whether one quantifier has scope over another. These may require explicit training of participants to identify multiple potential meanings, and/or explicitly constructed contexts to elicit one or more potential meanings.

Judgments involving relatively few lexical items: These are acceptability judgments about phenomena that occur with relatively few lexical items, such that the construction of 8 substantially distinct tokens, as was done for the phenomena tested in this study, would likely be impossible. This is not to say that these phenomena cannot be tested in formal experiments, but participants in such experiments may require special instruction to guard against potential repetition confounds.
Judgments involving prosodic manipulations: These are acceptability judgments that are based on specific prosodic properties of the sentence. They require either the presentation of auditory materials or the use of some notational conventions for conveying the critical prosodic properties in writing (e.g., the use of capital letters to indicate emphasis).

pairwise phenomena: two maximally similar sentence types differing in a way that (1) is relevant for theories of grammar and (2) lead to a significant difference in acceptability
judgment tasks: magnitude estimation (ME), 7-point Likert scale (LS), and two-alternative forced-choice (FC)





human acceptability judgments for grammaticality over the *Linguistic Inquiry* database, consistency over testing methods

LLMs on the LI-Adger dataset

- three BERT models fine-tuned using Corpus of Linguistic Acceptability (CoLA, 2019)
- with ADC (acceptability delta criterion) both BERT and the trigram model scored approximately 30% of minimal pairs correctly
- representative collection of 4177 sentences forming 2394 unique minimal pairs from LI-Adger
- comparison with human judgement data

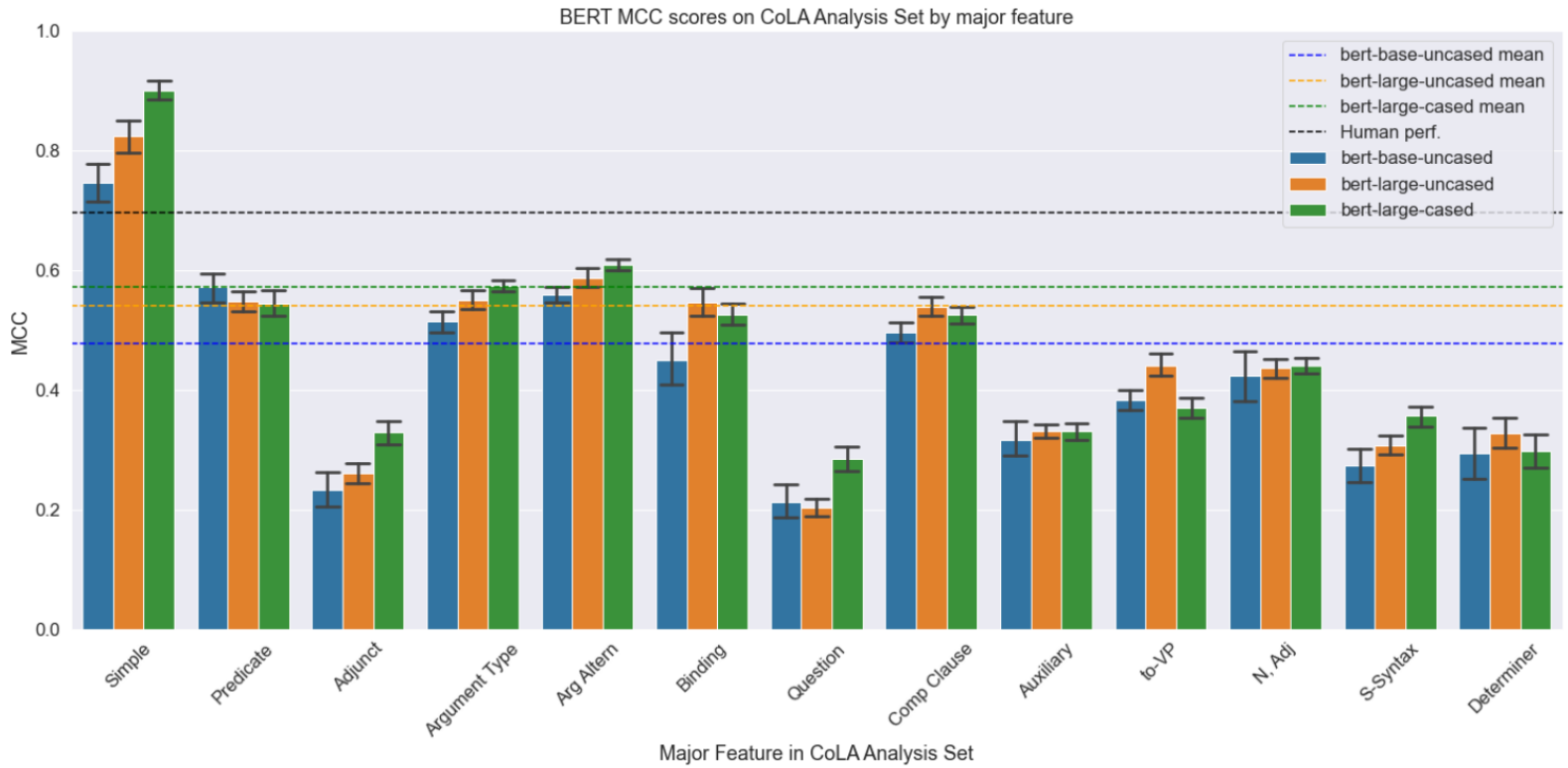
source:

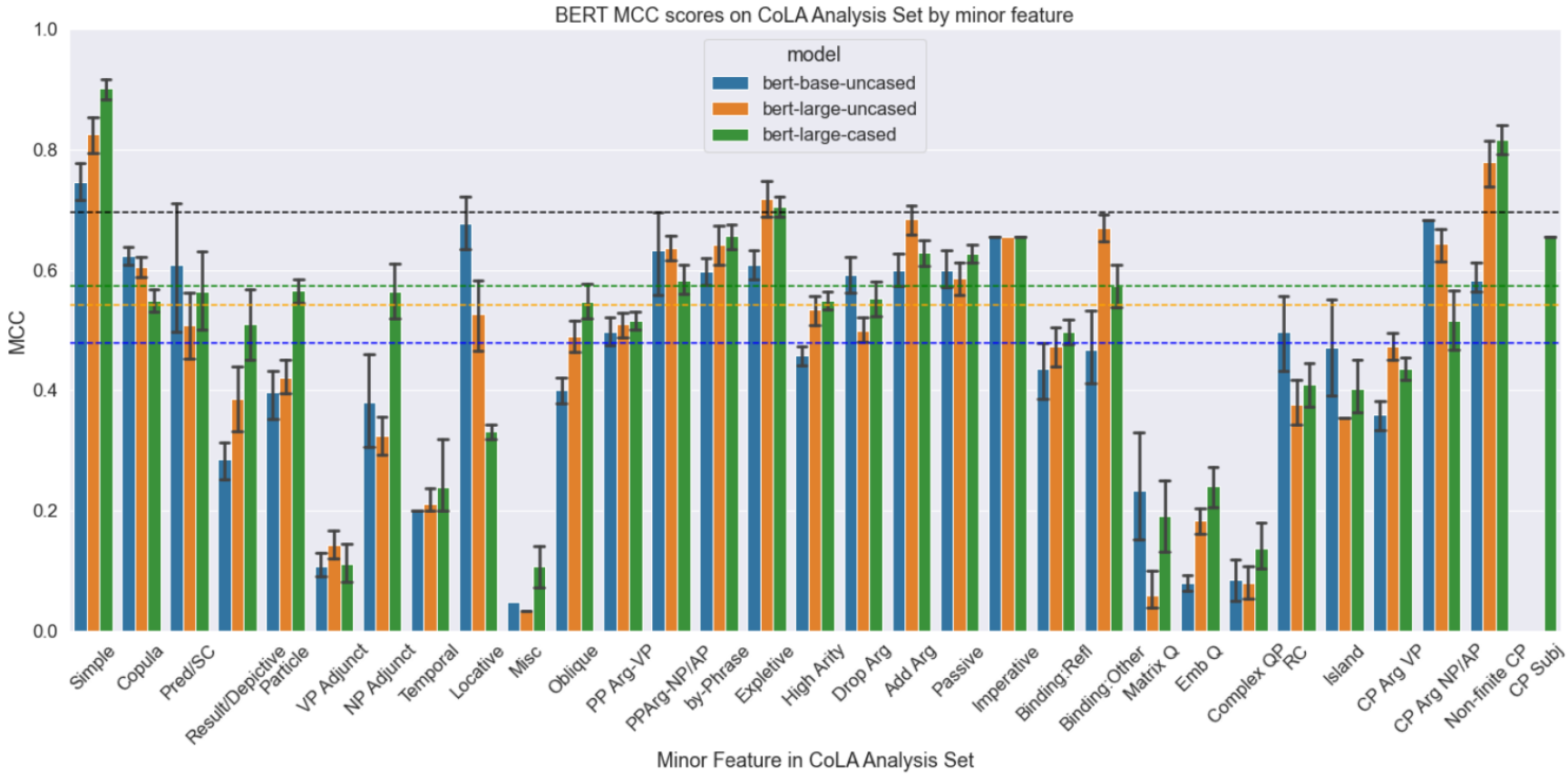
- H.J.Vazquez Martinez, "The Acceptability Delta Criterion: Testing Knowledge of Language using the Gradience of Sentence Acceptability", Fourth BlackboxNLP Workshop 2021.
- H.J.Vazquez Martinez, Master thesis CS, MIT 2021

