Deep Contextual Neural Word Representations: Linguistic Structure Discovery and Efficient Discriminative Training

Stanford

Christopher Manning

Stanford University and CIFAR Fellow

@chrmanning ❀ @stanfordnlp

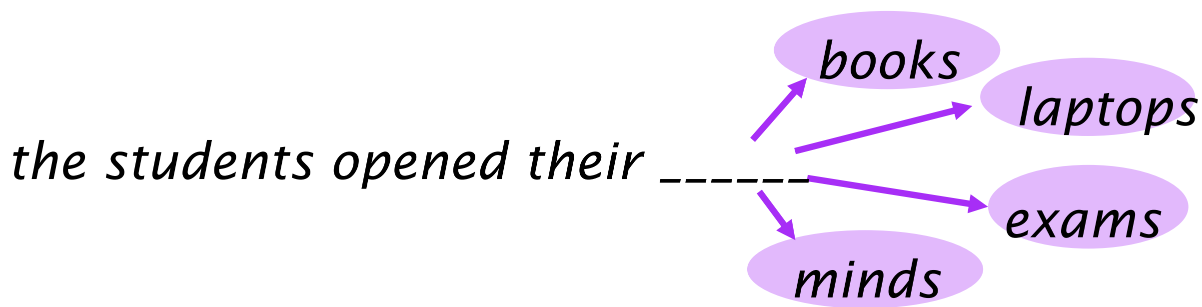
ElementAI/MILA, December 2019 (last talk of 2019!)

Plan

1. From recurrent sequence models to BERT transformers
2. BERT as a linguistic structure discovery machine
3. More efficient Discriminative Pre-training of Text Encoders

1. Language Modeling

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



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

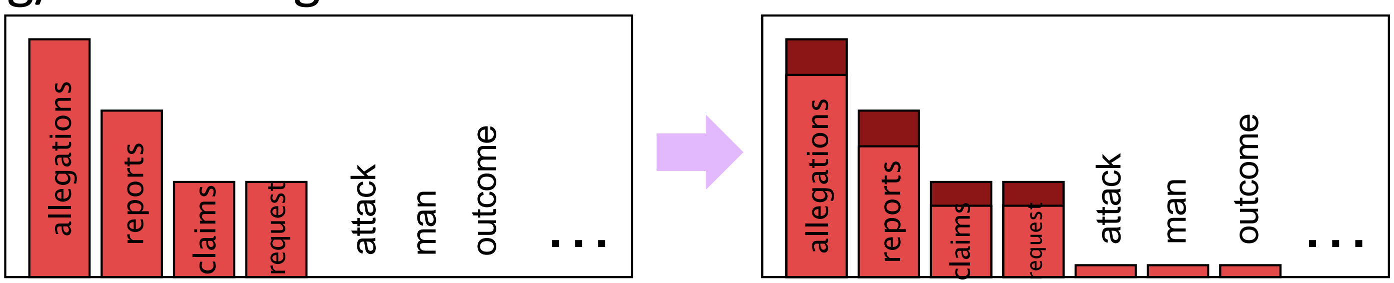
Count how often words follow word sequences; divide to get cond. prob.

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

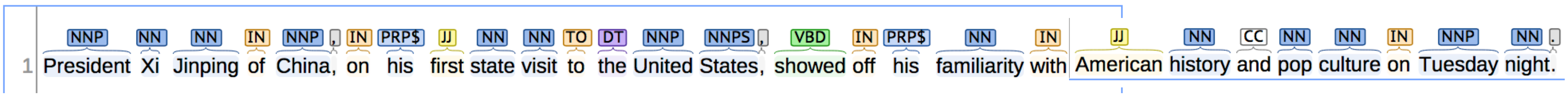
$$P_{bo}\left(x^{(3)}\middle|x^{(2)},x^{(1)}\right) \approx \lambda P\left(x^{(3)}\middle|x^{(2)},x^{(1)}\right) + (1-\lambda)P\left(x^{(3)}\middle|x^{(2)}\right)$$

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

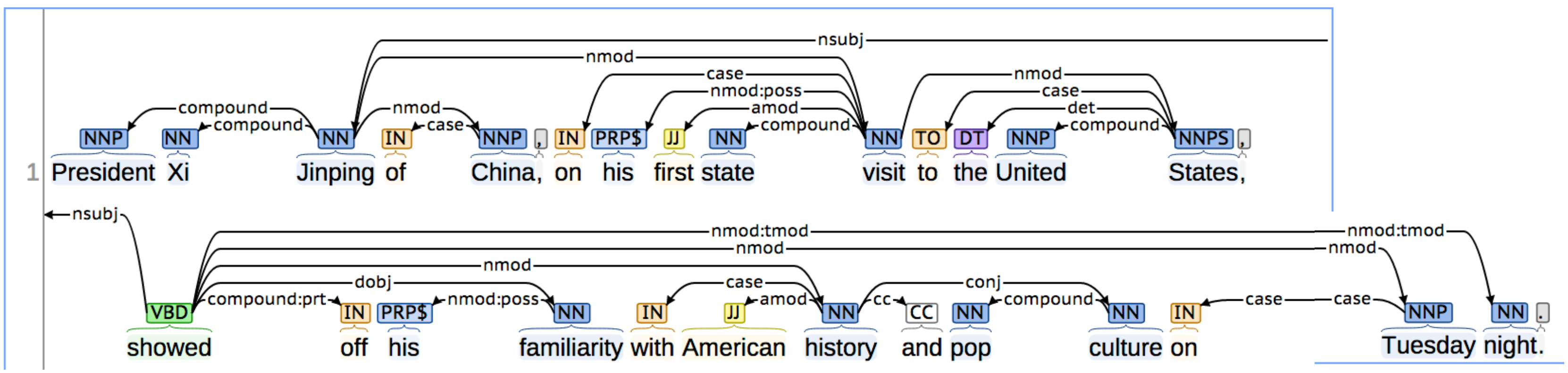
- (**Passionately argued!**) answer of linguists: **almost none**
  - Though they know quite a bit of simple world knowledge
    - The ship {sailed, sank, anchored, …}
  - And, in an unaggregated way, they know some low-level syntax
    - They know you tend to get sequences like:
      - preposition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

Capturing conventional linguistics in NLP

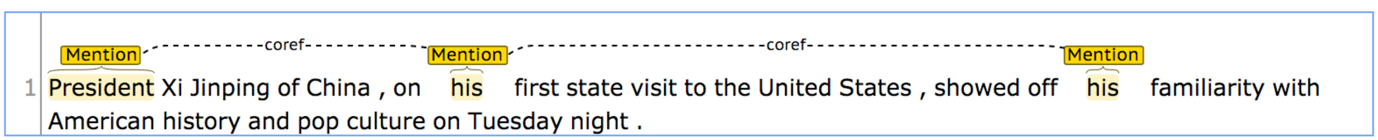
Part-of-Speech:



Basic Dependencies:



Coreference:



SCIENCE'S COMPASS • REVIEW

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.:

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

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***

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

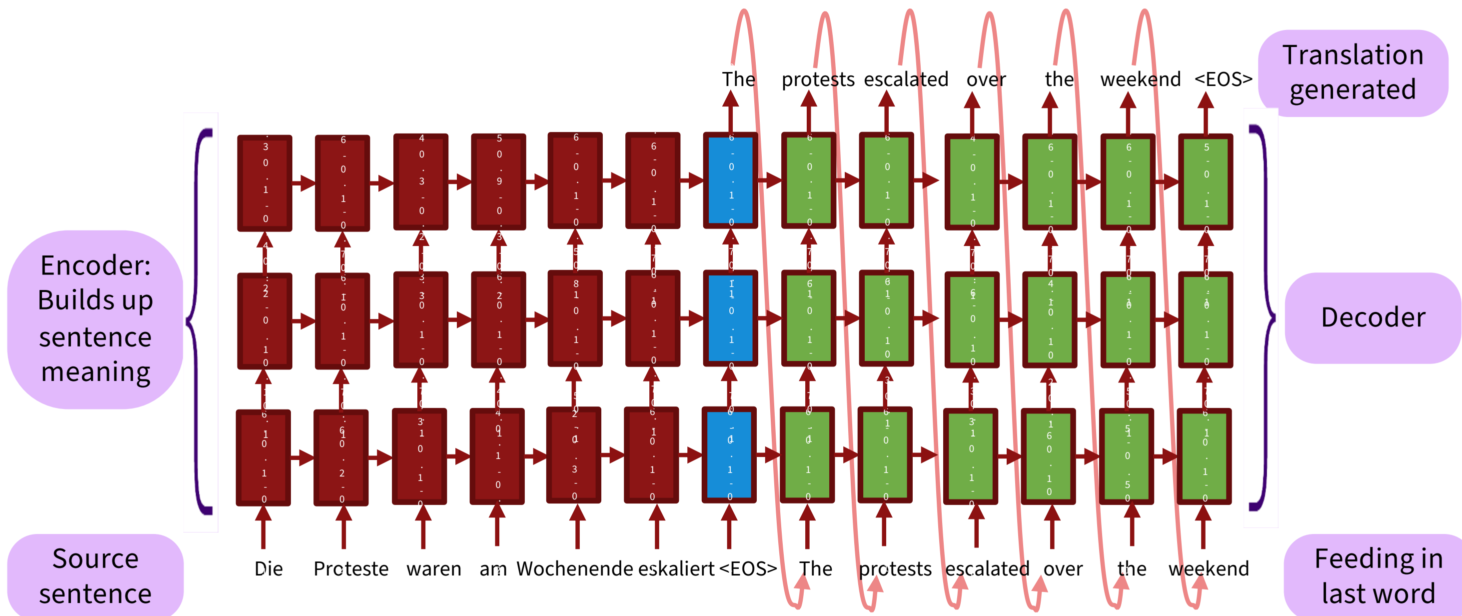
Flashback to 2017

The BiLSTM Hegemony

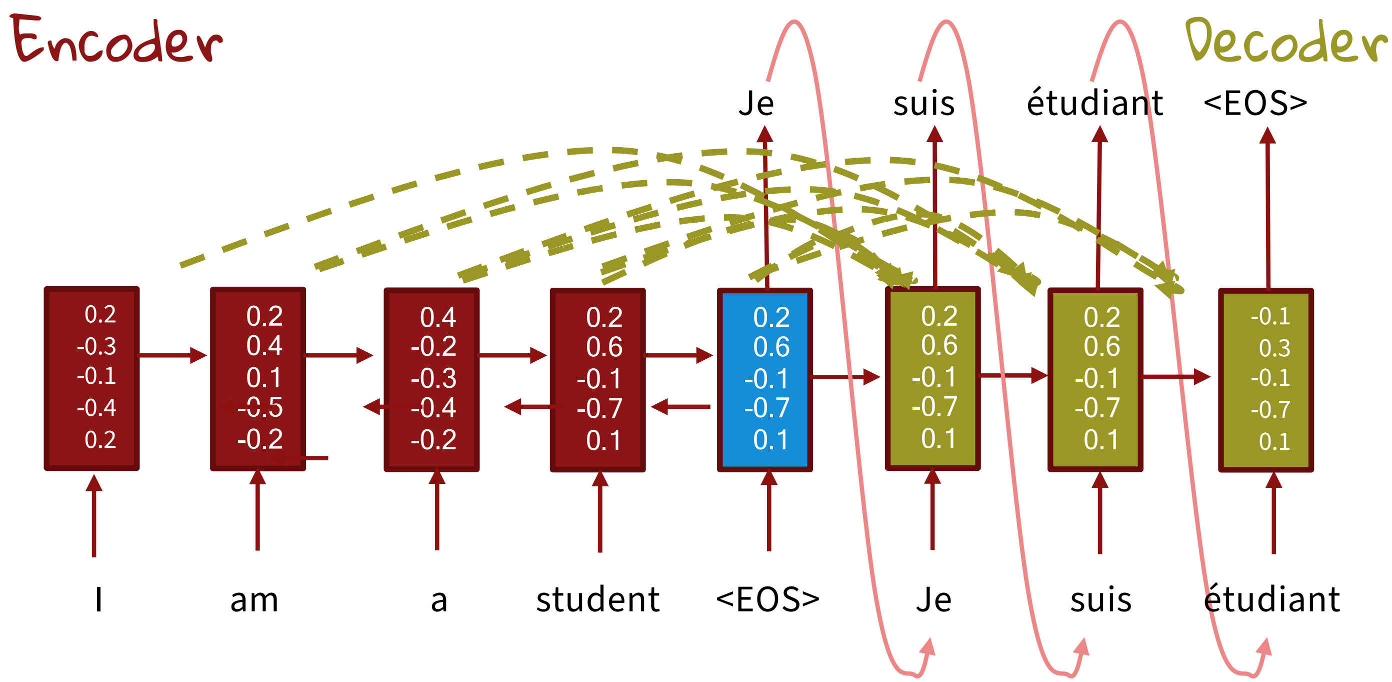
To a first approximation,
the de facto consensus in NLP in 2017 is
that no matter what the task,
you throw a BiLSTM at it, with
attention if you need information flow

An LSTM encoder-decoder network

[Sutskever et al. 2014]

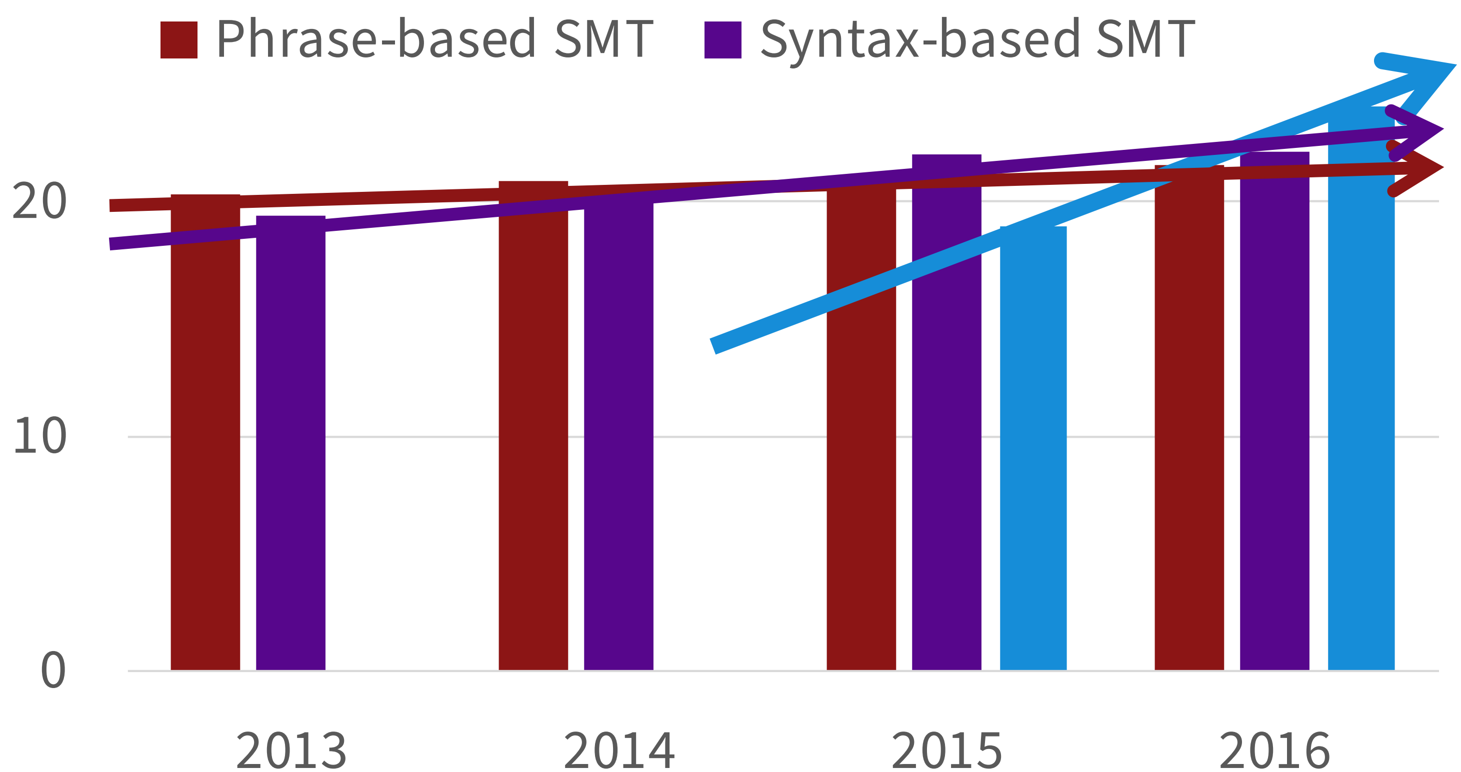


A BiLSTM encoder and LSTM-with-attention decoder



Progress in Machine Translation

[Edinburgh En-De WMT newstest2013 Cased BLEU; NMT 2015 from U. Montréal]



