Emergent linguistic structure in deep contextual neural word representations

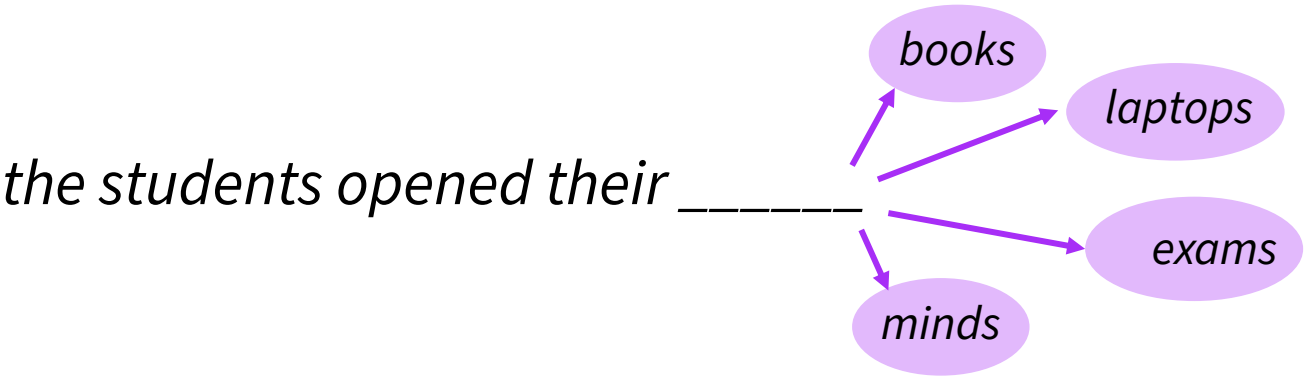
Christopher Manning
Stanford University
@chrmanning ❀ @stanfordnlp
Institute for Advanced Study, Princeton, NJ, 2019

Plan

1. From language models to contextual word representations
2. Transformers and BERT
3. What does BERT know? Observational evidence
4. What does BERT know? Experimental evidence

1. Language Modeling

A **Language Model (LM)** predicts a word in a context



For a word sequence $x^{(1)}, x^{(2)}, \ldots, x^{(t)}$, it gives the probability of $x^{(t+1)}$:

$$P(x^{(t+1)}|x^{(t)}, \ldots, x^{(1)})$$

An LM is a key part of decoding tasks like **speech recognition**, **spelling correction**, and any NL generation task, including **machine translation**, **summarization**, and **story generation**

LMs in The Dark Ages: *n*-gram models

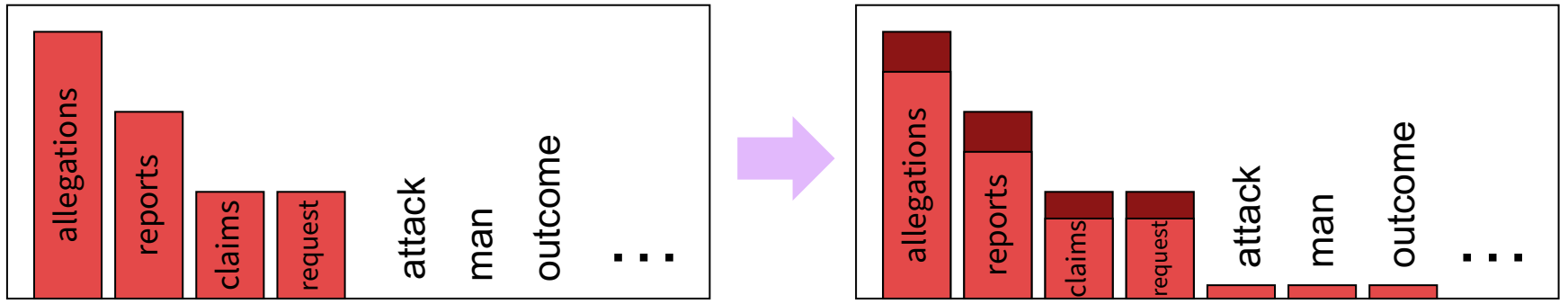
Count how often words follow word sequences; divide to get cond. prob.

Classic **curse of dimensionality** scenario: zillions of params

Markov assumption:

$$P(x^{(t+1)}|\text{President Trump denied the}) \approx P(x^{(t+1)}|\text{denied the})$$

Discounting/Smoothing



Mixture/Backoff

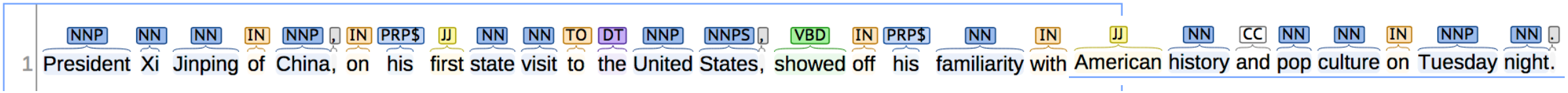
$$P_{bo}\left(x^{(3)}|x^{(2)},x^{(1)}\right) \approx \lambda P\left(x^{(3)}|x^{(2)},x^{(1)}\right) + (1-\lambda)P\left(x^{(3)}|x^{(2)}\right)$$

How much of the intricate structure of human languages do these language models know?

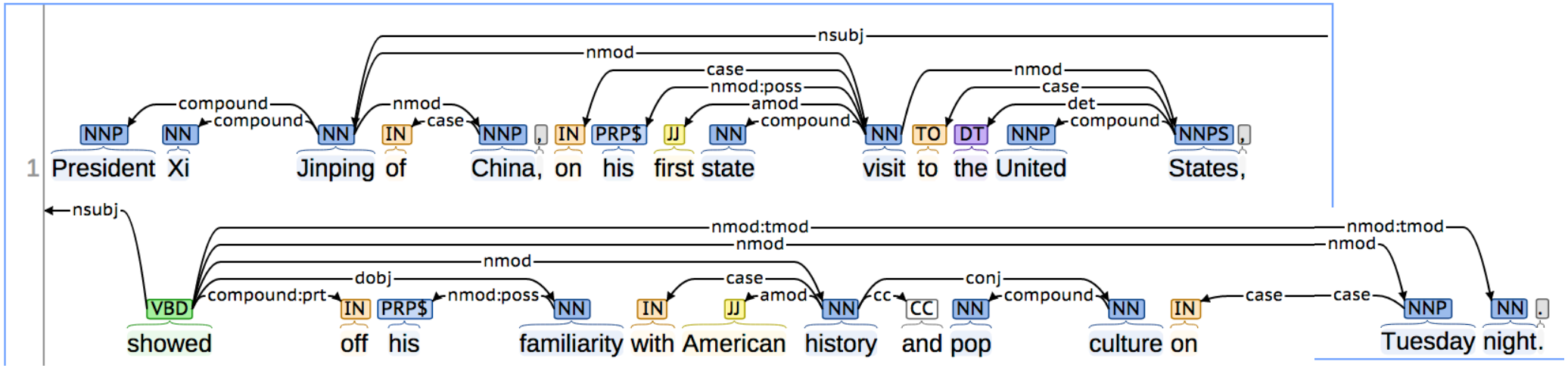
- (**Passionately argued!**) answer of linguists: **almost none**
  - Though they know quite a bit of simple world knowledge
    - The ship {sailed, sank, anchored, …}
  - And, in an unaggregated way, they know some low-level syntax
    - They know you tend to get sequences like:
      - preposition – article – noun
      - article – adjective – noun
  - But they don't know the concept "noun" or sentence structure rules

Capturing conventional linguistics in NLP

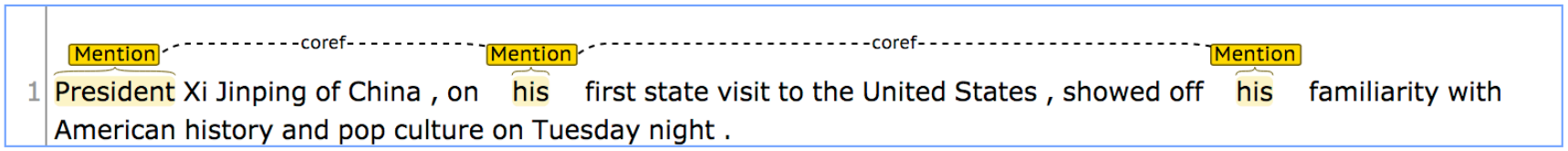
Part-of-Speech:



Basic Dependencies:



Coreference:



REVIEW: NEUROSCIENCE

The Faculty of Language: What Is It, Who Has It, and How Did It Evolve?

Marc D. Hauser,[1]* Noam Chomsky,[2] W. Tecumseh Fitch[1]

We argue that an understanding of the faculty of language requires substantial interdisciplinary cooperation. We suggest how current developments in linguistics can be profitably wedded to work in evolutionary biology, anthropology, psychology, and neuroscience. We submit that a distinction should be made between the faculty of language in the broad sense (FLB) and in the narrow sense (FLN). FLB includes a sensory-motor system, a conceptual-intentional system, and the computational mechanisms for recursion, providing the capacity to generate an infinite range of expressions from a finite set of elements. We hypothesize that FLN only includes recursion and is the only uniquely human component of the faculty of language. We further argue that FLN may have evolved for reasons other than language, hence comparative studies might look for evidence of such computations outside of the domain of communication (for example, number, navigation, and social relations).

If a martian graced our planet, it would be struck by one remarkable similarity among Earth's living creatures and a key difference. Concerning similarity, it would note that all

Enlightenment era neural language models (NLMs)

1. **Solve curse of dimensionality** by sharing of statistical strength via dense, low-dimensionality word vectors $v_1, v_2, \ldots, v_K$ [Bengio, Ducharme, Vincent & Jauvin JMLR 2003], etc.:

$$P\left(x^{(t+1)} \middle| x^{(t)}, x^{(t-1)}\right) = \text{softmax}(\text{FFNN}(v^{(t)}, v^{(t-1)}))$$

2. **Solve failure to exploit long contexts** via **recurrent NNs**

First, simple RNNs, soon usually LSTMs [Zaremba et al. 2014]

*the same **stump** which had impaled the car of many a guest*
*in the past thirty years and which he refused to have **removed***

$$P\left(x^{(t+1)} \middle| x^{(\leq t)}\right) = \text{LSTM}(h^{(t)}, x^{(t)})$$

A RNN Language Model

output distribution

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

hidden states

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$

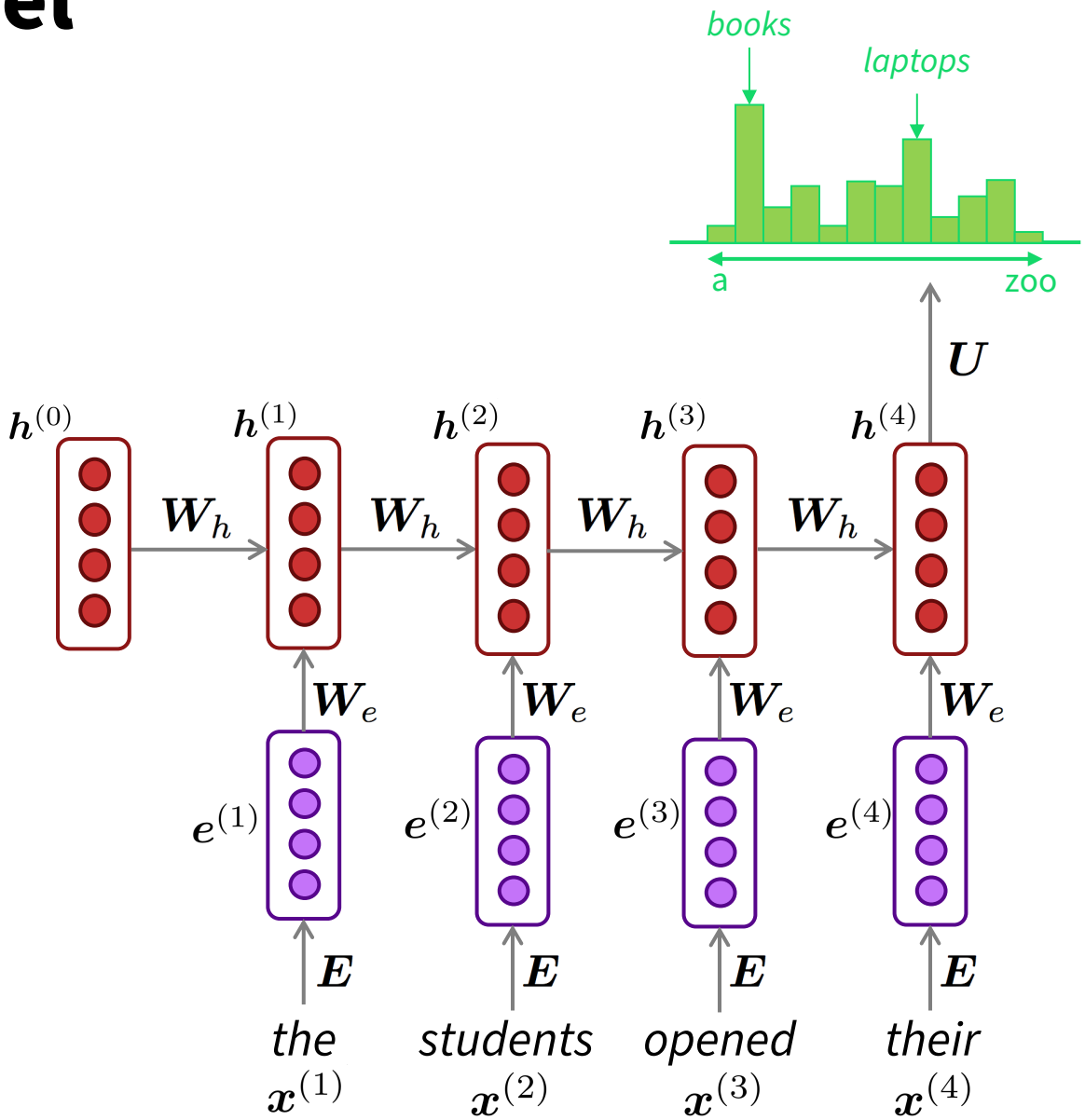
$\boldsymbol{h}^{(0)}$ is the initial hidden state

word embeddings

$$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$$

words / one-hot vectors

$$\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$$



Contextual word representations

output distribution

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

hidden states

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$

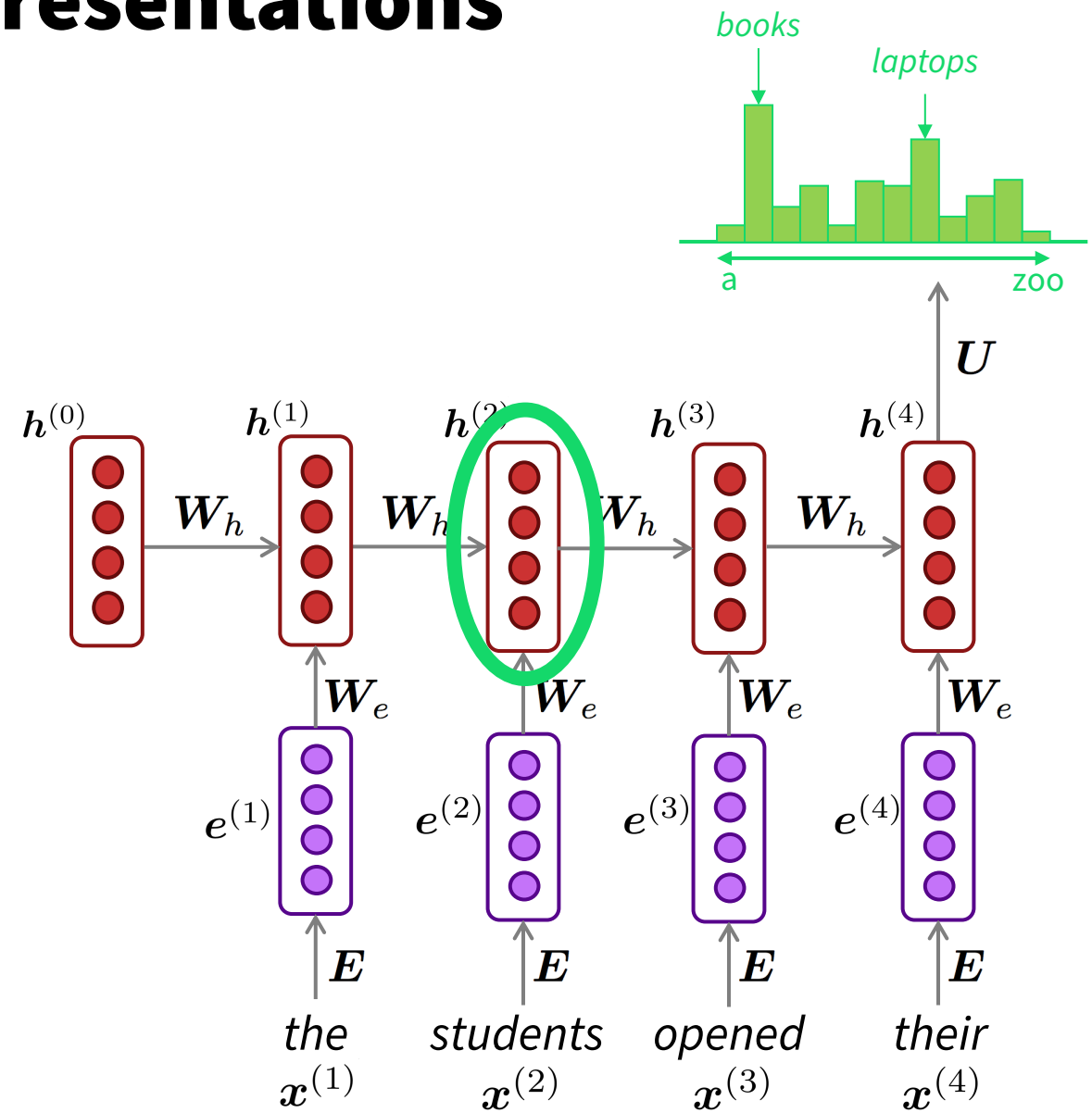
$\boldsymbol{h}^{(0)}$ is the initial hidden state

word embeddings

$$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$$

words / one-hot vectors

$$\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$$



Representing words by their context

- Distributional semantics: **A word's meaning is given by the words that frequently appear close-by**
  - *"You shall know a word by the company it keeps"* (J. R. Firth 1957: 11)
  - One of the most successful ideas of modern statistical NLP!
- When a word *w* appears in a text, its **context** is the set of words that appear nearby (within a fixed-size window).
- Use the many contexts of *w* to build up a representation of *w*

…government debt problems turning into ***banking*** *crises as happened in 2009…*
…saying that Europe needs unified ***banking*** *regulation to replace the hodgepodge…*
…India has just given its ***banking*** *system a shot in the arm…*

These context words will represent ***banking*** via a reconstruction loss

Word meaning

- I **broke** the vase.
- Dawn **broke**.
- He **broke** the silence.
- The news **broke**.
- The machine **broke**.
- Sandy **broke** the world record.
- Sandy **broke** the law.
- The burglar **broke** into the house.
- The newscaster **broke** into the movie broadcast.
- We **broke** even.

Based on an example of Chris Potts

2. Let's scale it up!

ELMo
Oct 2017
Training:
800M words
42 GPU days

GPT
June 2018
Training
800M words
240 GPU days

BERT
Oct 2018
Training
3.3B words
256 TPU days
~320–560 GPU days

GPT-2
Feb 2019
Training
40B words
~2048 TPU v3 days according to a reddit thread

XL-Net,
ERNIE,
Grover
RoBERTa, …
July 2019 on

GPT-2 language model (cherry-picked) output

SYSTEM PROMPT (HUMAN-WRITTEN)

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.

The incident occurred on the downtown train line, which runs from Covington and Ashland stations.

In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief.

“The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,” said Tom Hicks, the U.S. Energy Secretary, in a statement. “Our top priority is to secure the theft and ensure it doesn’t happen again.”

The stolen material was taken from the University of Cincinnati’s Research Triangle Park nuclear research site, according to a news release from Department officials.