- major syntactic feature in the CoLA analysis set: Matthew's Correlation Coefficient (MCC) scores
  - confusion matrix: true positives (TP), false positives (FP), true negatives (TN), false negatives (FN)

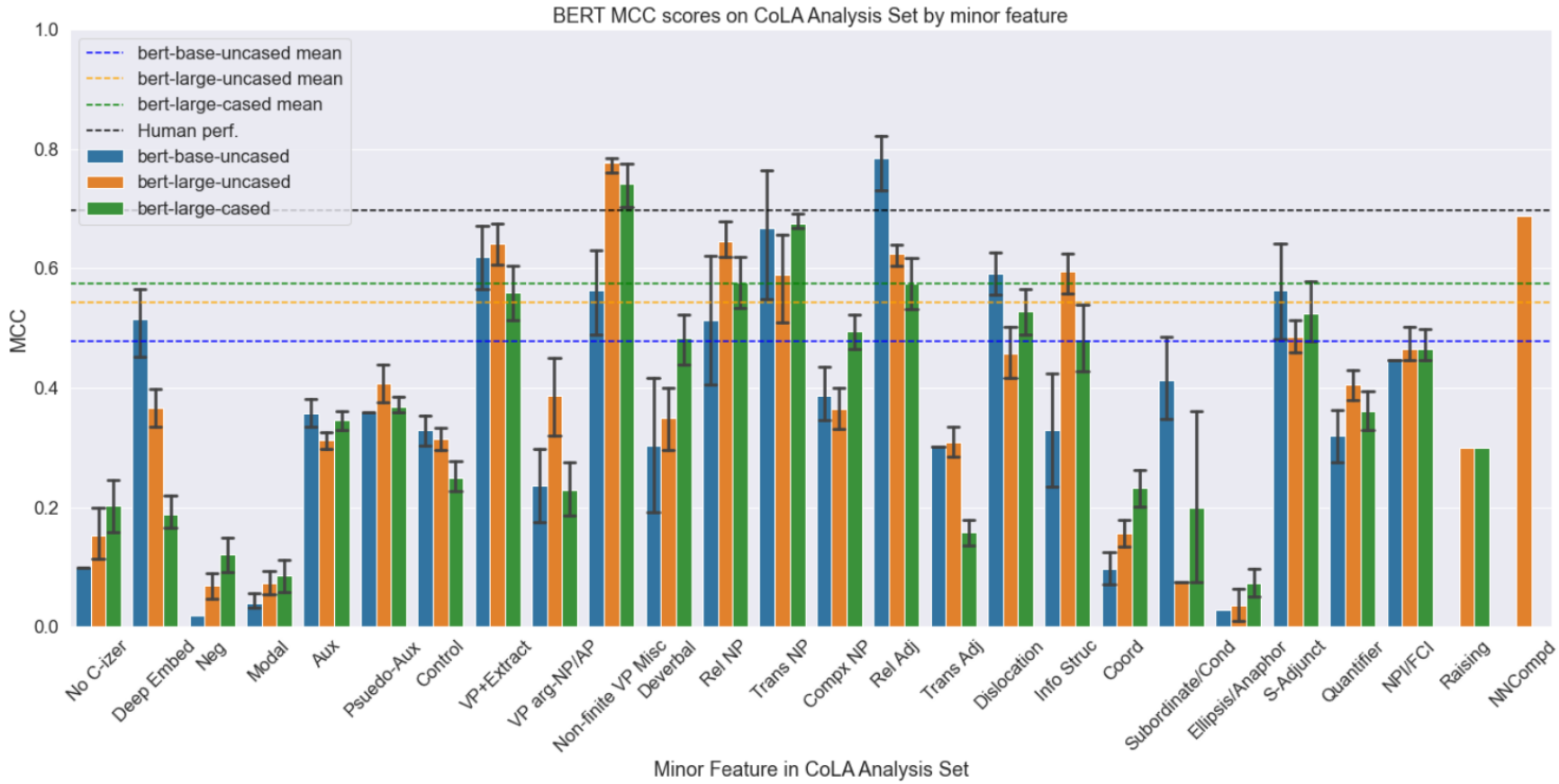
$$\mathrm{MCC} = \frac{\mathrm{TP}\cdot\mathrm{TN}-\mathrm{FP}\cdot\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})\cdot(\mathrm{TP}+\mathrm{FN})\cdot(\mathrm{TN}+\mathrm{FP})\cdot(\mathrm{TN}+\mathrm{FN})}}$$

(worst and minimum value -1; best and maximum value +1)





H.J.Vazquez Martinez, "The Acceptability Delta Criterion: Testing Knowledge of Language using the Gradience of Sentence Acceptability" 2921

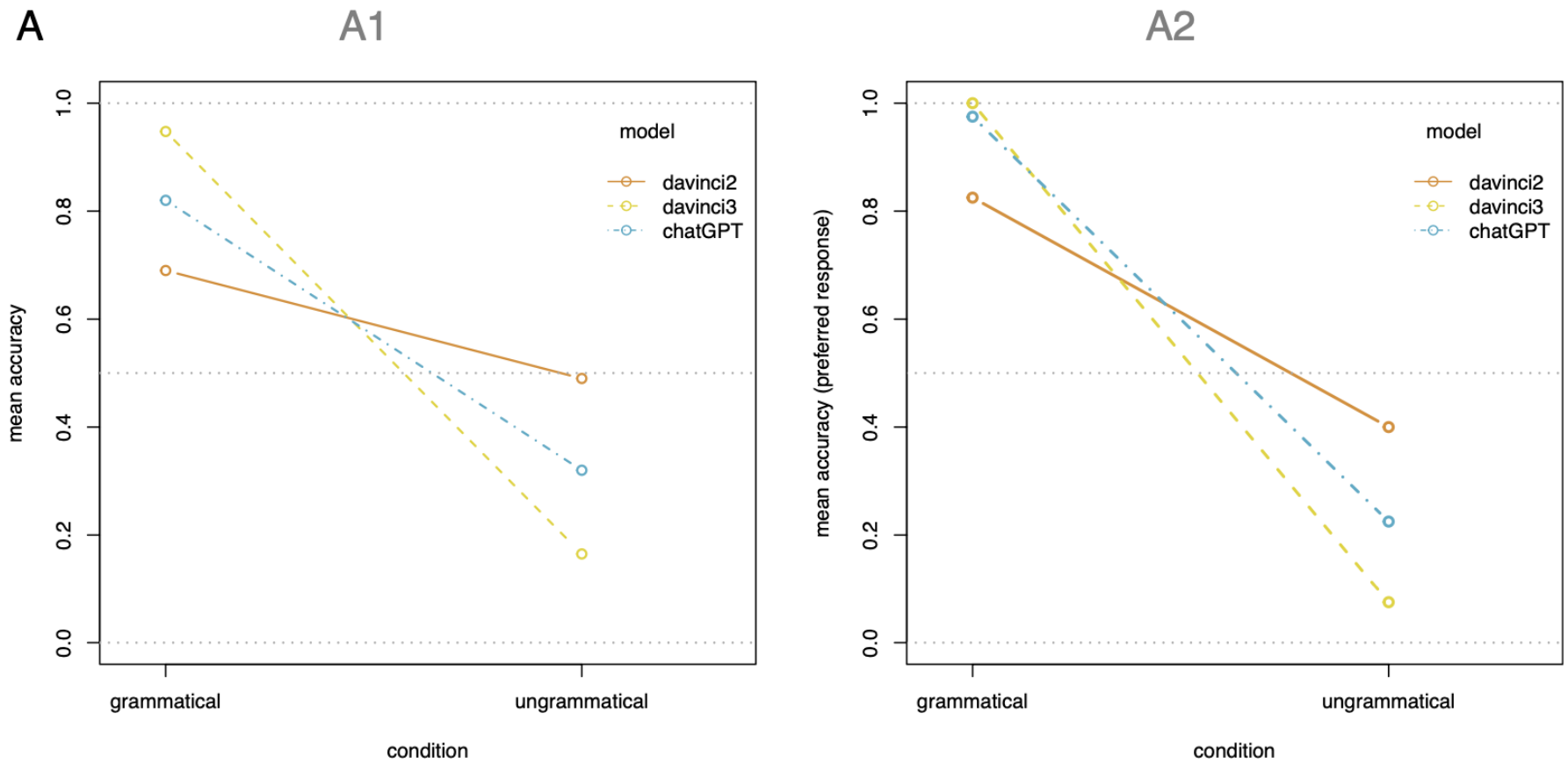


H.J.Vazquez Martinez, "The Acceptability Delta Criterion: Testing Knowledge of Language using the Gradience of Sentence Acceptability" 2921