From [Sennrich 2016, http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf]

2018 NLP breakthrough with big language models

All these models are Transformer models

ELMo,
ULMfit
Jan 2018
Training:
103M words
1 GPU day

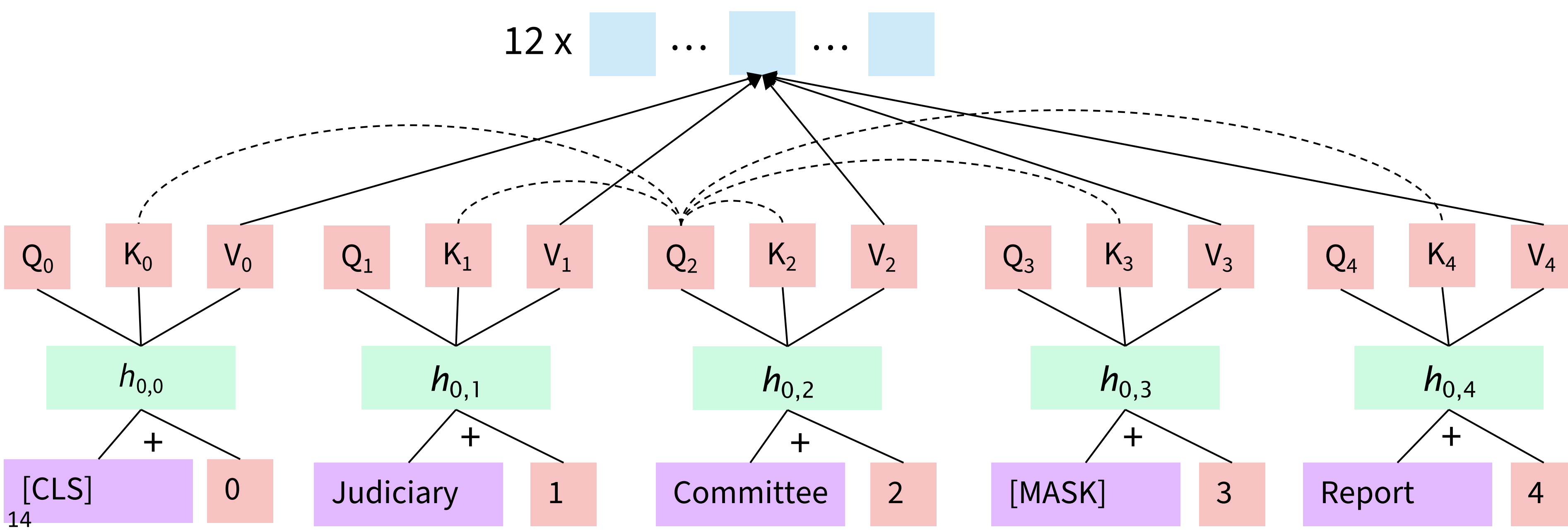
GPT
June 2018
Training
800M words
240 GPU days

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

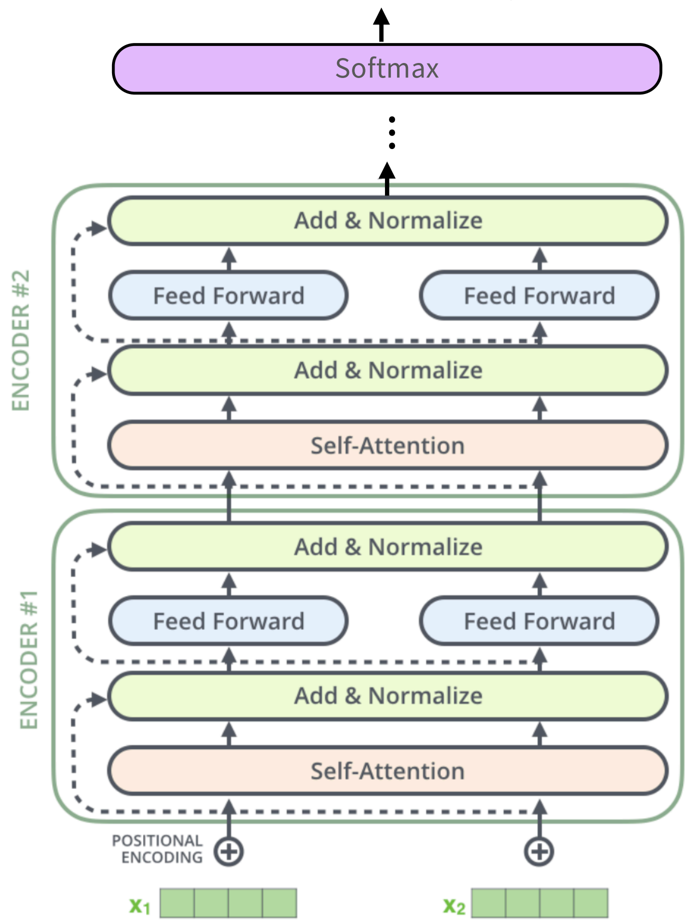
GPT-2
Feb 2019
Training
40B words
~2048 TPU v3 days
according to [a reddit thread]

XL-Net, ERNIE,
Grover, AlBERT,
Megatron-LM, T5,
RoBERTa, GPT-3
July 2019–

Transformer (Vaswani et al. 2017)
BERT (Devlin et al. 2018)



Transformer (Vaswani et al. 2017)
BERT (Devlin et al. 2018)



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

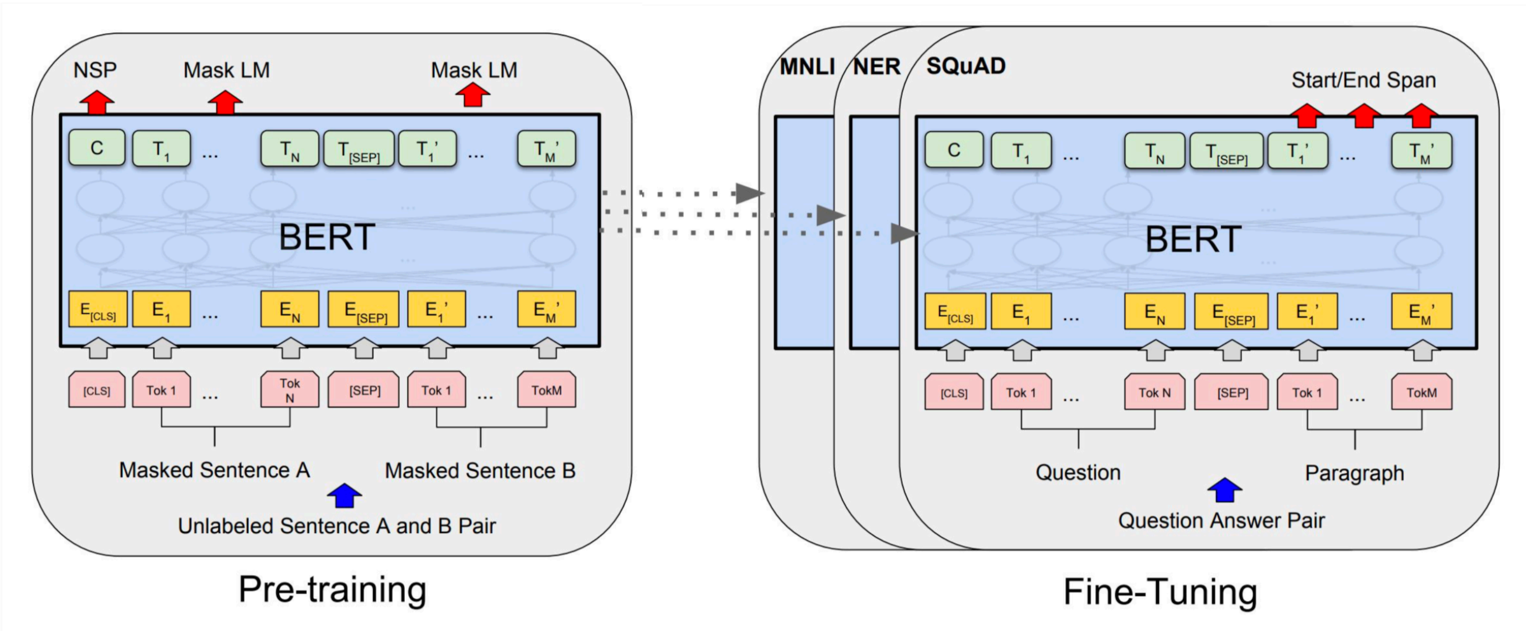
↑ ↑

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

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

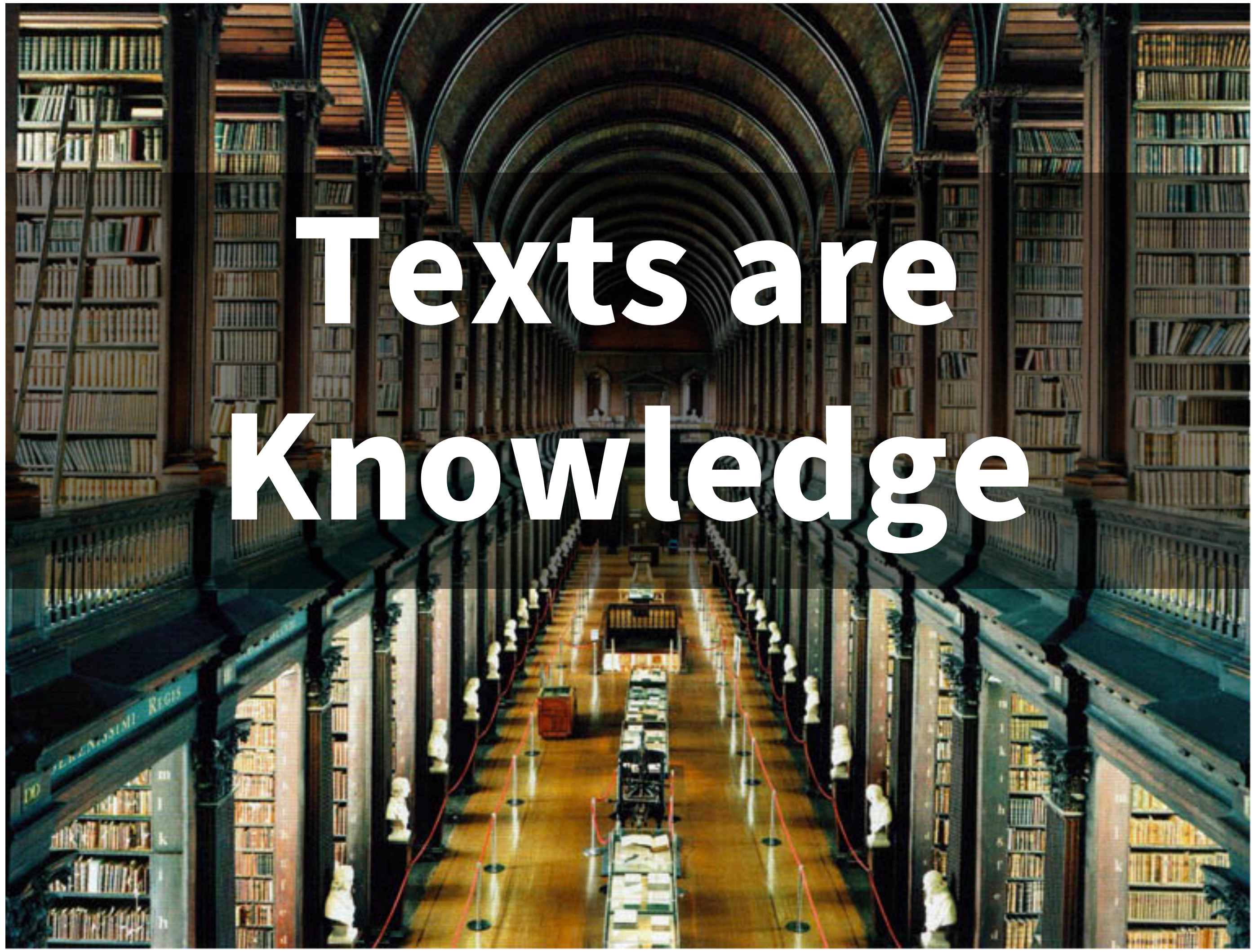
Passage

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

My talk
at the
Automated
Knowledge
Base
Construction
(AKBC)
workshop
2013



From ‘F’ to ‘A’ on the N.Y. Regents Science Exams: An Overview of the Aristo Project. Peter Clark, Oren Etzioni, Daniel Khashabi, Tushar Khot, Bhavana Dalvi Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, Sumithra Bhakthavatsalam, Dirk Groeneveld, Michal Guerquin, Michael Schmitz

AllenAI ARISTO: Answering Science Exam Questions

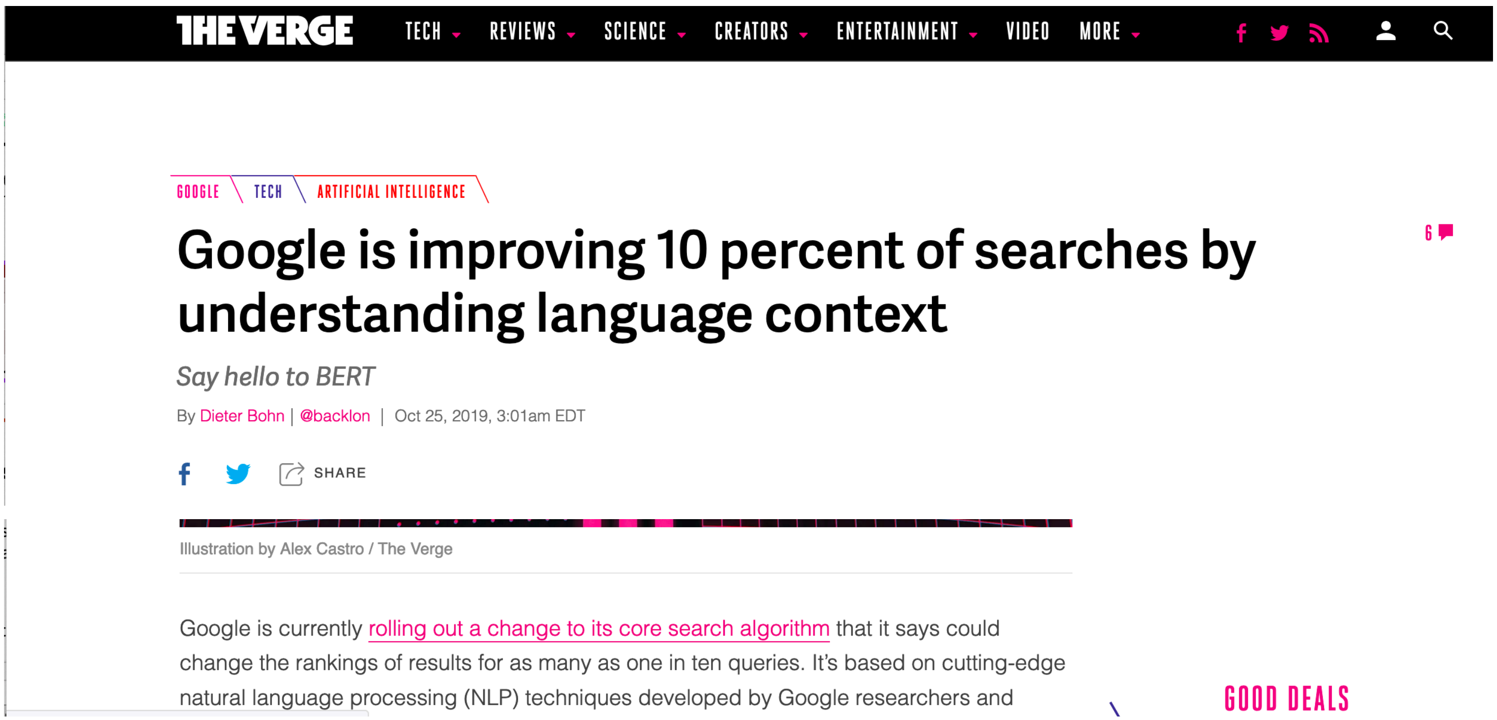
Which equipment will best separate a mixture of iron filings and black pepper? **(1)** magnet **(2)** filter paper **(3)** triplebeam balance **(4)** voltmeter

Which process in an apple tree primarily results from cell division? **(1)** growth **(2)** photosynthesis **(3)** gas exchange **(4)** waste removal

Test Set	IR	TupInf	Multee	AristoBERT	AristoRoBERTa	ARISTO
Regents 4th	64.5	63.5	69.7	86.2	88.1	**89.9**
Regents 8th	66.6	61.4	68.9	86.6	88.2	**91.6**
Regents 12th	41.2	35.4	56.0	75.5	82.3	**83.5**
ARC-Challenge	0.0	23.7	37.4	57.6	**64.6**	64.3