METRO
NEWS... BUT NOT AS YOU KNOW IT

Elon Musk's OpenAI builds artificial intelligence so powerful it must be kept locked up for the good of humanity

Jasper Hamill Friday 15 Feb 2019 10:06 am

Transformer models

All of these models are Transformer models

ELMo
Oct 2017
Training:
800M words
42 GPU days

GPT
June 2018
Training
800M words
240 GPU days

BERT
Oct 2018
Training
3.3B words
256 TPU days
~320–560 GPU days

GPT-2
Feb 2019
Training
40B words
~2048 TPU v3 days according to a reddit thread

XL-Net,
ERNIE,
Grover
RoBERTa, …
July 2019

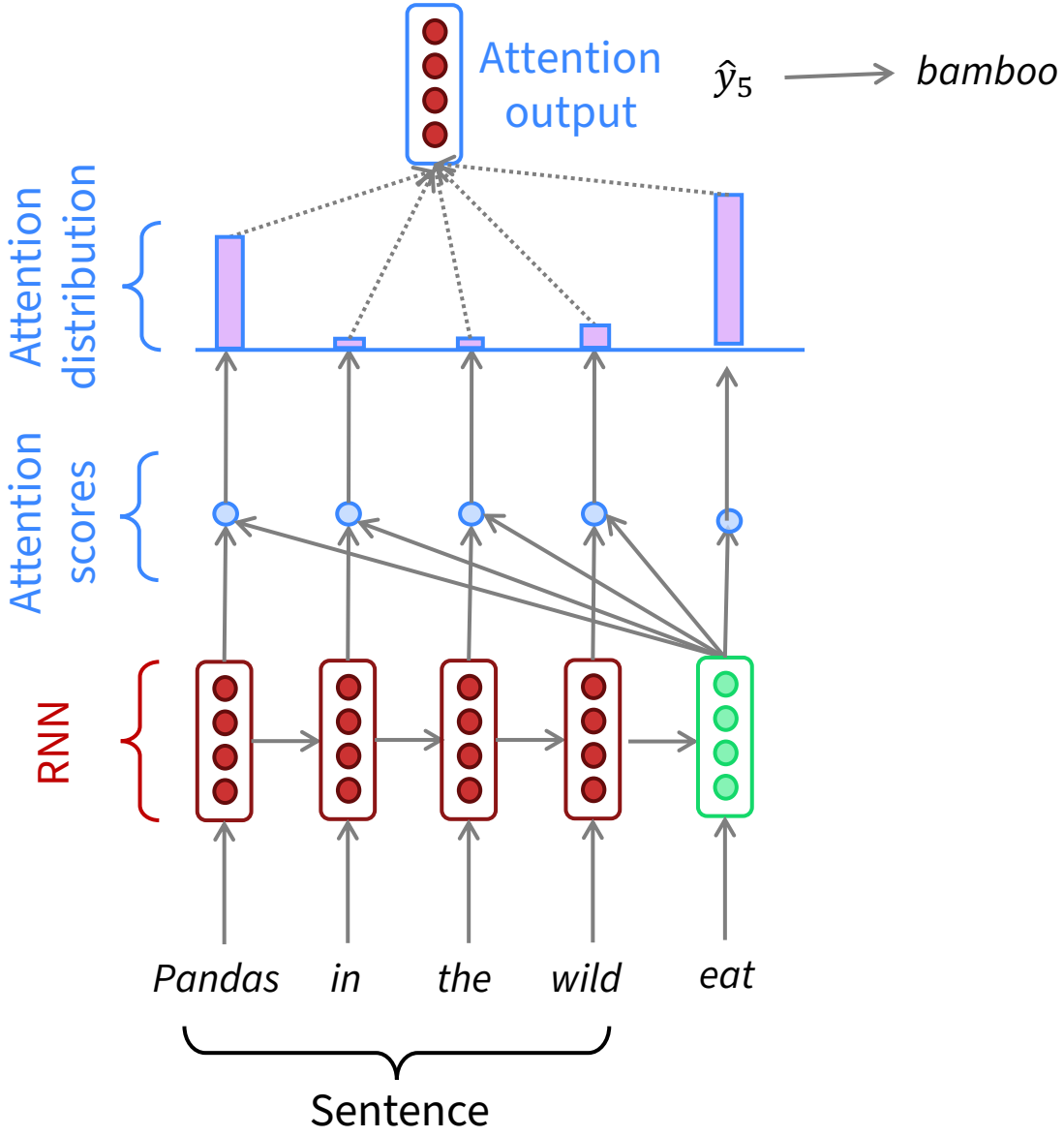
Recurrent models with (self-)attention

$$\boldsymbol{c}_t = \sum_s \boldsymbol{a}_t(s)\bar{\boldsymbol{h}}_s$$

$$\boldsymbol{a}_t(s) = \frac{\mathrm{e}^{\mathrm{score}(s)}}{\sum_{s'} \mathrm{e}^{\mathrm{score}(s')}}$$

$$\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s) = \boldsymbol{h}_t^\top \boldsymbol{W}_{\boldsymbol{a}} \bar{\boldsymbol{h}}_s$$

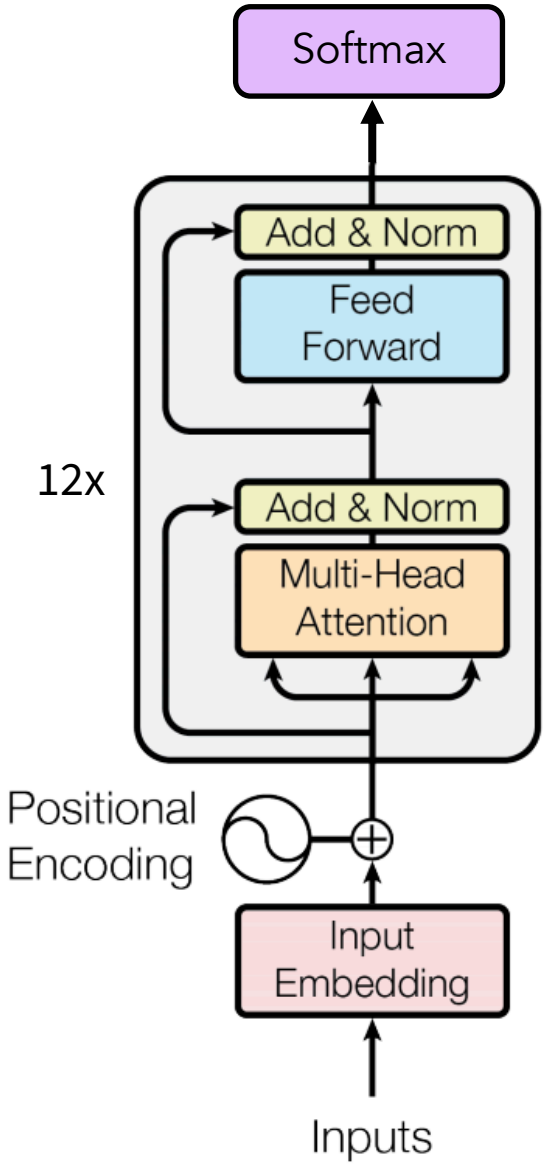
Bilinear attention



Transformer (Vaswani et al. 2017)

https://arxiv.org/pdf/1706.03762.pdf

- **Non-recurrent** sequence model (or sequence-to-sequence model)
- A **deep** model with a sequence of **attention**-based transformer blocks
- Depth allows a certain amount of lateral information transfer in understanding sentences, in slightly unclear ways
- Final cost/error function is standard cross-entropy error on top of a softmax classifier



BERT: Devlin, Chang, Lee, Toutanova (2018)

BERT (Bidirectional Encoder Representations from Transformers):

Pre-training of Deep Bidirectional Transformers for Language Understanding, which is then fine-tuned for a particular task

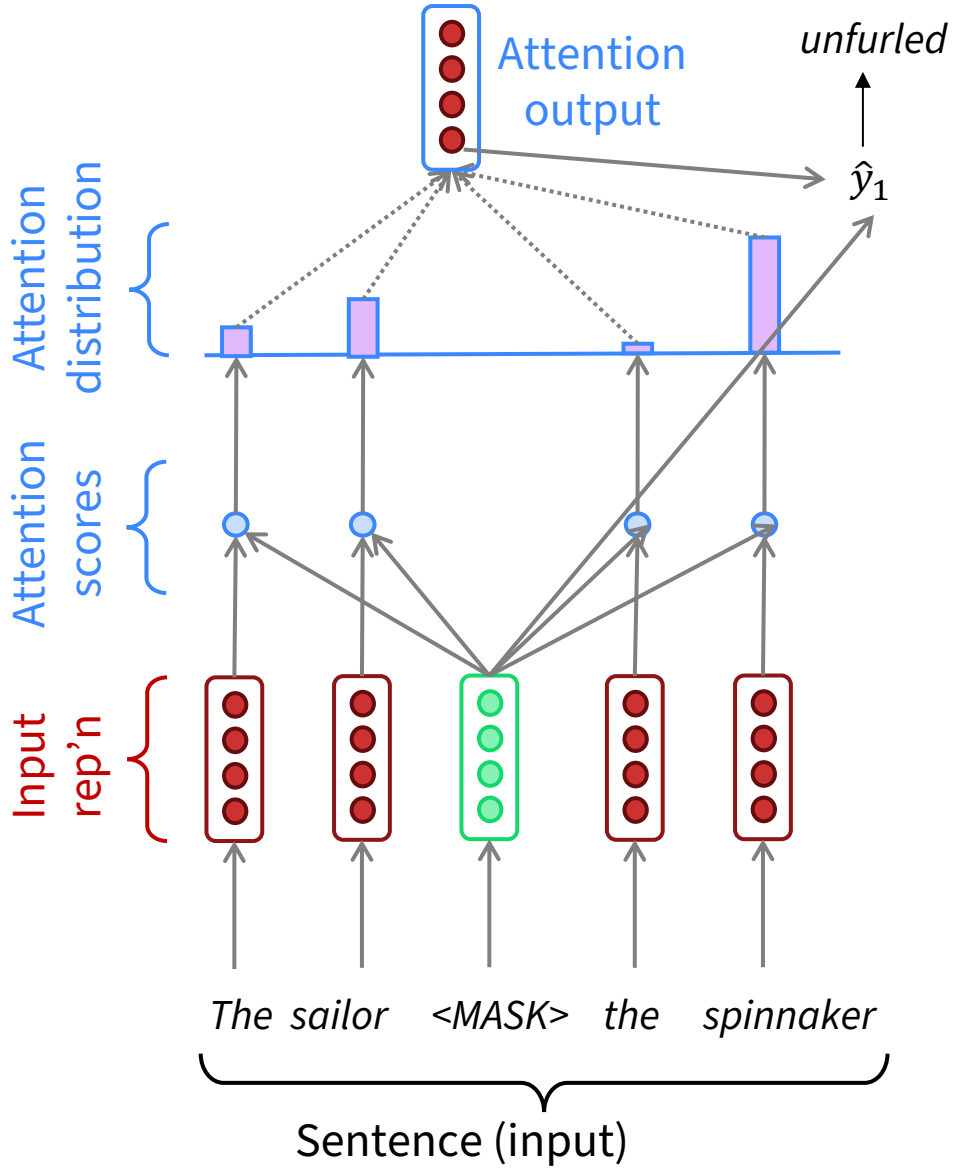
Pre-training uses a cloze task formulation where 15% of words are masked out and predicted:

store gallon

↑ ↑

the man went to the [MASK] to buy a [MASK] of milk

Self-attention in masked sequence model



We get the attention scores e^t for step t

$$\boldsymbol{e}^t = [\boldsymbol{s}_t^T \boldsymbol{h}_1, \ldots, \boldsymbol{s}_t^T \boldsymbol{h}_N] \in \mathbb{R}^N$$

We take softmax to get the attention (prob.) distribution α^t for step t

$$\alpha^t = \text{softmax}(\boldsymbol{e}^t) \in \mathbb{R}^N$$

We use α^t to take weighted sum of the hidden states to get attention output

$$\boldsymbol{a}_t = \sum_{i=1}^{N} \alpha_i^t \boldsymbol{h}_i \in \mathbb{R}^h$$

Finally we join (sum or concatenate) the attention output a^t with the decoder hidden state s_t and proceed in model

Multi-head (self) attention

With simple self-attention: Only one way for a word to interact with others

Solution: Multi-head attention

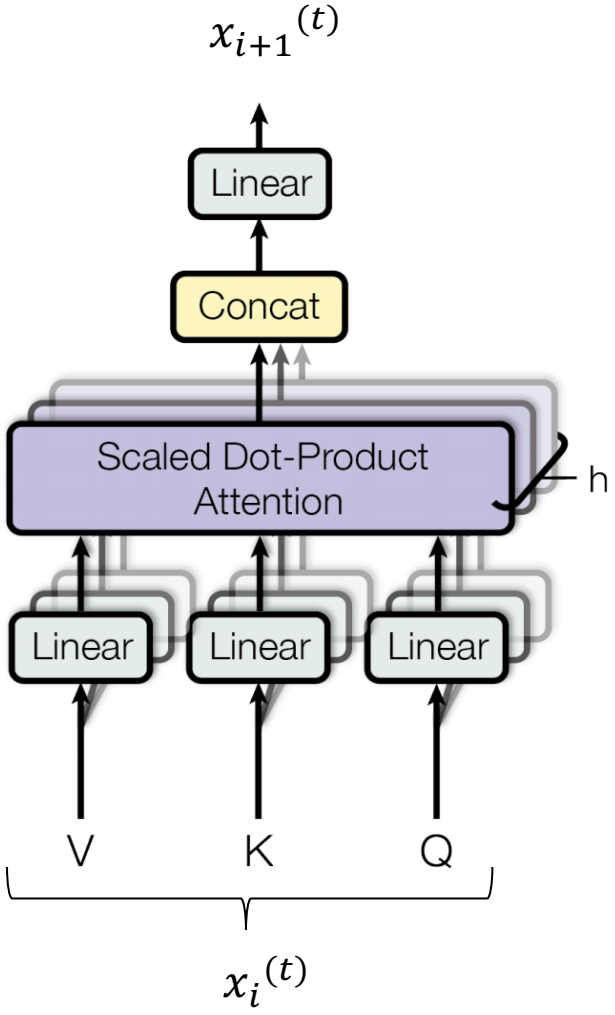
Map input into $h = 12$ many lower dimensional spaces via W_h matrices

Then apply attention, then concatenate outputs and pipe through linear layer

$$\text{Multihead}\left(x_i^{(t)}\right) = \text{Concat}(head_j)W^O$$

$$head_j = \text{Attention}(x_i^{(t)}W_j^Q, x_i^{(t)}W_j^K, x_i^{(t)}W_j^V)$$

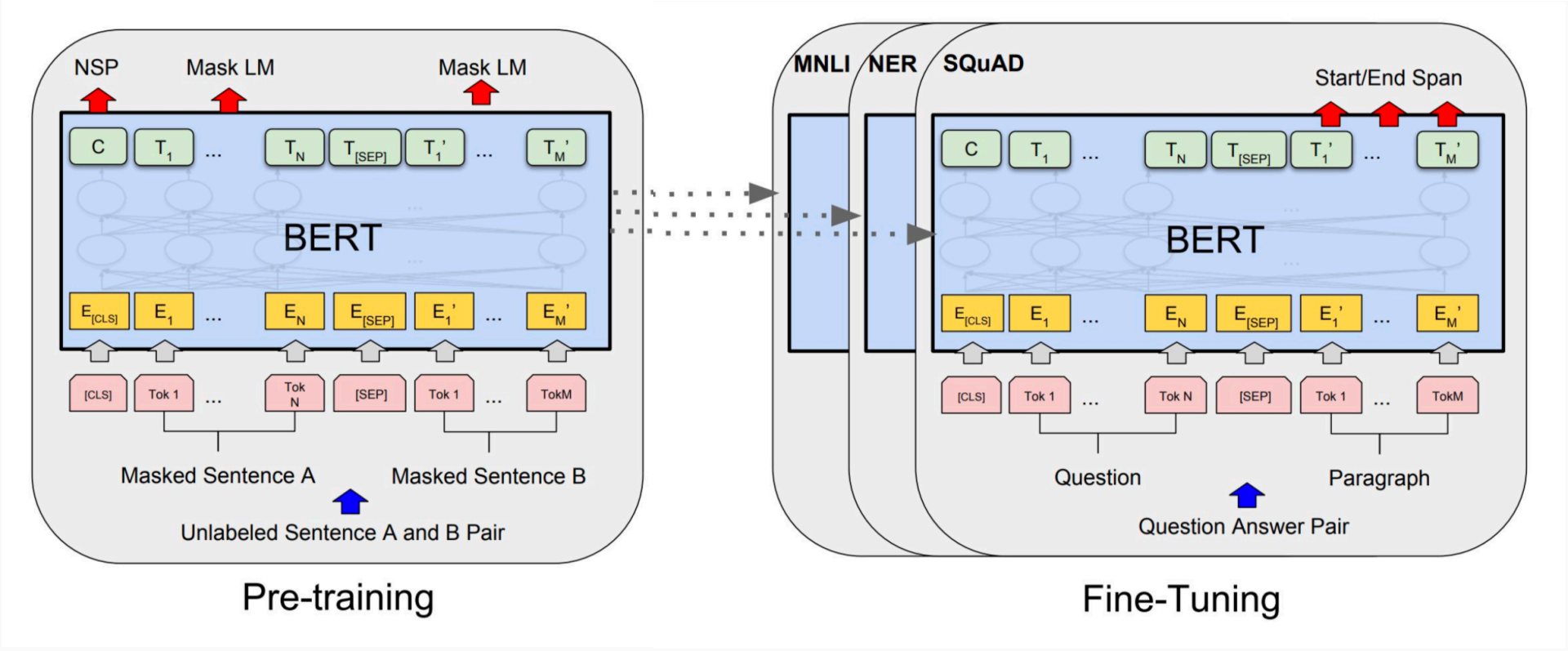
So attention is like bilinear: $x_i^{(t)}(W_j^Q(W_j^K)^T)x_i^{(l)}$



BERT model

Pre-train contextual word vectors in a LM-like way with transformers

Learn a classifier built on the top layer for each task that you fine tune for



SQuAD Question Answering leaderboard 2017-02-07

Passage

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

Question: Which team won Super Bowl 50?

System	F1
Human performance	91.2
r-net (MSR Asia) [Wang et al., ACL 2017]	79.7
DrQA (Chen et al. 2017)	79.4
Multi-Perspective Matching (IBM)	78.7
BiDAF (UW & Allen Institute)	77.3
Fine-Grained Gating (Carnegie Mellon U)	73.3
Logistic regression	51.0

SQuAD 2.0 Question Answering leaderboard 2019-02-07

Passage

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

Question: Which team won Super Bowl 50?

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jan 15, 2019	BERT + MMFT + ADA (ensemble) *Microsoft Research Asia*	**85.082**	**87.615**
2 Jan 10, 2019	BERT + Synthetic Self-Training (ensemble) *Google AI Language* https://github.com/google-research/bert	84.292	86.967
3 Dec 13, 2018	BERT finetune baseline (ensemble) *Anonymous*	83.536	86.096
4 Dec 16, 2018	Lunet + Verifier + BERT (ensemble) *Layer 6 AI NLP Team*	83.469	86.043
4 Dec 21, 2018	PAML+BERT (ensemble model) *PINGAN GammaLab*	83.457	86.122
5 Dec 15, 2018	Lunet + Verifier + BERT (single model) *Layer 6 AI NLP Team*	82.995	86.035

SQuAD 2.0 Question Answering leaderboard 2019-10-09

Passage

Super Bowl 50 was an American football game to determine the champion of the National Football League (NFL) for the 2015 season. The American Football Conference (AFC) champion Denver Broncos defeated the National Football Conference (NFC) champion Carolina Panthers 24–10 to earn their third Super Bowl title. The game was played on February 7, 2016, at Levi's Stadium in the San Francisco Bay Area at Santa Clara, California.

Question: Which team won Super Bowl 50?

