# Natural Language Processing with Deep Learning

## CS224N/Ling284

Stanford | NLP

John Hewitt

# Analysis and Interpretability of Neural NLP

# Course Logistics

**Final Project Report**

- Due Saturday, March 14 at 11:59 PM PST
  - 1 late day: Sunday, March 15 at 11:59 PM PST
  - 2 late days: Monday, March 16 at 11:59 PM PST
  - 3 late days: Tuesday, March 17 at **4:30 PM PST**

**Final Project Poster**

- Zoom 3-minute poster presentations with 2 TAs and a cohort of ~14 other teams
  - Monday, March 16 at 5PM - 7PM PST
  - Monday, March 16 at 7.30PM - 9:30PM PST
  - Tuesday, March 17 at 9AM - 11AM PST
- Fill out the form on Piazza with your time preferences

# Lecture Outline

Lecture 20: Analysis and Interpretability of Neural NLP

1. **Motivation: what are our models doing? (10 mins)**
2. Neural networks as linguistic test subjects (10 mins)
3. Careful ablation studies and architecture modifications (5 mins)
4. Analysis of inherently interpretable architectures (5 mins)
5. Playing the adversary: breaking NLP models (5 mins)
6. Analyzing representations using supervised methods (35 mins)
7. Aggregating analysis insights across studies (10 mins)

# Motivation: what are our models doing?

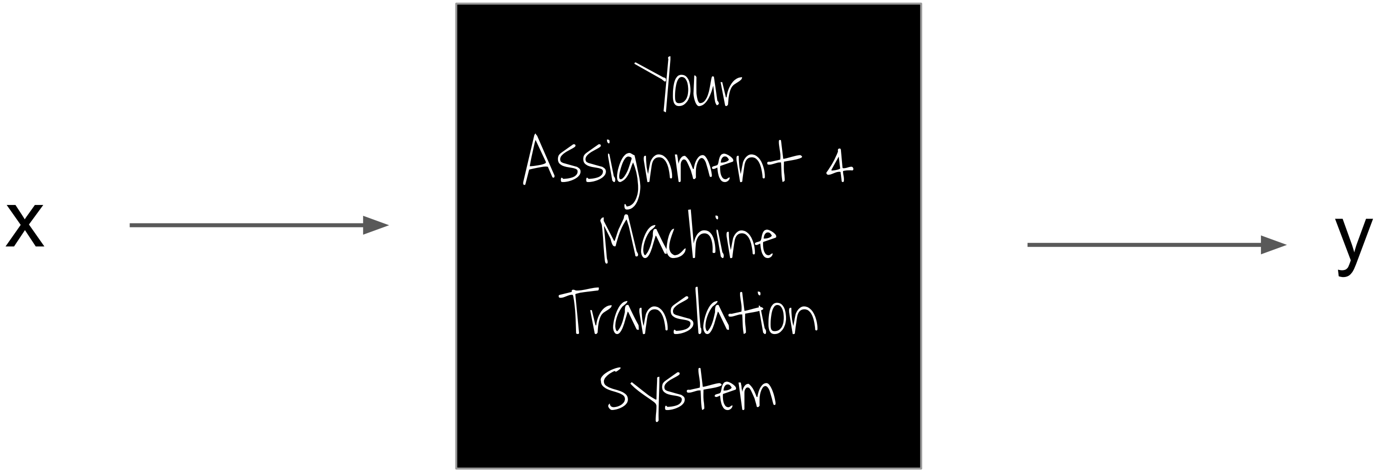
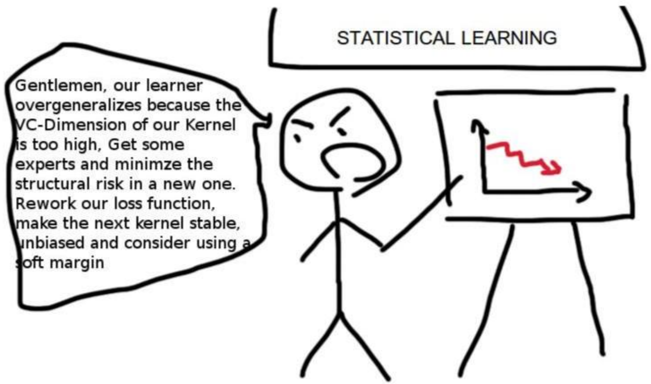



*Fig 1: A black box*

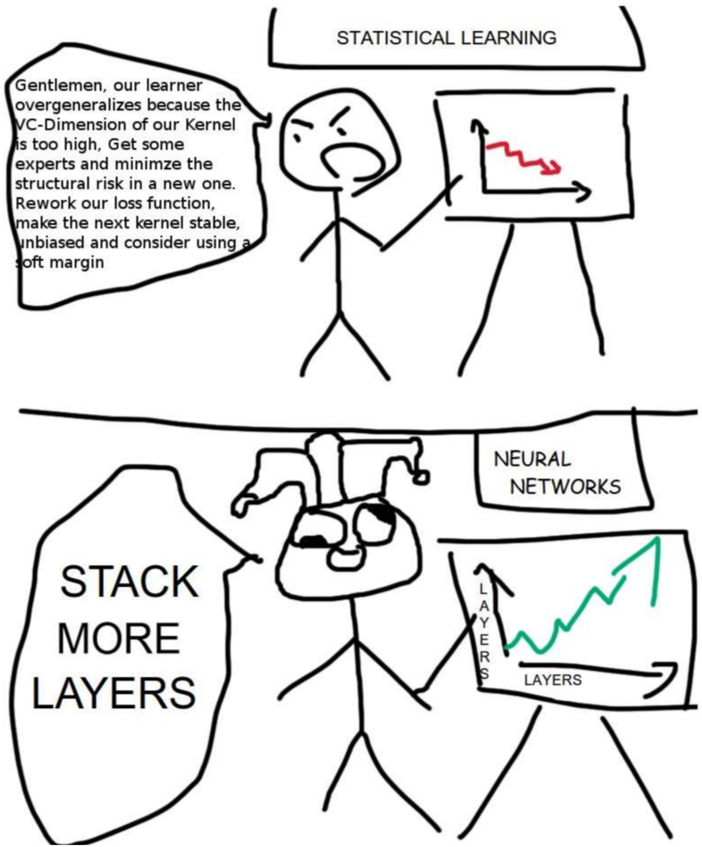
It is not clear what *functions* our algorithms learn, and their complexity precludes exact understanding