- Acceptability Delta Criterion ($\delta = 0.5$): enforces that models' predictions be within a set number of standard deviation units δ from the human ME judgements
- then BERT only correctly evaluates 726 out of 2365 (31%) minimal pairs, whereas trigram model correctly evaluates 712 out of 2365 (30%)
- when it comes to tracking acceptability of sentences across minimal pairs, BERT does not go much farther than Shannon's *N*-gram models of the 1940s

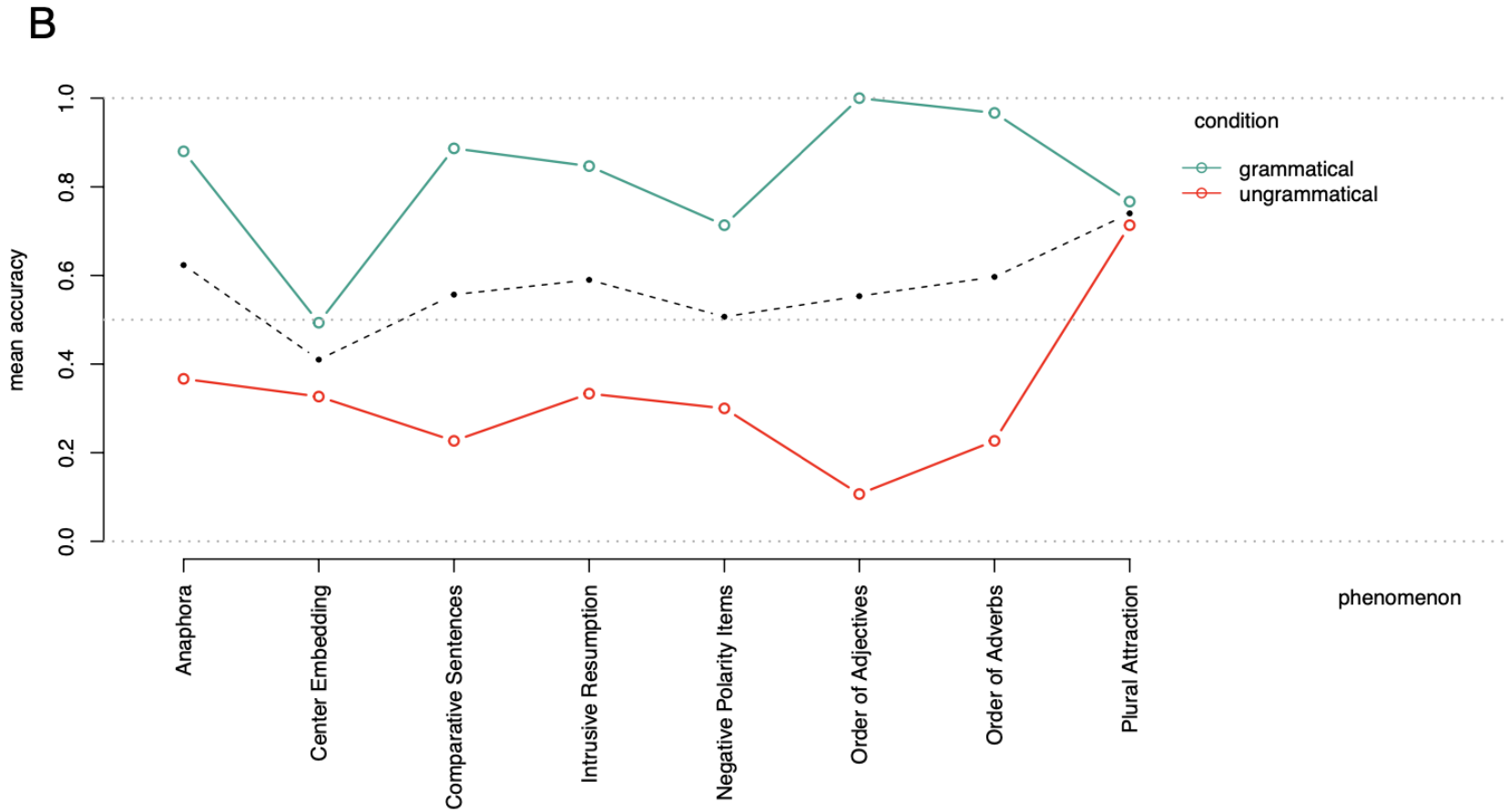
LLMs handling of syntactic phenomena (Dentella et al.)

- 8 linguistic phenomena: plural attraction; anaphora; center embedding; comparative sentences; intrusive resumption; negative polarity items; order of adjectives; and order of adverbs
- all evaluable without context
- each phenomenon 10 sentences: 5 grammatical and 5 ungrammatical (ungrammatical involve violation of one specific rule of English syntax)
- prompt used: "Is the following sentence grammatically correct in English?"
- LLMs tested: GPT-3/text-davinci-002, Nov 2022 (davinci2); GPT-3/text-davinci-003, Jan 2023 (davinci3); ChatGPT Feb 2023
- comparative judgments from humans



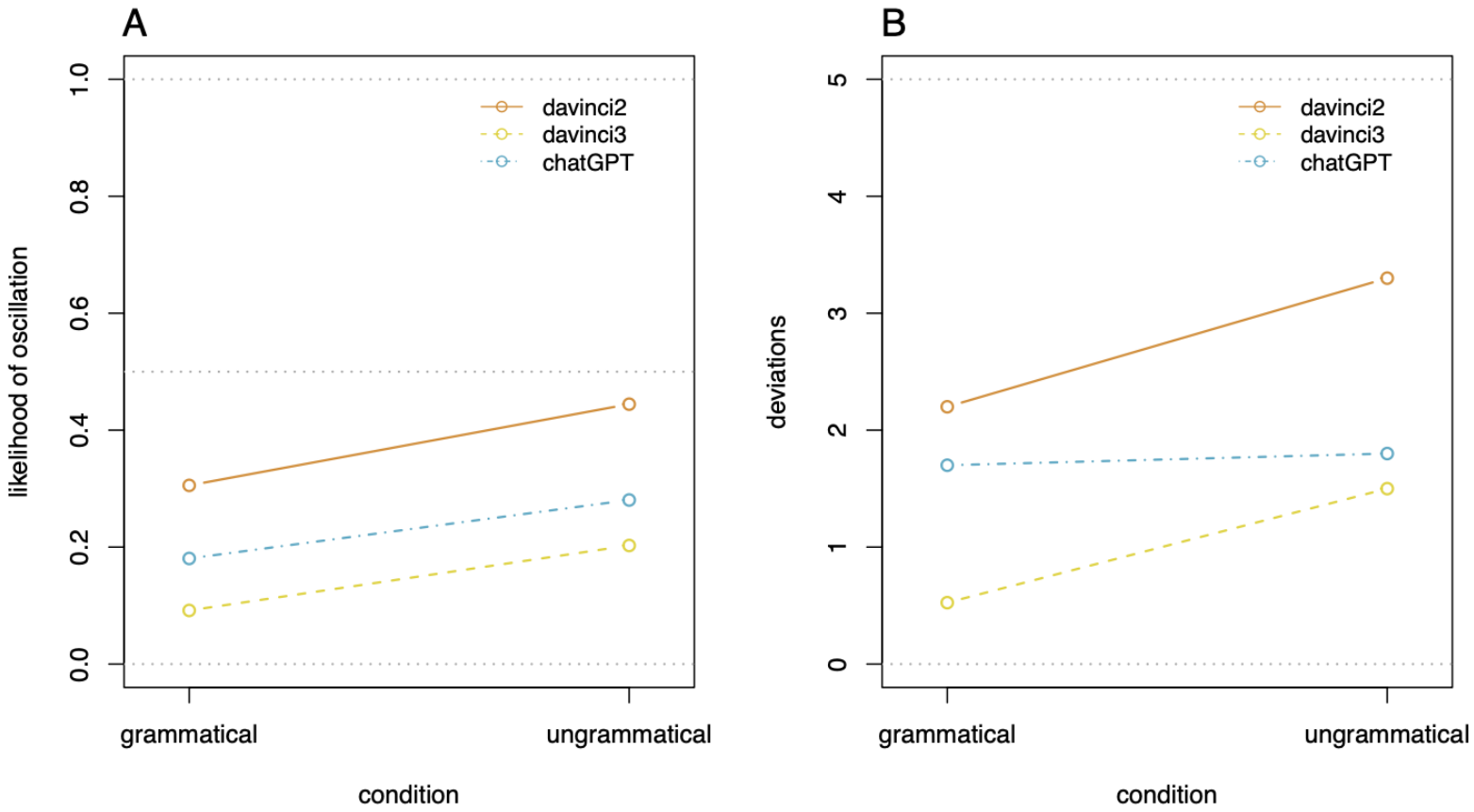
mean accuracy different LLM models

V.Dentella, et al. "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", PNAS 2023