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Sep 18, 2019	ALBERT (ensemble model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	**89.731**	**92.215**
2 Jul 22, 2019	XLNet + DAAF + Verifier (ensemble) *PINGAN Omni-Sinitic*	88.592	90.859
2 Sep 16, 2019	ALBERT (single model) *Google Research & TTIC* https://arxiv.org/abs/1909.11942	88.107	90.902
2 Jul 26, 2019	UPM (ensemble) *Anonymous*	88.231	90.713
3 Aug 04, 2019	XLNet + SG-Net Verifier (ensemble) *Shanghai Jiao Tong University & CloudWalk* https://arxiv.org/abs/1908.05147	88.174	90.702
4 Aug 04, 2019	XLNet + SG-Net Verifier++ (single model) *Shanghai Jiao Tong University & CloudWalk* https://arxiv.org/abs/1908.05147	87.238	90.071
5 Jul 26, 2019	UPM (single model) *Anonymous*	87.193	89.934
6 Mar 20, 2019	BERT + DAE + AoA (ensemble) *Joint Laboratory of HIT and iFLYTEK Research*	87.147	89.474
6 Jul 20, 2019	RoBERTa (single model) *Facebook AI*	86.820	89.795

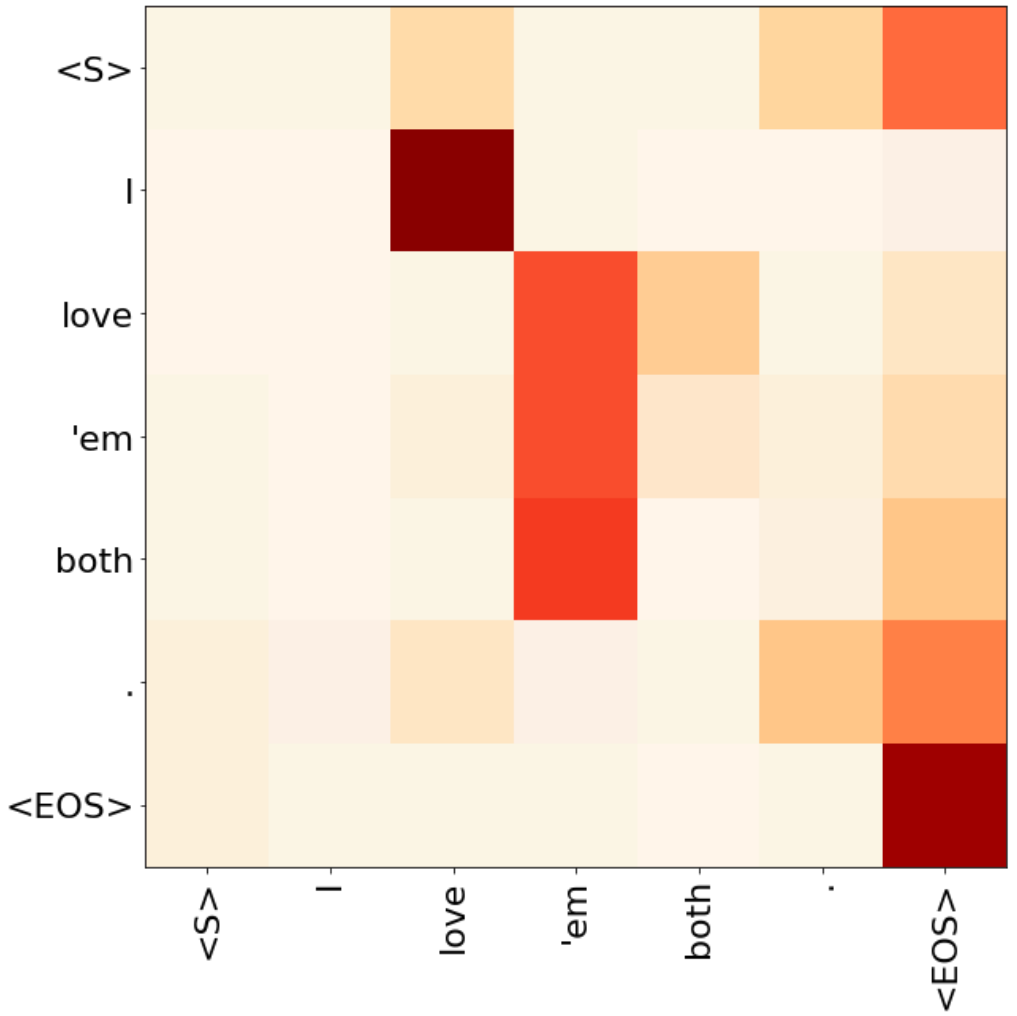
3. What does BERT know? Observational evidence

Kevin Clark, Urvashi Khandelwal, Omer Levy, & Christopher Manning (BlackBoxNLP 2019 workshop at ACL 2019 best paper)

- BERT works really well and calculates clearly useful context-dependent word representations
- Directly observe what BERT is looking at
- We find that BERT induces a lot of structure similar to conventional linguistic structure … because it helps predict

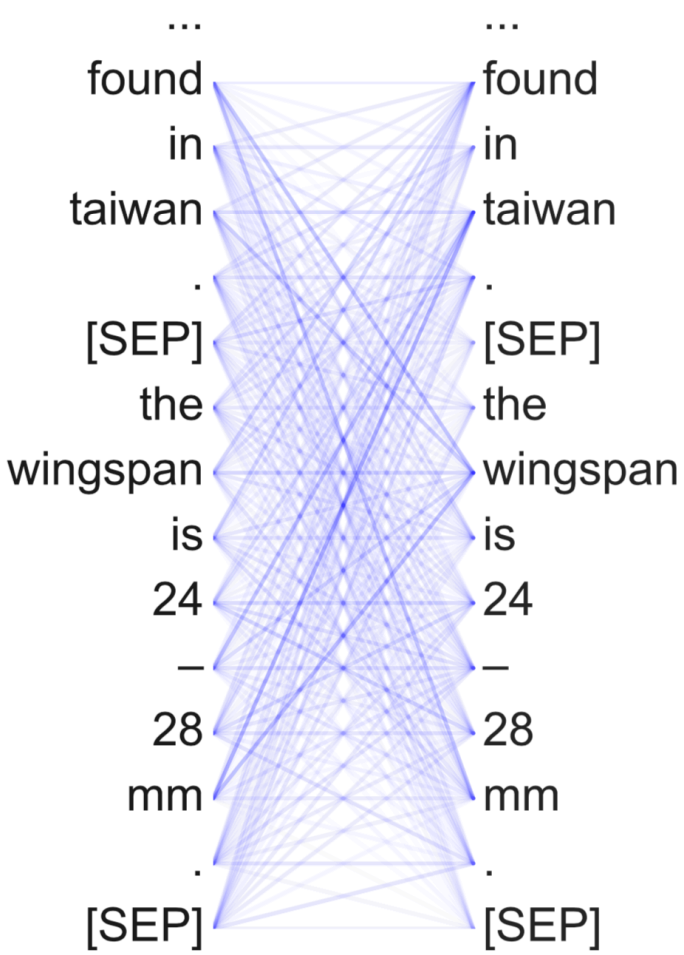
BERT Attention Heads

- For each of many attention heads, for each word position, see where BERT pays attention
- Look at the most-attended-to word for each head
- How does what BERT attends to correspond to linguistics?



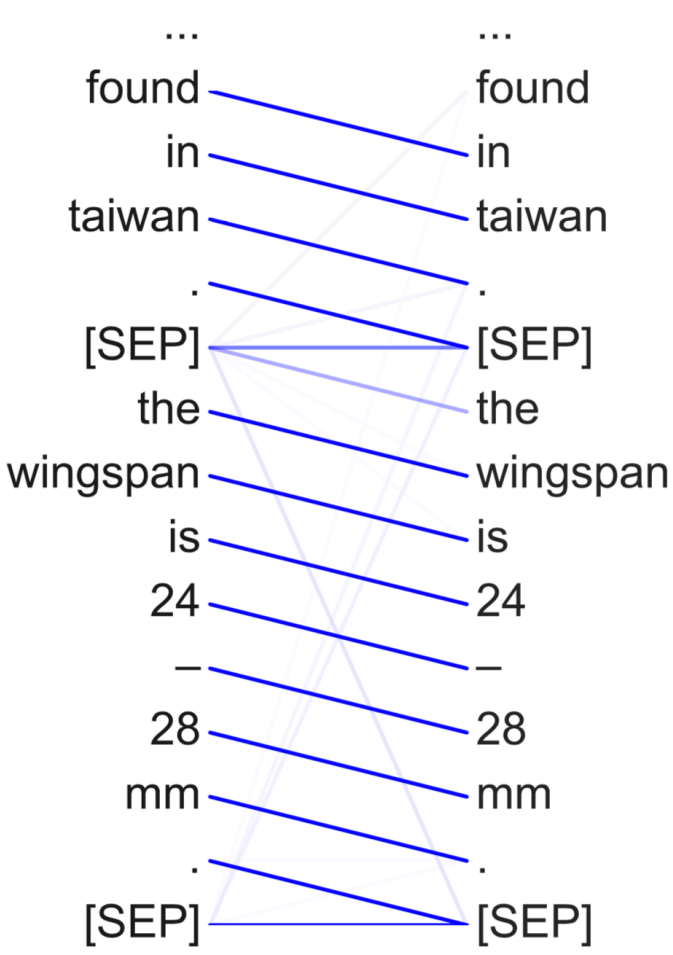
What do BERT attention heads do?

1-1: Attend broadly (“BoW head”)

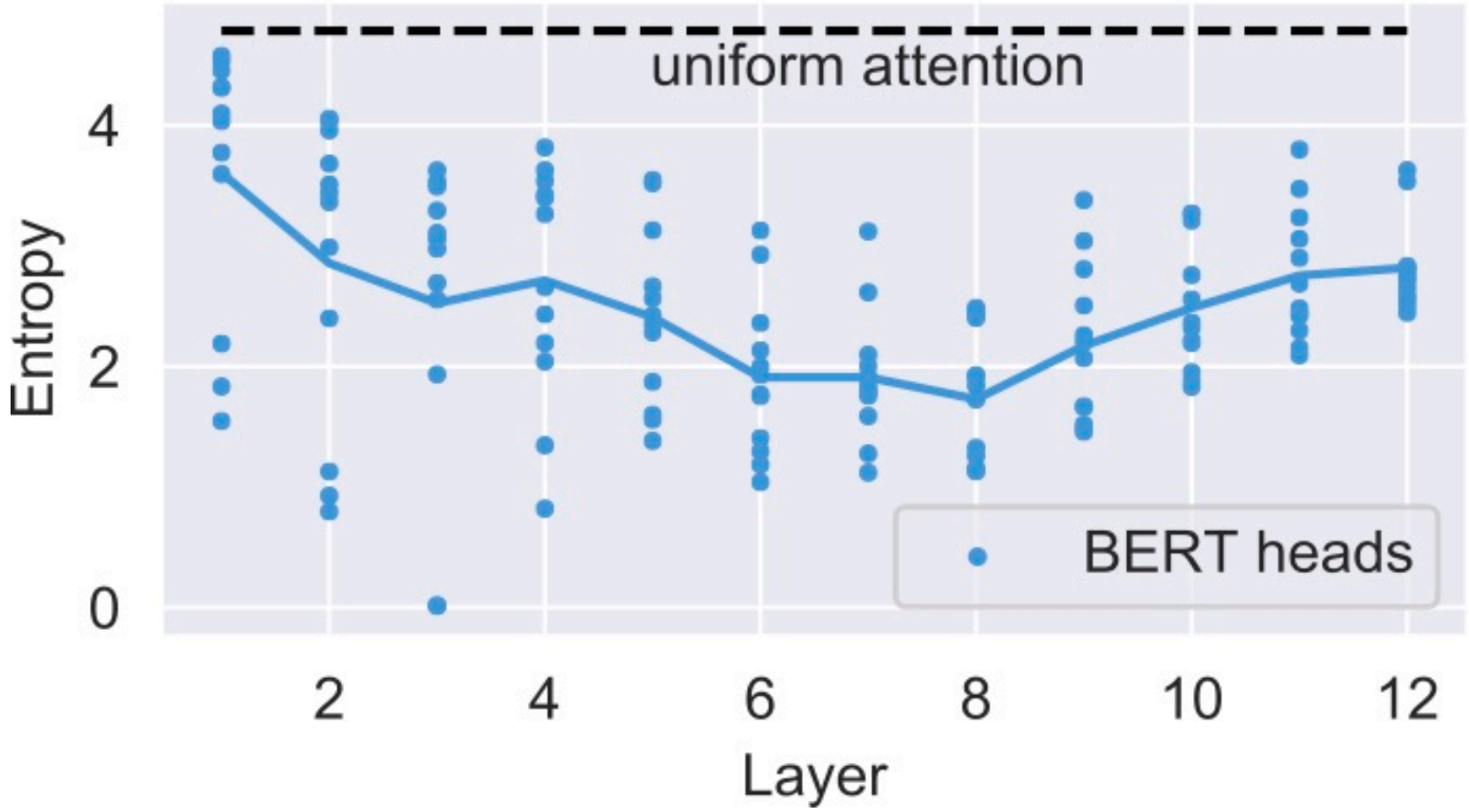


word attention target

3-1: Attend to next (or prev) word

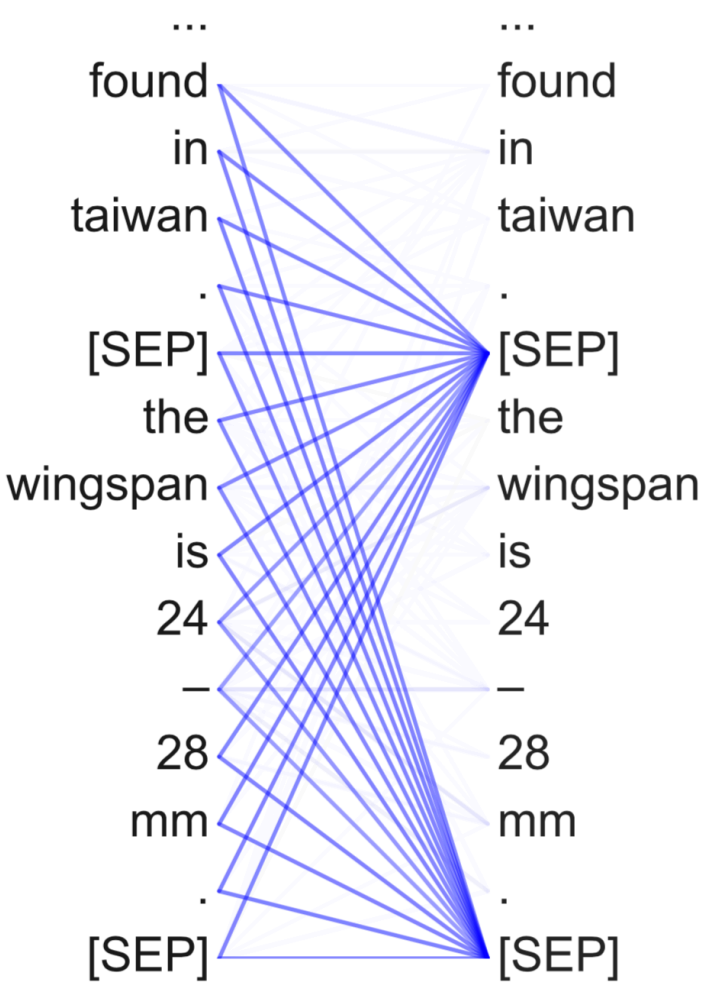


First layer heads mainly average

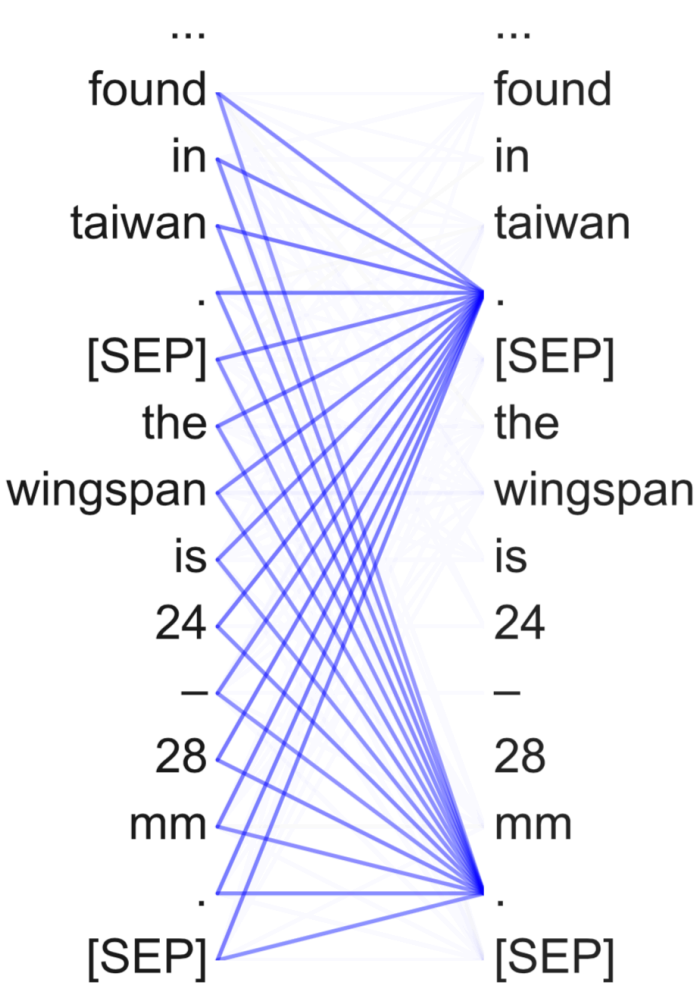


What do BERT attention heads do?

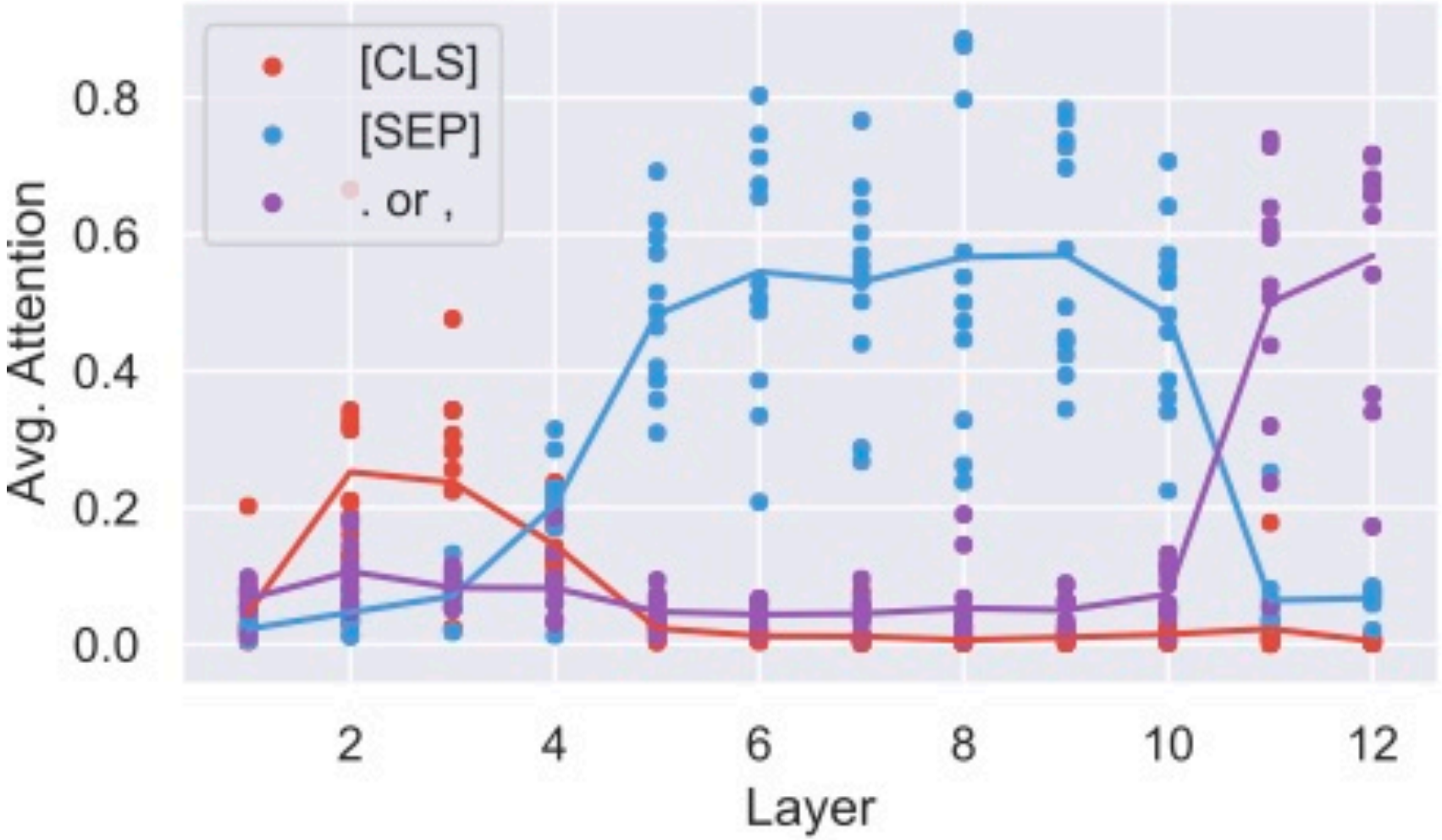
8-7: Attend to SEP (or CLS)



11-6: Attend to periods (or commas)