# Motivation: how do we make models better?



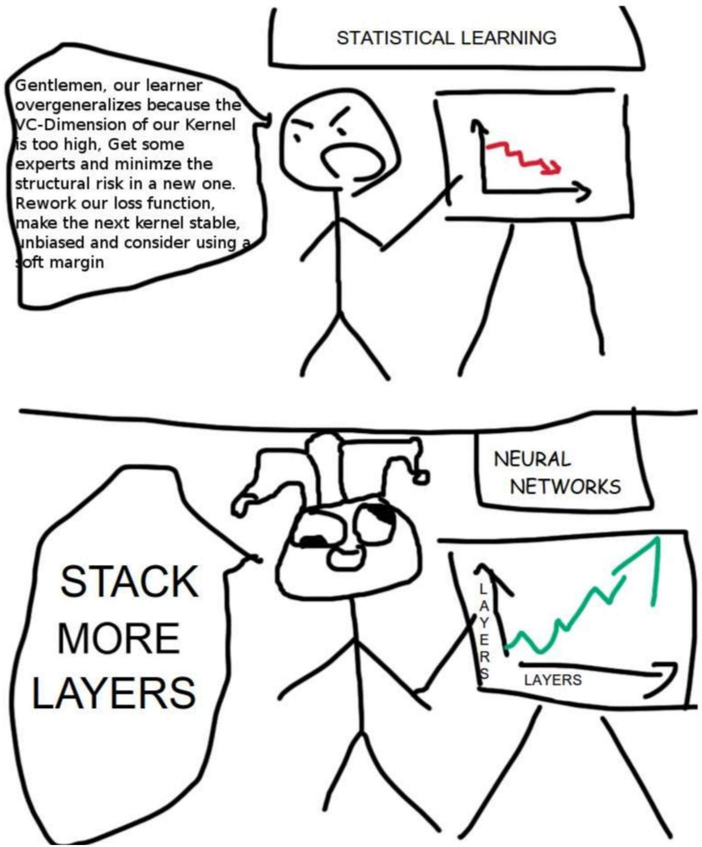

[Reddit; source unknown]

# Motivation: how do we make models better?




[Reddit; source unknown]

# Motivation: how do we make models better?



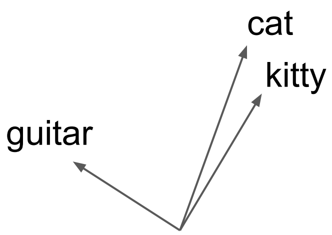

[Reddit; source unknown]


[xkcd.com]

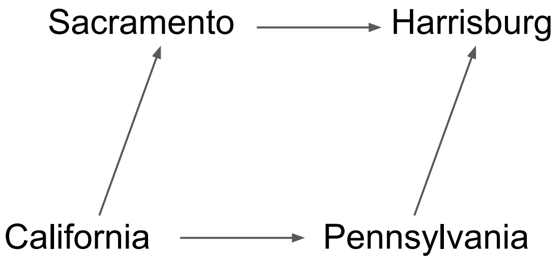
# What we’ve seen: simple analyses of word2vec

**Bold type: Math property**

*Italic type: interpretation*




We interpret **cosine similarity** as *semantic similarity*



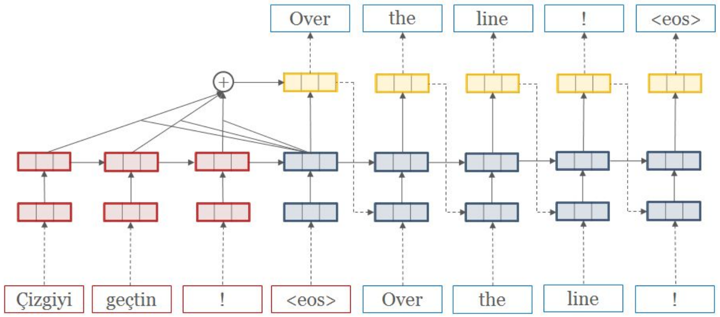

Some *relationships* are encoded as **vector differences**

Knowing what *properties* word embeddings have: useful for practitioners!

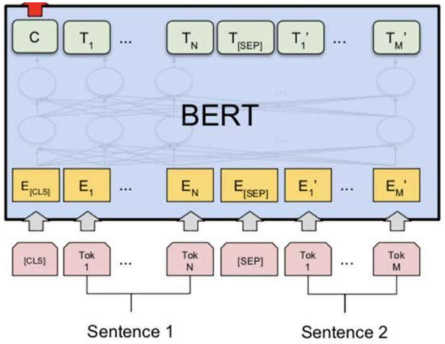
Knowing that word embeddings encode *undesirable social biases*: useful for everyone!