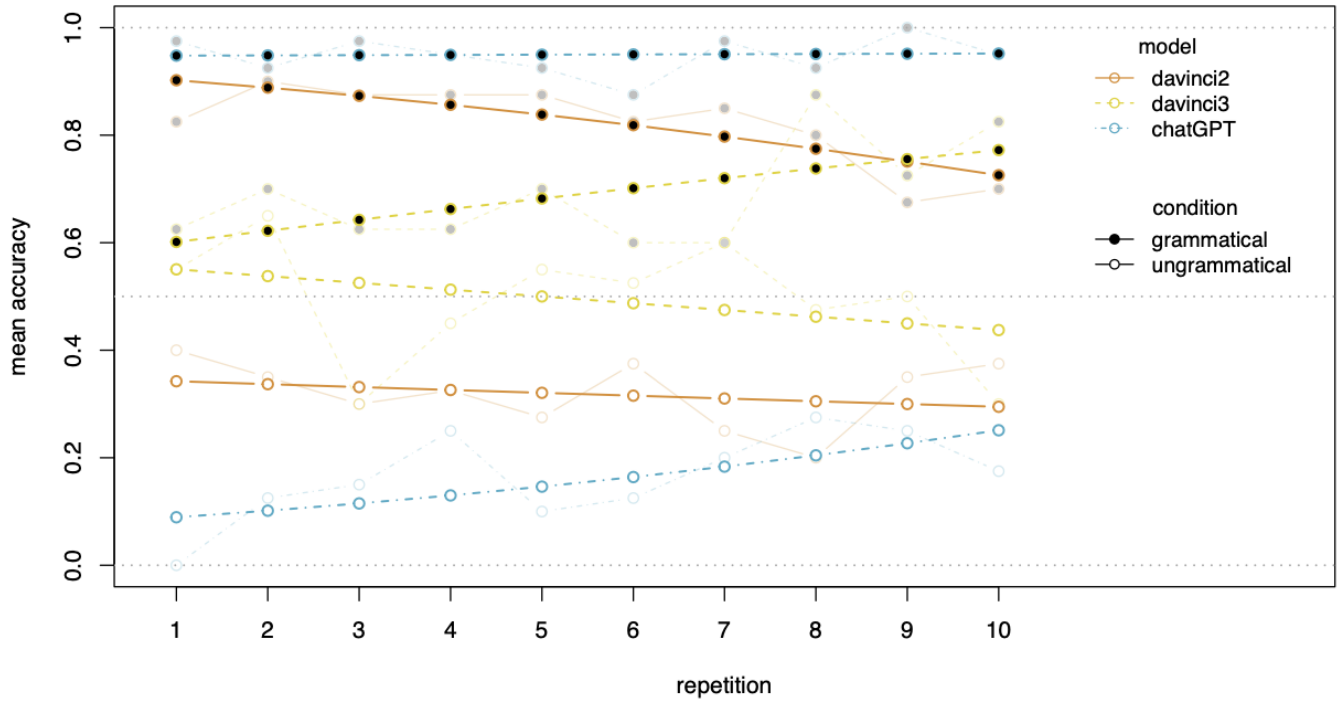
mean accuracy by syntactic phenomenon

V.Dentella, et al. "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", PNAS 2023



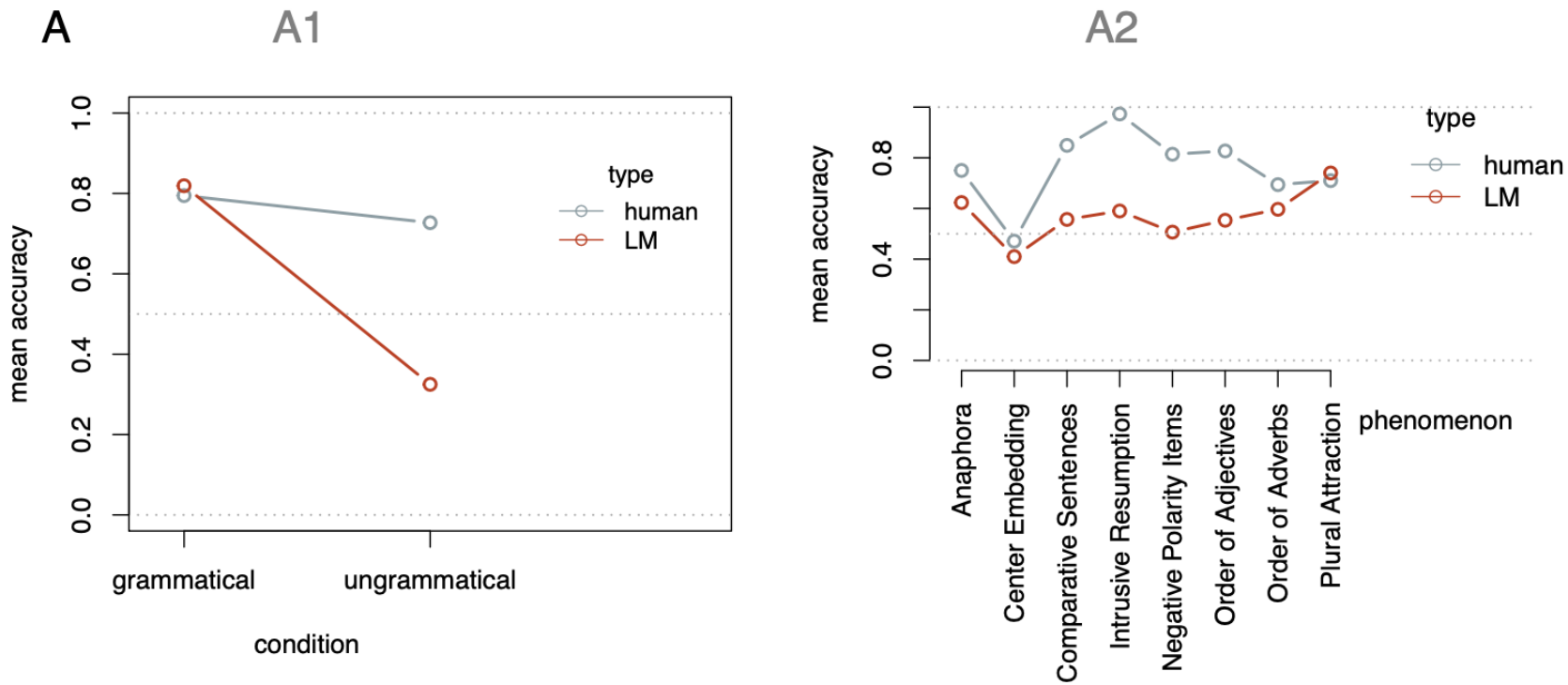
response instability, different LLM models

V.Dentella, et al. "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", PNAS 2023



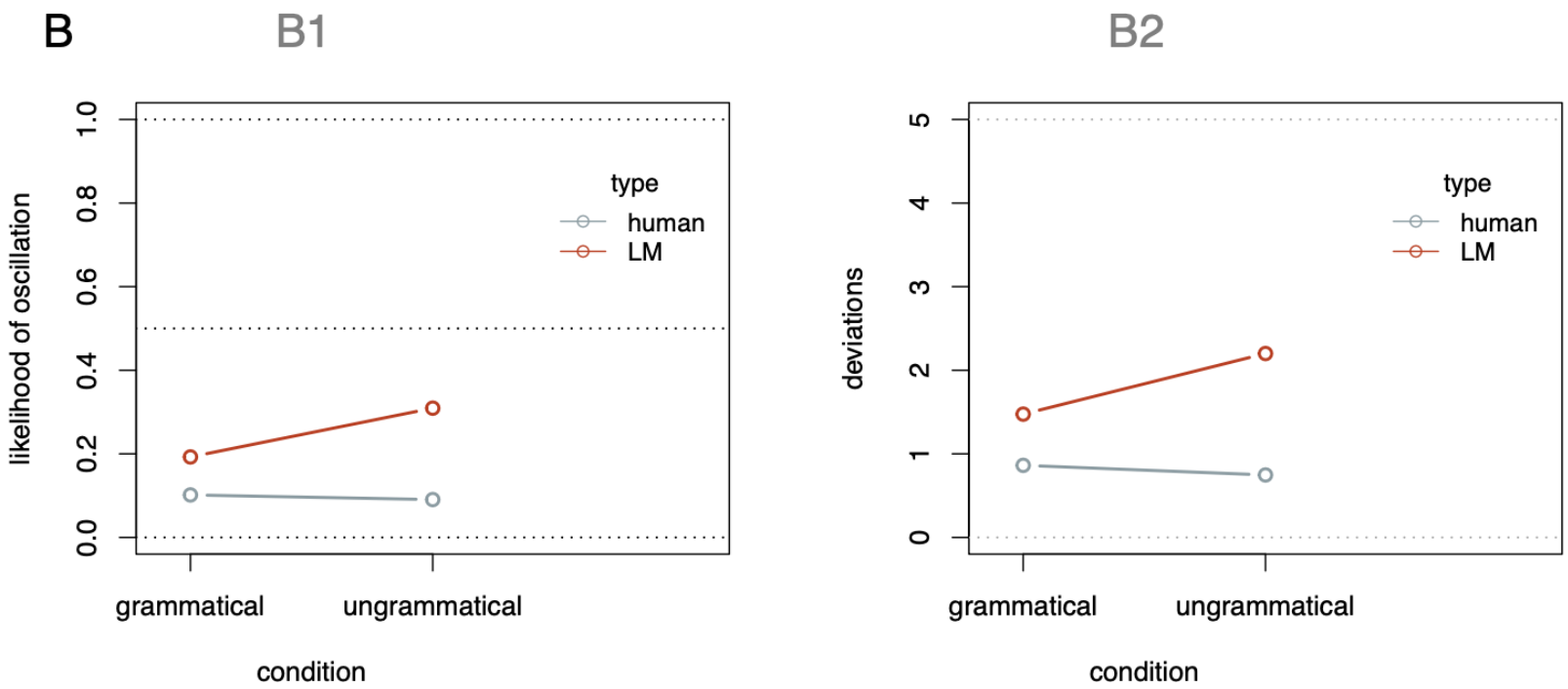
effect of repetitions on mean accuracy, different LLM models and different syntactic phenomena

V.Dentella, et al. "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", PNAS 2023



mean accuracy, comparison with human judgment

V.Dentella, et al. "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", PNAS 2023



likelihood of oscillations, comparison with human judgment

V.Dentella, et al. "Systematic testing of three Language Models reveals low language accuracy, absence of response stability, and a yes-response bias", PNAS 2023

What does this say so far?