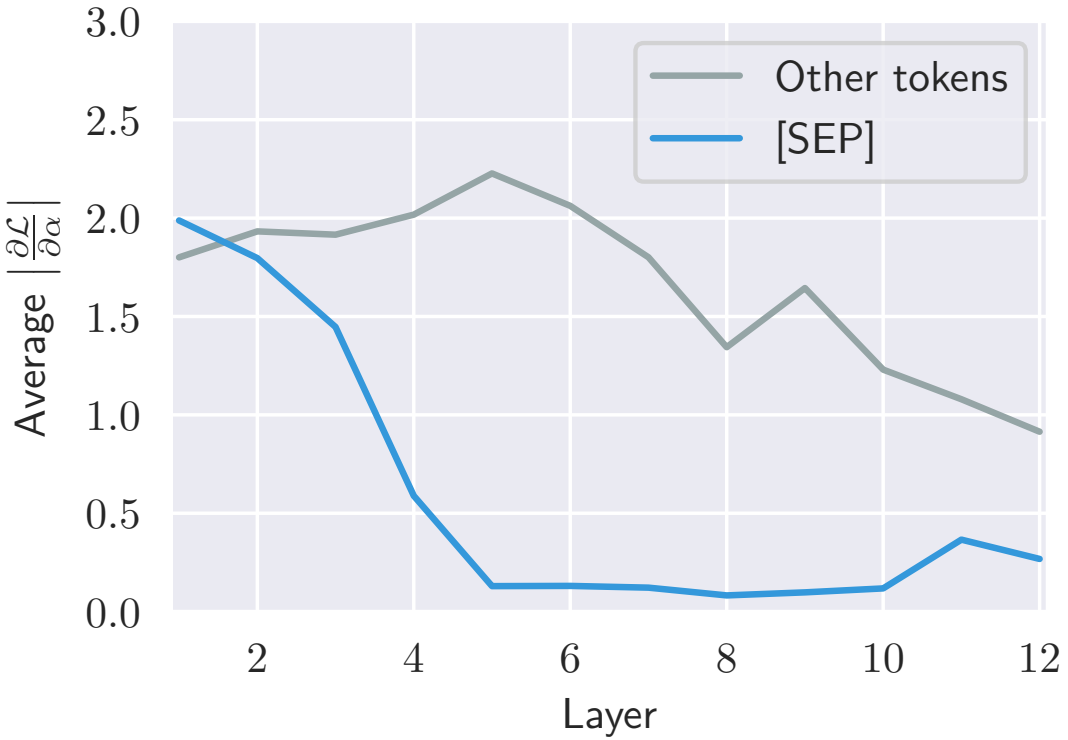


Many heads much of the time attend to special tokens

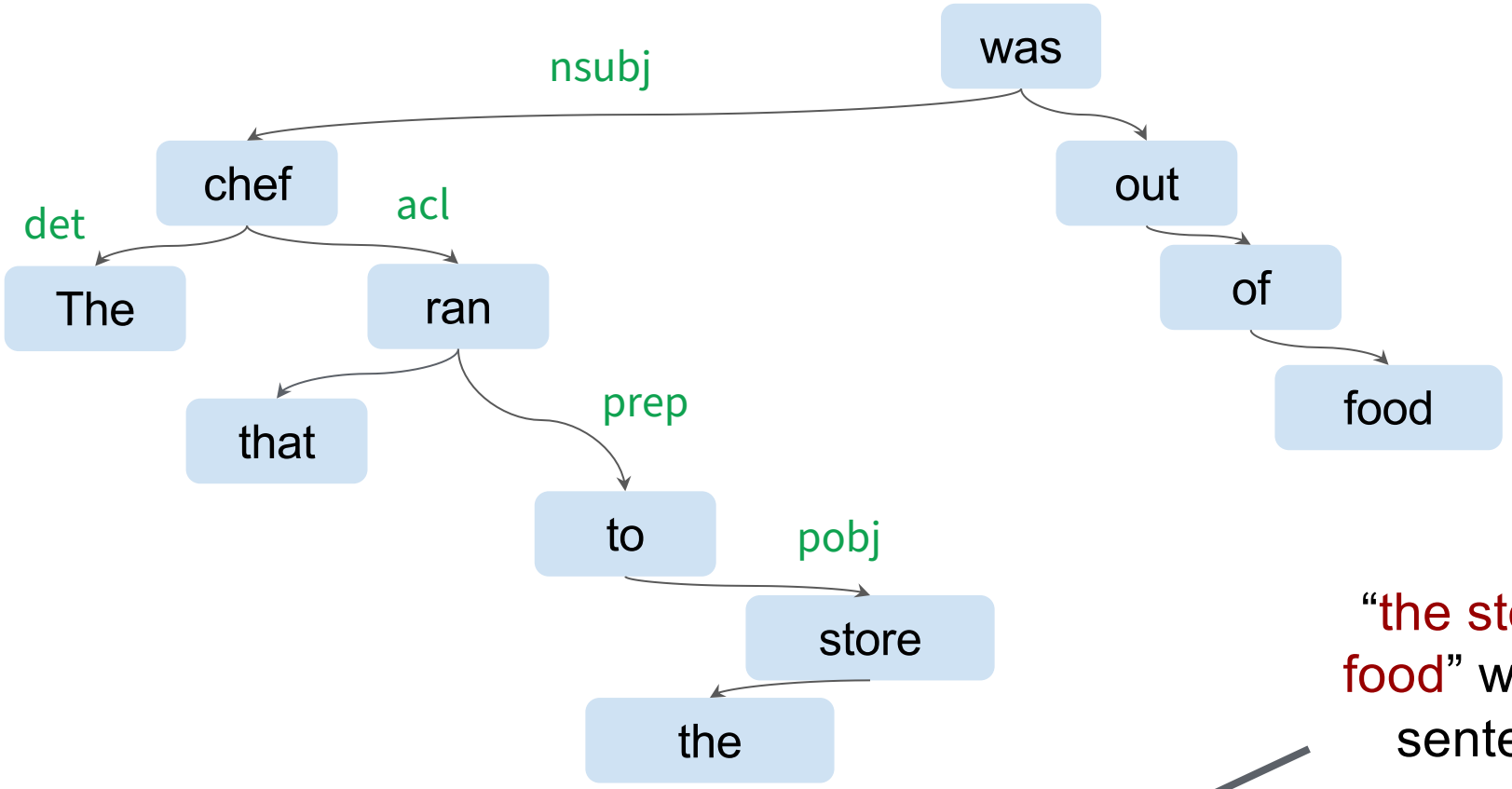


Attention to [SEP] is used as a no-op when a feature isn't firing

Gradient-based importance measure: how much does increasing attention to this token change BERT's output?



A sentence's meaning is composed via its syntax tree

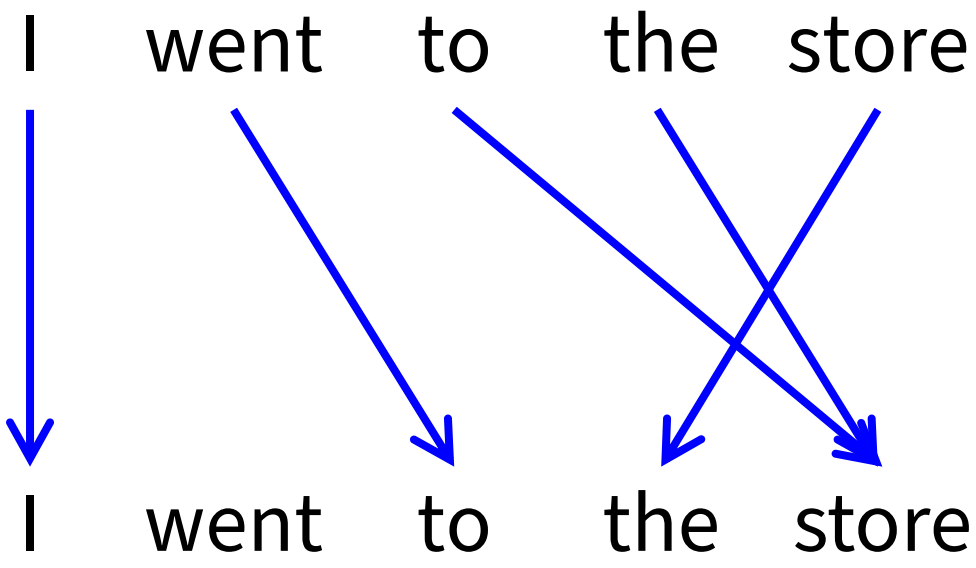


"the store was out of food" would be a valid sentence by itself

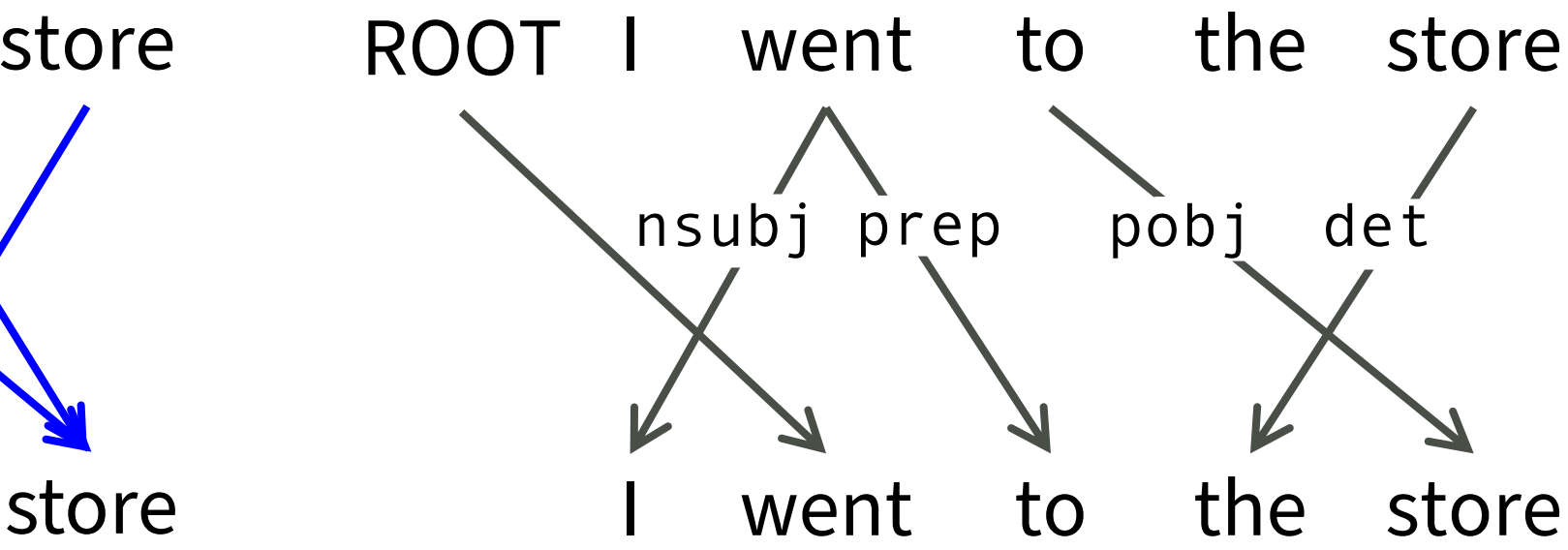
The chef that ran to ~~the store~~ was out of food

The chef that ran to the store was out of food

Does some of BERT attention resemble dependency syntax?

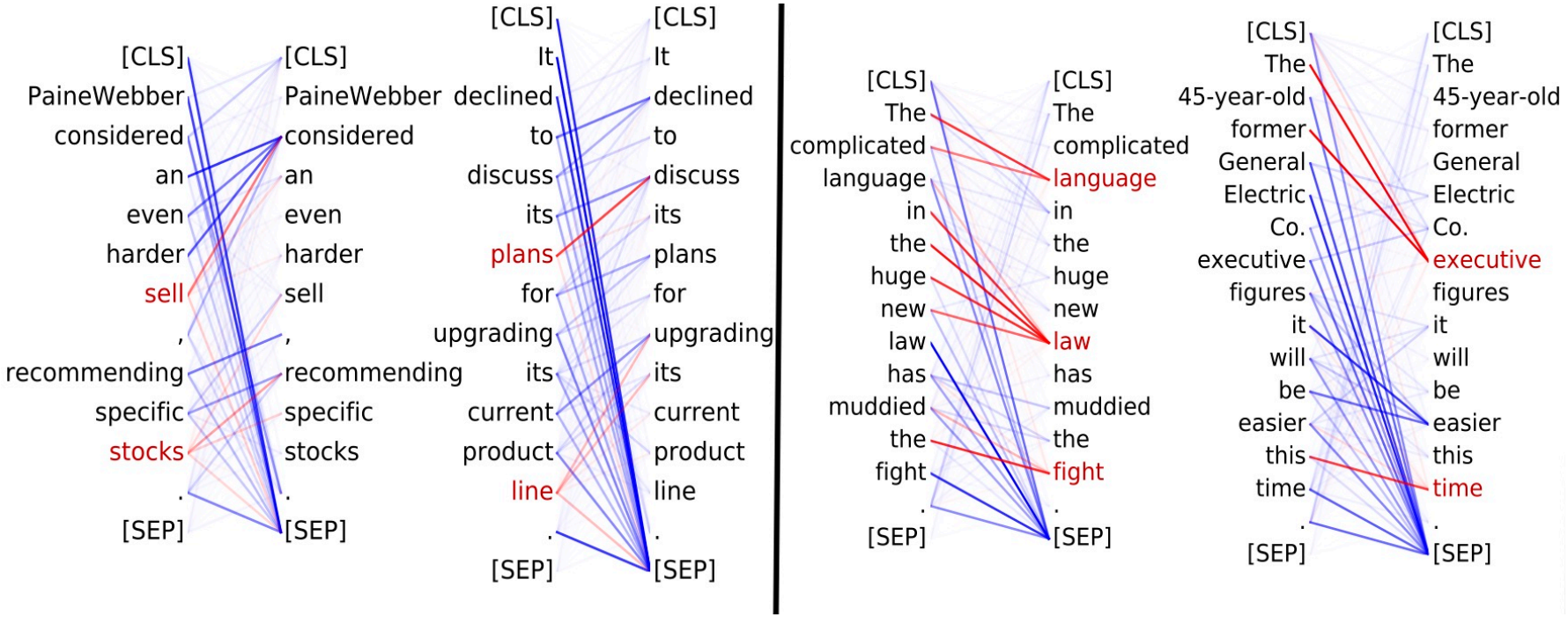


Take the most-attended-to words



Compare with dependency tree

A bunch of heads specialize on a syntactic relation (!)



Head 8-10
Direct objects attend to verbs
86.8% on dobj relation

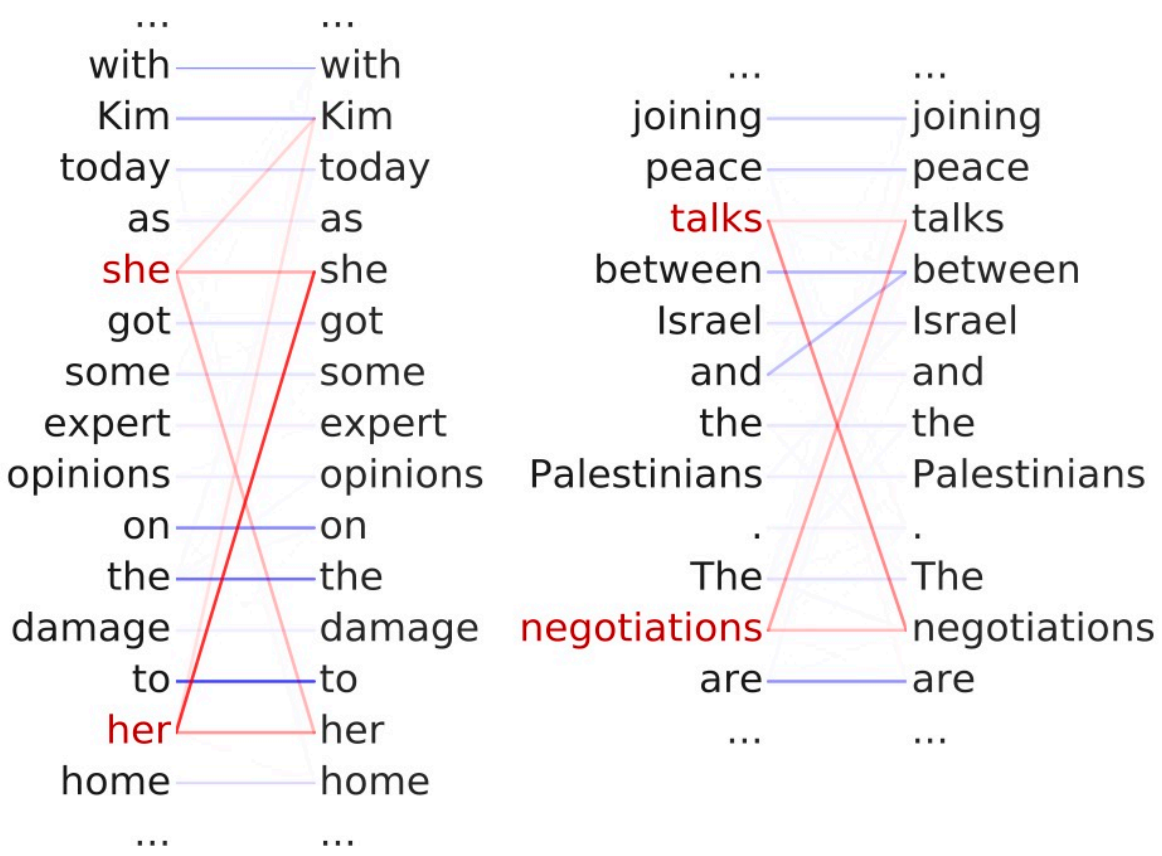
Head 8-11
Noun modifiers (det, adj) attend to head noun
94.3% on det relation

Overall, a combination of these heads can give an okay dependency parser: 77 UAS
(Cf. 26 from right branching, 58 from GloVe word vecs + distance.)

BERT attention heads capture many dependency relations remarkably well

Relation	Best head's accuracy	Best baseline's accuracy
ALL	35	26
pobj	**76**	**35**
det	**94**	**52**
dobj	**87**	**40**
poss	**81**	**48**
auxpass	**83**	**41**

There’s a coreference head (!)



Coreferent mentions attend to their antecedent; for not a mention words: no-op attention 85% on [SEP].
Head 5-4: **65.1%** accuracy at linking to head of antecedent
Cf. vs. 69% for a 4-sieve, rule-based system (cf. Lee et al. 2011)
choosing nearest {full string, headword, PNG match; any NP}

4. What does BERT know? Experimental evidence

Hewitt and Manning (NAACL 2019)

tl;dr

Does BERT encode syntax (dependency trees) in its contextual representations?

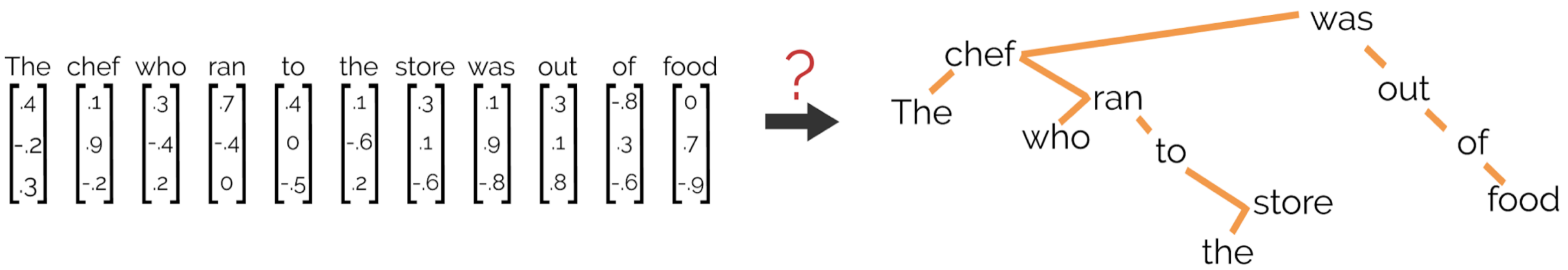
Yes, approximately

How can we tell whether its vector representations encode trees?