# Neural networks are worthy subjects of study

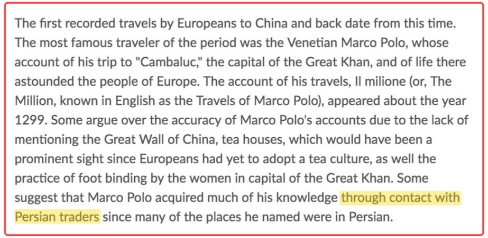
***Machine Translation***




***Language Modeling***




***Question Answering***




How did some suspect that Polo learned about China instead of by actually visiting it?

**Answer:** through contact with Persian traders

It's wild that any of our models work at all

- Their behavior is an emergent property of *data* and our *design decisions*
- Accuracy on a held out test set is not sufficient to fully characterize them

# Lecture Outline

Lecture 20: Analysis and Interpretability of Neural NLP

1. Motivation: what are our models doing?
2. **Neural networks as linguistic test subjects**
3. Careful ablation studies and architecture modifications
4. Analysis of inherently interpretable architectures
5. Playing the adversary: breaking NLP models
6. Analyzing representations using supervised methods
7. Aggregating analysis insights across studies

# Neural networks as linguistic test subjects

# ~~Neural networks~~ Humans as linguistic test subjects

# ~~Neural networks~~ Humans as linguistic test subjects

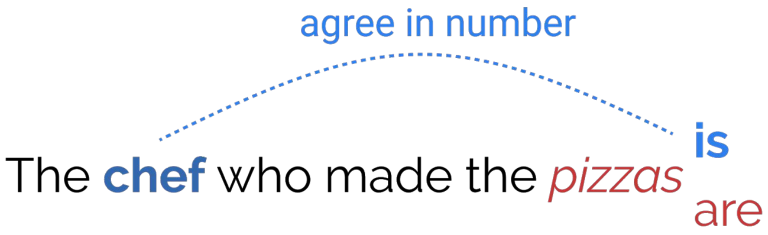
How do we understand language behavior in **humans**?

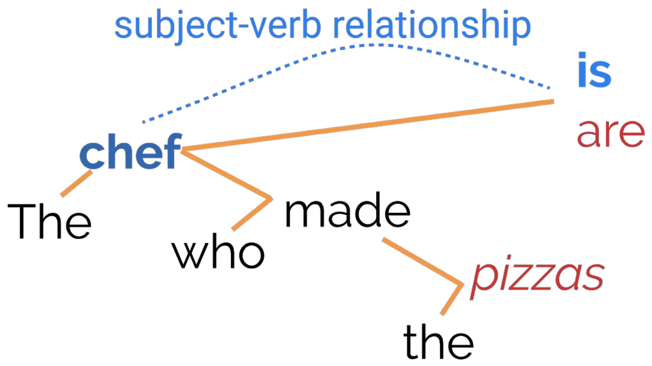
One method: *minimal pairs. What sounds "okay" to a speaker?*

The chef who made the pizzas is ← "*Acceptable*"

*The chef who made the pizzas are ← "*Unacceptable*"

Idea: English present-tense verbs *agree in number* with their subject.



# Neural networks as linguistic test subjects

How do we understand language behavior in **language models**?

One method: *minimal pairs. Is the acceptable sentence higher-probability?*

The chef who made the pizzas is → Language Model → 0.0001

\>

The chef who made the pizzas are → Language Model → 0.0000001

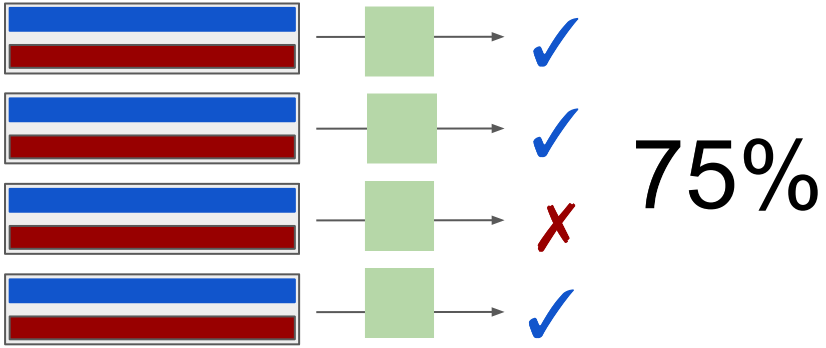
**Premise:** A language model should assign higher probability to the acceptable sentence in any minimal pair.

Linzen et al., 2016

# Neural networks as linguistic test subjects

Steps to conduct a *minimal pairs* test on a language model:

1. Gather or construct a test set of minimal pairs which require specific aspects of understanding to distinguish.
2. Run your language model on the pairs, and report percent of pairs the model predicts as desired.



# Neural networks as linguistic test subjects

Example: Do LMs show Subject-Verb number agreement across attractors?