Google web search

BERT brings big gains to web search



THE VERGE TECH REVIEWS SCIENCE CREATORS ENTERTAINMENT VIDEO MORE

GOOGLE \ TECH \ ARTIFICIAL INTELLIGENCE

Google is improving 10 percent of searches by understanding language context

6

Say hello to BERT

By Dieter Bohn | @backlon | Oct 25, 2019, 3:01am EDT

SHARE

Illustration by Alex Castro / The Verge

Google is currently rolling out a change to its core search algorithm that it says could change the rankings of results for as many as one in ten queries. It's based on cutting-edge natural language processing (NLP) techniques developed by Google researchers and

GOOD DEALS

2. What does BERT know? Observational evidence

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

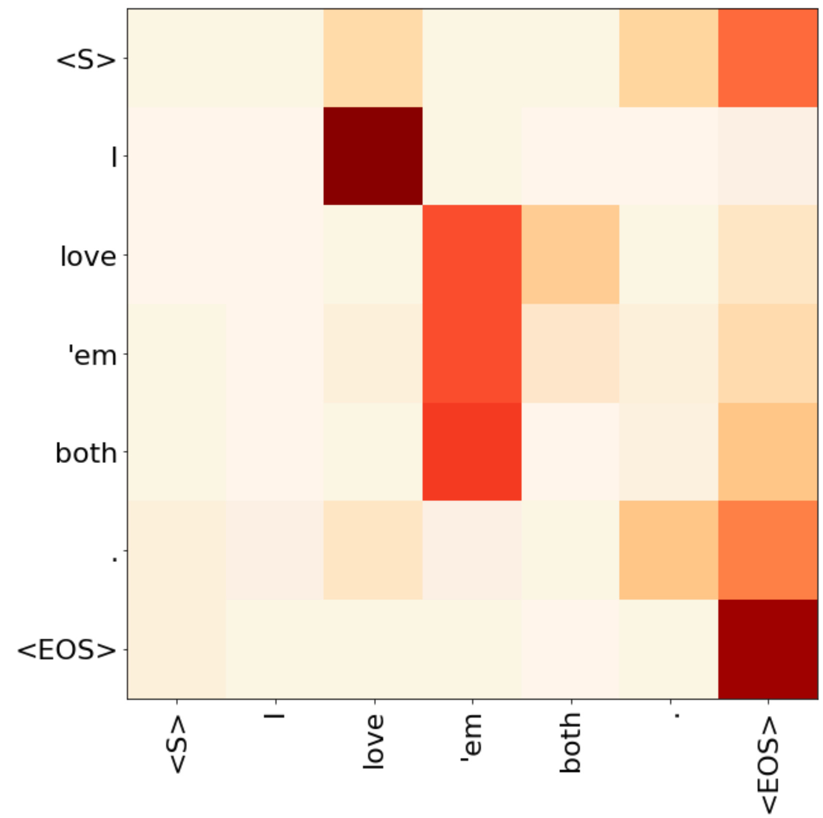
- BERT works really well and calculates clearly useful context-dependent word representations
- Directly observe what BERT is looking at
- We find that BERT induces a lot of structure similar to conventional linguistic structure … because it helps predict

BERT Attention Heads

- For each of many attention heads, for each word position, see where BERT pays attention

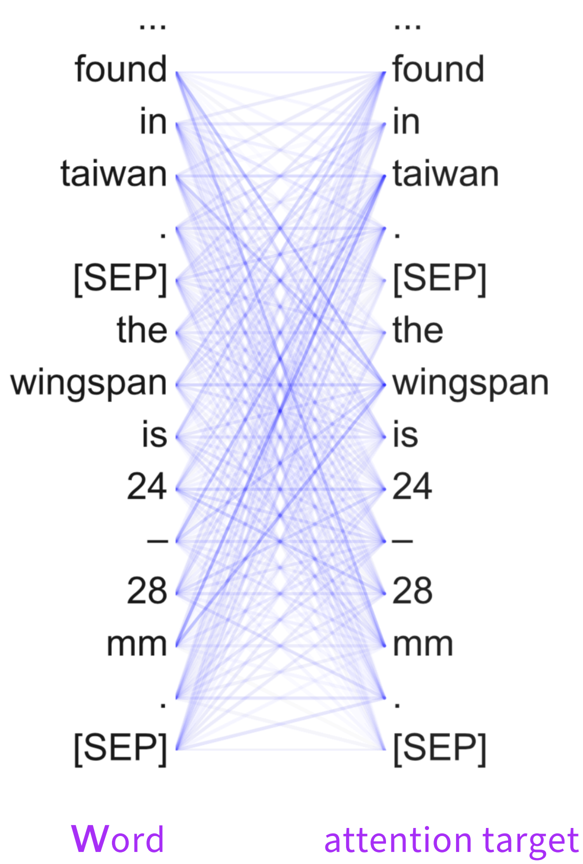
- Look at the most-attended-to word for each head

- How does what BERT attends to correspond to linguistics?

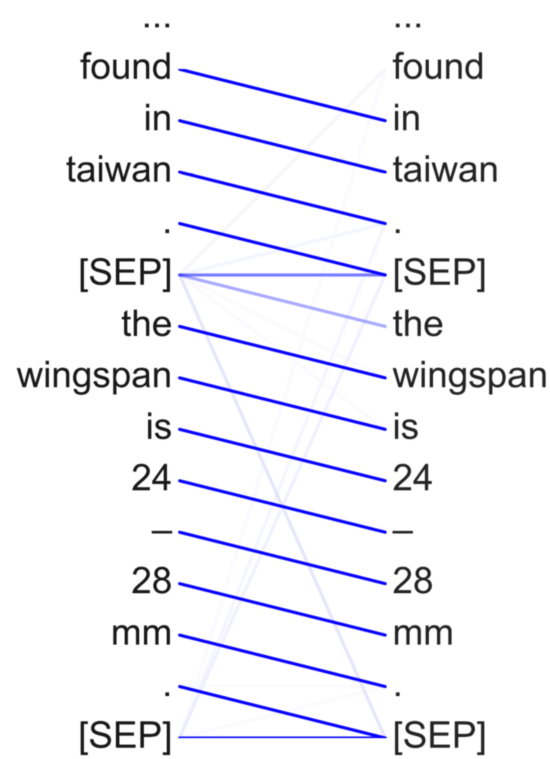


What do BERT attention heads do?

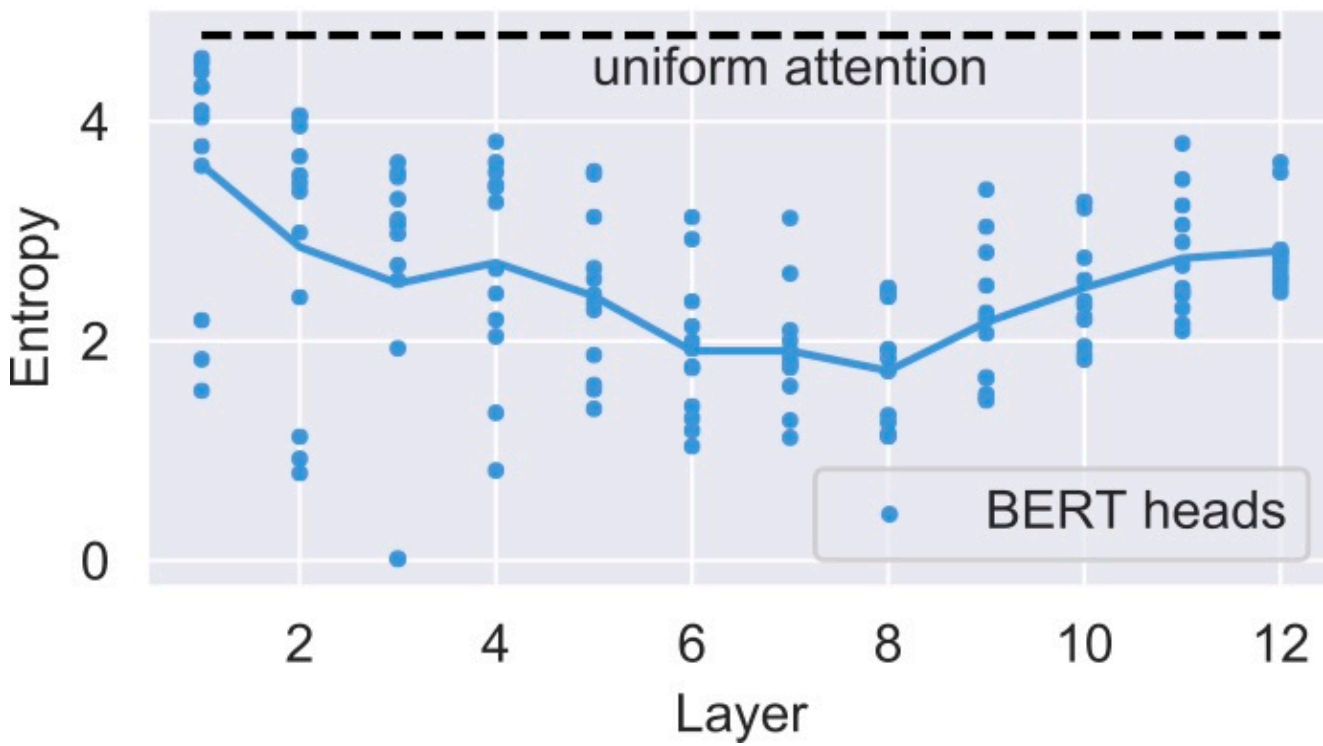
1-1: Attend broadly ("BoW head")



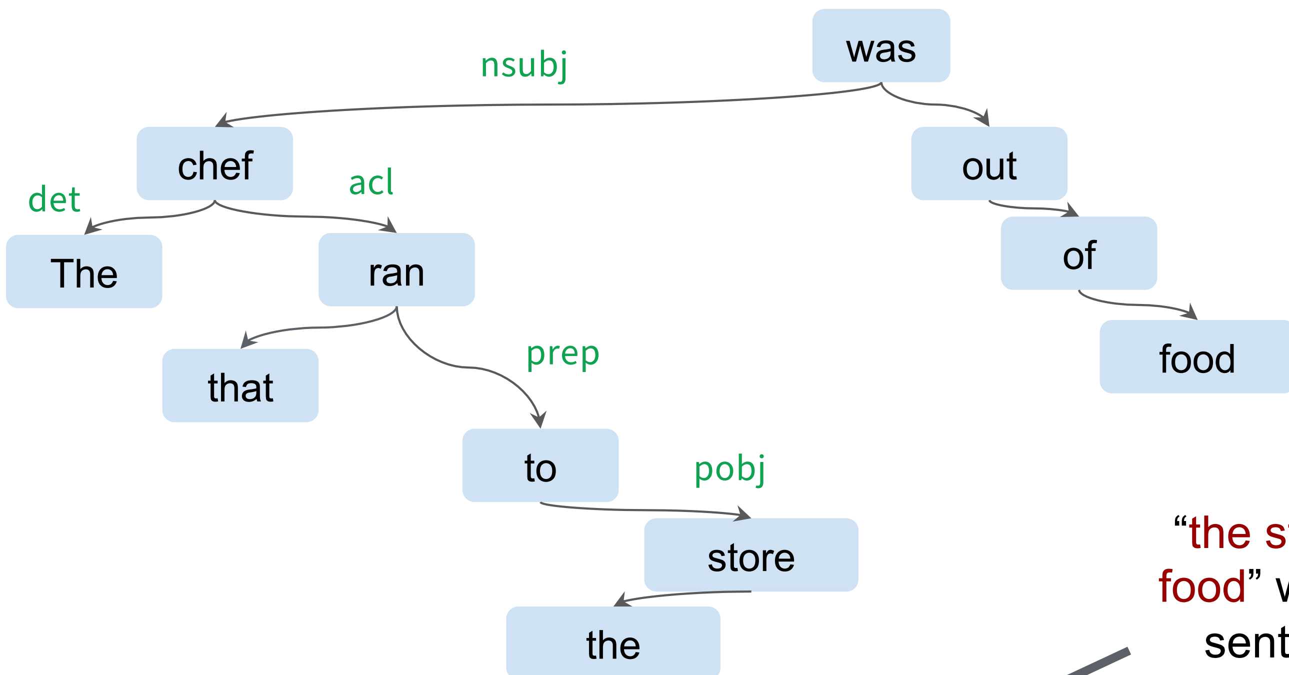
3-1: Attend to next (or prev) word



First layer heads mainly average



A sentence's meaning is composed via its syntax tree

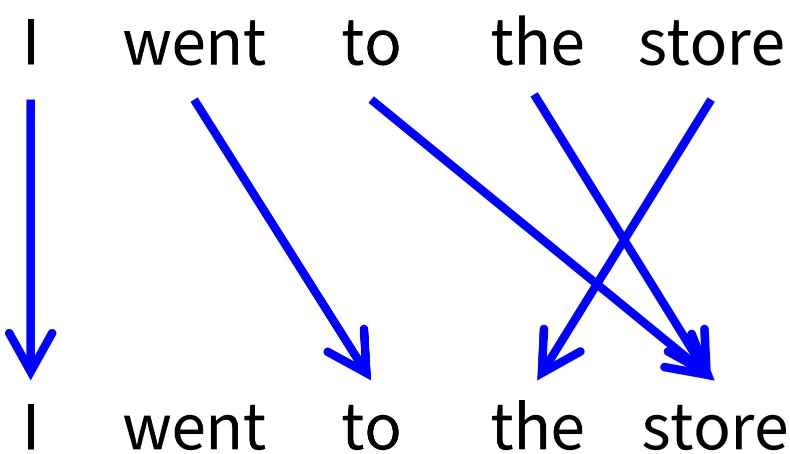


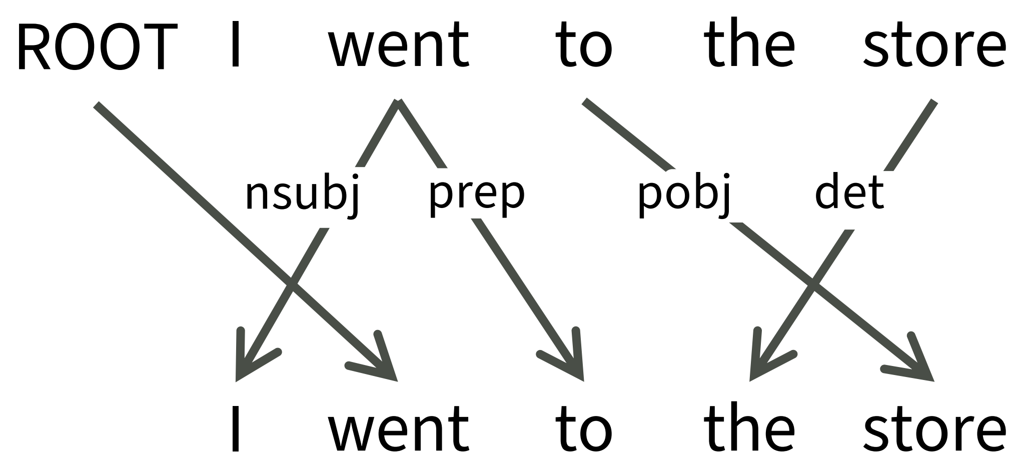
"the store was out of food" would be a valid sentence by itself

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?