Using a **structural probe** to look at the geometry

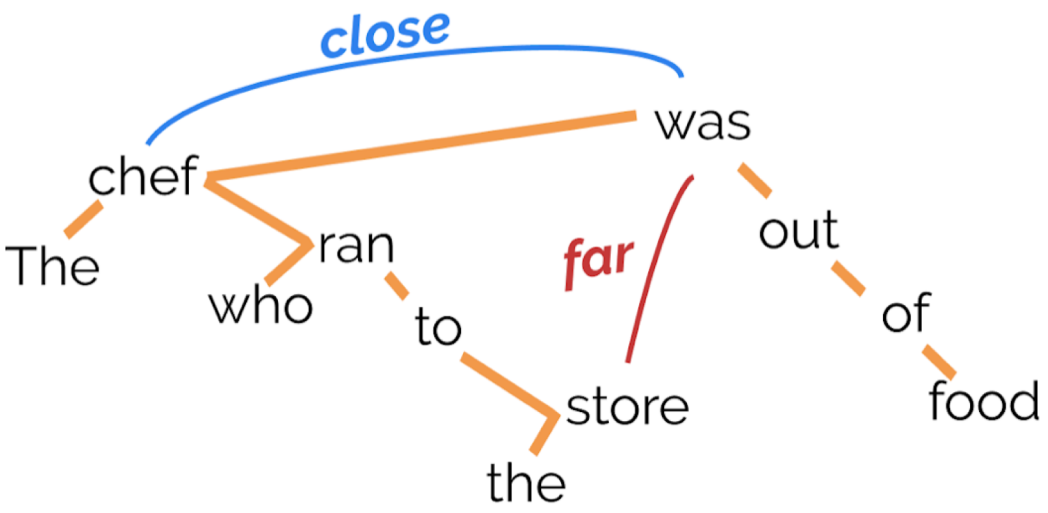
Are vector spaces and trees reconcilable?

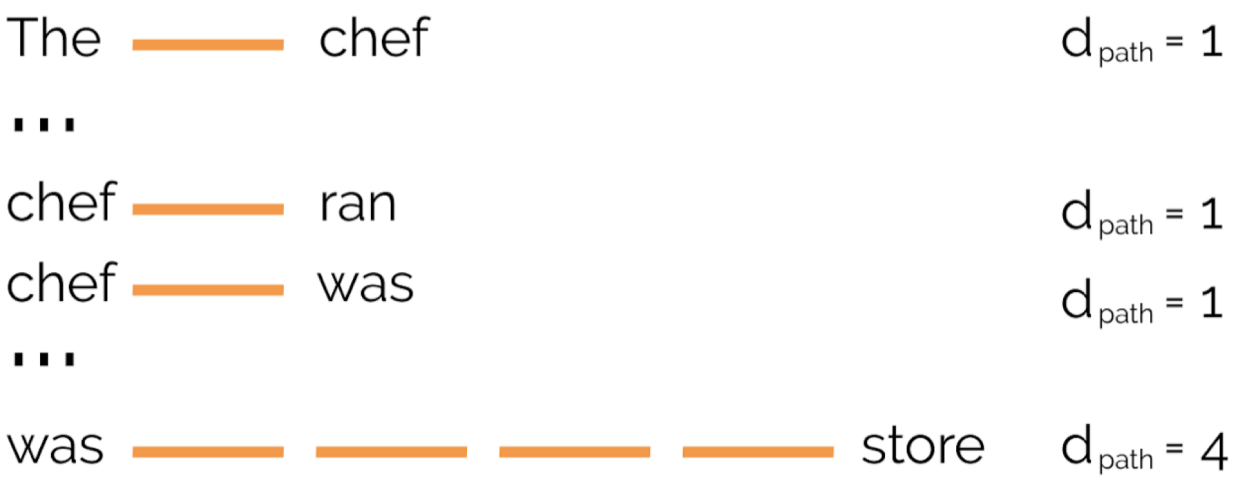
Are the vector space representations in NLP reconcilable with the discrete syntactic tree structures hypothesized for language?



Distance metrics unify trees and vectors

An **undirected tree** defines a **distance metric** on pairs of words, the path metric: the number of edges in the path between the words.

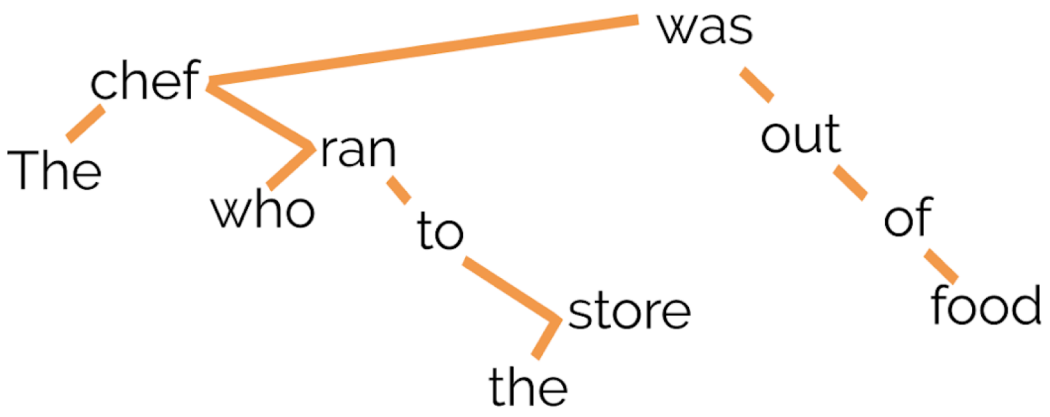


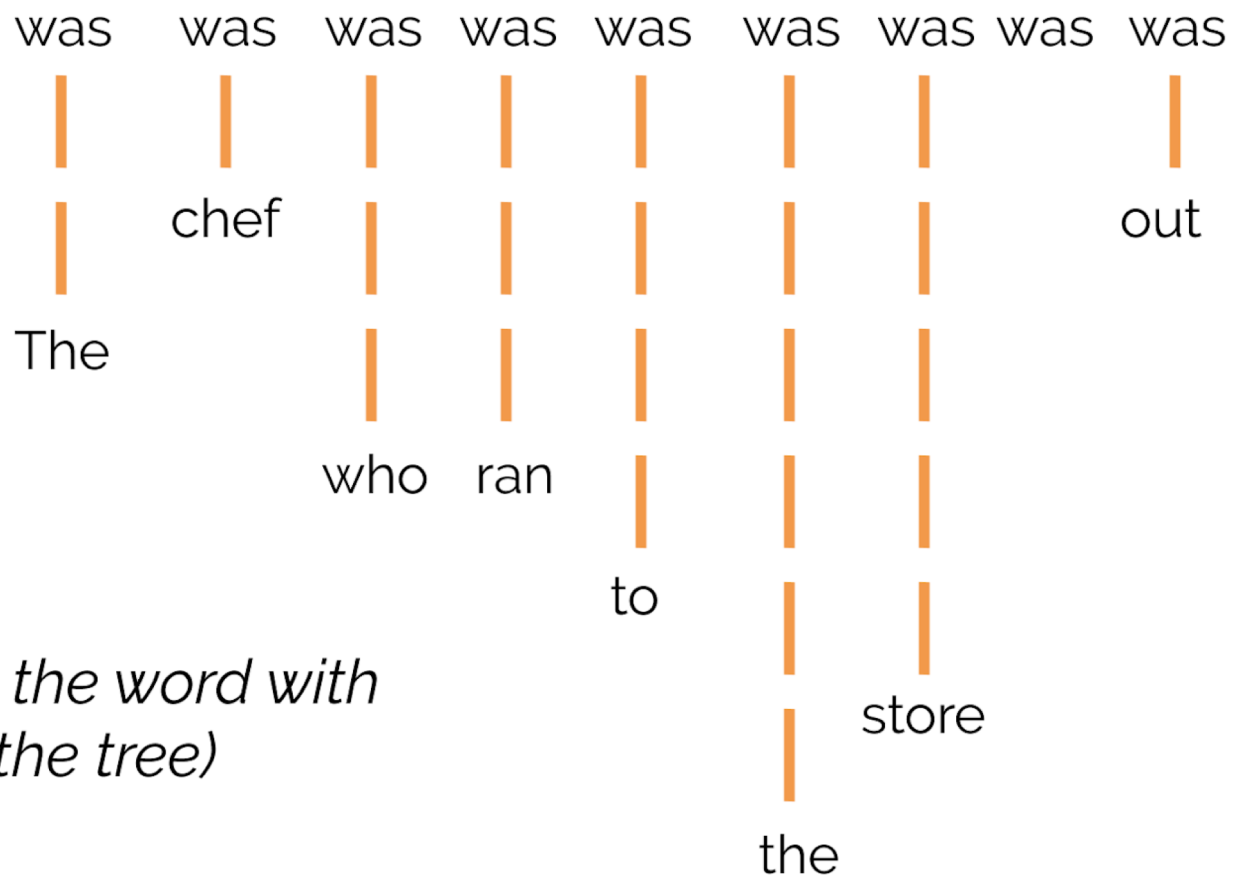


The edges of the tree can be recovered by looking at all distance=1 pairs.

Norms unify edge directions and vectors

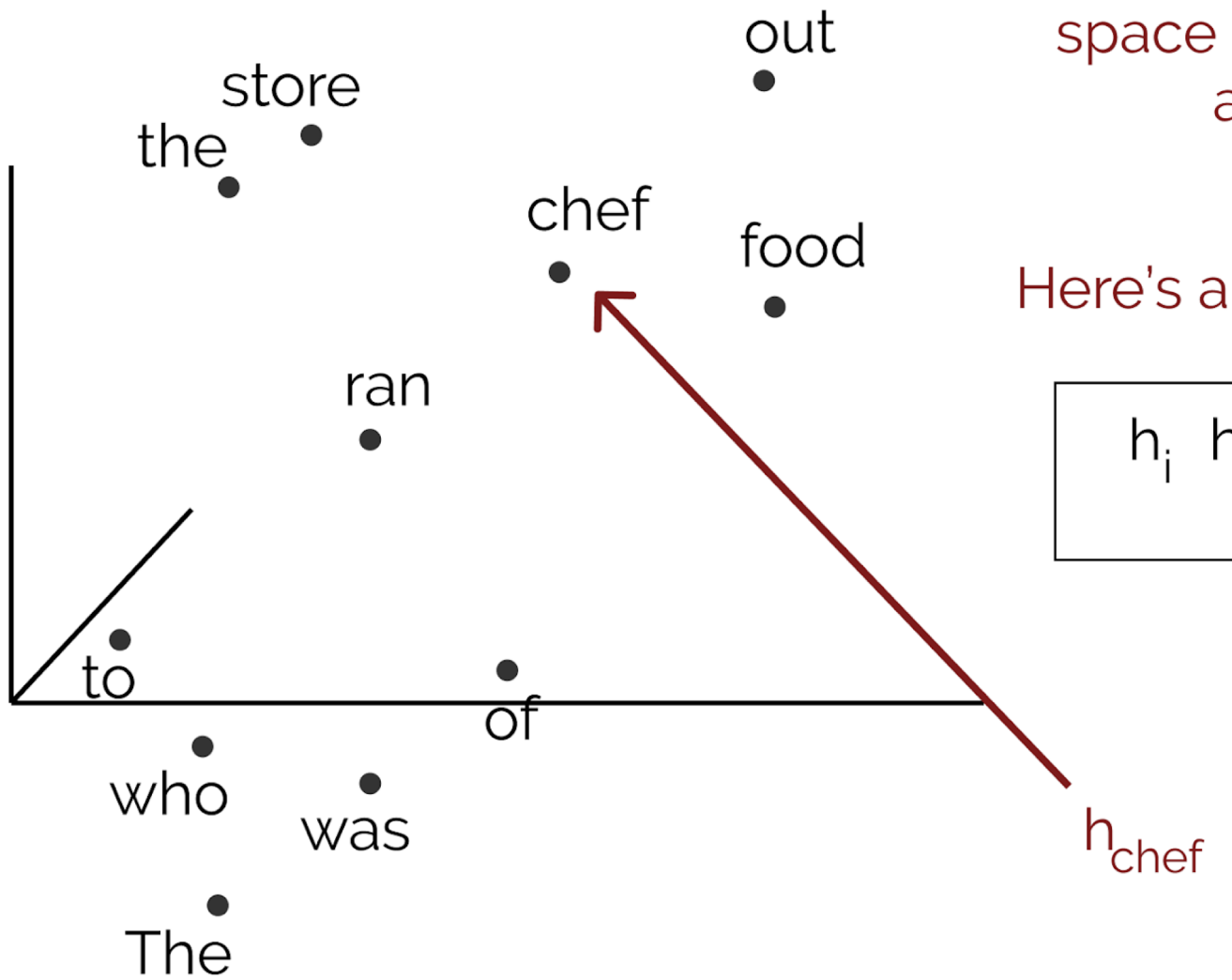
A **rooted tree** defines a **norm** on the words, the parse depth:
the number of edges from each word to ROOT.





Each edge is directed towards the word with greater norm (deeper in the tree)

Finding trees in vector spaces

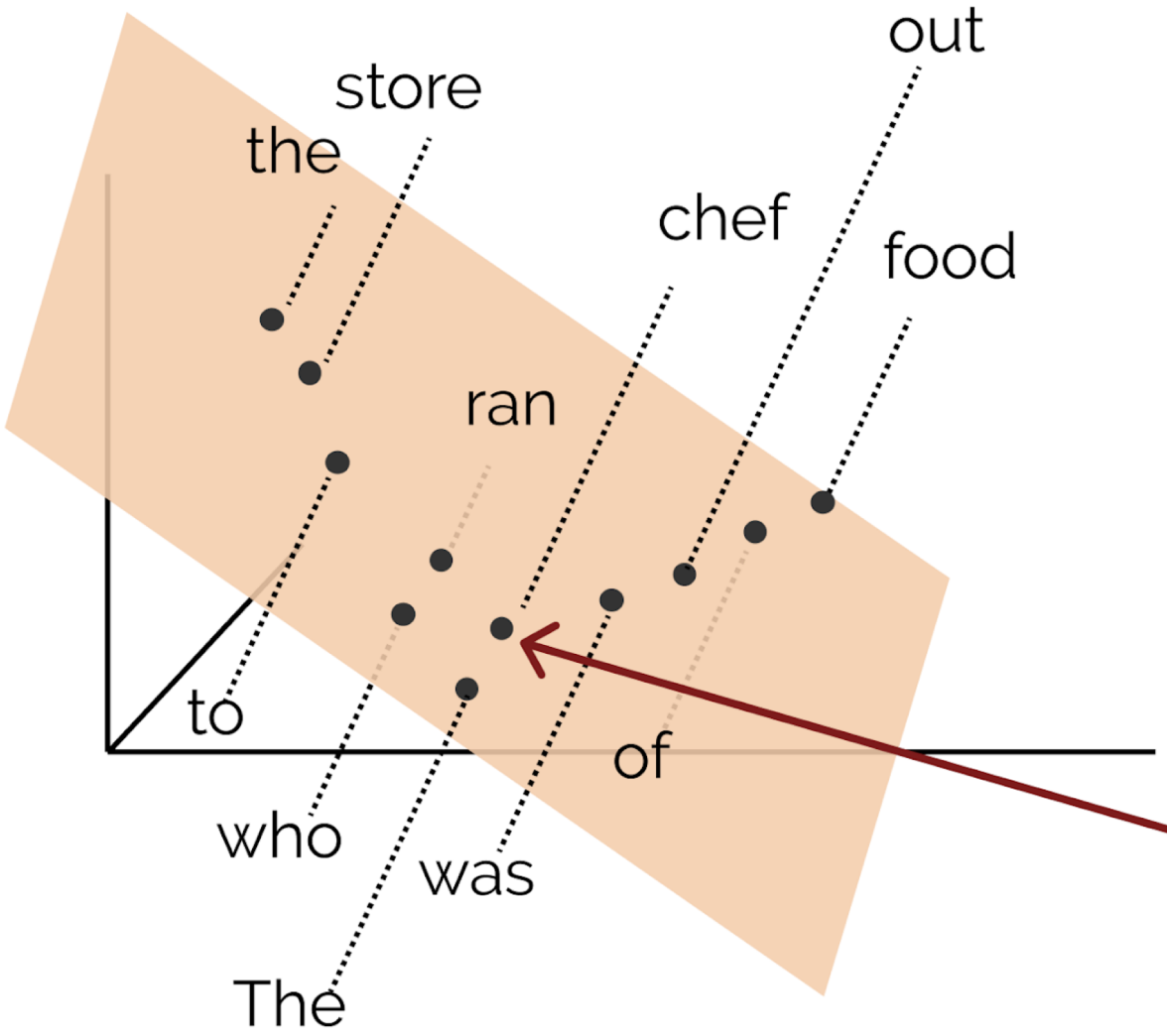


We can look for trees in the vector space by looking for their **distances** and **norms** in the space.

Here's a sentence embedded by a NN!

h_i h_j : vector representation of words *i* and *j*.

Finding trees in vector spaces



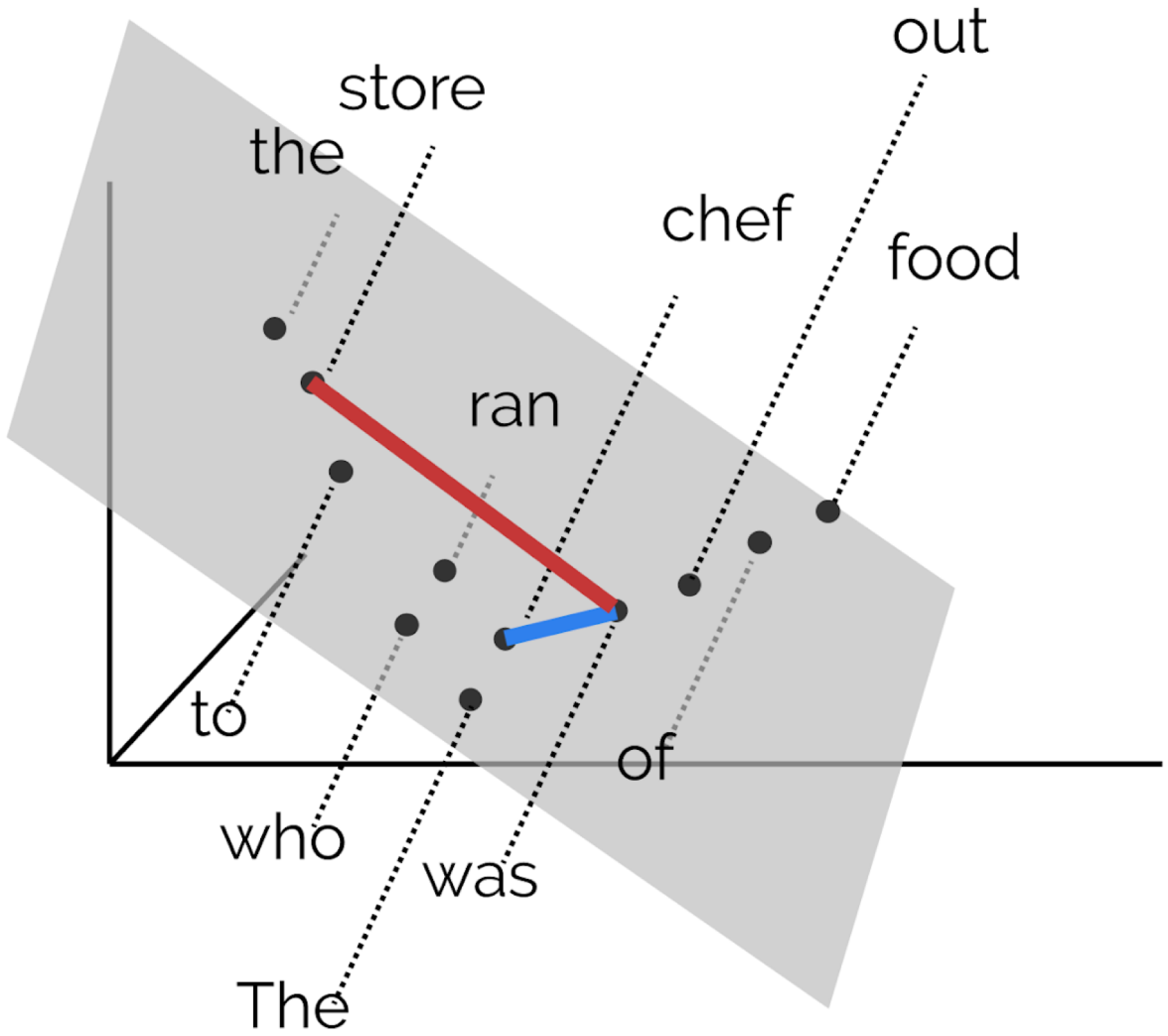
We don't expect all dimensions of the vector space to encode syntax -- NNs have a lot to encode!

We find the linear transformation that encodes syntax best.