The chef who made the pizzas and talked to the customers is

*subject* *attractor* *attractor* *verb*

| | n=0 | n=1 | n=2 | n=3 | n=4 |
|---|---|---|---|---|---|
| Random | 50.0 | 50.0 | 50.0 | 50.0 | 50.0 |
| Majority | 32.0 | 32.0 | 32.0 | 32.0 | 32.0 |
| LSTM, H=50† | 6.8 | 32.6 | ≈50 | ≈65 | ≈70 |
| Our LSTM, H=50 | 2.4 | 8.0 | 15.7 | 26.1 | 34.65 |
| Our LSTM, H=150 | 1.5 | 4.5 | 9.0 | 14.3 | 17.6 |
| Our LSTM, H=250 | 1.4 | 3.3 | 5.9 | **9.7** | 13.9 |
| Our LSTM, H=350 | **1.3** | **3.0** | **5.7** | **9.7** | **13.8** |
| 1B Word LSTM (repl) | 2.8 | 8.0 | 14.0 | 21.8 | 20.0 |
| Char LSTM | **1.2** | 5.5 | 11.8 | 20.4 | 27.8 |

# of attractors between subject and verb

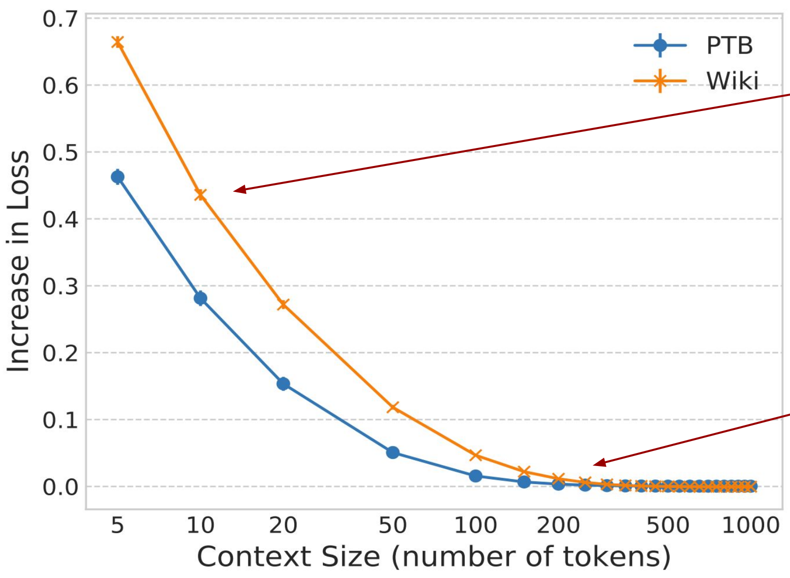
Error rate on a large corpus of minimal pairs

LMs do *really* well!?

[Kuncoro et al., 2018]

# Neural networks as linguistic test subjects

Method: Modify the test set to remove long contexts, or replace them with longer words. Evaluate whether the LM perplexity changes.




Only giving the LM 10 words of context at test time makes the test error go up.

Only giving the LM 250 words of context *doesn't change its loss*, so it's not using contexts longer than 250 words much.

[Khandelwal et al., 2018]

# Neural networks as linguistic test subjects

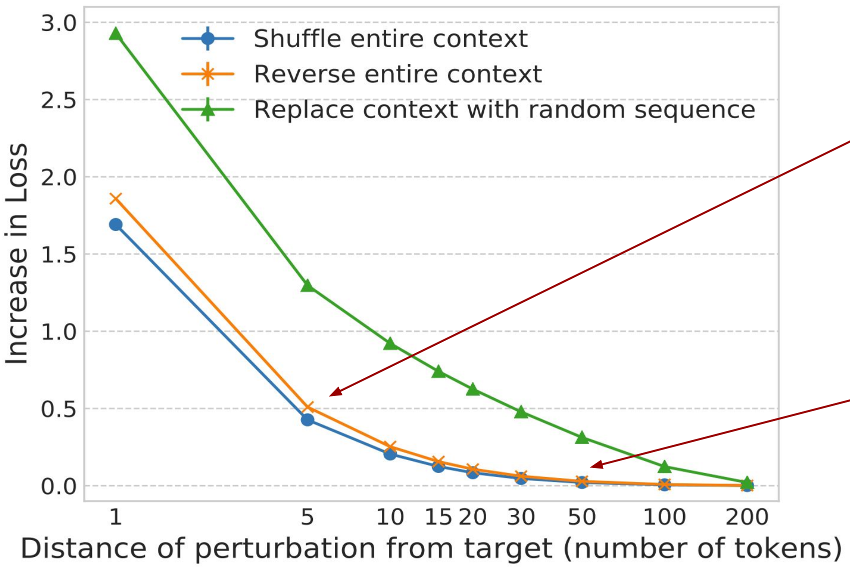
Question: How does an LSTM language model use its long-distance contexts?

Method: Modify the test set to remove long contexts, or replace them with longer words. Evaluate whether the LM perplexity changes.

[Khandelwal et al., 2018]

# Neural networks as linguistic test subjects

Method: Modify the test set to remove long contexts, or replace them with longer words. Evaluate whether the LM perplexity changes.




Shuffling the order of the context further than 5 words away increases loss, so the LM cares about word order past 5 words.