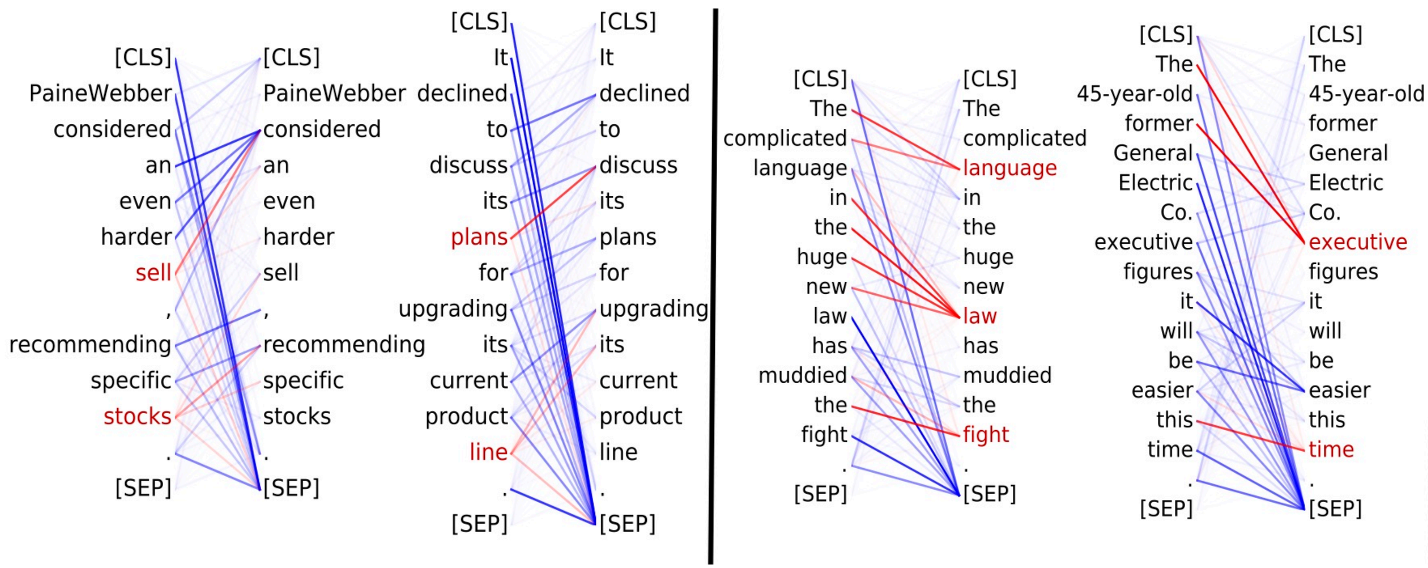




Take the most-attended-to words

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

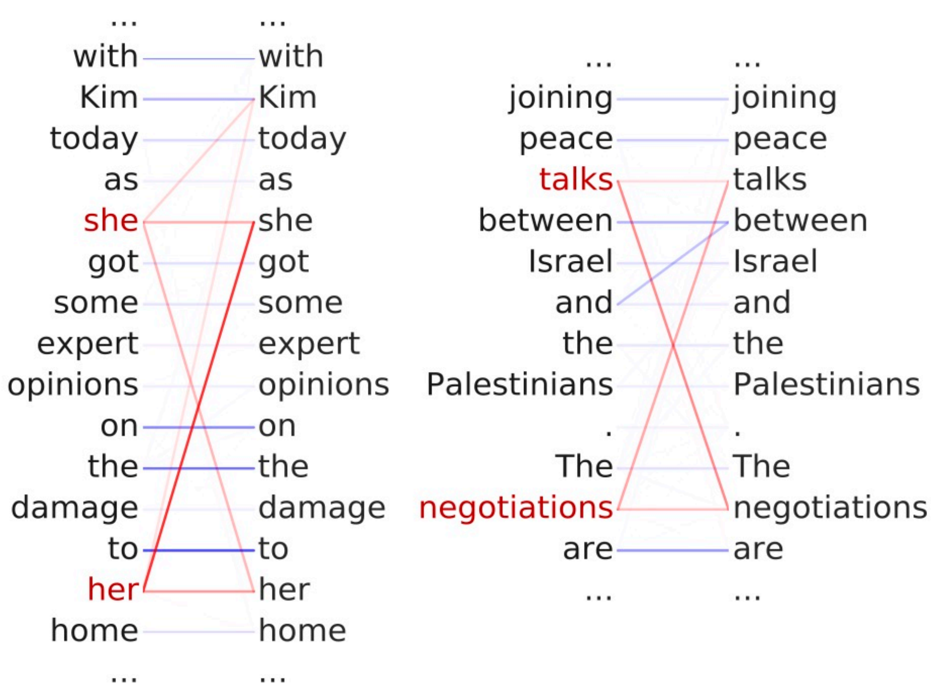
Head 8-11
Noun modifiers (det, adj) attend to head noun. 94.3% on det relation

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}

Experimental evidence

Hewitt and Manning (NAACL 2019)

tl;dr

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

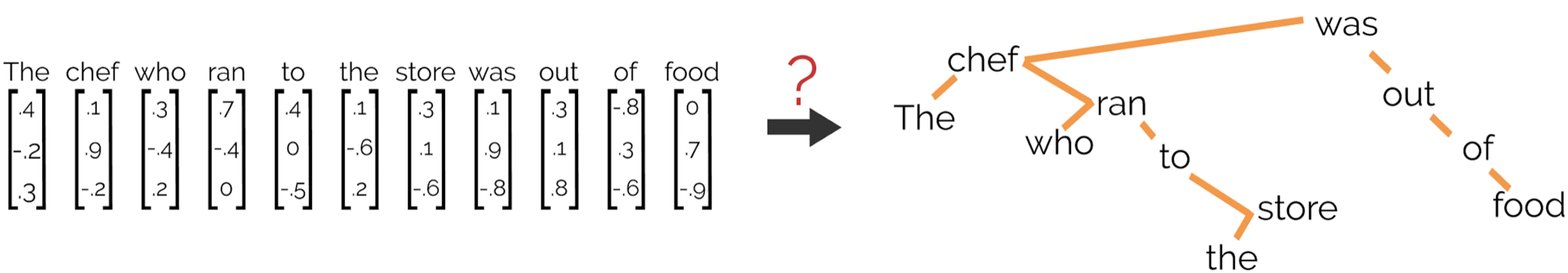
Yes, approximately

How can we tell whether its vector representations encode trees?

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

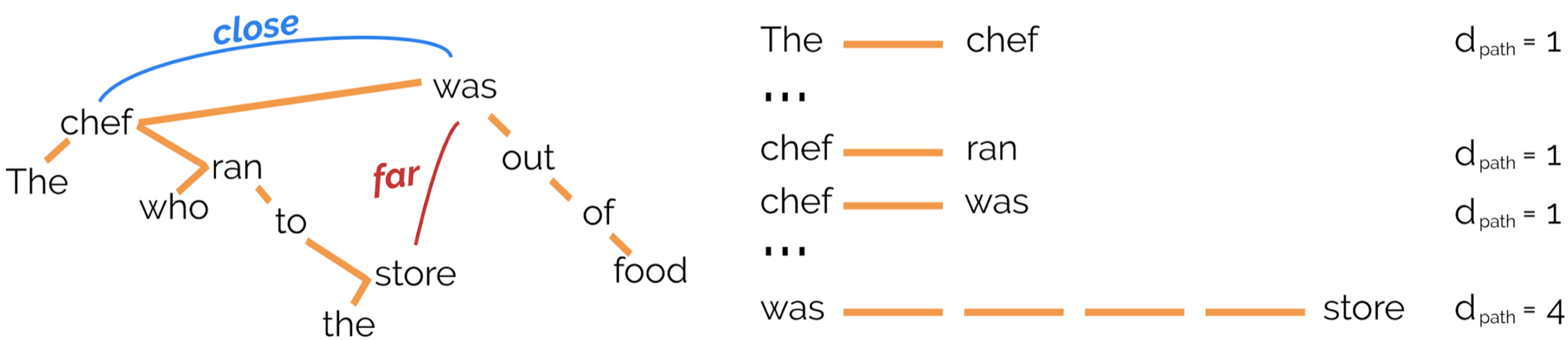
Are vector spaces and trees reconcilable?

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



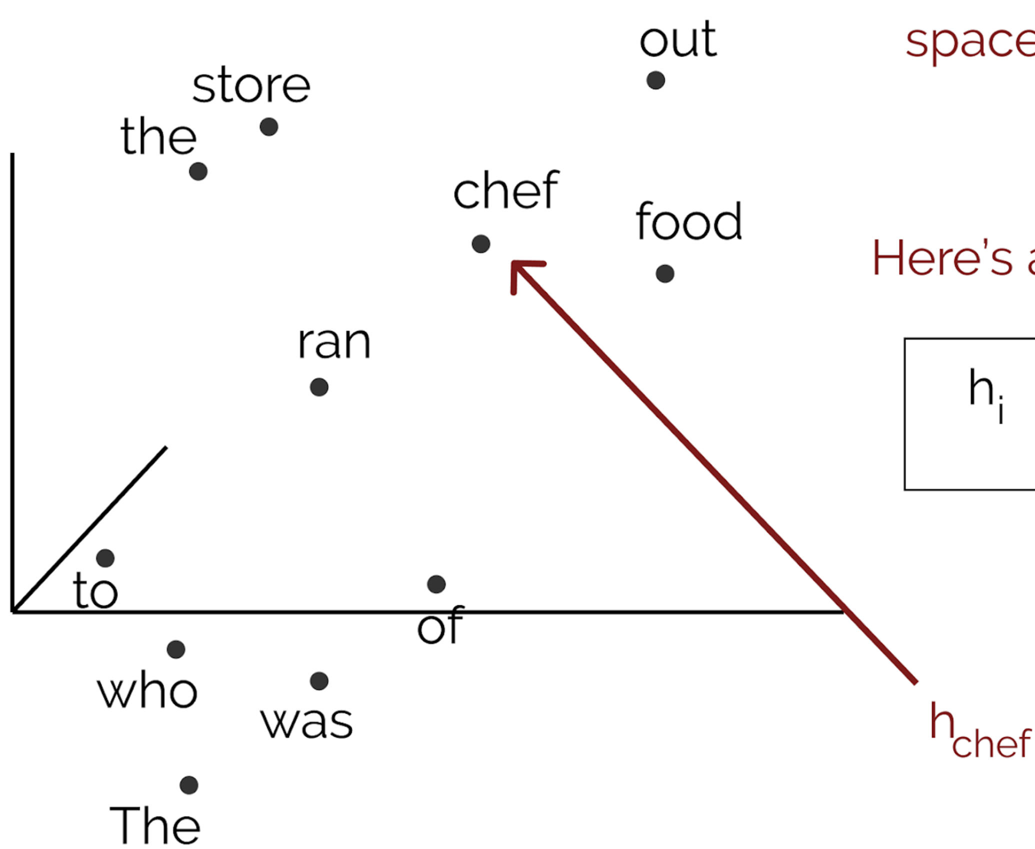
Distance metrics unify trees and vectors

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.

Finding trees in vector spaces

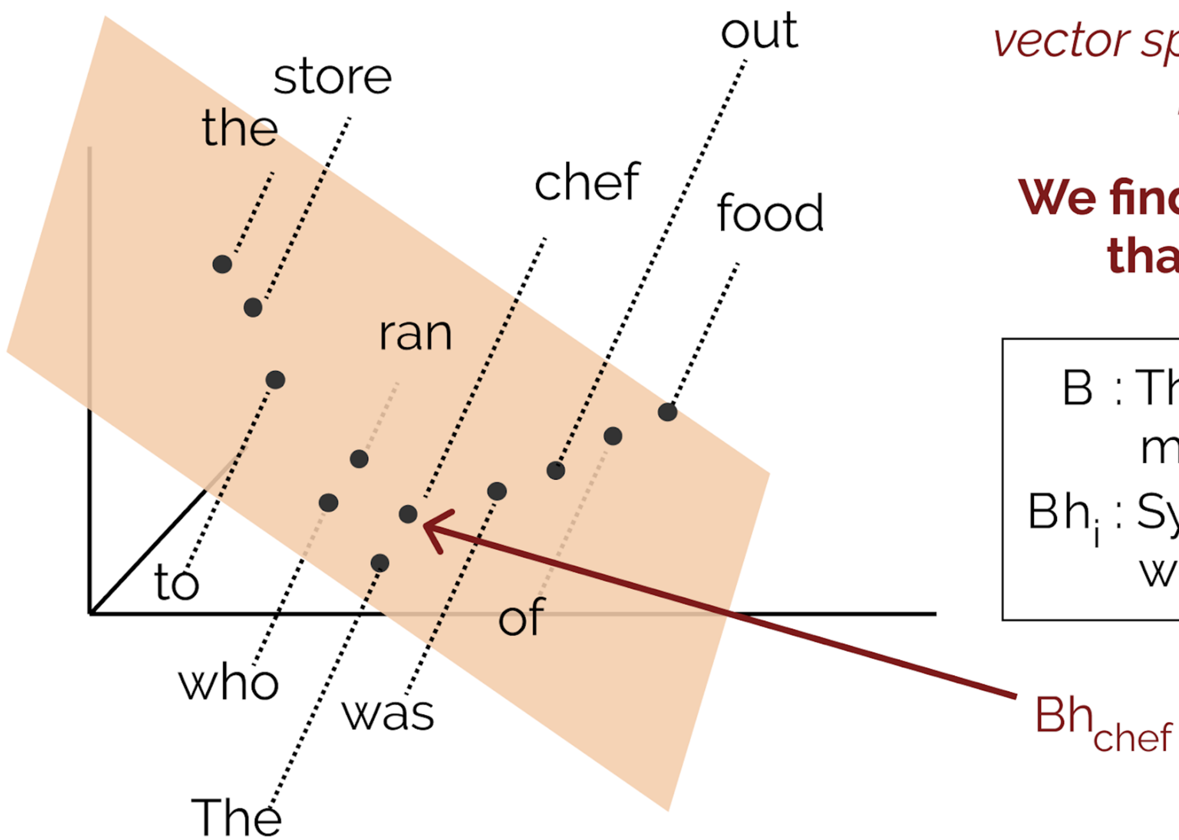


We can look for trees in the vector space by looking for their **distances** and **norms** in the space.

Here's a sentence embedded by a NN!

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

Finding trees in vector spaces



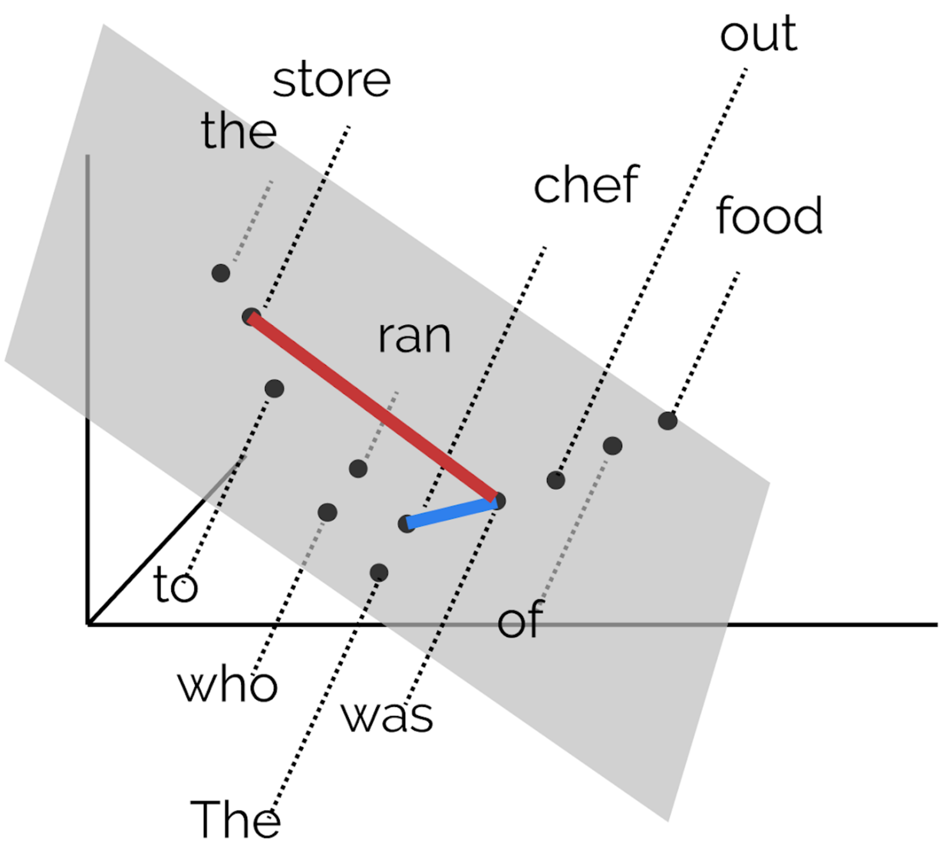
We don't expect all dimensions of the vector space to encode syntax -- NNs have a lot to encode!

We find the linear transformation that encodes syntax best.