- pessimist view: the most extensive and most expensive experiment ever to show that Zellig Harris' distributional model of syntax is inaccurate (which was known since 1955)
- optimist view: what additional information about the "inverse problem of syntax" can be derived from this LLM experimental apparatus?

pessimism of the intellect, optimism of the will

...so let's keep going

detecting the "inverse problem of syntax" in LLMs

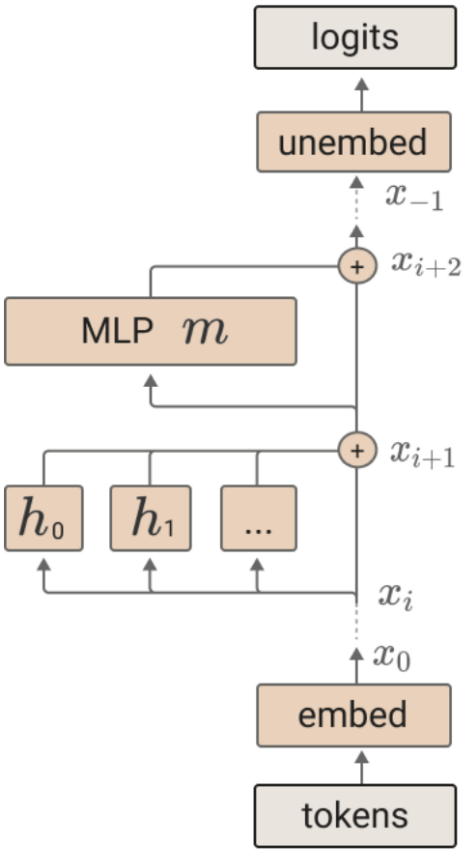
beyond identifying an embedding of syntactic trees in a semantic space determined by attention modules, want to understand to what extent the actual computational mechanism of syntax (Merge) is reconstructed in this inverse problem

- mechanistic interpretability: reverse engineer computations performed by transformers
- identify simple algorithmic patterns (motifs)
- *Note*: again similar to Zellig Harris' idea of "mechanical procedures" for discovering basic elements of language and transformation rules from probabilistic distributions
- more likely to work on "small models" (e.g. studied for transformers with at most two layers and only attention blocks – by comparison GPT-3 has 96 layers)
- a notion of *induction head*: in-context learning algorithms (C.Olsson et al "In-context learning and induction heads")

mechanistic interpretability: a closer look at transformers

N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

- type of model: (1) autoregressive, decoder-only (like GPT-3, not encoder-decoder structure like translation); (2) attention-only (rather than attention and MLP layers – multi-layer-perceptron)
- transformer operations:
  1. token embedding,
  2. a series of "residual blocks" (attention layer with multiple attention heads in parallel and MLP layer)
  3. token unembedding



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_{-1}$$

An MLP layer, m, is run and added to the residual stream.

$$x_{i+2} = x_{i+1} + m(x_{i+1})$$

Each attention head, h, is run and added to the residual stream.

$$x_{i+1} = x_i + \sum_{h \in H_i} h(x_i)$$

One residual block

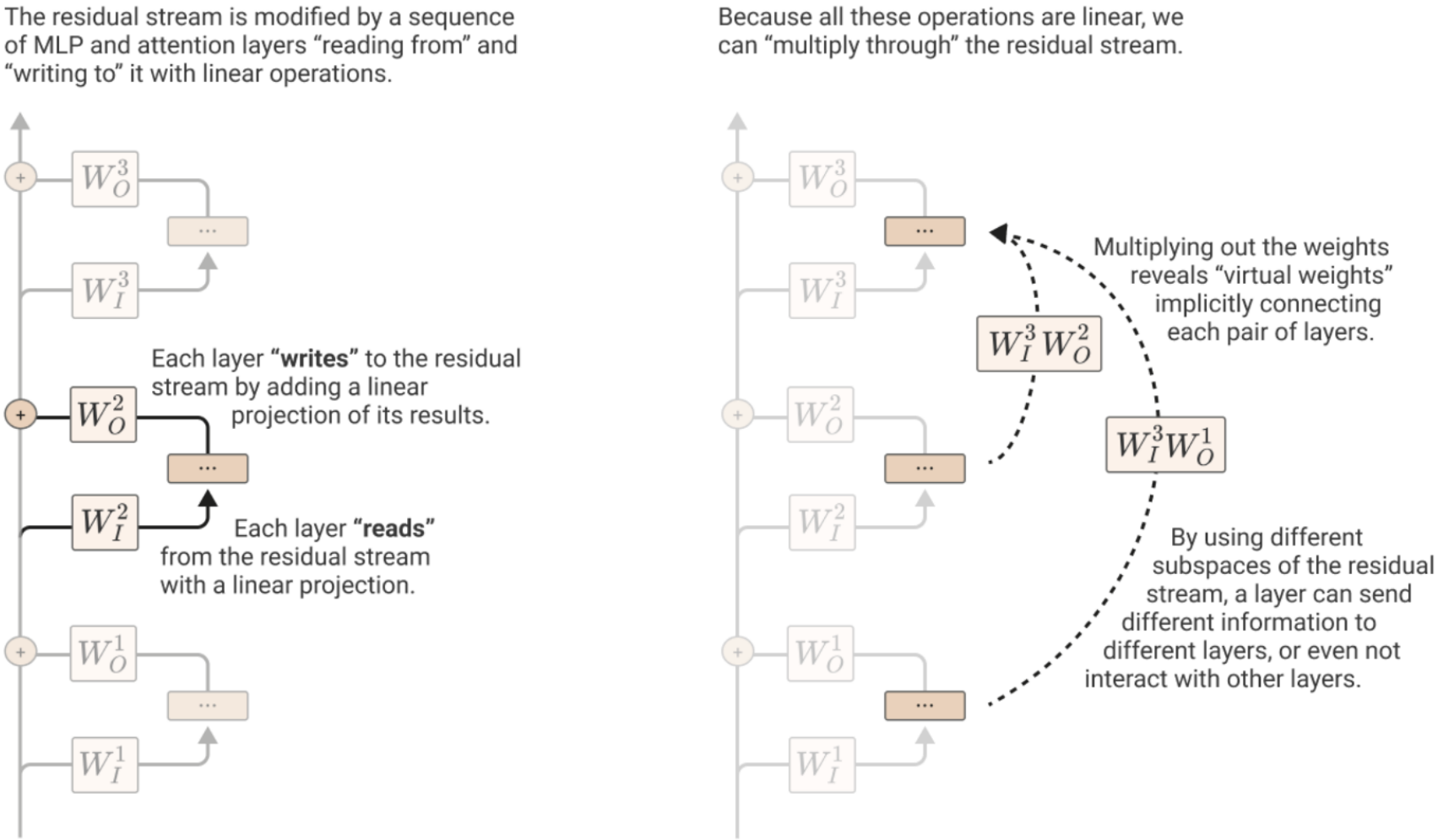
Token embedding.

$$x_0 = W_E t$$

N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

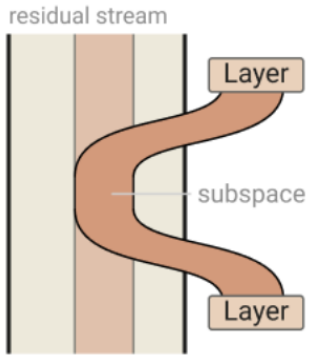
residual stream and linearity

- each layer adds result into the "residual stream" (residual stream vectors also referred to as "embedding")
- residual stream is run by *linear* operations
- linear structure of residual stream means can encode how later layers read information in previous layers though "virtual weights" (matrix entries of a linear transformation)
- residual stream is a high-dimensional vector space (10^2 for small models, 10^4 for large)
- different information stored in different subspaces sent to different layers
- in attention modules each attention head operates on a small subspace (e.g. 64-dim); different attention heads can write to different subspaces
- subspaces of the residual stream are like memory storage, lots of additional subspaces to store from other layers

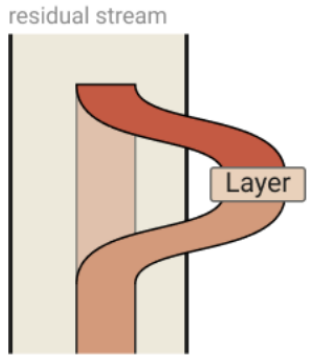


N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

The residual stream is high dimensional, and can be divided into different subspaces.