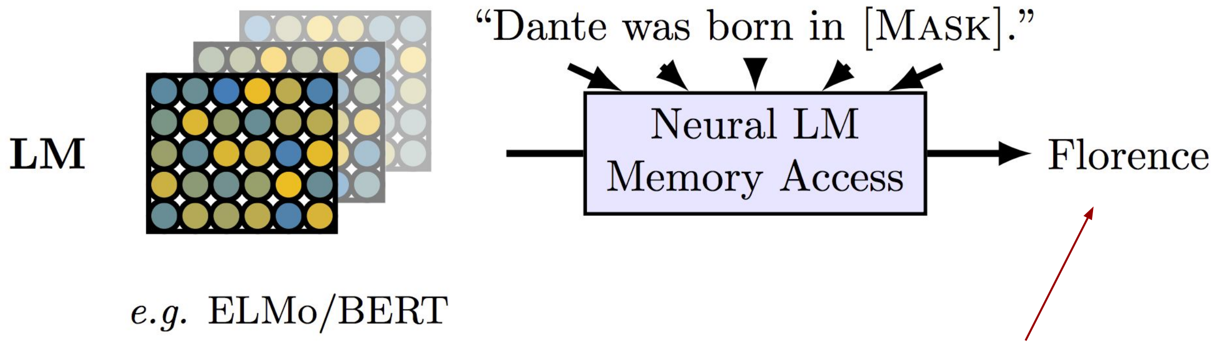
Shuffling the word order of the context further than 50 words away *doesn't* increase loss, so the LM treats words 50-250 effectively as a bag-of-words.

[Khandelwal et al., 2018]

# Neural networks as linguistic test subjects

Question: Do LMs memorize factual relations?

Method:




Check if most likely word under the LM is a correct answer.

**Eval**: % of these relations for which this holds.

[Petroni et al., 2019]

# Neural networks as linguistic test subjects

Question: Do LMs memorize factual relations?

Evaluation:

Baseline: Return word that shows up most with the subject (Dante) and the relation (born in)

BERT-base and BERT-large: memorize a surprising number of facts

| Corpus | Relation | Statistics | | Baselines | | KB | | LM | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | #Facts | #Rel | Freq | DrQA | $RE_n$ | $RE_o$ | Fs | Txl | Eb | E5B | Bb | Bl |
| Google-RE | birth-place | 2937 | 1 | 4.6 | - | 3.5 | 13.8 | 4.4 | 2.7 | 5.5 | 7.5 | 14.9 | **16.1** |
| | birth-date | 1825 | 1 | 1.9 | - | 0.0 | **1.9** | 0.3 | 1.1 | 0.1 | 0.1 | 1.5 | 1.4 |
| | death-place | 765 | 1 | 6.8 | - | 0.1 | 7.2 | 3.0 | 0.9 | 0.3 | 1.3 | 13.1 | **14.0** |
| | Total | 5527 | 3 | 4.4 | - | 1.2 | 7.6 | 2.6 | 1.6 | 2.0 | 3.0 | 9.8 | **10.5** |

[Khandelwal et al., 2018]

# Lecture Outline

Lecture 20: Analysis and Interpretability of Neural NLP

1. Motivation: what are our models doing?
2. Neural networks as linguistic test subjects
3. **Careful ablation studies and architecture modifications**
4. Analysis of inherently interpretable architectures
5. Playing the adversary: breaking NLP models
6. Analyzing representations using supervised methods
7. Aggregating analysis insights across studies

# Viewing model studies as network analysis

Question: What is necessary, or even *good*, about my network design?

Method: Make targeted model changes; observe validation accuracy

Ex: The Transformer interleaves *self-attention* with *feed-forward* layers


[Press et al., 2019]

# Viewing model studies as network analysis

Question: What is necessary, or even *good*, about my network design?
Method: Make targeted model changes; observe validation accuracy

Ex: The Transformer interleaves *self-attention* with *feed-forward* layers

s f s f s f s f s f s f s f s f s f s f s f s f s f s f → 18.40 PPL

But what if we re-ordered them?

s s s s s s s f s f s f s f s f s f s f s f f f f f f f → **17.96 PPL**
*(Better!)*

Bunch of self-attention — Bunch of feed-forward

[Press et al., 2019]

# Lecture Outline

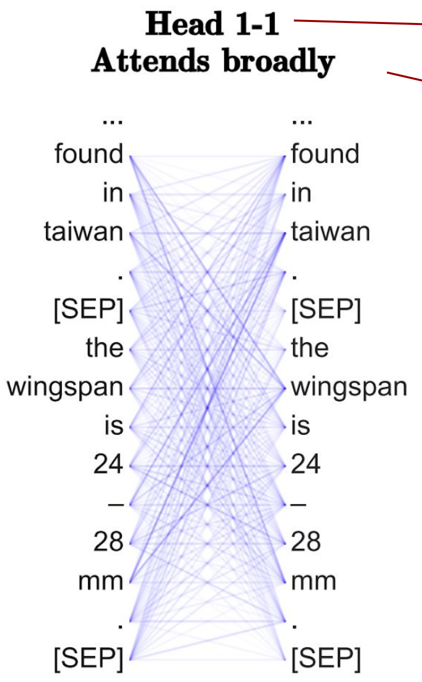
Lecture 20: Analysis and Interpretability of Neural NLP

1. Motivation: what are our models doing?
2. Neural networks as linguistic test subjects
3. Careful ablation studies and architecture modifications
4. **Analysis of inherently interpretable architectures**
5. Playing the adversary: breaking NLP models
6. Analyzing representations using supervised methods
7. Aggregating analysis insights across studies

# Analysis of "interpretable" architectures

Some architectures have components that lend themselves to inspection

Example: Try to characterize each attention head of BERT.




Attention head 1 of layer 1

Not just on this sentence, but on most sentences, this head demonstrates a behavior like this.