B : The syntax transformation matrix

Bh_i : Syntax-transformed vector word representation

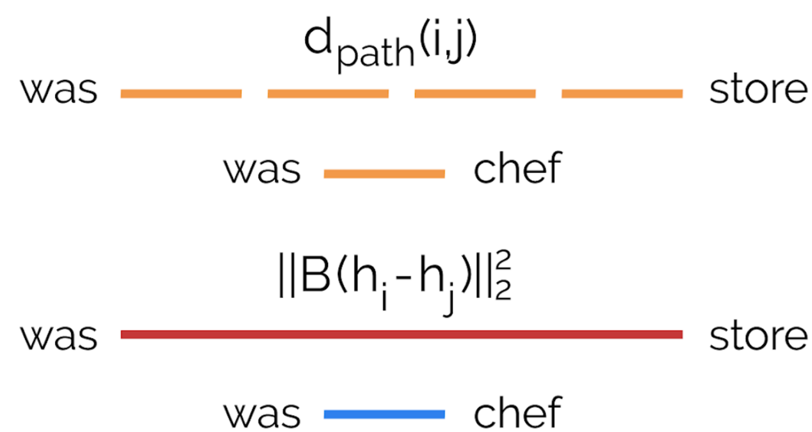
Finding trees in vector spaces



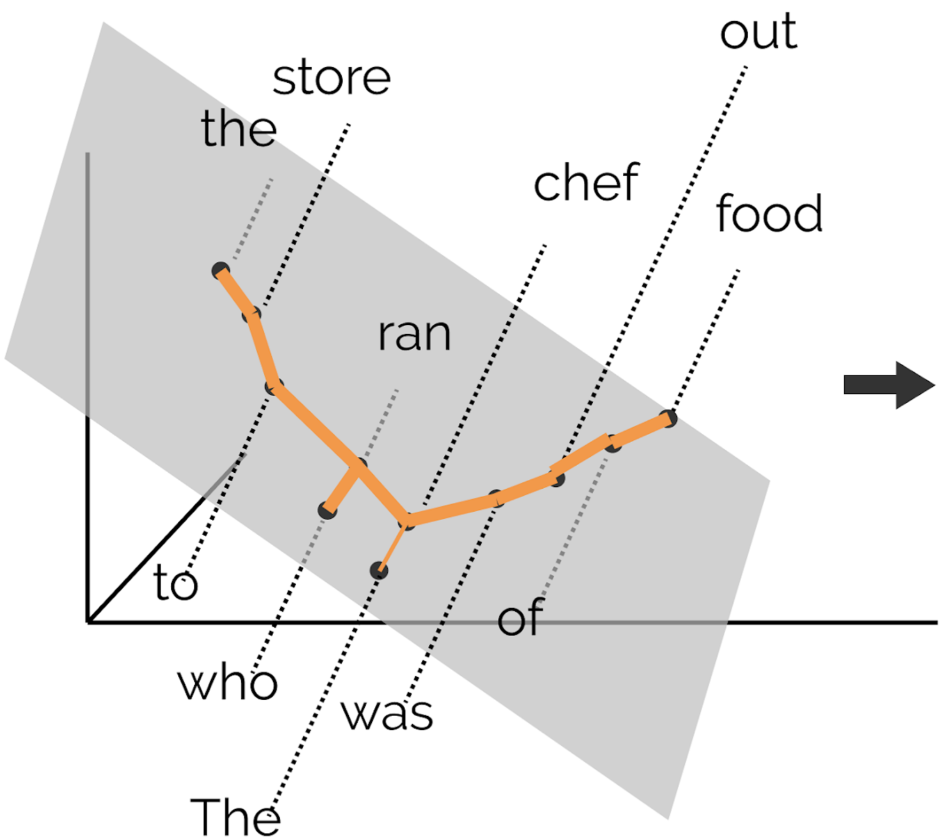
In the transformed space, (squared) L2 distance approximates tree distance.

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

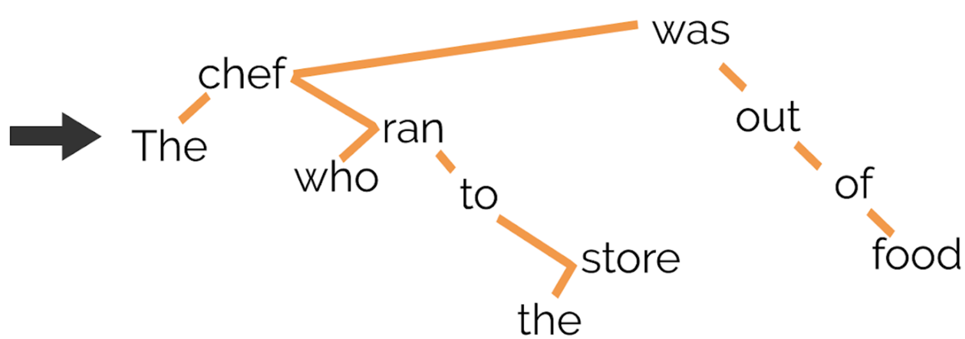
$\|B(h_i - h_j)\|_2^2$: Squared Vector space distance ($\|h_i - h_j\|_B^2$)



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?

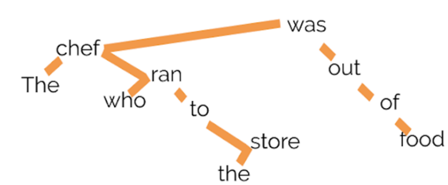
$$\arg\min_{B} \sum_{\ell \in \mathrm{PTB}} \frac{1}{|s^{\ell}|^2} \sum_{i,j} \left| d^{\ell}_{\mathrm{path}}(i,j) - \|B(h_i^{\ell} - h_j^{\ell})\|_2^2 \right|$$

Find a single transformation *B*

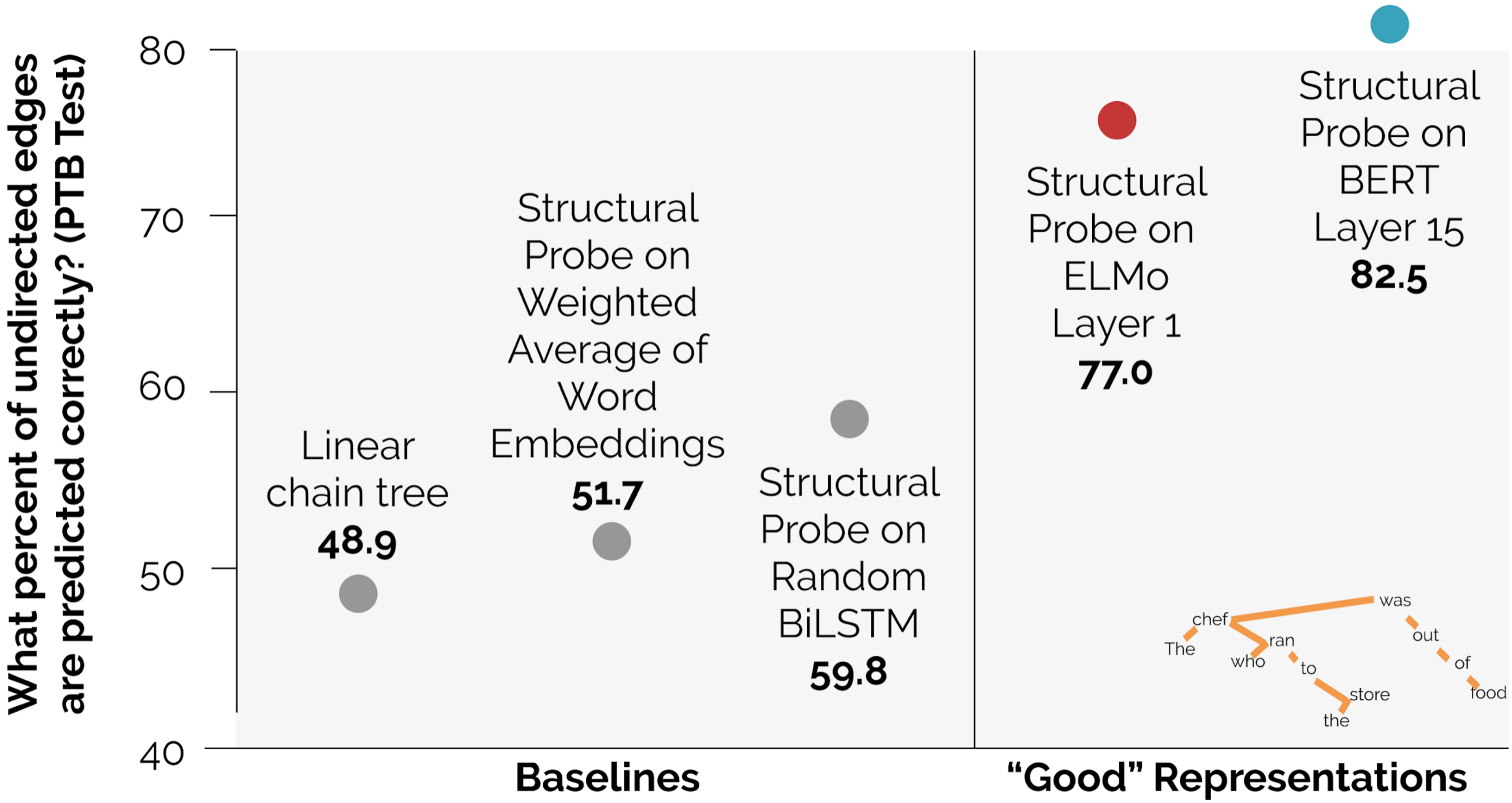
such that over all sentences in PTB training

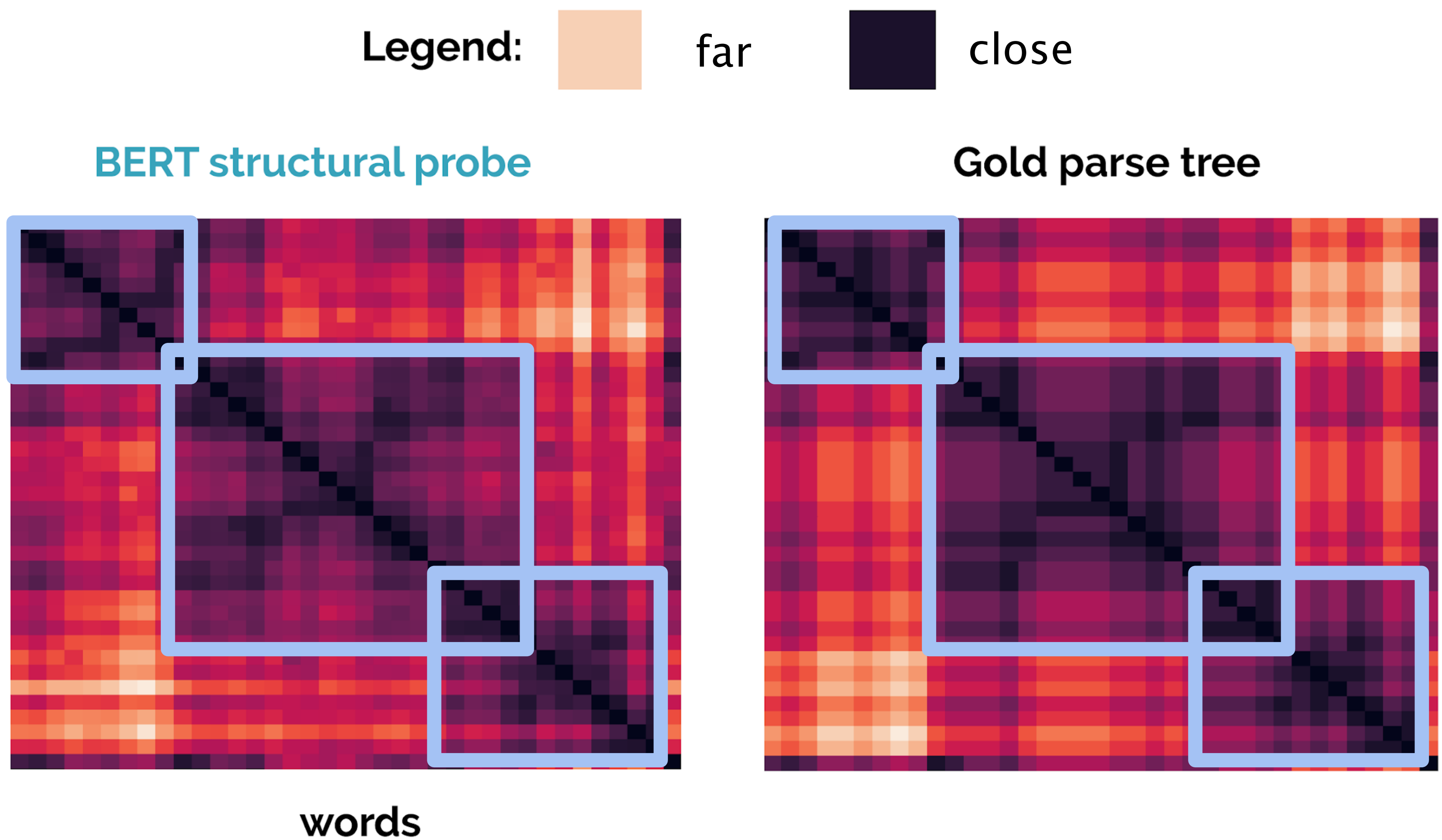
Over all word pairs in each sentence

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



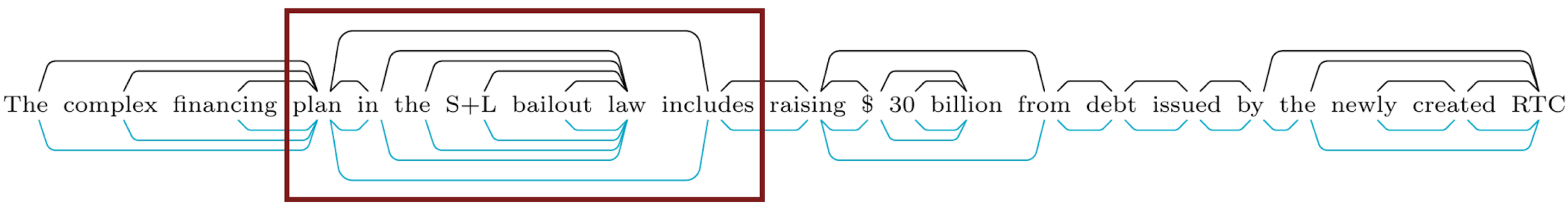
Trees are encoded well in these representations