Layers can interact by writing to and reading from the same or overlapping subspaces. If they write to and read from disjoint subspaces, they won't interact. Typically the spaces only partially overlap.



Layers can delete information from the residual stream by reading in a subspace and then writing the negative verison.

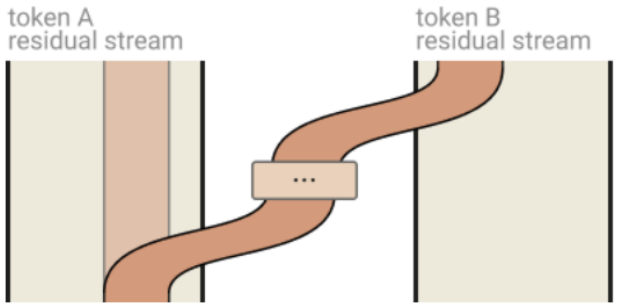
N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

closer look at attention heads

- primary function is "moving information" between different parts of the residual stream
- multiple independent attention heads in an attention module: operate completely in parallel, each adding its output back in the residual stream

$$\sum_i W_O^{h_i} r^{h_i}$$

r^{h_i} = result vector of the i-th attention head, $W_O^{h_i}$ = output matrix i-th block



Attention heads copy information from the residual stream of one token to the residual stream of another. They typically write to a different subspace than they read from.

operations of attention head

- compute value vector for each token embedded in the residual stream vector space

$$v_i = W_V x_i$$

- compute result vector from attention matrix and value vectors

$$r_i = \sum_j A_{ij} v_j$$

- compute output vector

$$h(x)_i = W_O r_i$$

- combined operation

$$h(x) = (\mathrm{id} \otimes W_O) \circ (A \otimes \mathrm{id}) \circ (\mathrm{id} \otimes W_V)\, x = (A \otimes W_O W_V)\, x$$

A mixes between tokens, $W_O W_V$ acts on each independently

- for the attention matrix A part: $\sigma =$ softmax

$$A = \sigma(x^t W_Q^t W_K x) = \sigma(q^t k)$$

- query vectors $q = W_Q x$
- key vectors $k = W_K x$

$$A(q, K, V) = \frac{\sum_j \exp\left[\frac{\langle q, k_j \rangle}{\sqrt{d_K}}\right] v_j}{\sum_j \exp\left[\frac{\langle q, k_j \rangle}{\sqrt{d_K}}\right]}$$

$$Q = W^Q X = \{q_i = W^Q x_i\},$$
$$K = W^K X = \{k_i = W^K x_i\},$$
$$V = W^V X = \{v_i = W^V x_i\},$$

$$h_j(x_i, X) = A(W_j^Q x_i, W_j^K X, W_j^V X)$$

$$Z = \{z_i\} = M(X) = \left\{ W^O \oplus_j \frac{\sum_l \exp\left[\frac{\langle W_j^Q x_i, W_j^K x_l \rangle}{\sqrt{d_K}}\right] W_j^V x_l}{\sum_l \exp\left[\frac{\langle W_j^Q x_i, W_j^K x_l \rangle}{\sqrt{d_K}}\right]} \right\}$$

- two separate circuits: $W_O W_V$ and $W_Q^t W_K$ ("vector-per-token side" and "position side"): extremely sparse matrices
- $W_O W_V$ reading source token writing destination token
- $W_Q^t W_K$ move information between different tokens (note: A nonlinear in $W_Q^t W_K$)
- the W_O, W_V, W_Q, W_K only occur through the compositions $W_O W_V$ and $W_Q^t W_K$ (so any other factorization with same compositions would do the same)
- multiple attention heads functionally equivalent to single

$$(A^{h_n} \otimes W_{OV}^{h_n}) \circ \cdots \circ (A^{h_1} \otimes W_{OV}^{h_1}) = A^{h_n} \circ \cdots \circ A^{h_1} \otimes W_{OV}^{h_n} \circ \cdots \circ W_{OV}^{h_1}$$

- on two sides of tensor product: position variables (left) and token vectors (right)

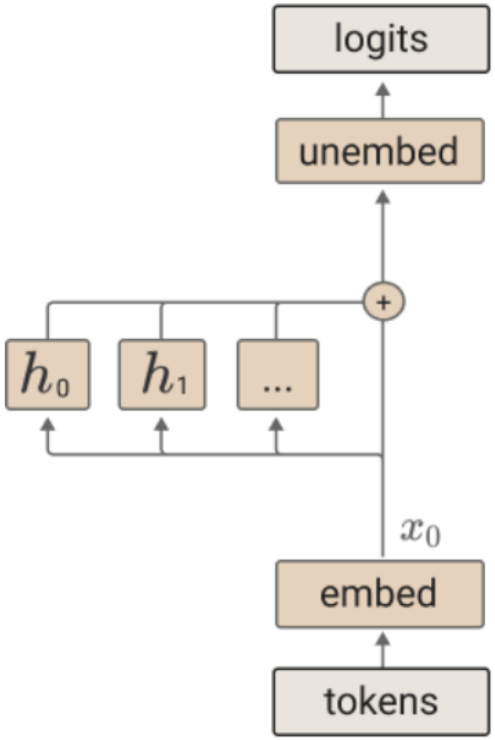
residual stream as basic digram: zero-layer transformer

- if no attention modules: just take token, embed it, unembed it
- linear map $T = W_U W_E$
- optimal behavior possible: $W_U W_E$ digram log-likelihood
- even when other parts of model (attention modules) are present, this residual stream part will contribute digram model log-likelihood and can be seen to detect such digram correlations not related to grammatical rules and syntax structures

N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

Attention head circuits: one-layer attention-only transformer

- embedding, attention module, unembedding



The final logits are produced by applying the unembedding.

$$T(t) = W_U x_1$$

Each attention head, h, is run and added to the residual stream.

$$x_1 = x_0 + \sum_{h \in H} h(x_0)$$

Token embedding.

$$x_0 = W_E t$$

N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

$$T = \underbrace{\mathrm{Id} \otimes W_U}_{\text{The token unembedding maps residual stream vectors to logits.}} \cdot \underbrace{\left(Id + \sum_{h \in H_1} A^h \otimes W_{OV}^h\right)}_{\text{The attention layer has multiple heads. The result of each is added into the residual stream.}} \cdot \underbrace{\mathrm{Id} \otimes W_E}_{\text{The token embedding maps tokens to residual stream vectors.}}$$

where $$A^h = \underbrace{\mathrm{softmax}^*}_{\text{Softmax with autoregressive masking}} \left(\underbrace{t^T \cdot W_E^T W_{QK}^h W_E \cdot t}_{\text{Attention pattern logits are produced by multiplying pairs of tokens through different sides of } W_{QK}^h.} \right)$$

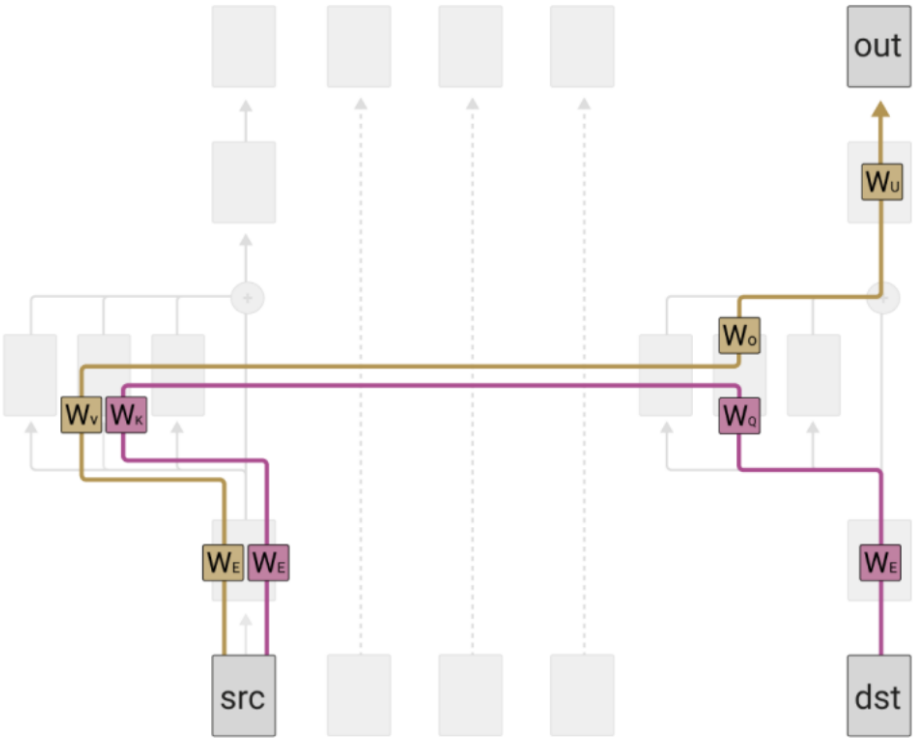
N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

$$T = \underbrace{\mathrm{Id} \otimes W_U W_E}_{\text{"Direct path" term contributes to bigram statistics.}} + \underbrace{\sum_{h \in H} A^h \otimes (W_U W_{OV}^h W_E)}_{\text{The attention head terms describe the effects of attention heads in linking input tokens to logits. } A^h \text{ describes which tokens are attended to while } W_U W_{OV}^h W_E \text{ describes how each token changes the logits if attended to.}}$$