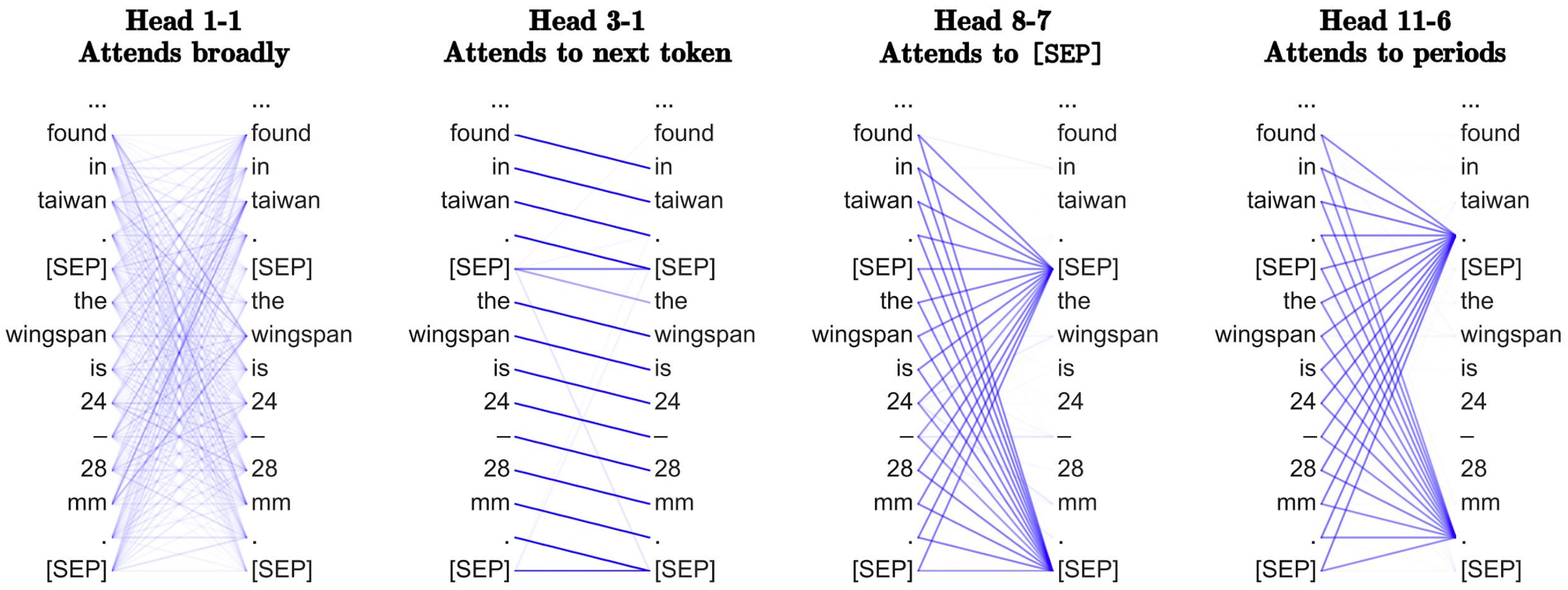


[Clark et al., 2019]

# Analysis of “interpretable” architectures

Some architectures have components that lend themselves to inspection

Example: Try to characterize each attention head of BERT.



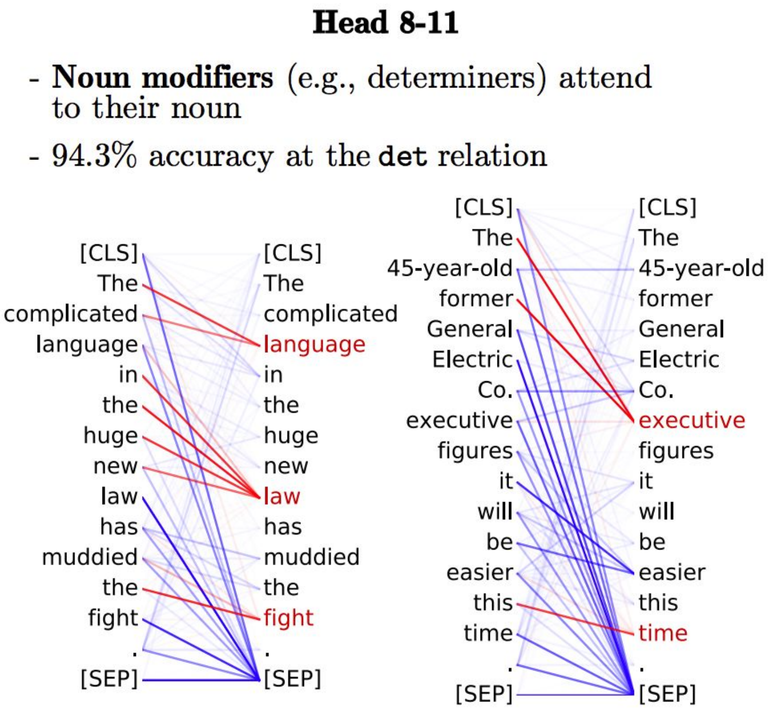



[Clark et al., 2019]

# Analysis of “interpretable” architectures

Some architectures have components that lend themselves to inspection

Example: Try to characterize each attention head of BERT.