B : The syntax transformation matrix

Bh_i : Syntax-transformed vector word representation

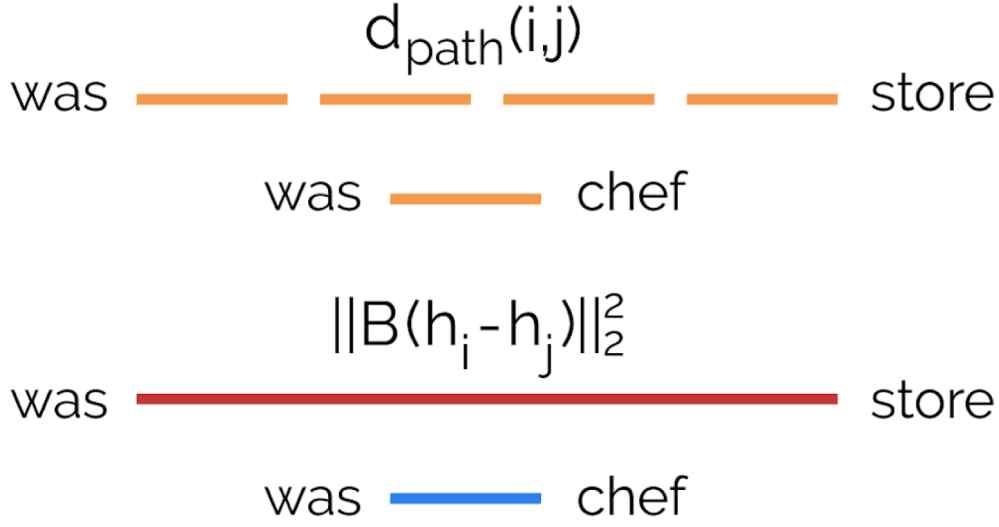
Finding trees in vector spaces



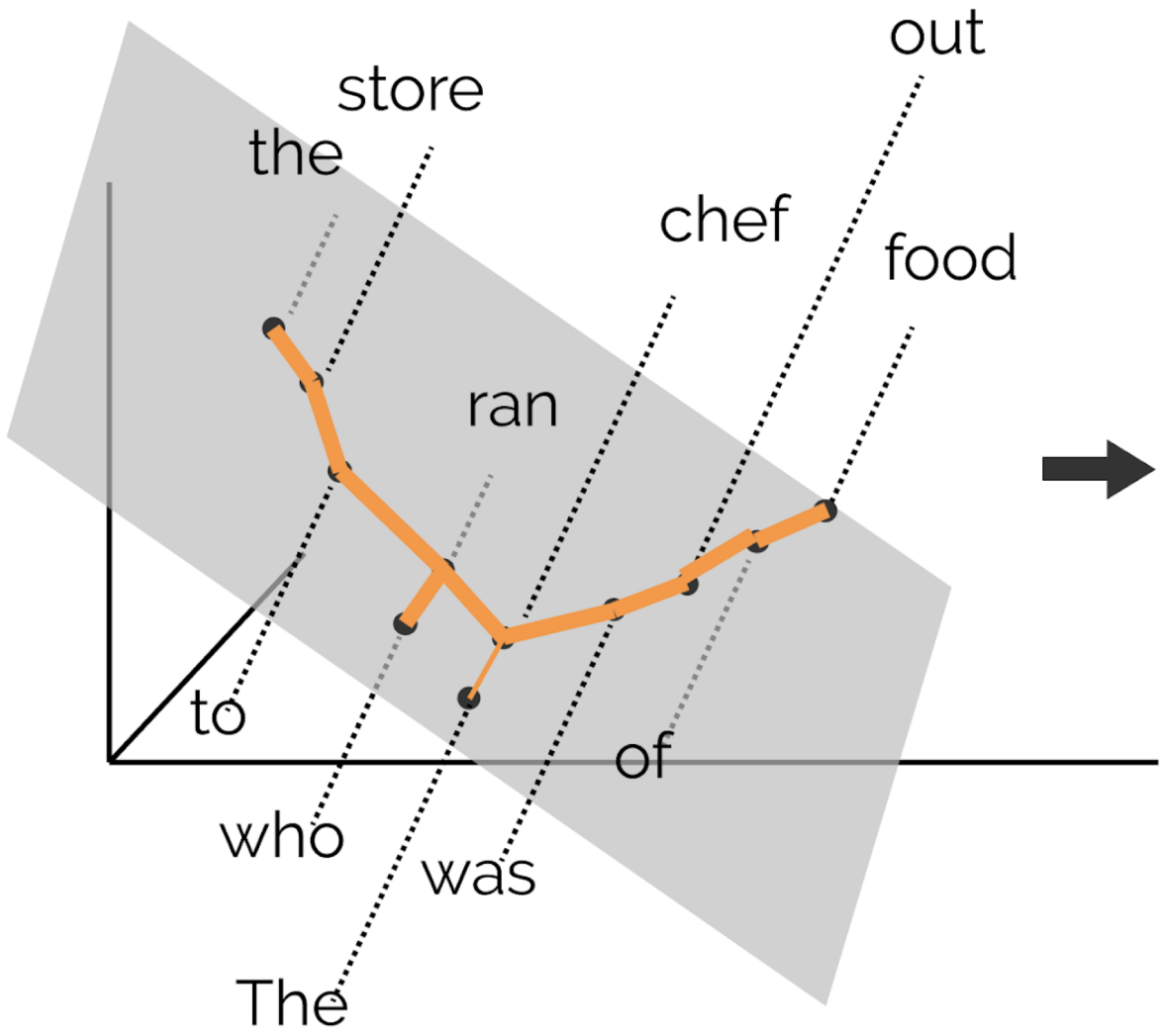
In the transformed space, (squared) L2 distance approximates tree distance.

$d_{path}(i,j)$: Tree path distance

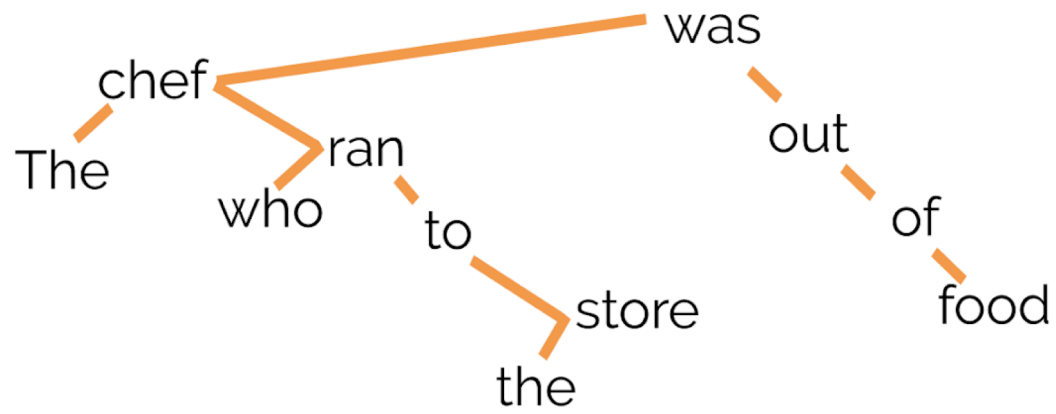
$||B(h_i - h_j)||_2^2$: Squared Vector space distance ($||h_i - h_j||_B^2$)



Finding trees in vector spaces



With this property, a minimum spanning tree in the vector space distance recovers the tree.



Does BERT encode undirected parse trees -> does there exist a *distance* transformation?

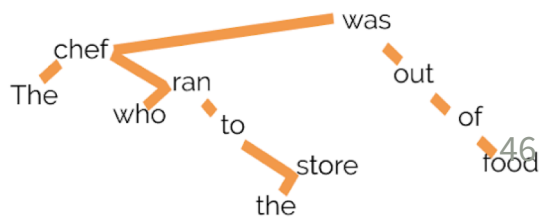
$$\arg\min_{B} \sum_{\ell \in \text{PTB}} \frac{1}{|s^\ell|^2} \sum_{i,j} \left| d^\ell_{\text{path}}(i,j) - \|B(h^\ell_i - h^\ell_j)\|_2^2 \right|$$

Find a single transformation *B*

such that over all sentences in PTB training

Over all word pairs in each sentence

The difference between **tree distance** and **squared vector distance** is *minimized*



Does BERT encode edge directions -> does there exist a *depth* transformation?

$$\arg\min_{B} \sum_{\ell \in \text{PTB}} \frac{1}{|s^\ell|} \sum_{i} \left| \text{depth}^\ell(i) - \|Bh_i^\ell\|_2^2 \right|$$

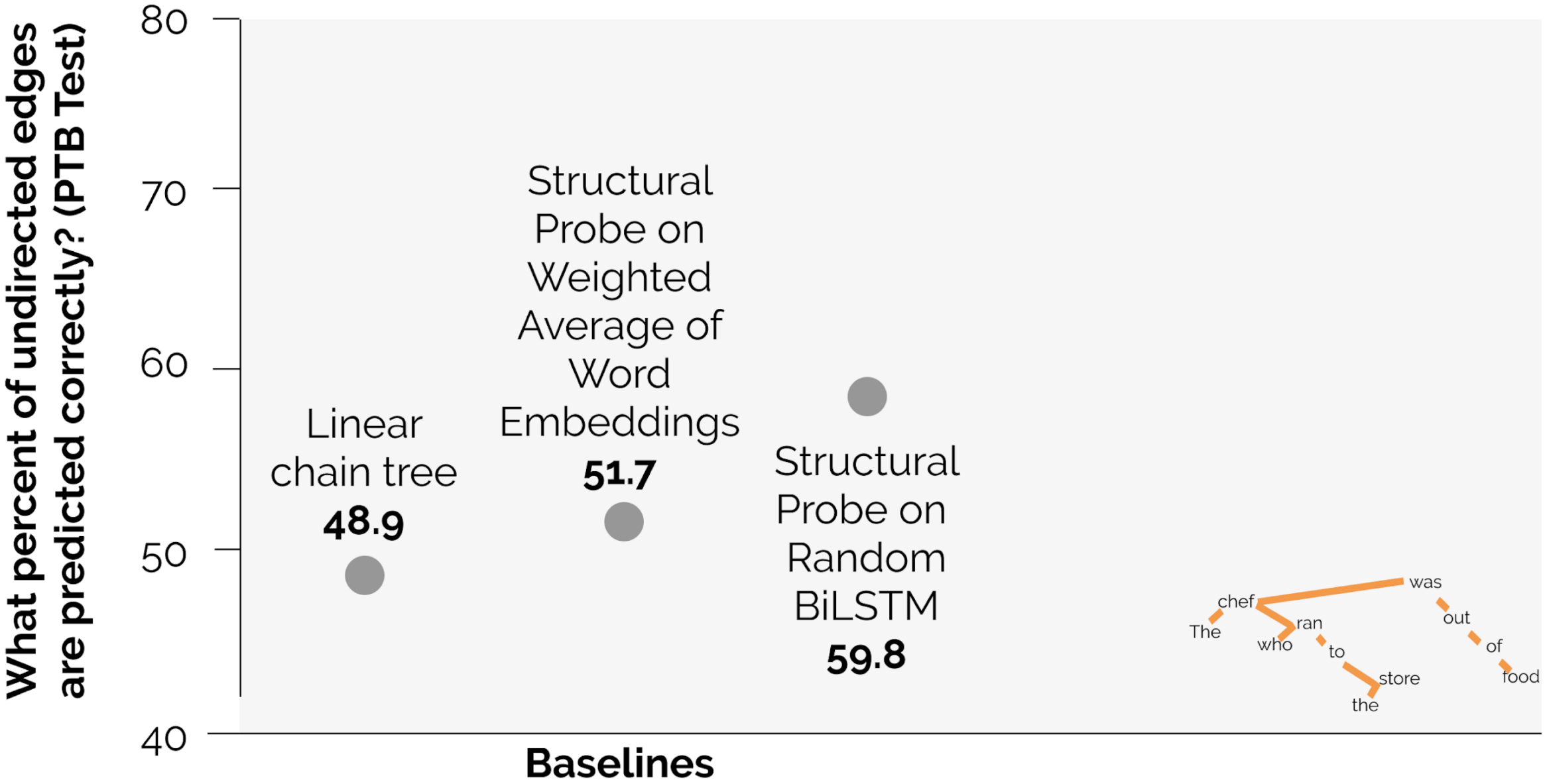
Find a single transformation *B*

such that over all sentences in PTB training

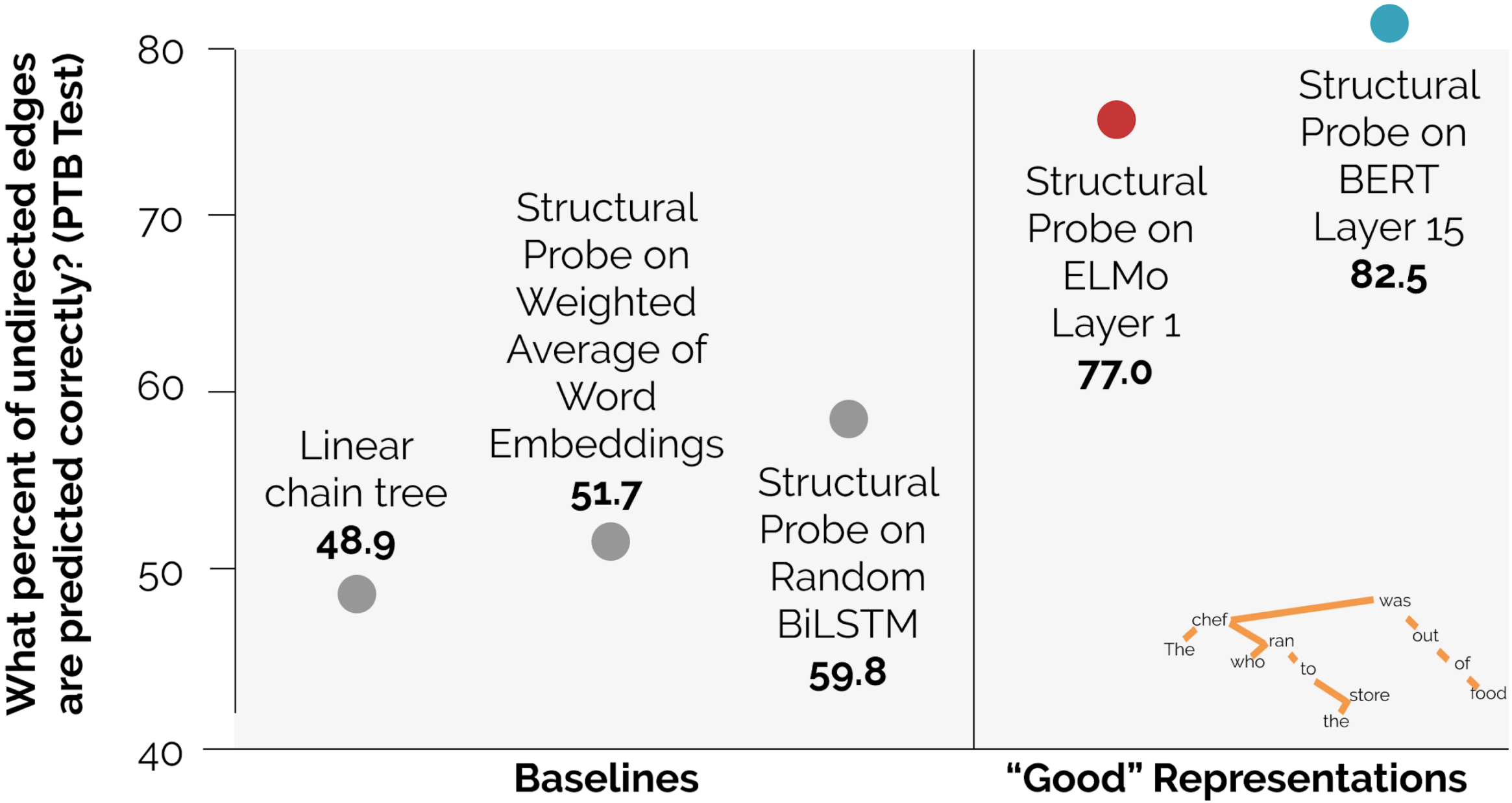
Over all words in each sentence

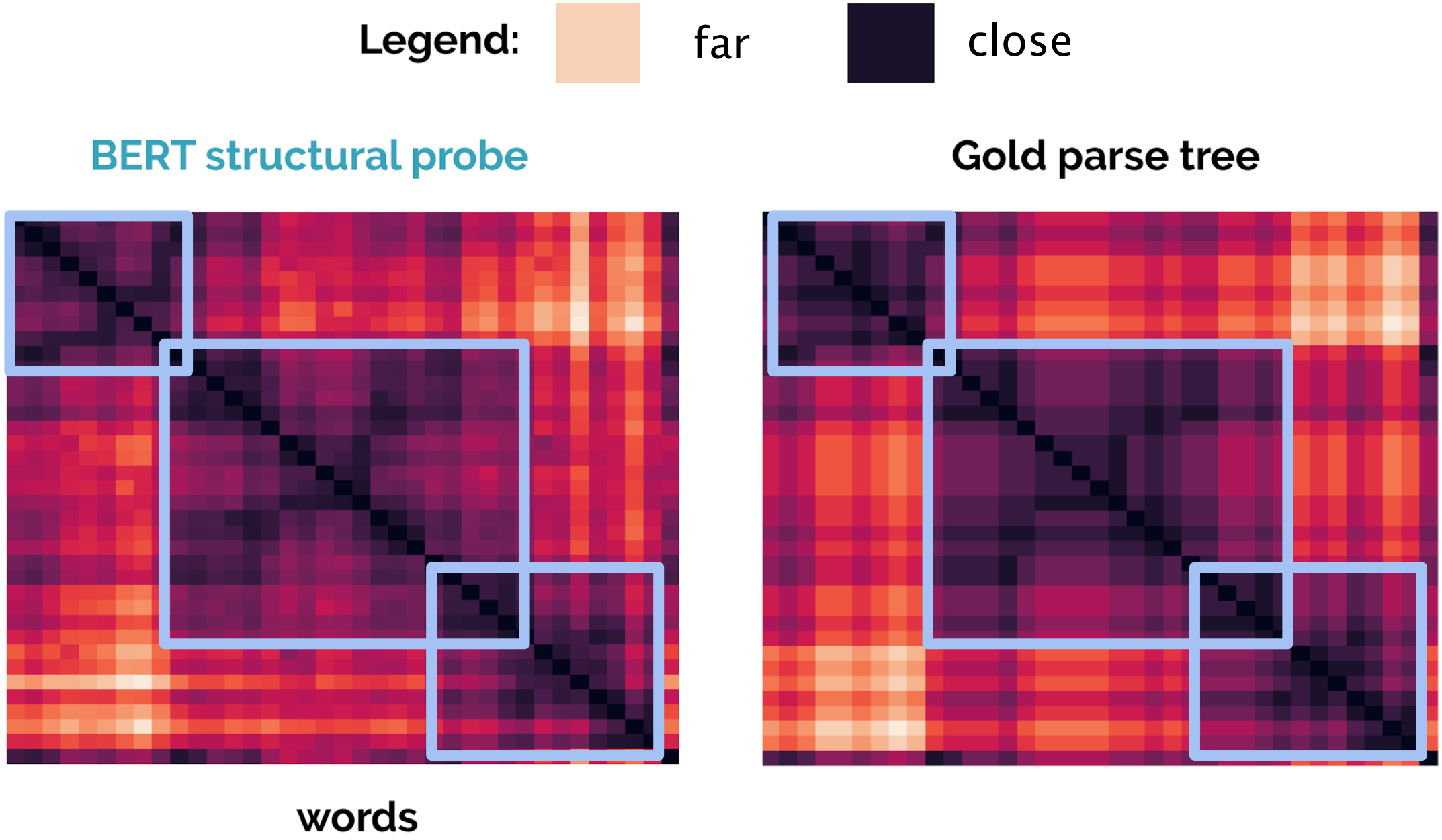
The difference between **tree depth** and **squared vector norm** is *minimized*