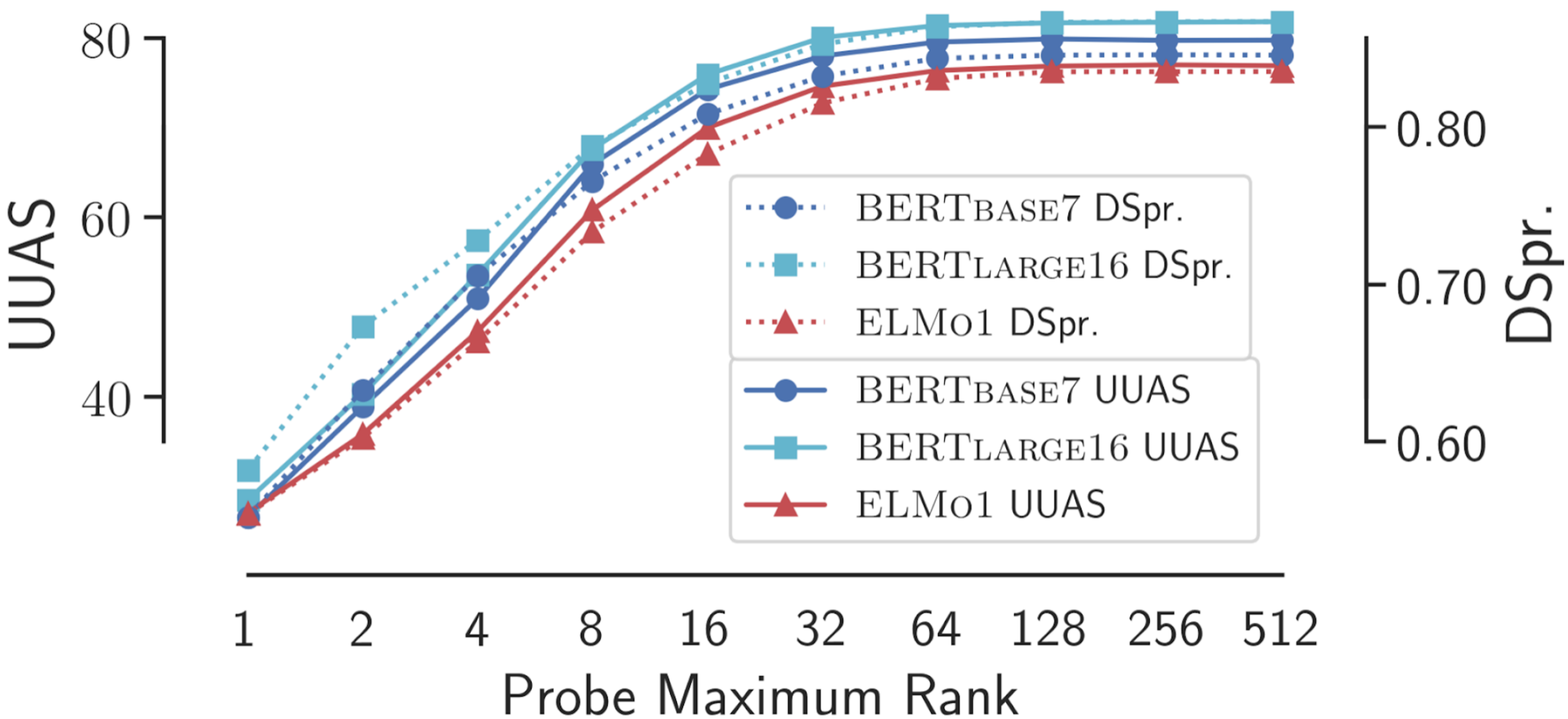

Legend:
far
close
BERT structural probe
Gold parse tree
words

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



Syntax geometry is quite low rank



Visualizing and Measuring the Geometry of BERT

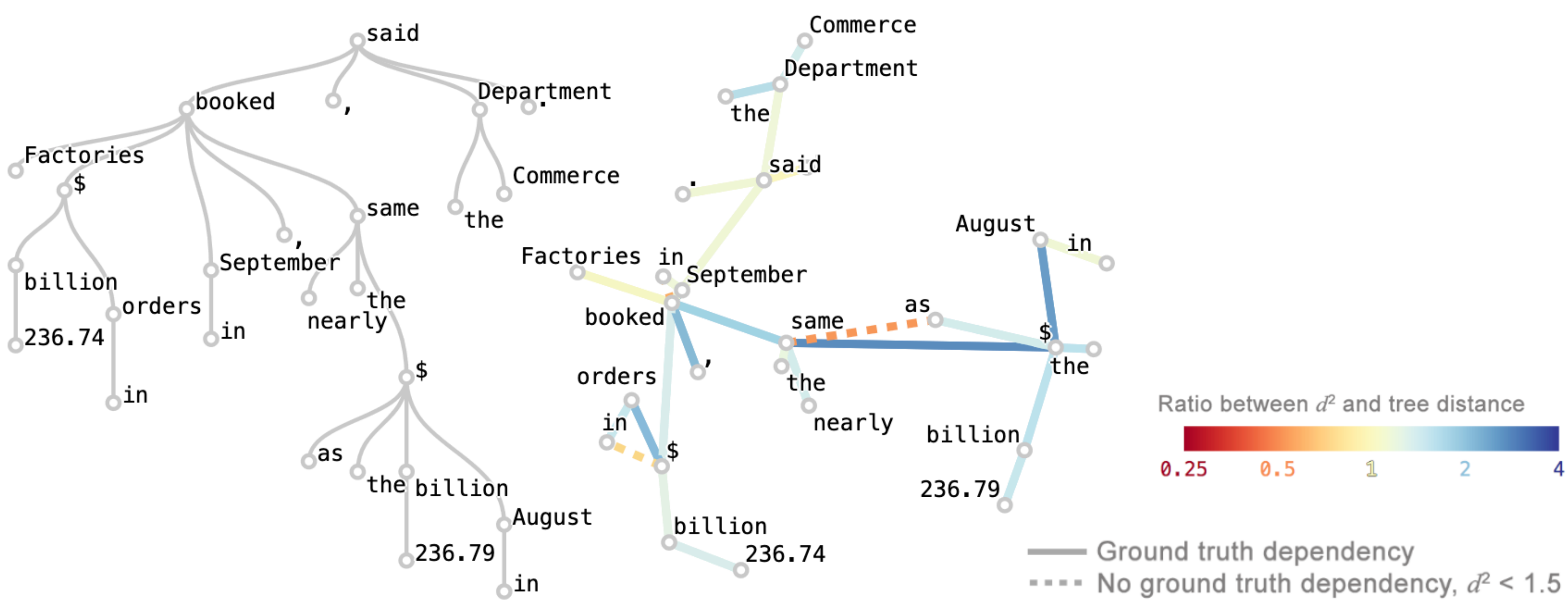
[Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg, NeurIPS 2019]

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

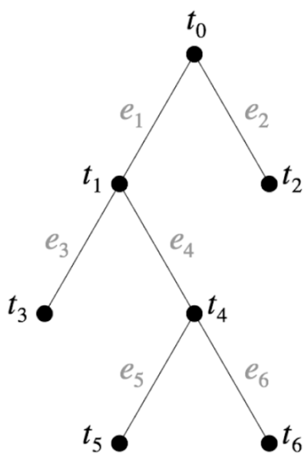
Visualizing and Measuring the Geometry of BERT

"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.

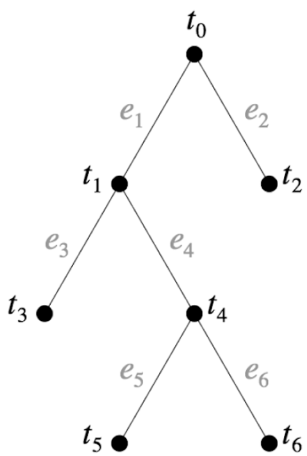


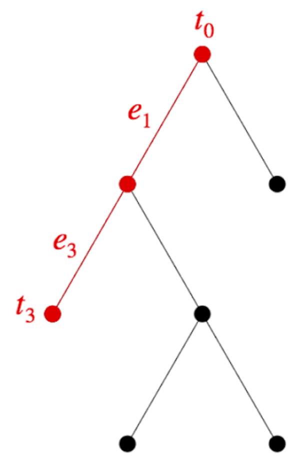
1. Assign edges orthogonal unit embeddings.

[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.

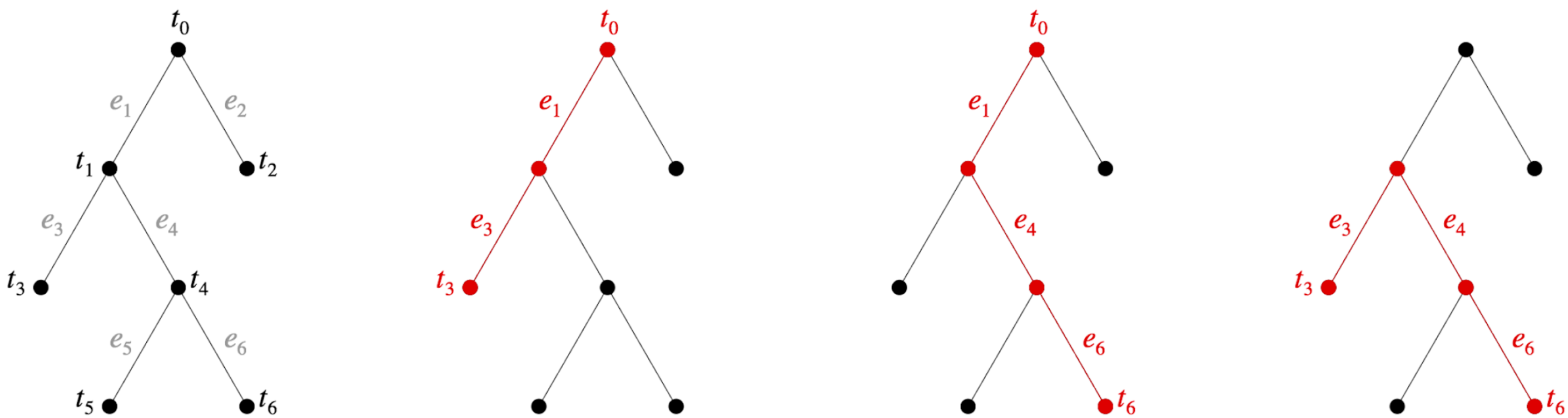
2. Assign each edge a direction (say, root-> leaf)

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



$f(t_3) =$

$e_1 + e_3 =$

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

$f(t_6) =$

$e_1 + e_4 + e_6 =$

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

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

$e_3 - e_4 - e_6 =$

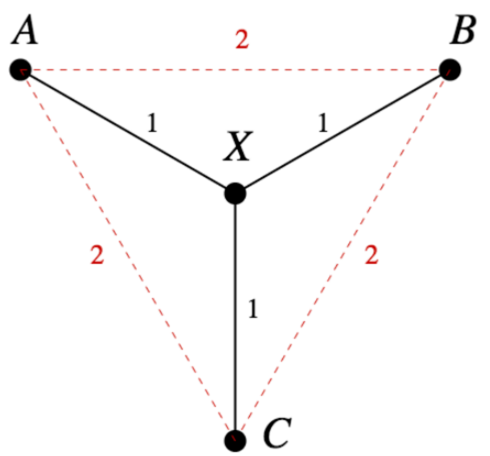
$(0, 0, 1, -1, 0, -1)$

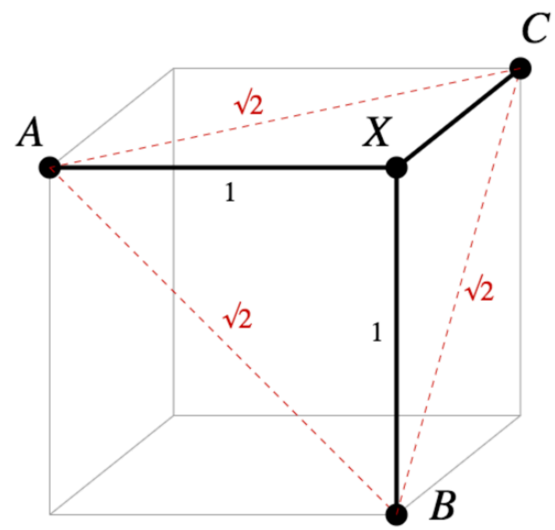
$\| f(t_3) - f(t_6) \|^2 = 3$

[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





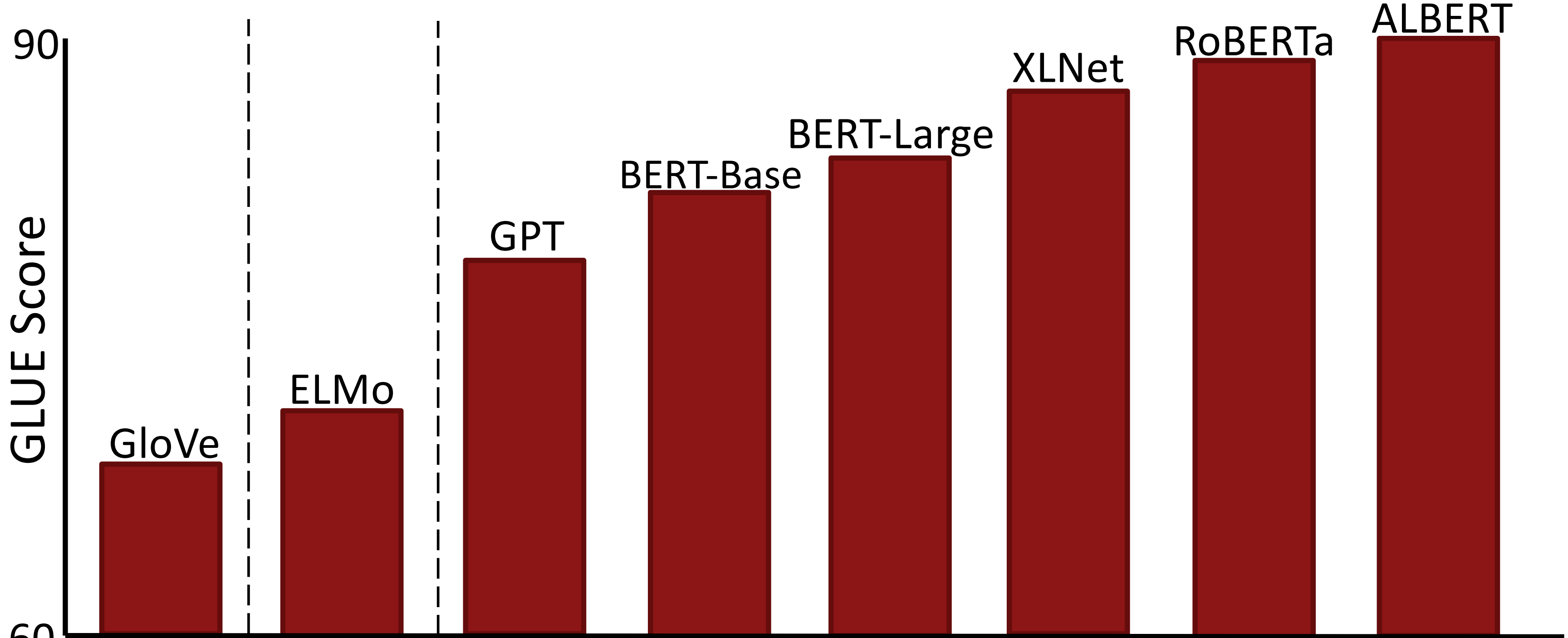
You can encode it in a "Pythagorean embedding"

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

3. Electra: Efficient Discriminative Pre-training of Text Encoders

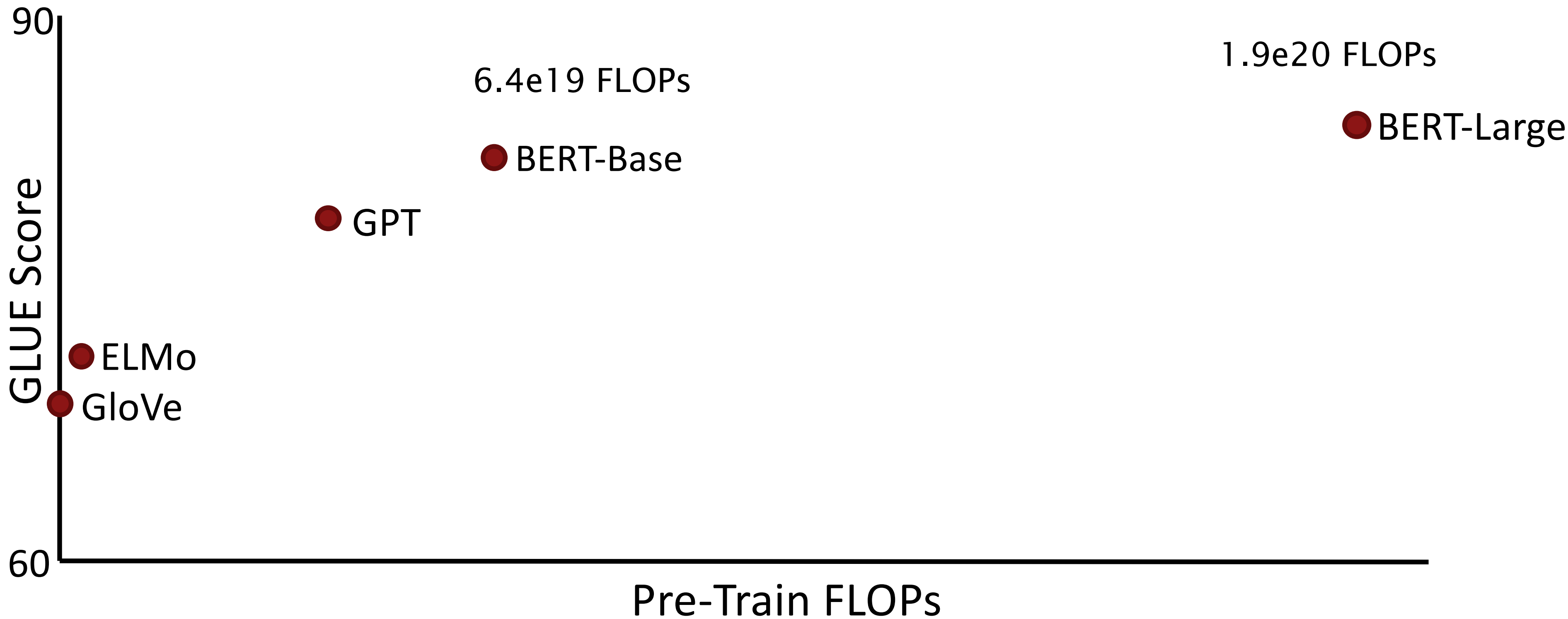
- Kevin Clark and **Christopher Manning**

Rapid Progress from Pre-Training (GLUE benchmark)



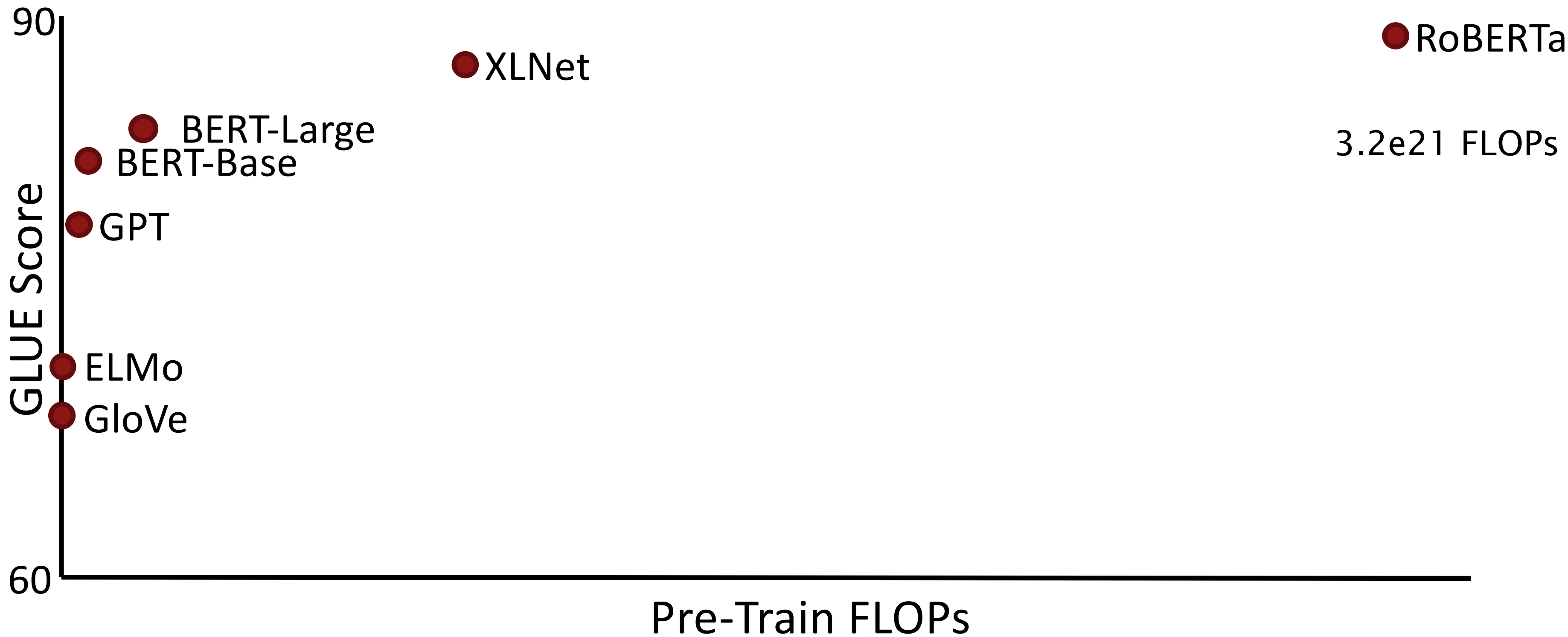
Over 3x reduction in error in 2 years, “superhuman” performance

But let’s change the x-axis to compute ...