Interpretation + Quantitative Analysis

Qualitative Model behavior

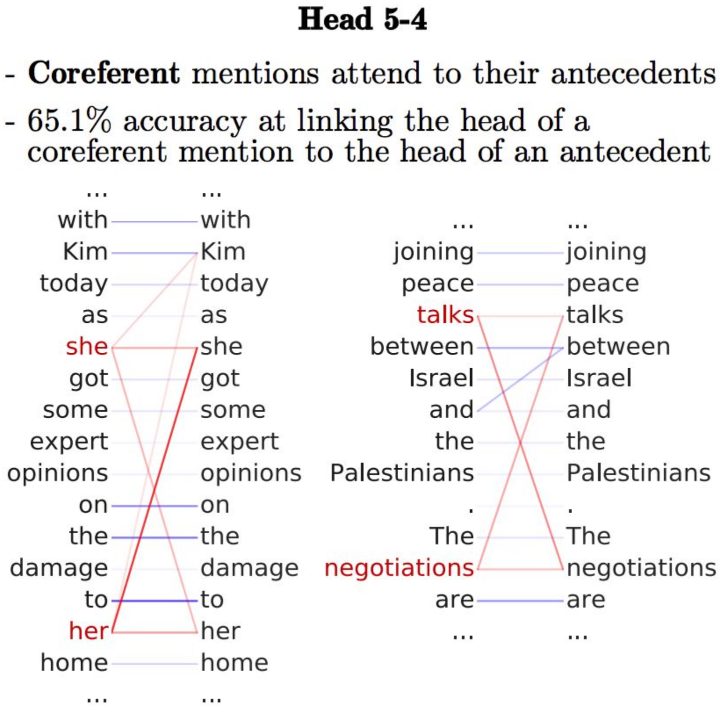


[Clark et al., 2019]

# Analysis of “interpretable” architectures

Some architectures have components that lend themselves to inspection

Example: Try to characterize each attention head of BERT.




Interpretation + Quantitative Analysis

Qualitative Model behavior

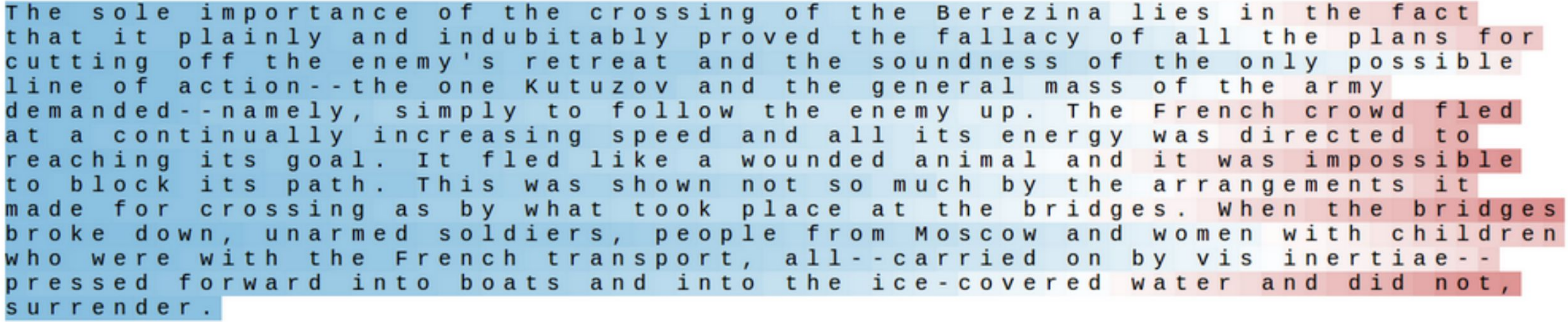


[Clark et al., 2019]

# Understanding representations by inspection

Are individual hidden units in recurrent neural networks interpretable?

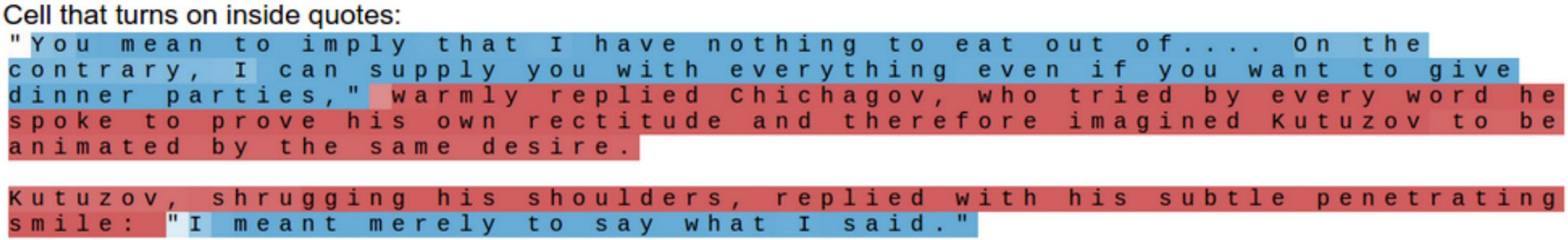
Cell sensitive to position in line:




[Karpathy et al., 2016]