Trees aren't well-encoded in baselines

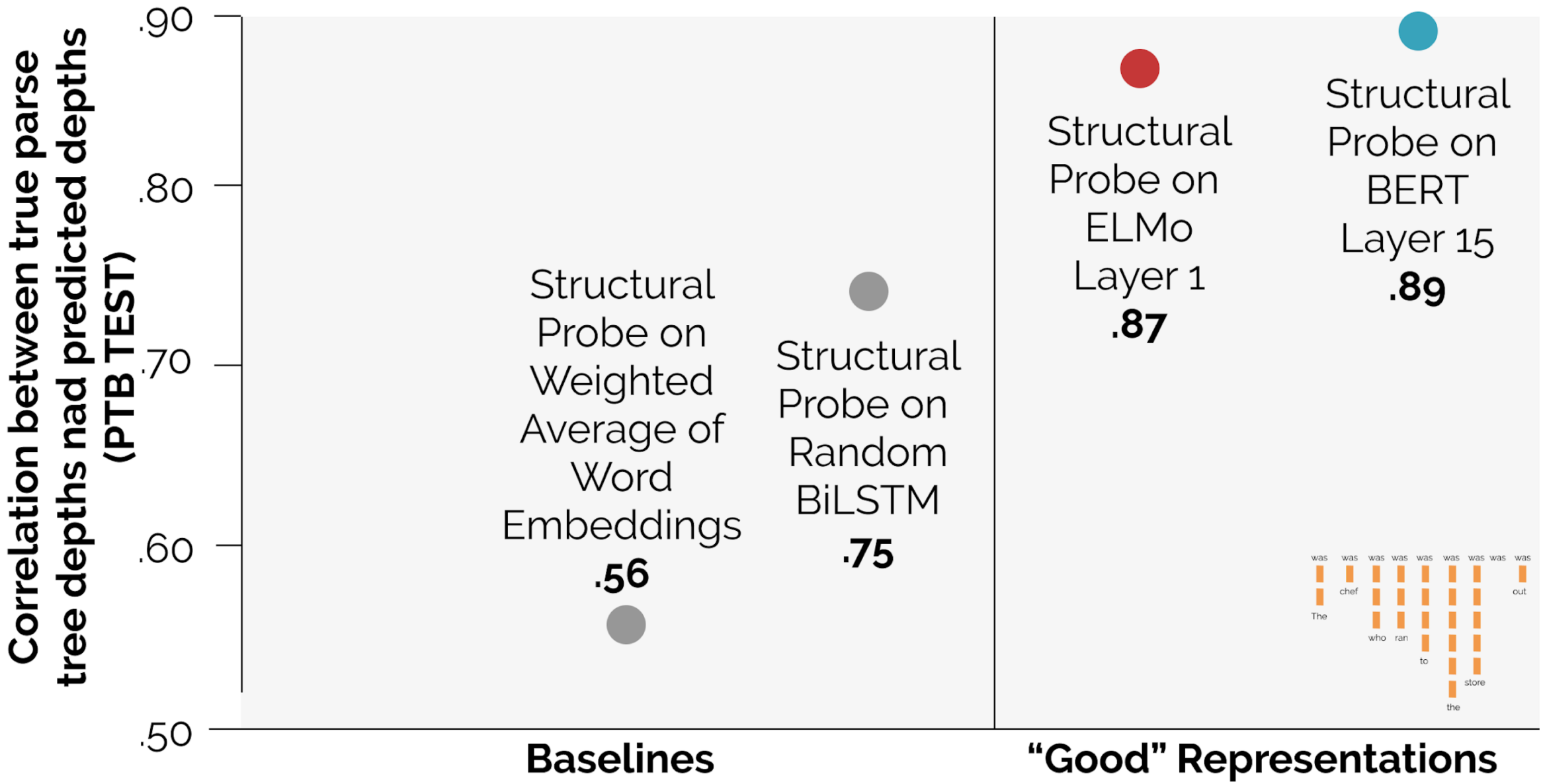


But they are in trained representations!




Legend:
far
close
BERT structural probe
Gold parse tree
words

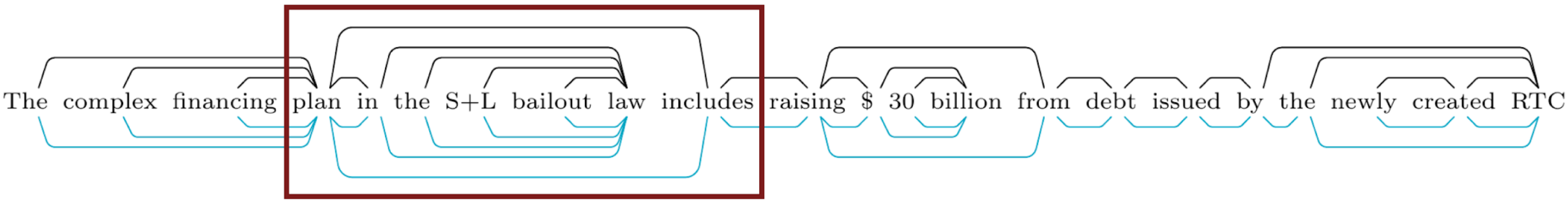
But it is in trained representations!



Trees from structural probe parse distances approximate parse trees pretty well!

Black (above sentence): Human-annotated parse tree

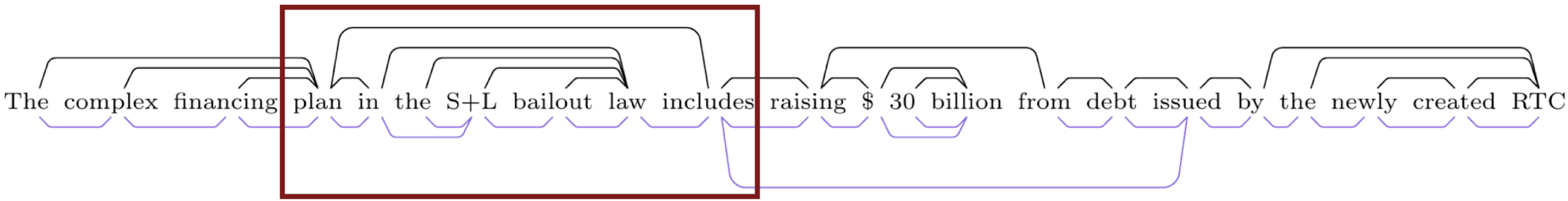
Teal (below sentence): Minimum spanning tree, structural probe on BERT



Trees on baseline representations don't approximate gold trees well!

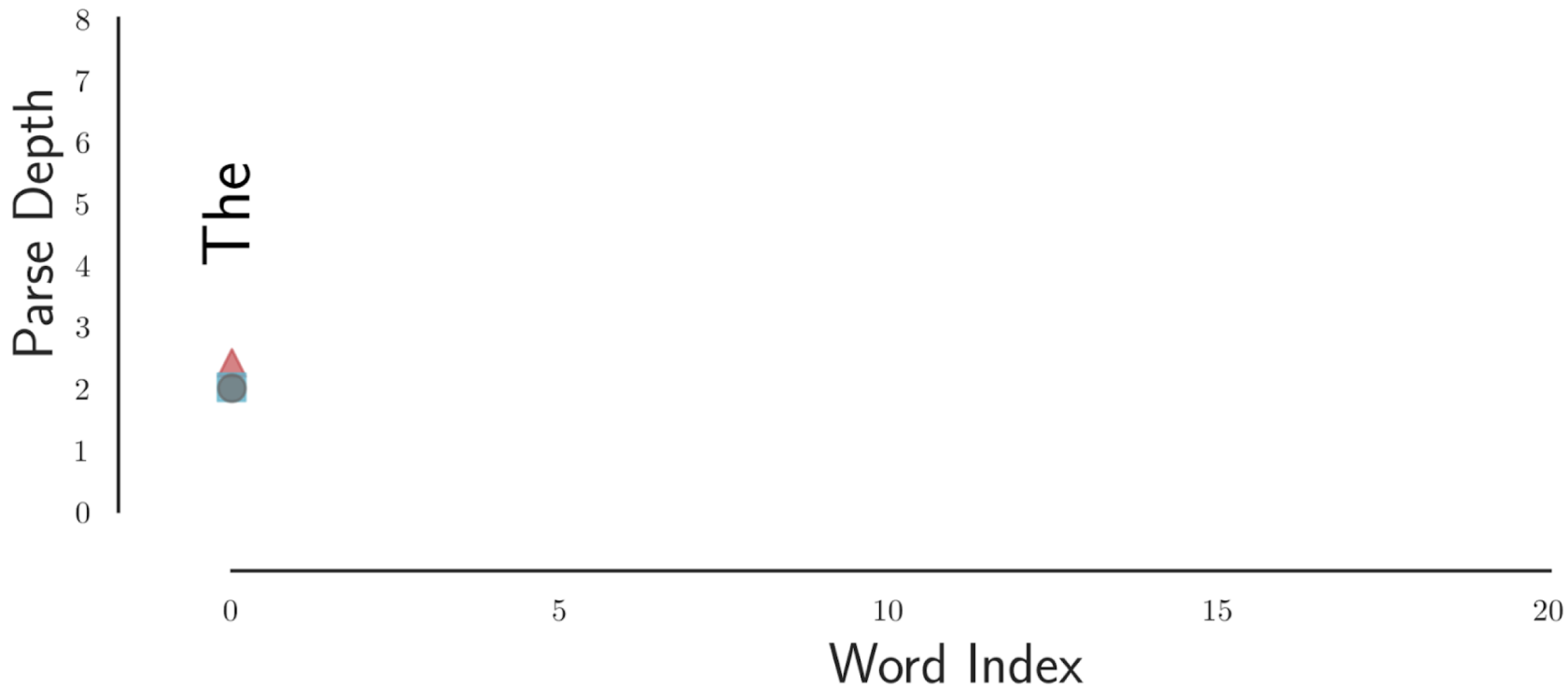
Black (above sentence): Human-annotated parse tree

Purple (below sentence): MST, structural probe on random-weights BiLSTM



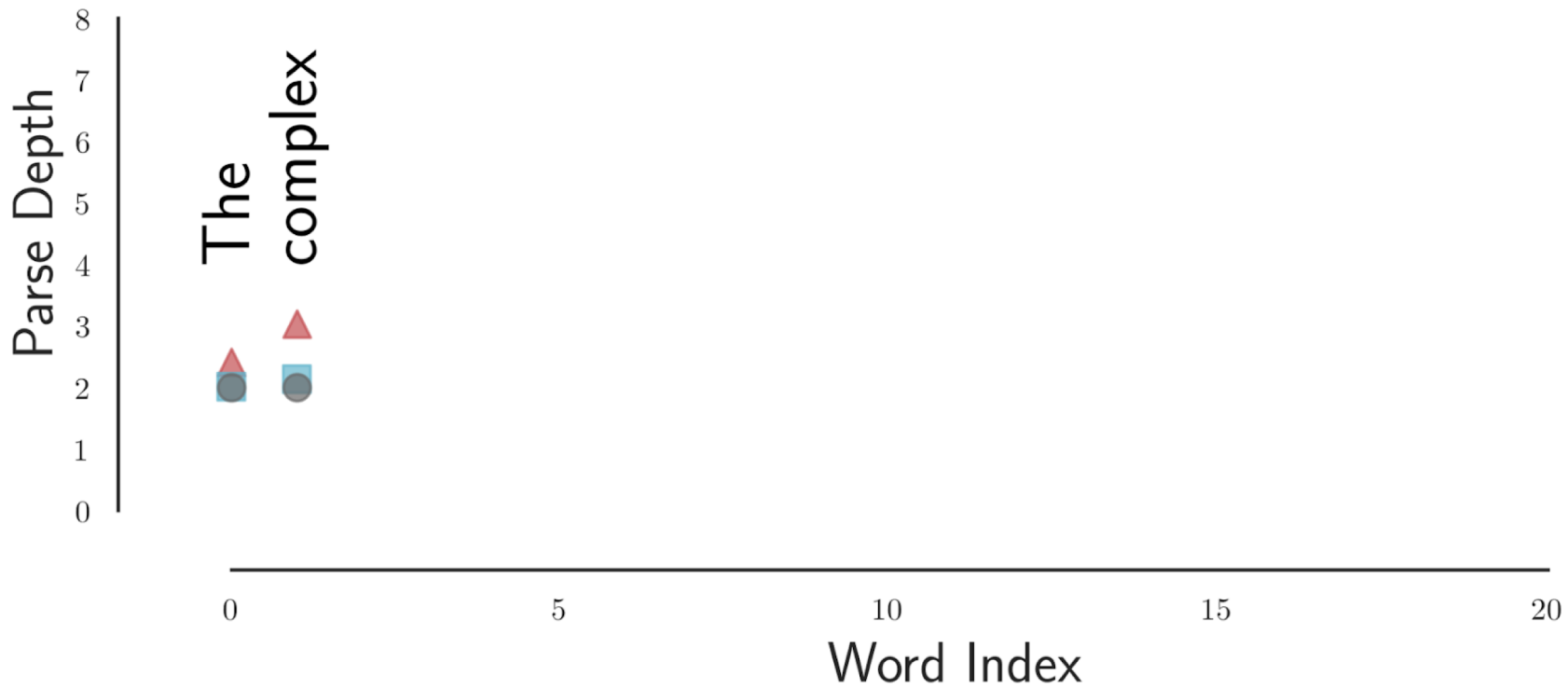
Predicted depths on BERT + ELMo reconstruct parse depths well!

grey circle: gold parse depth
red triangle: ELMo1 squared norm
blue square: BERT large 15 squared norm



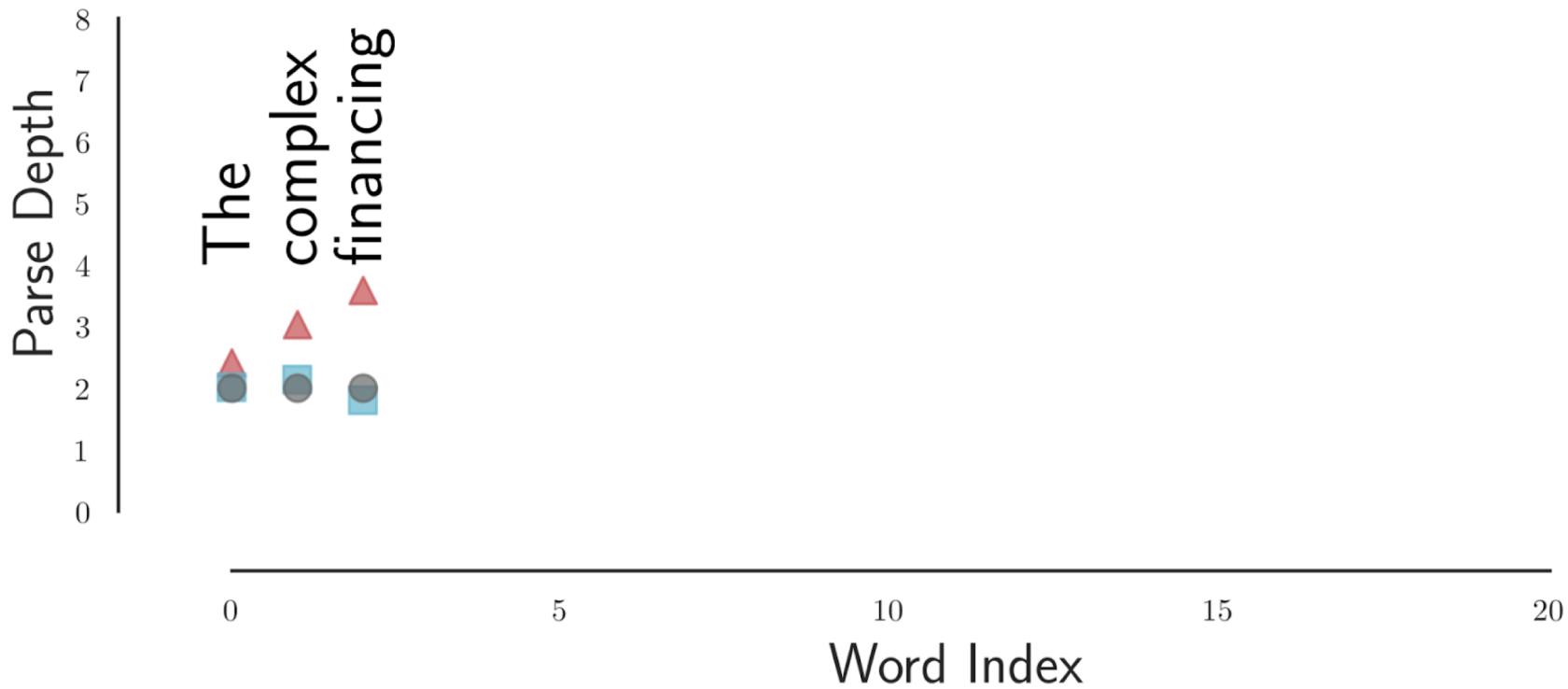
Predicted depths on BERT + ELMo reconstruct parse depths well!

grey circle: gold parse depth
red triangle: ELMo1 squared norm
blue square: BERT large 15 squared norm



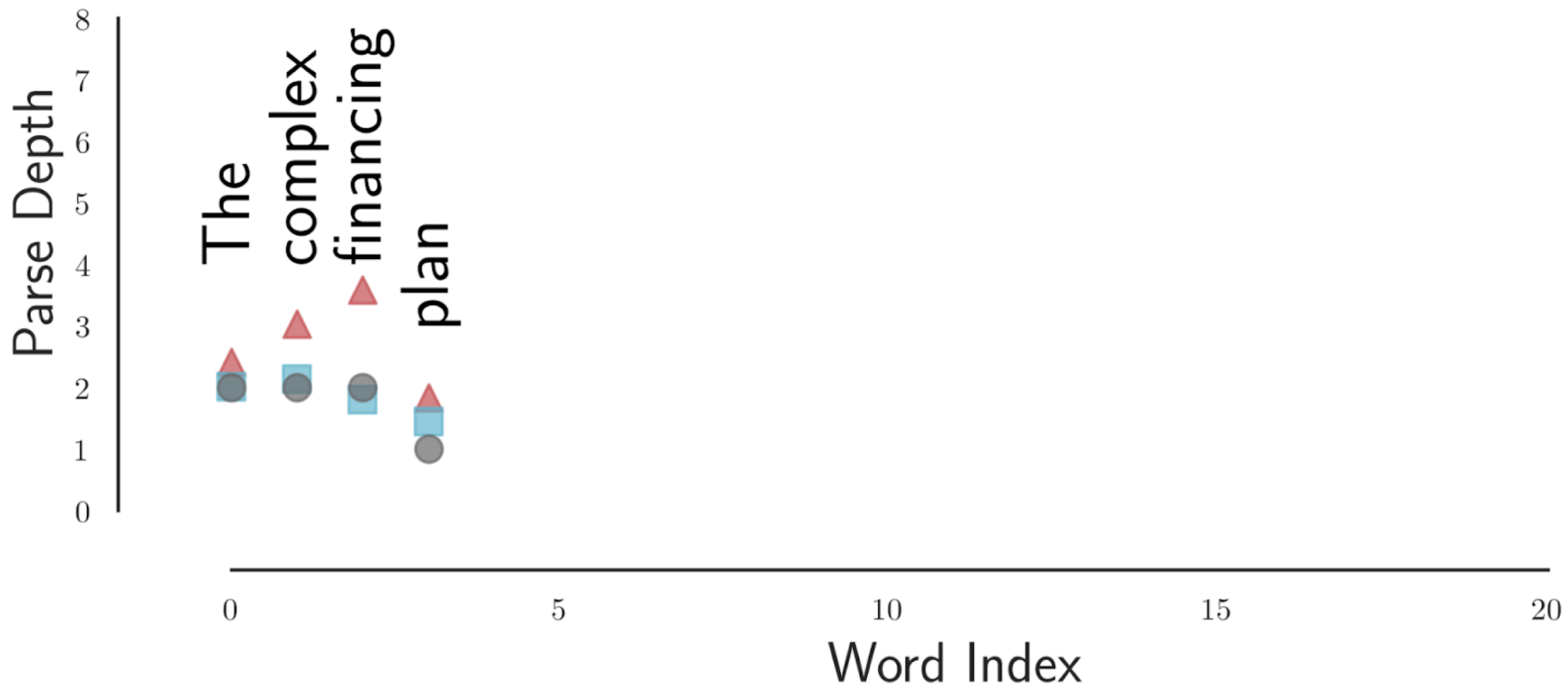
Predicted depths on BERT + ELMo reconstruct parse depths well!

grey circle: gold parse depth
red triangle: ELMo1 squared norm
blue square: BERT large 15 squared norm