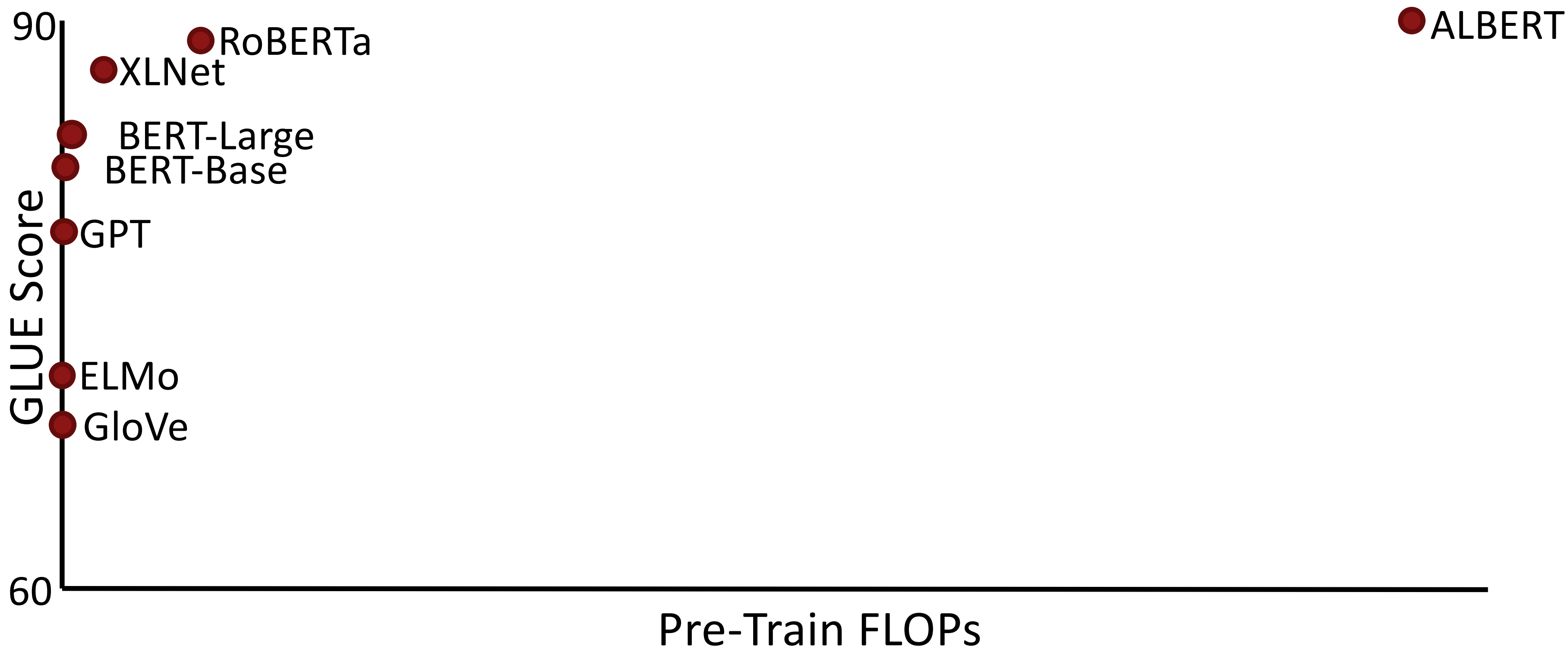
BERT-Large uses 60x more compute than ELMo

But let’s change the x-axis to compute …



RoBERTa uses 16x more compute than BERT-Large

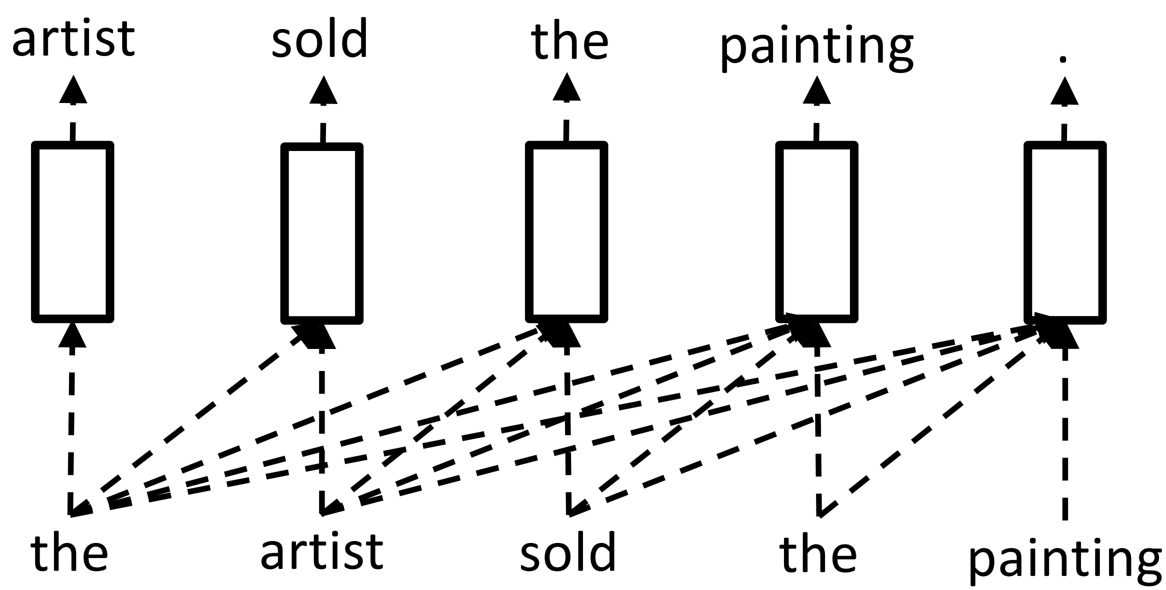
More compute, more better?



ALBERT uses 10x more compute than RoBERTa

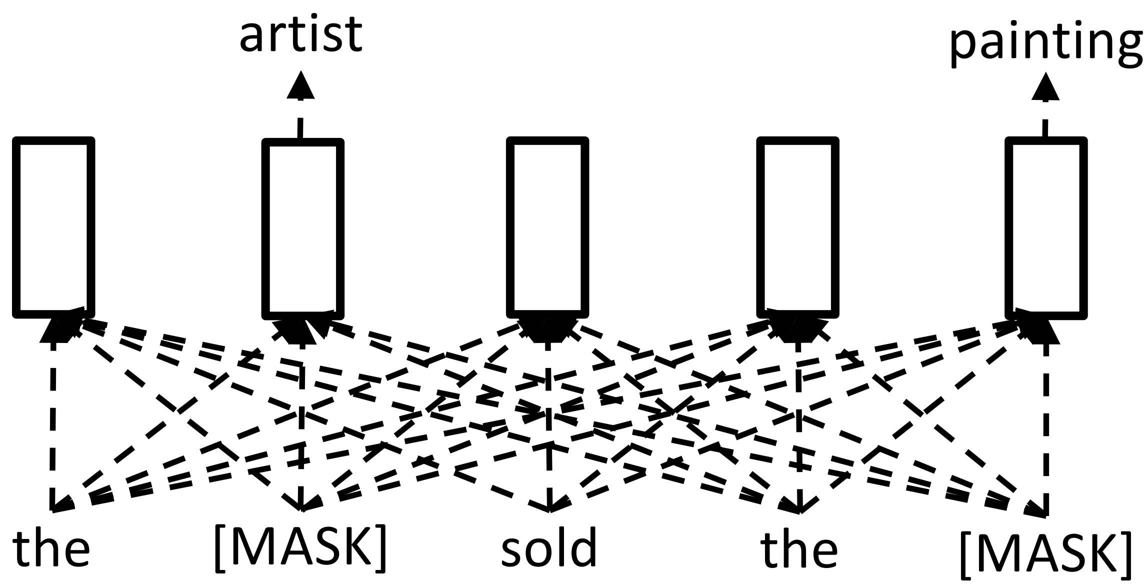
Language Model Pretraining

- ULMFit, ELMo, GPT, …



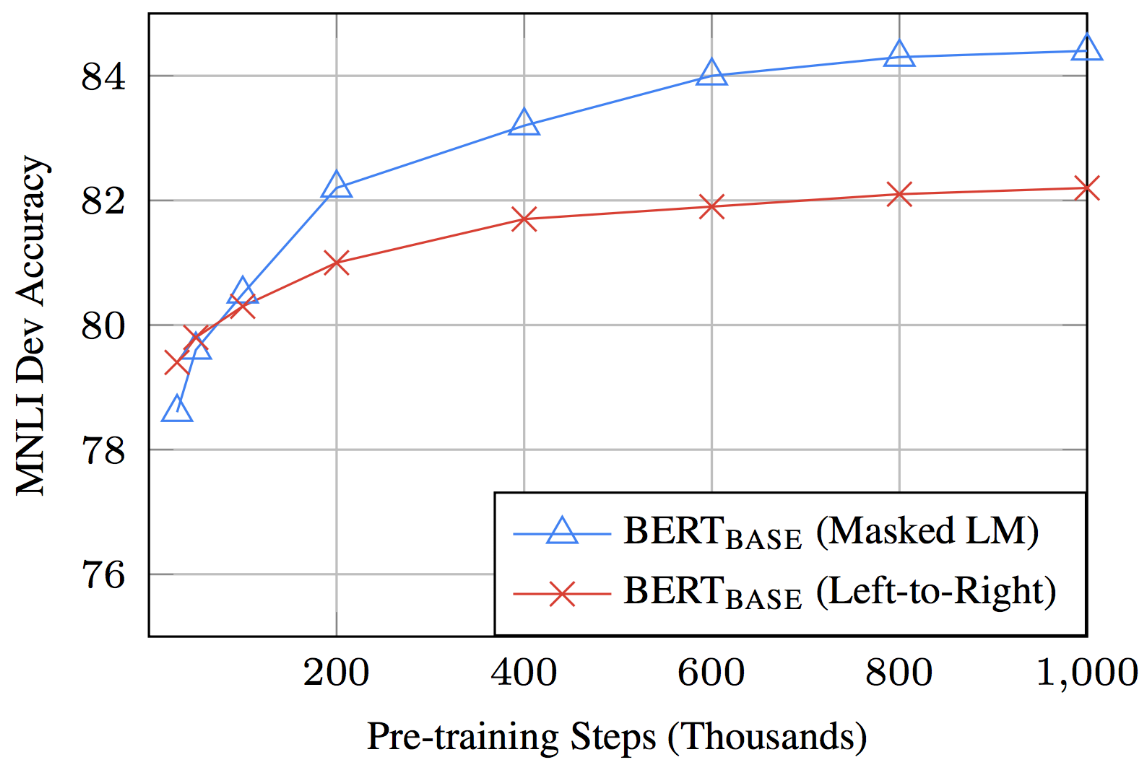
Masked Language Model Pretraining

- BERT, XLNet, RoBERTa, ...



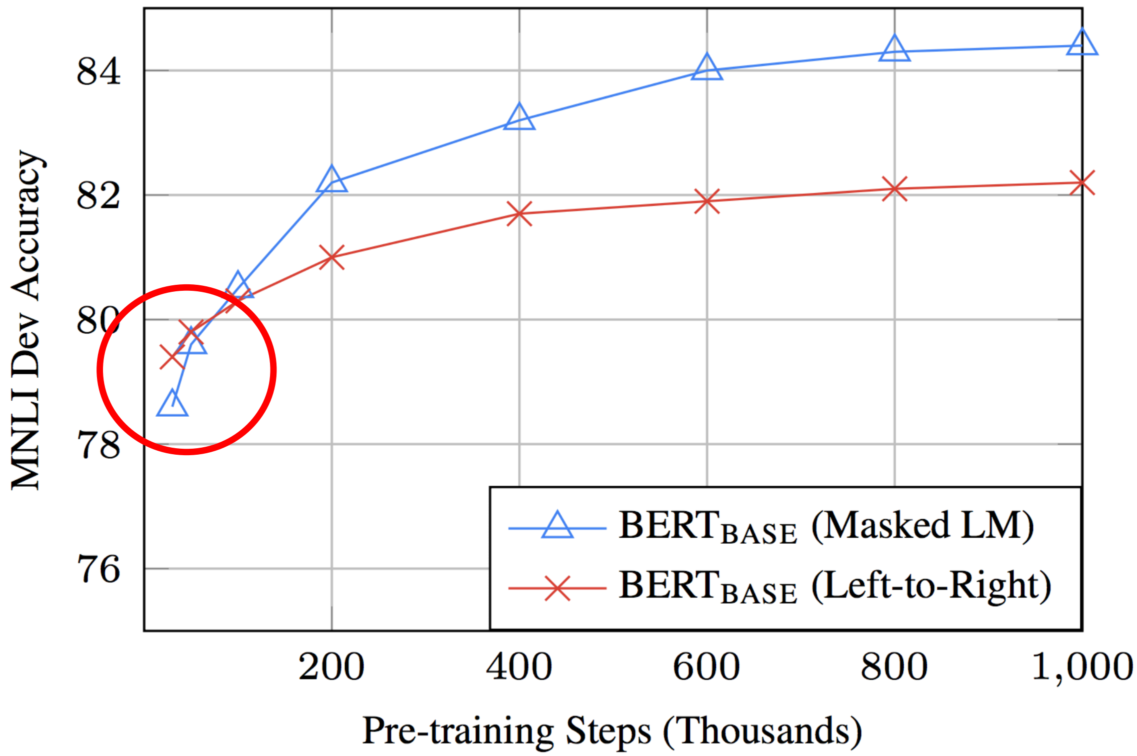
Masked Language Model Pretraining

- Bidirectional gives better performance



Masked Language Model Pretraining

- Bidirectional gives better performance
- But less efficient because only learn from 15% of tokens per example
- **Our method: best of both worlds**



New Pre-Training Task: Replaced Token Detection

- Instead of [MASK], replace tokens with plausible alternatives

the artist sold the painting

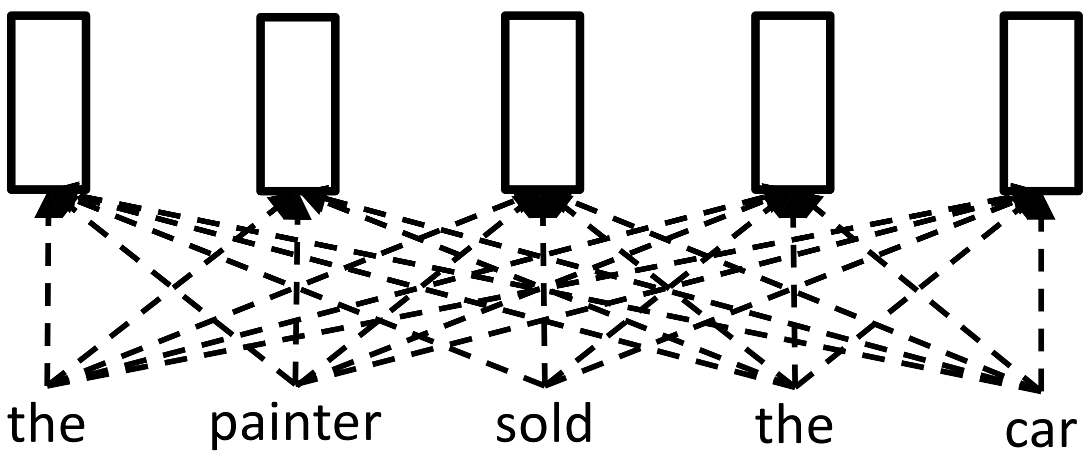
New Pre-Training Task: Replaced Token Detection