N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

- separate out the digram effect $W_U W_E$
- the attention module effect is in the $A^h \otimes W_U W_{OV}^h W_E$
- consists of two separate circuits A^h and $W_U W_{OV}^h W_E$ acting on different sets of variables

- A moves information between different tokens: contextual word embedding (vector in the residual stream) of a token has components in subspaces with information copied from other tokens
- query-key circuit: $W_E^t W_{QK} W_E$ which tokes head preferably attends to
- output value circuit: $W_U W_{OV} W_E$ how a given token will affect the output if attended to



The **OV ("output-value") circuit** determines how attending to a given token affects the logits.

$$W_U W_O W_V W_E$$

The **QK ("query-key") circuit** controls which tokens the head prefers to attend to.

$$W_E^T W_Q^T W_K W_E$$

one-layer transformer as skip-trigram

- QK-circuit: which source token a present destination token attends back to and copies information from; OV-circuit: resulting effect on the out-predictions for the next token
- k-skip N-gram: subset of an unordered N-gram using non-contiguous substrings with skips of length k
- skip-trigram with 3 tokens: source, destination, output (last one is modified)
- matrices themselves are enormous but *very sparse* (50k x 50k but rank 64)

- searching for large entries shows behavior:
  - most attention heads in one-layer perform *copying*
  - tokens are copied to places where *digram-statistics* make them plausible
  - other skip-trigram behavior: identifies classical trigrams ("back and forth", "eat and drink", "day and night", "keep in mind", "keep at bay", etc)
  - but because of factored QK and OV, not quite 3-way interactions: eg high probability for "keep in mind" and "keep at bay" also causes high probability for "keep at mind", "keep in bay"
  - most heads heads attend to previous token, but essentially none that attend two tokens back or more
- this last fact: would totally miss syntactic structures (unless trivial enough to correlate strongly with immediately adjacent words in linear order), but most syntactic phenomena depend on structural relations (in the tree structure) between tokens distant in linear ordering

when attention layers compose two-layer transformer

$$T = \underbrace{\mathrm{Id} \otimes W_U}_{\text{The second attention layer has multiple attention heads which add into the residual stream}} \cdot \underbrace{\left(Id + \sum_{h \in H_2} A^h \otimes W_{OV}^h\right)}_{\text{The first attention layer has multiple attention heads which add into the residual stream}} \cdot \left(Id + \sum_{h \in H_1} A^h \otimes W_{OV}^h\right) \cdot \mathrm{Id} \otimes W_E$$

$$= \underbrace{\mathrm{Id} \otimes W_U W_E}_{\text{"Direct path" term contributes to bigram statistics.}} + \underbrace{\sum_{h \in H_1 \cup H_2} A^h \otimes (W_U W_{OV}^h W_E)}_{\text{The individual attention head terms describe the effects of individual attention heads in linking input tokens to logits, similar to those we saw in the one layer model.}} + \underbrace{\sum_{h_2 \in H_2} \sum_{h_1 \in H_1} (A^{h_2} A^{h_1}) \otimes (W_U W_{OV}^{h_2} W_{OV}^{h_1} W_E)}_{\text{The virtual attention head terms correspond to V-composition of attention heads. They function a lot like individual attention heads, with their own attention patterns (the compositon of the heads patterns) and own OV matrix.}}$$

direct path term and individual head terms same way as in one-layer case; but new effect from composition of attention matrices $A^{h_2} A^{h_1}$ and its own OV-circuit

N.Elhage et al. *A Mathematical Framework for Transformer Circuits*, 2021

main observation here

- $A^{h_2 \circ h_1} := A^{h_2} A^{h_1}$ and $W_{OV}^{h_2 \circ h_1} := W_{OV}^{h_2} W_{OV}^{h_1}$
- virtual attention heads $h_2 \circ h_1$
- see at this level virtual attention heads that attend two tokens back or to other positions (beginning of sentence, subject, etc)
- there starts to be signs of syntactic rules being detected
- "induction heads" visible from weights: compositions attending to previous copies of token (even on completely random repeated patterns)

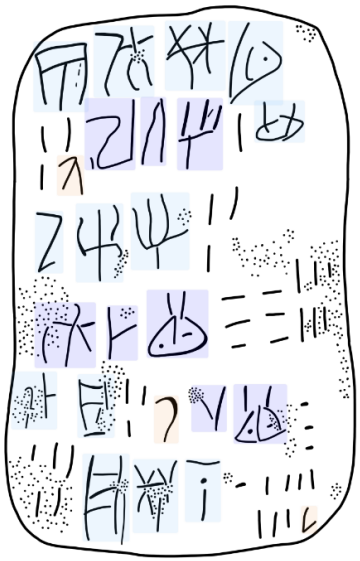
Note: main advantage with respect to the old Behaviorism approach: *now we* **know** *what we are looking for*!

natural questions

1. can see where the syntactic trees detected in attention weights are located? virtual attention heads at what level in number of layers?
2. does depth of syntactic trees relate to layers?
3. can identify a circuit performing Merge?

cautionary tale N.1: prediction versus explanation

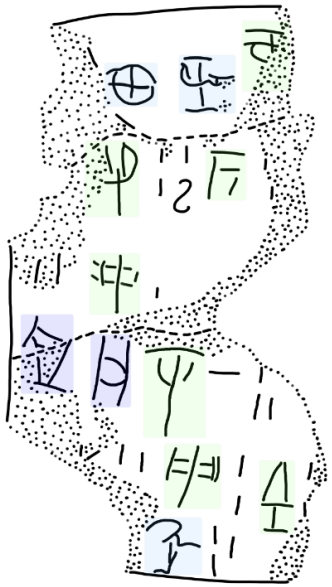
- one can train an LLM on all the 772 existing texts of the Linear A language (SigLA database)
- Linear A is the *undeciphered* language of the Minoan civilization of Crete
- automated generation of next word *prediction* in Linear A will add nothing to our *understanding* of the language



HT 6b

HT 7a

ARKH 2



ARKH 3a

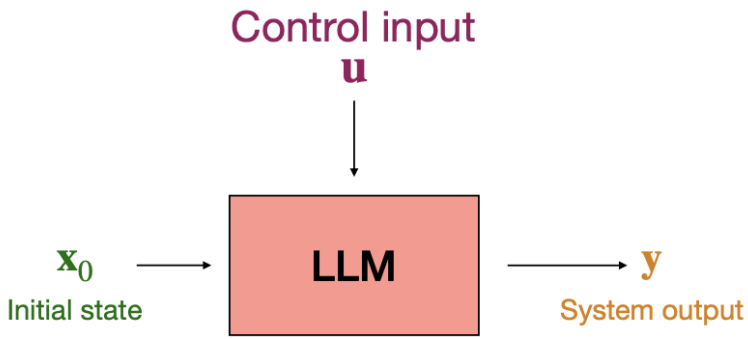
- compare with the story of the decipherment of Linear B
- Alice Kober's three papers (1945–1948): use of distributional model for *testing of scientific hypotheses*
  1. proved that Linear B is an inflected language: roots modified by suffixes (name declension, verb conjugation)
  2. identification of phonetic relations between sets of Linear B syllabic signs
  3. final step (completed by Ventris and Chadwick, 1952) comparison with a well known language: Ancient Greek
- conclusion: Linear B is Greek (Mycenaean Greek)
- conclusion: can successfully use distributional/statistical models *to test scientific hypotheses*; they do not in themselves constitute a viable "theory of language"

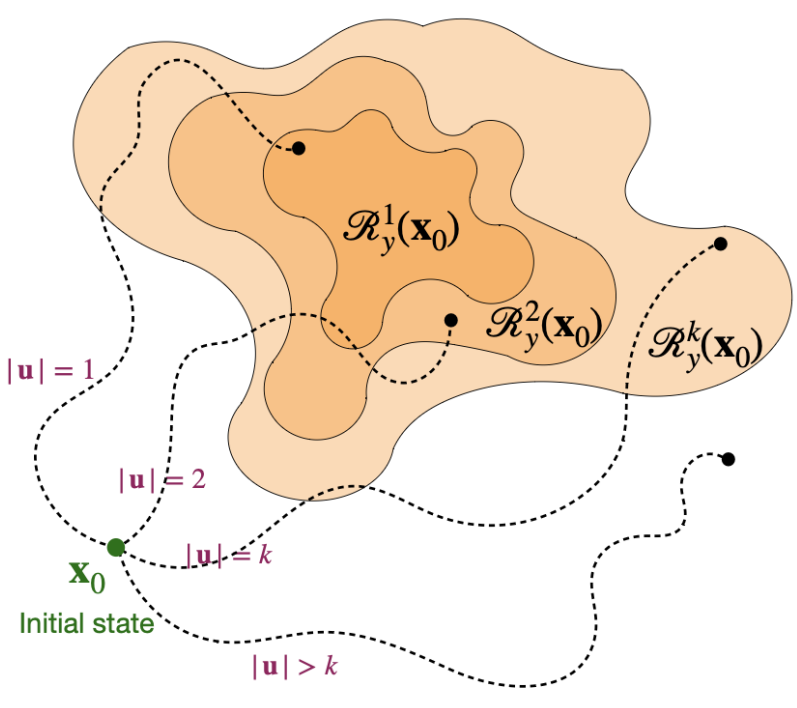
cautionary tale N.2: control theory

- developed in the context of *alignment* of LLMs and adversarial techniques
- how to add a control sequence (of shortest number of token) to prompt to ensure a desired next output of LLM
- extensive search over *all* single-token substitutions, minimizing a loss function (of distance to desired output), greedy gradient-based search
- *unlike human language*: optimal control sequences are *not* semantically/syntactically related to output but gibberish combinations of tokens

source

- Aman Bhargava, Cameron Witkowski, Manav Shah, Matt Thomson, *What's the Magic Word? A Control Theory of LLM Prompting*, arXiv:2310.04444