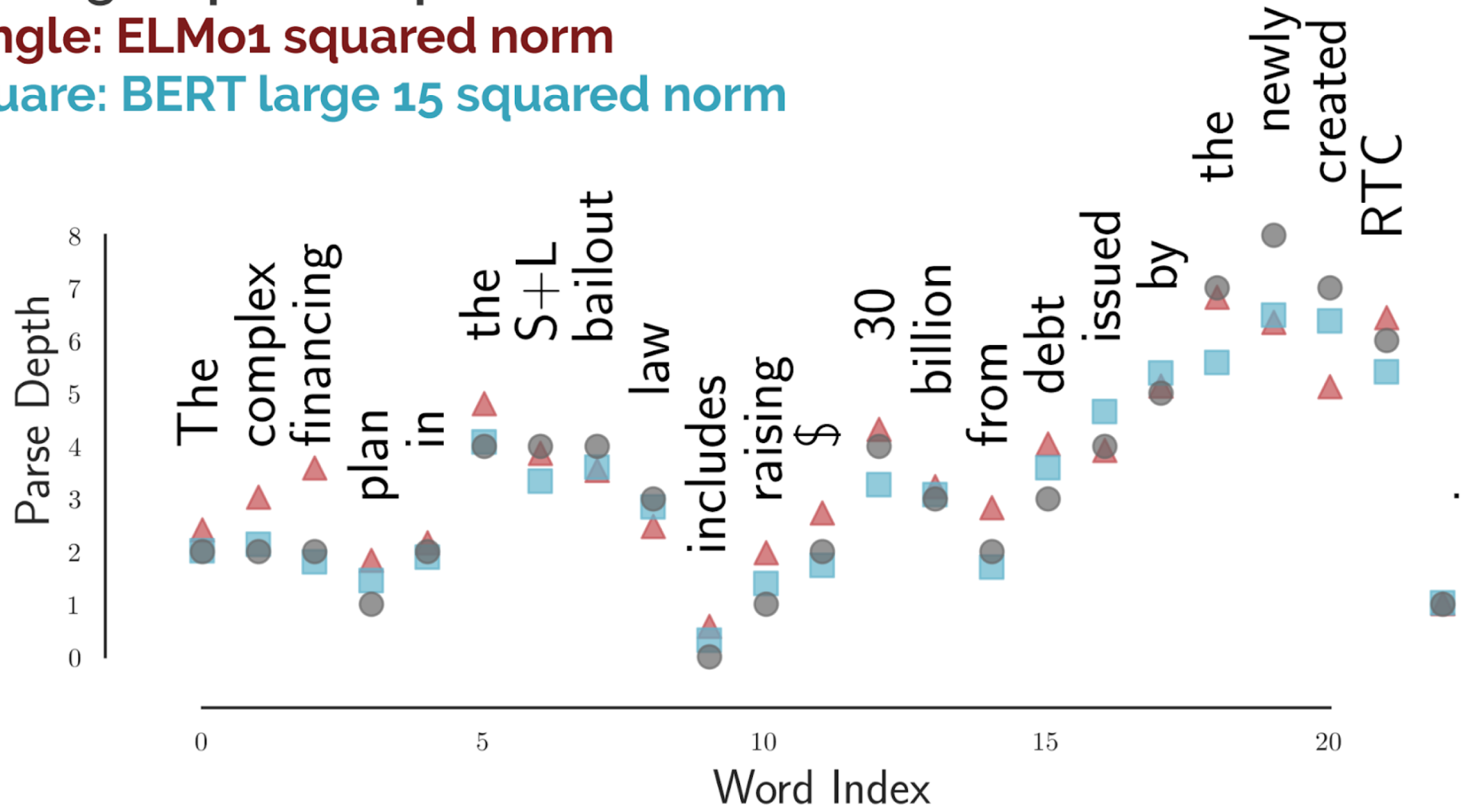
Predicted depths on BERT + ELMo reconstruct parse depths well!

grey circle: gold parse depth
red triangle: ELMo1 squared norm
blue square: BERT large 15 squared norm

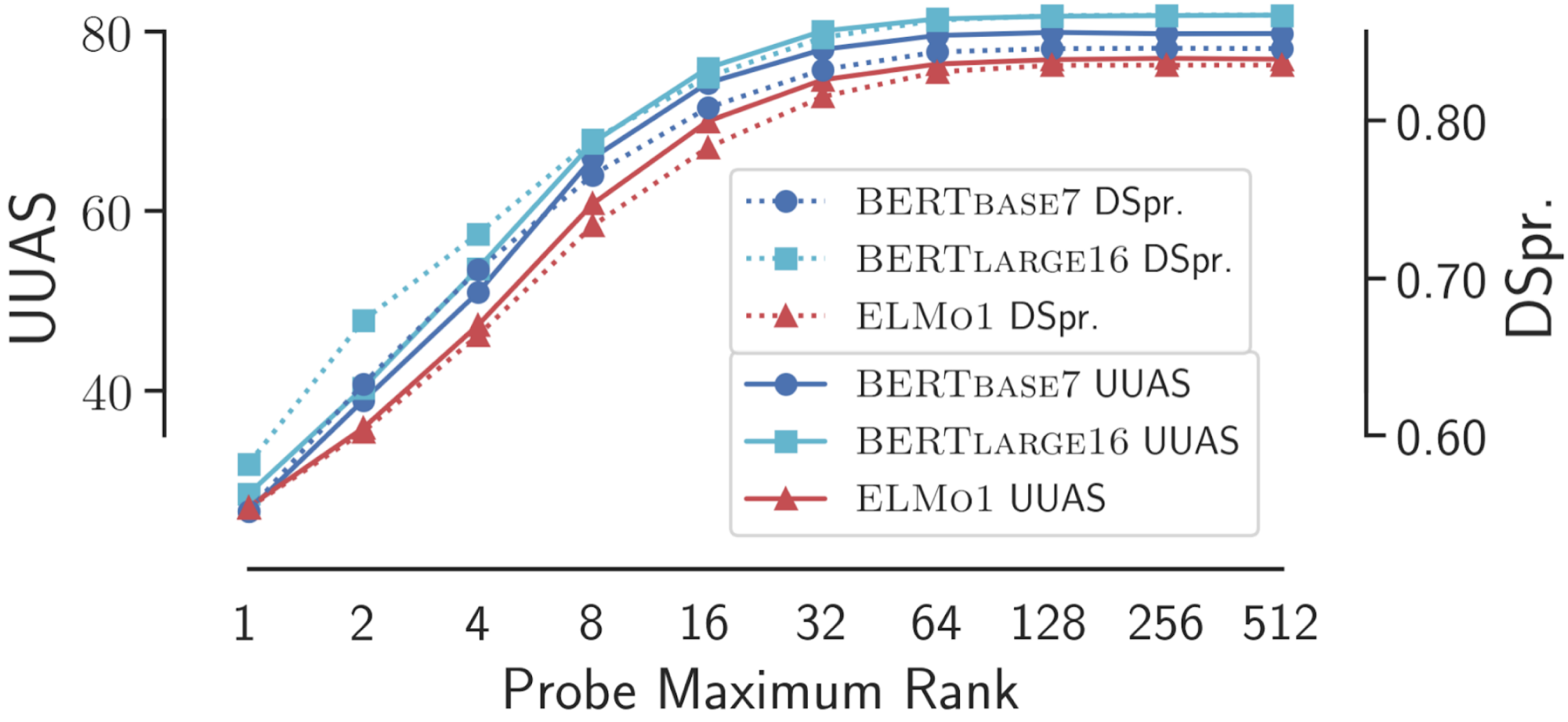


Predicted depths on BERT + ELMo reconstruct parse depths well!

grey circle: gold parse depth
red triangle: ELMo1 squared norm
blue square: BERT large 15 squared norm



Syntax geometry is quite low rank



Visualizing and Measuring the Geometry of BERT

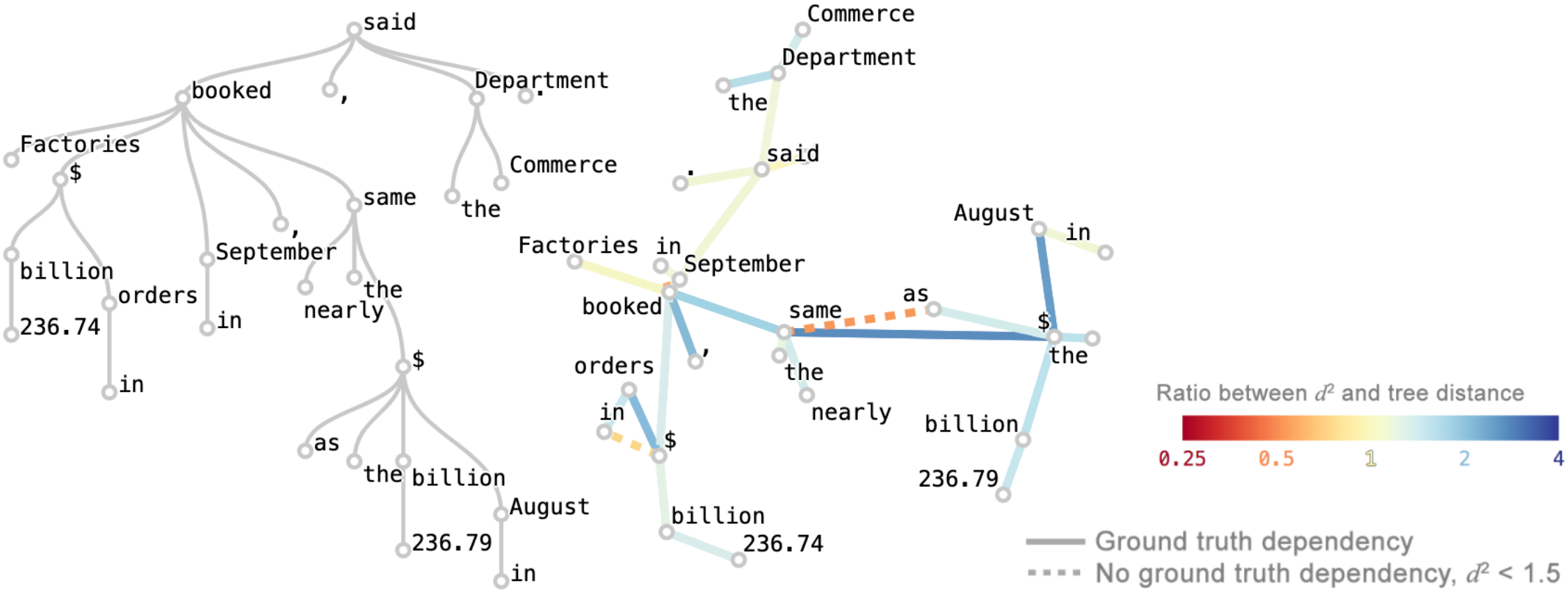
[Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg, 2019]

https://pair-code.github.io/interpretability/bert-tree/

- What does syntax geometry look like?
- Why are trees encoded in **squared** vector distance?
- Geometry + structural probes for understanding BERT syntax
- Representation of word senses in BERT

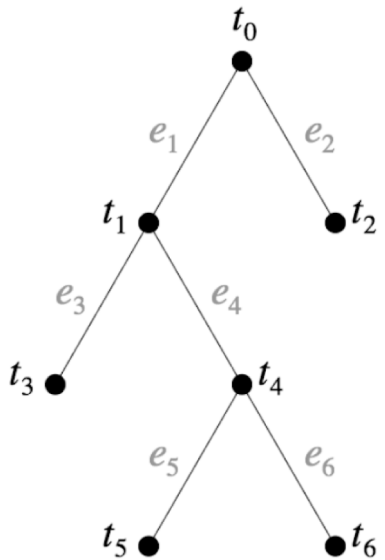
Visualizing and Measuring the Geometry of BERT

"Factories booked $236.74 billion in orders in September, nearly the same as the $236.79 billion in August, the Commerce Department said."



Why are trees encoded in *squared* vector distance?

Nodes in trees have a natural vector embedding.

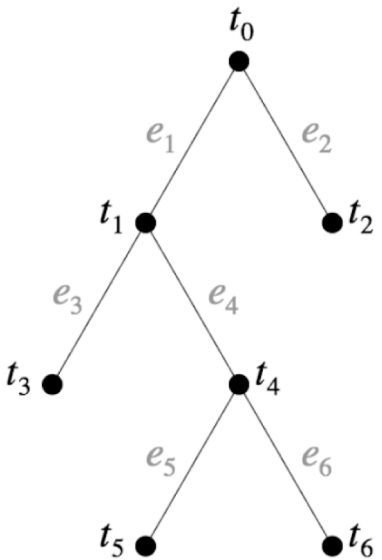


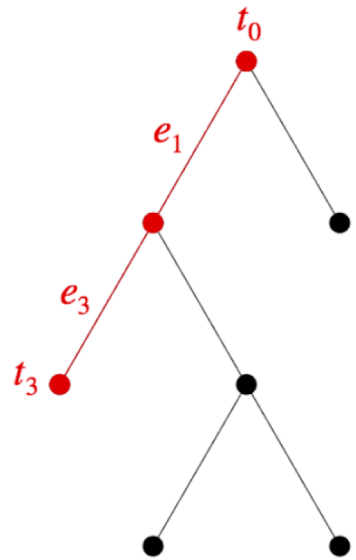
1. Assign edges orthogonal unit embeddings.

[Coenen et al.., 2019]; https://pair-code.github.io/interpretability/bert-tree/

Why are trees encoded in *squared* vector distance?

Nodes in trees have a natural vector embedding.





$f(t_3) =$

$e_1 + e_3 =$

$(1, 0, 1, 0, 0, 0)$

1. Assign edges orthogonal unit embeddings.

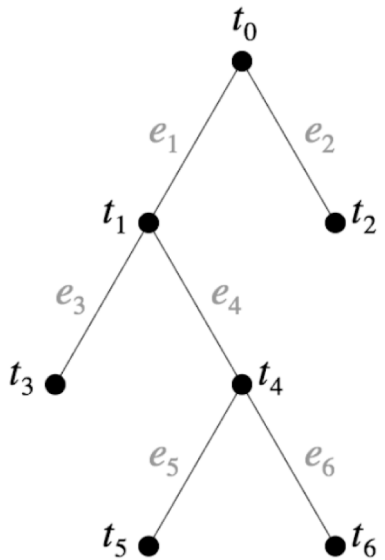
2. Assign each edge a direction (say, root-> leaf)

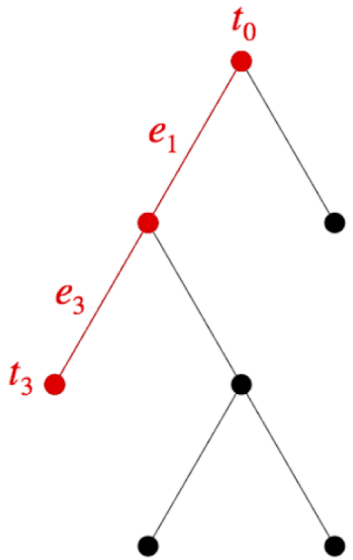
3. Assign each node sum of embeddings of edges pointing "towards" it

[Coenen et al.., 2019]; https://pair-code.github.io/interpretability/bert-tree/

Why are trees encoded in *squared* vector distance?

Squared L2 distance preserves tree distances

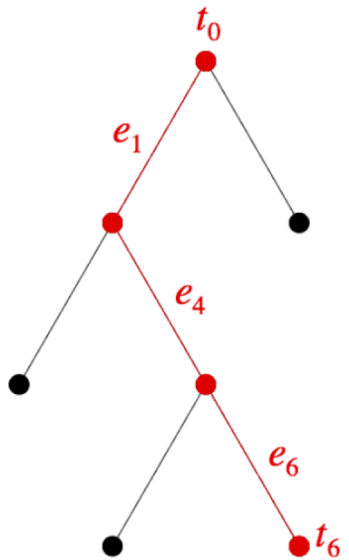




$f(t_3) =$

$e_1 + e_3 =$

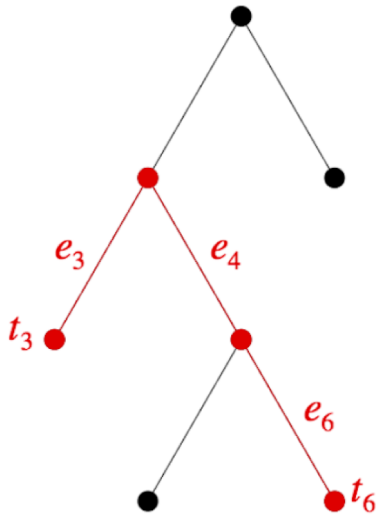
$(1, 0, 1, 0, 0, 0)$



$f(t_6) =$

$e_1 + e_4 + e_6 =$

$(1, 0, 0, 1, 0, 1)$



$f(t_3) - f(t_6) =$

$e_3 - e_4 - e_6 =$

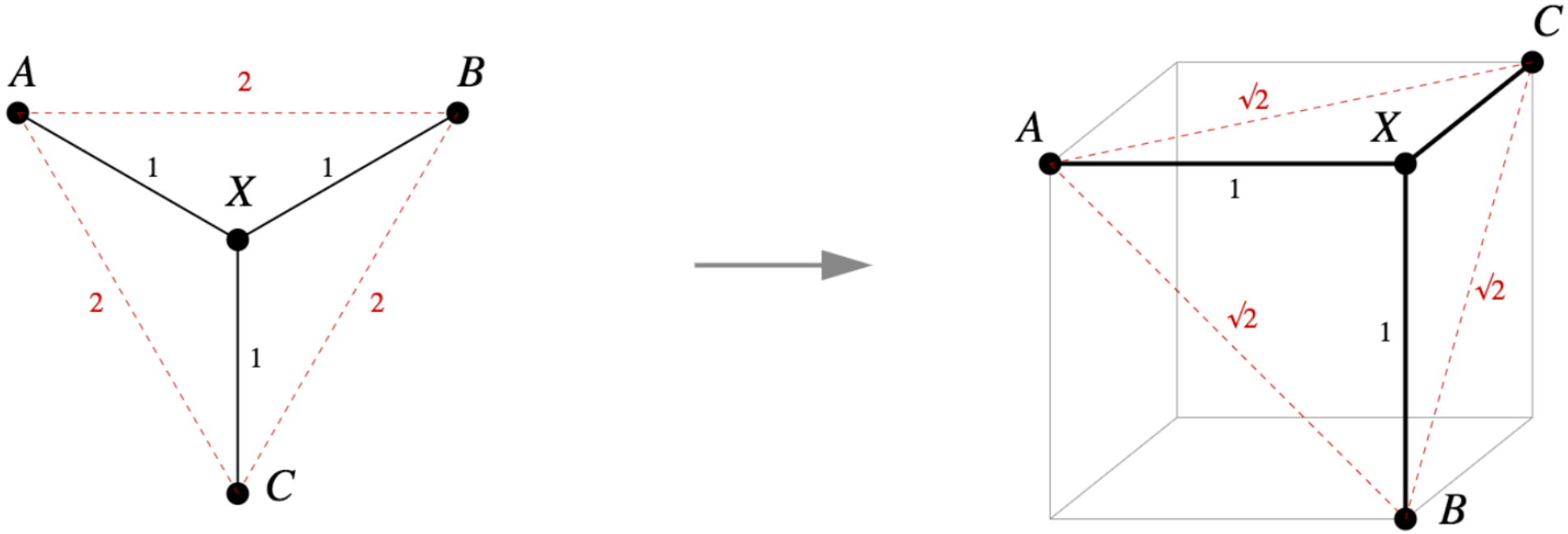
$(0, 0, 1, -1, 0, -1)$

$\| f(t_3) - f(t_6) \|^2 = 3$

[Coenen et al.., 2019]; https://pair-code.github.io/interpretability/bert-tree/

Why are trees encoded in squared vector distance?

You can’t isometrically embed tree distance in Euclidean space



You can encode it in a “Pythagorean embedding”

$f:M\to\mathbb{R}^n$ is a *Pythagorean embedding* if for all $x,y\in M$, $d(x,y)=\|f(x)-f(y)\|^2$

Final thoughts

- "Unsupervised" (self-supervised) learning is **very** successful here
  - More so than conventional multi-task learning
- Has annotating lots of linguistic data all been a mistake?
  - Language model learning exploits the richness of the task
- Deep contextual word representations have phase-shifted from statistical association learners to **language discovery devices**!
  - Syntax emerges (approximately) in the geometry of BERT
- Going big stretches computational resources and energy
  - And maybe also the analogy to child language acquisition?

Thank you!

Relevant papers:

Kevin Clark, Urvashi Khandelwal, Omer Levy, & Christopher Manning. 2019. What Does BERT Look At? An Analysis of BERT's Attention. BlackBoxNLP.

John Hewitt and Christopher Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. NAACL.

Emergent linguistic structure in deep contextual neural word representations

Christopher Manning
Stanford University
@chrmanning ❀ @stanfordnlp
Institute for Advanced Study, Princeton, NJ, 2019