- Instead of [MASK], replace tokens with plausible alternatives

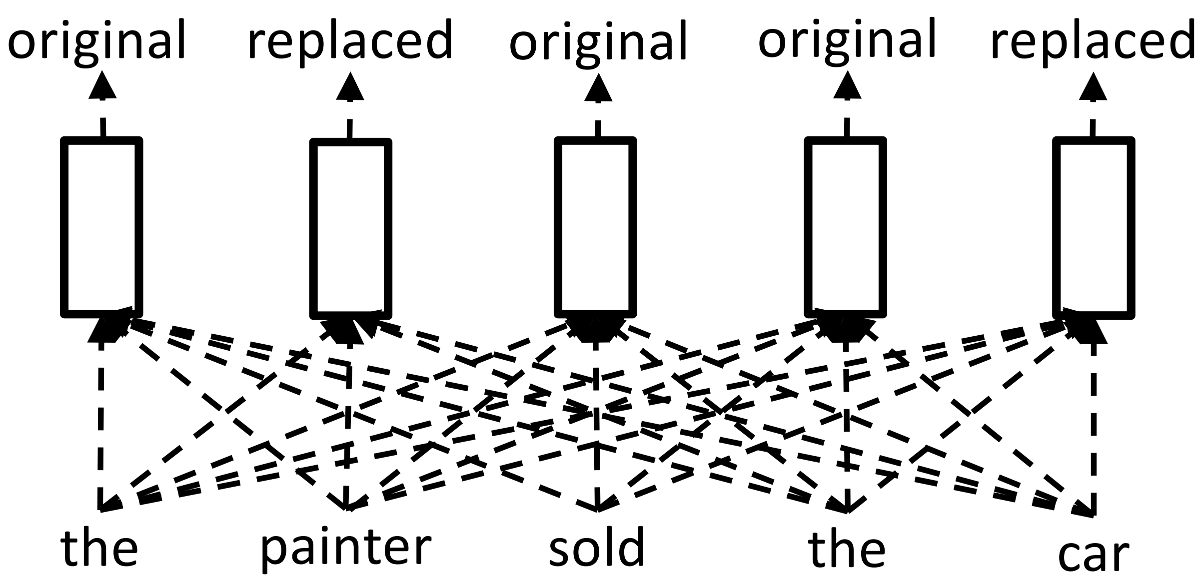
painter car

the ~~artist~~ sold the ~~painting~~

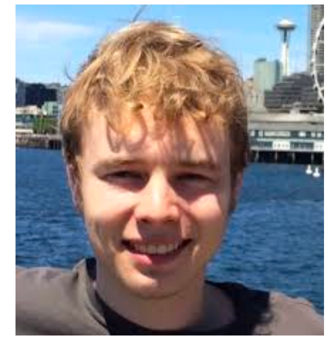
New Pre-Training Task: Replaced Token Detection



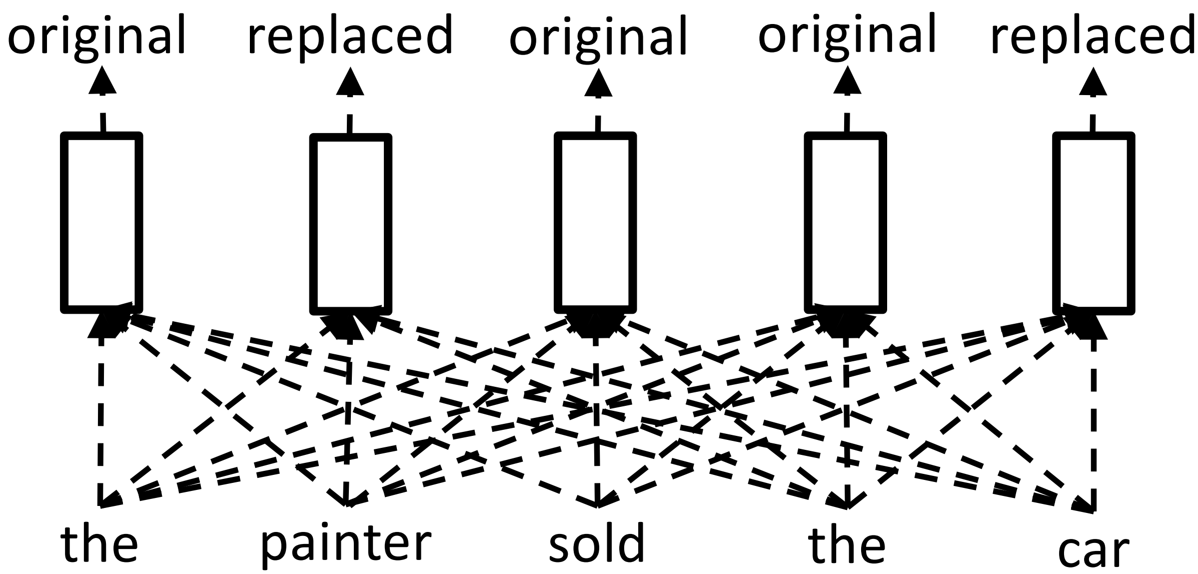
New Pre-Training Task: Replaced Token Detection



ELECTRA: Efficiently Learning an Encoder to Classify Token Replacements Accurately



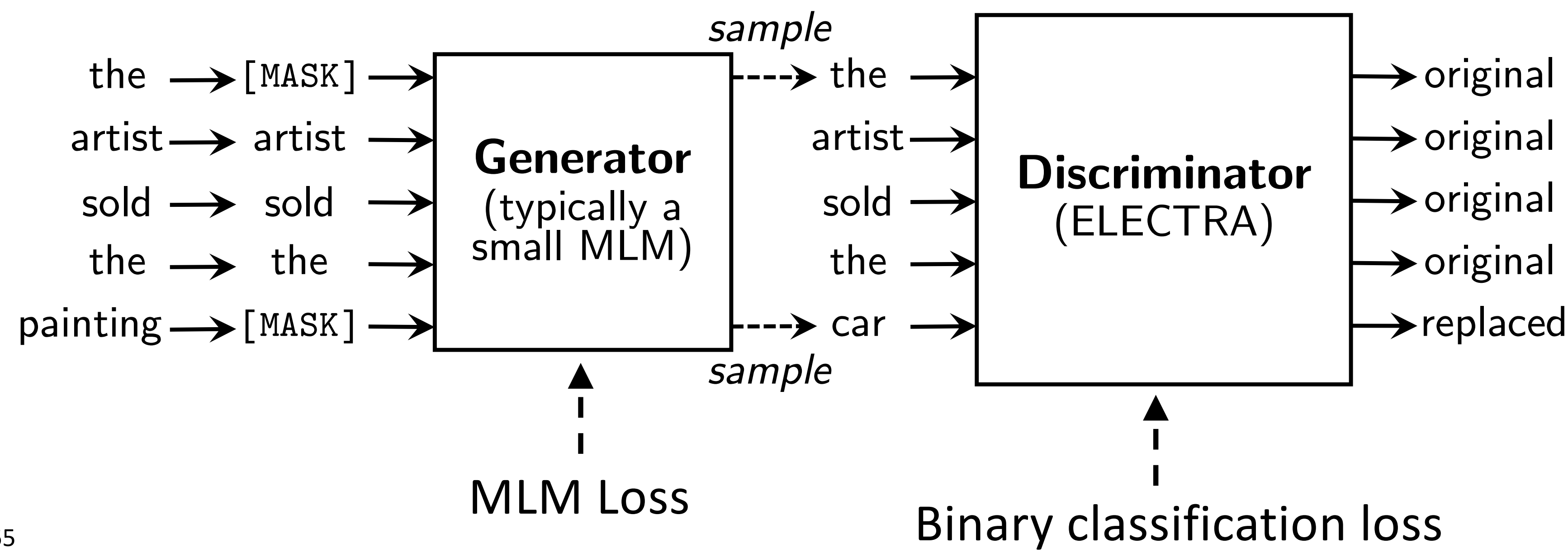
Bidirectional model but learn from all tokens



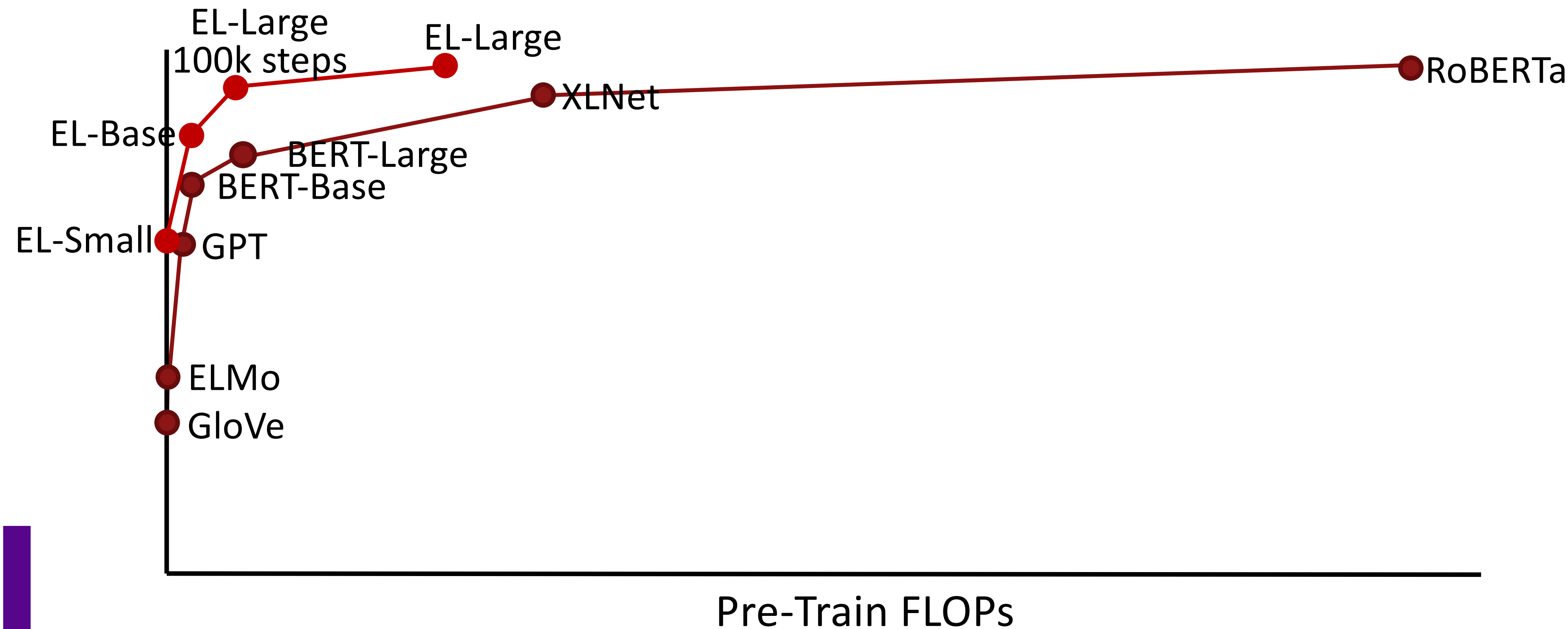
Clark, Luong, Le, and Manning (2020)

Generating Replacements

Plausible alternatives come from small masked language model (the “generator”) trained jointly with ELECTRA



Results: Glue Score vs Compute



GLUE Results: ELECTRA-Small and smaller and smaller

Model	Train/Infer Speedup over BERT-Base	GLUE Score	Train time / hardware
ELMo	19x / 1.2x	71.2	14d on 3 1080s
ELECTRA 6.25%	**722x / 8x**	**74.1**	**6h on 1 V100**
BERT-Small (ours)	45x / 8x	75.1	4d on 1 V100
ELECTRA 25%	**181x / 8x**	**77.7**	**1d on 1 V100**
DistilBERT	- / 2x	77.8	
GPT	1.6x / 1x	78.8	
ELECTRA-Small	**45x / 8x**	**79.0**	**4d on 1 V100**
BERT-Base	1x / 1x	82.2	4d on 16 TPUv3s

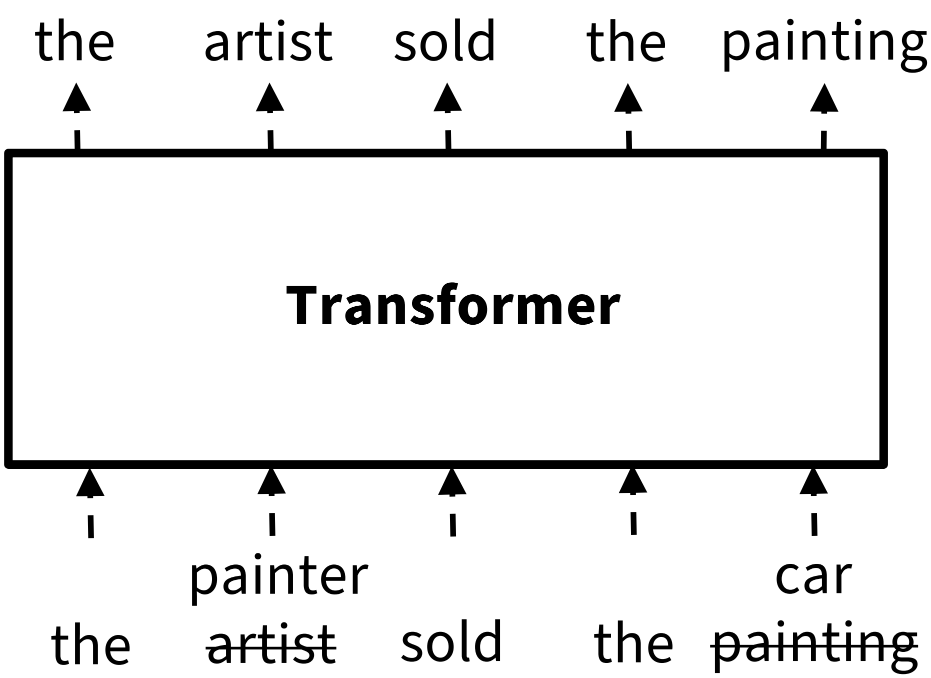
SQuAD 2.0 dev Results: ELECTRA-Large

- BERT-Large architecture, trained on XLNet data

Model	Train FLOPs	F1 Score
BERT	0.3x	81.8
XLNet	1.3x	88.8
RoBERTa (100k steps)	0.9x	87.7
RoBERTa	4.5x	89.4
BERT-large (ours)	1x	87.5
ELECTRA	**1x**	**89.6**

Clark, Luong, Le, and Manning (2020)

Efficiency Ablations: All-Tokens MLM



Model	GLUE Score
BERT	82.2
Replace MLM	82.4
ELECTRA 15%	82.4
All-Tokens MLM	**84.3**
ELECTRA	**85.0**

Clark, Luong, Le, and Manning (2020)

Electra

- Recent pre-training methods let models benefit from unprecedented compute scale
  - But our environment/energy use doesn't benefit!
  - It is important to be sensitive to compute when reporting results
- Replaced token detection is a more effective pre-training task then masked language modeling
  - Can provide good results on a single GPU in hours/days
  - At larger scale, trains over 4x faster

Final thoughts

- Self-supervised (or “unsupervised”) learning is very successful for doing natural language understanding tasks
  - More successful than multi-task learning (if only because of data supply)
- However, one key limitation has been the size/cost of models
- Was annotating lots of linguistic data all a mistake?
  - Maybe. Language model learning exploits a much richer task compared to the categories in typical annotations
  - Of course, we still fine tune, test, etc.

Final thoughts

- Is linguistic structure all a mistake?
  - No! Deep contextual word representations have phase-shifted from statistical association learners to **language discovery devices**!
  - Syntax, coref, etc. emerges (approximately) in the geometry of BERT! See:
    - 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.
- Does going big stretch any analogy to child language acquisition?
  - Maybe, but it's more that acquisition without grounding is unrealistic

Deep Contextual Neural Word Representations: Linguistic Structure Discovery and Efficient Discriminative Training

Stanford

Christopher Manning

Stanford University and CIFAR Fellow

@chrmanning ❀ @stanfordnlp

ElementAI/MILA, December 2019 (last talk of 2019!)