Aman Bhargava, Cameron Witkowski, Manav Shah, Matt Thomson, *What's the Magic Word? A Control Theory of LLM Prompting*, arXiv:2310.04444

- reachable set of outputs of an LLM: set $R_y(x_0)$ of output sequences y for which $\exists$ control input sequence u that stirs LLM from initial state x_0 to output y
- bounds on reachable output set for a self-attention head as function of singular values of its parameter matrices
- tested on Falcon-7b, Llama-7b, Falcon-40b (dataset of 5k state-ouput sequences with states of length 8–32)
- sample initial states x_0 from Wikitext dataset and probe reachable output tokens y under length-constrained control input sequences $|u| \leq k$
- top 75 most likely output tokens y are reachable at least 85% of the time with $k \leq 10$ control sequence
- interesting facts:
  - some least likely output tokens controllable: most likely output with controls $k \leq 4$
  - control sequences maximizing $P(y|x_0 + u)$ are gibberish

if LLMs are "*a theory of language*" (as some people claim) then there's some very serious problem there!

Что делать?

as scientists, let's try doing some science

- non-behaviorist mechanistic interpretability: circuit investigation looking for more precise information on the embedded image of syntax (syntactic objects and Merge operation)
- direct comparison of mathematical models: the Harris distributional model LLMs are based on can be given a modern mathematical formulation in category theory language (Gaubert-Vlassopoulos, Bradley–Terilla–Vlassopoulos), this *can* be compared directly to the mathematical model of Merge

stay tuned for more to come...

mathematical model of Z.Harris' distributional theory in LLMs

Mathematical models of LLMs

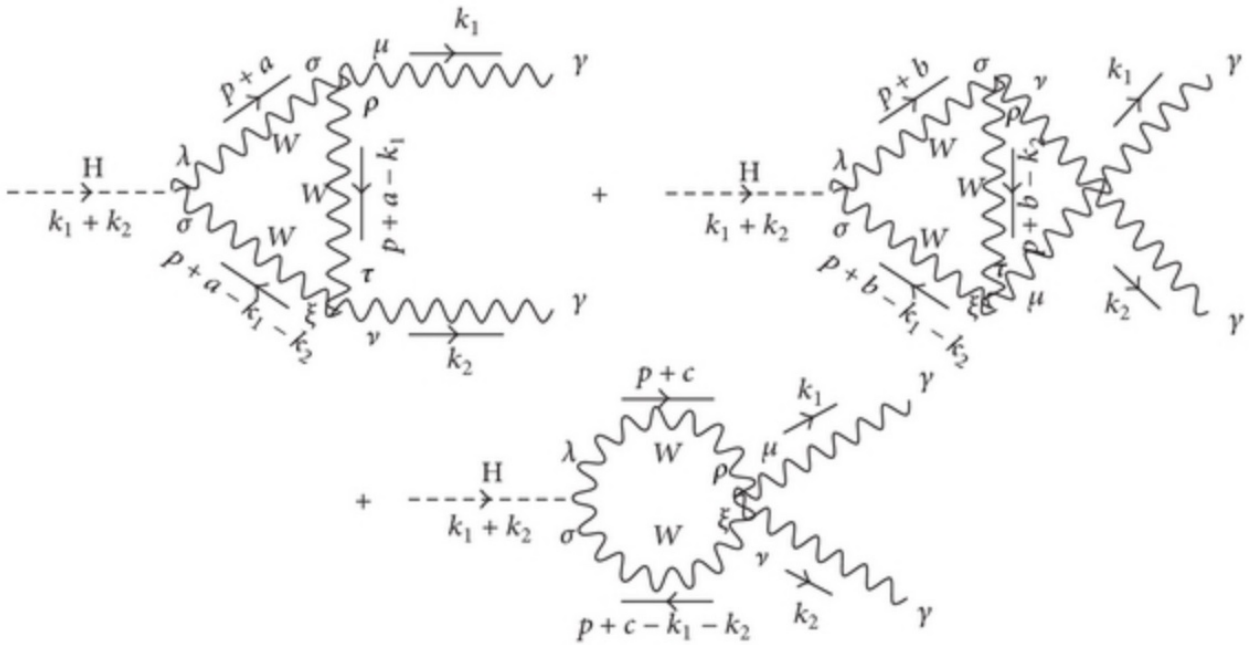
- S. Gaubert, Y. Vlassopoulos, *Directed metric structures arising in large language models*, preprint 2024.
- Tai-Danae Bradley, John Terilla, Yiannis Vlassopoulos, *An enriched category theory of language: from syntax to semantics*, arXiv:2106.07890

empirical evidence of this model in LLMs:

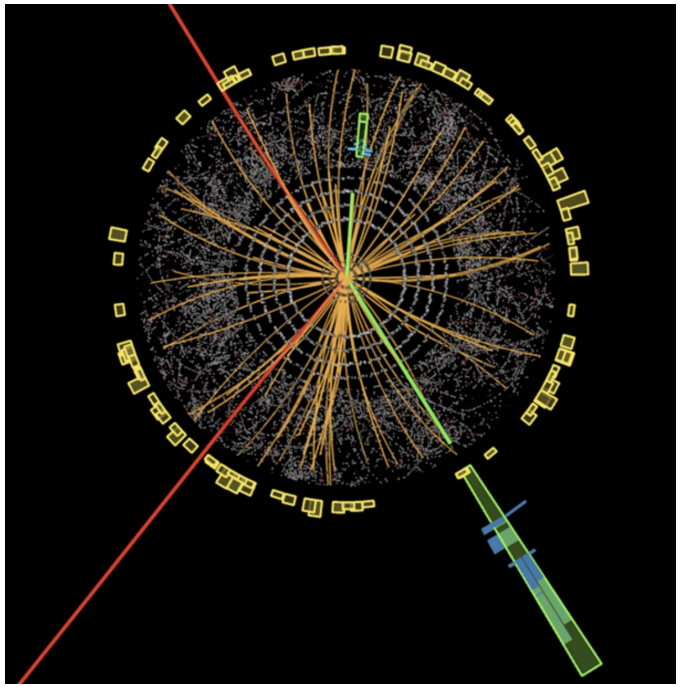
- Tian Yu Liu, Matthew Trager, Alessandro Achille, Pramuditha Perera, Luca Zancato, Stefano Soatto, *Meaning Representations from Trajectories in Autoregressive Models*, arXiv:2310.18348

Concluding remarks: physics as metaphor

- Quantum Field Theory: generative process of Feynman diagrams, assignment of meaningful physical values (renormalization) $\Rightarrow$ perturbative computation of Higgs boson production cross sections



- Particle accelerators and detectors: solving an *inverse problem* that identifies inside enormous set of data traces of the correct diagrams/processes involving creation/decay of a Higgs particle through interactions of other particles



sees "an image" of the QFT objects embedded into the set of data collected by detectors, against a noise background of a huge number of other simultaneous events

- the generative process of syntax is embedded in LLMs in a conceptually similar way: its image is scattered in a probabilistic smear across large number of weights and vectors, trained over large data sets
- signals of linguistic structures detectable against a background of probabilistic noise
- LLMs do not "invalidate" generative syntax any more than particle detectors would "invalidate" Quantum Field Theory: quite the opposite
  consequently:
- LLMs are *not* a language theory, generative syntax is
- LLMs are an *experimental apparatus* for the study of the inverse problem of the syntax-semantic interface
- data and technology without *theory* do *not* constitute science
- Where is the *explanatory* power? Where is the *understanding*?

The purpose of science is to obtain a coincise conceptual explanation of natural phenomena, that should be testable, predictive, and essential (*entia non sunt multiplicanda praeter necessitatem*)

Predictions are needed for *falsifiability* of scientific theory, but are not the goal in themselves, the goal of science is conceptual explanation

Generative linguistics aims at producing such explanations for the structure and functioning of language

what is actually happening in LLMs *should* be understood by a careful mathematical modeling of what they compute and comparing it with mathematical models of generative syntax as produced by human brains

- mathematics is a powerful explanatory tool, because it is both highly constrained and highly flexible
- this is why it is the language of science (or as Galileo said, the language in which the universe is written)

Thank You!