# Understanding representations by inspection

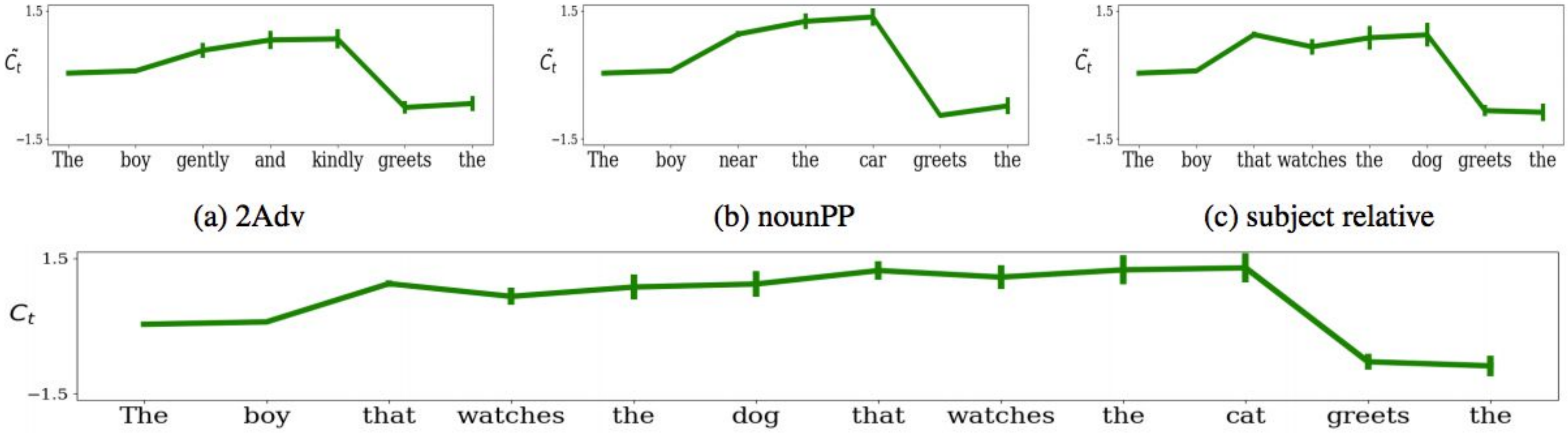
Are individual hidden units in recurrent neural networks interpretable?




[Karpathy et al., 2016]

# Understanding representations by inspection

Are individual hidden units in recurrent neural networks interpretable?




Interpretation: this LSTM cell unit fires approximately between a subject and its verb

[Lakretz et al., 2019]

# Lecture Outline

Lecture 20: Analysis and Interpretability of Neural NLP

1. Motivation: what are our models doing?
2. Neural networks as linguistic test subjects
3. Careful ablation studies and architecture modifications
4. Analysis of inherently interpretable architectures
5. **Playing the adversary: breaking NLP models**
6. Analyzing representations using supervised methods
7. Aggregating analysis insights across studies

# Understanding models by breaking them

Question: Are our models robust to innocuous changes in their input?

By robust, in this case we mean their outputs do not change.

**Article:** Super Bowl 50

**Paragraph:** "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager.*

**Question:** "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"

**Original Prediction:** John Elway

The performance of this QA model on this input looks good!

[Jia et al., 2019]

# Understanding models by breaking them

Question: Are our models robust to innocuous changes in their input?

By robust, in this case we mean their outputs do not change.

**Article:** Super Bowl 50
**Paragraph:** "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
**Question:** "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
**Original Prediction:** John Elway

The performance of this QA model on this input looks good!

This sentence is irrelevant; adding it does not change the answer.



[Jia et al., 2019]

# Understanding models by breaking them

Question: Are our models robust to innocuous changes in their input?

By robust, in this case we mean their outputs do not change.

**Article:** Super Bowl 50
**Paragraph:** "*Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*"
**Question:** "*What is the name of the quarterback who was 38 in Super Bowl XXXIII?*"
**Original Prediction:** John Elway
**Prediction under adversary:** Jeff Dean

The performance of this QA model on this input looks good!

This sentence is irrelevant; adding it does not change the answer.

But it changes the model's prediction :(

Interpretation: model is not really working



[Jia et al., 2019]

# Understanding models by breaking them

Question: Are our models robust to innocuous changes in their input?

In the United States especially, several high-profile cases such as Debra LaFave, Pamela Rogers, and Mary Kay Letourneau have caused increased scrutiny on teacher misconduct.

(a) Input Paragraph

**Q:** What has been the result of this publicity?
**A:** increased scrutiny on teacher misconduct

(b) Original Question and Answer

The performance of this QA model on this input looks good!



[Ribeiro et al., 2018]

# Understanding models by breaking them

## Question: Are our models robust to innocuous changes in their input?

In the United States especially, several high-profile cases such as Debra LaFave, Pamela Rogers, and Mary Kay Letourneau have caused increased scrutiny on teacher misconduct.

(a) Input Paragraph

**Q:** What has been the result of this publicity?
**A:** increased scrutiny on teacher misconduct

(b) Original Question and Answer

**Q:** What haL been the result of this publicity?
**A:** teacher misconduct

(c) Adversarial Q & A (Ebrahimi et al., 2018)

The performance of this QA model on this input looks good!

This typo is annoying, but a reasonable language learner would be robust to it.



[Ribeiro et al., 2018]

# Understanding models by breaking them

Question: Are our models robust to innocuous changes in their input?

In the United States especially, several high-profile cases such as Debra LaFave, Pamela Rogers, and Mary Kay Letourneau have caused increased scrutiny on teacher misconduct.

(a) Input Paragraph

The performance of this QA model on this input looks good!

**Q:** What has been the result of this publicity?
**A:** increased scrutiny on teacher misconduct

(b) Original Question and Answer

**Q:** What haL been the result of this publicity?
**A:** teacher misconduct

(c) Adversarial Q & A (Ebrahimi et al., 2018)

This typo is annoying, but a reasonable language learner would be robust to it.

**Q:** What's been the result of this publicity?
**A:** teacher misconduct

(d) **Semantically Equivalent Adversary**

Changing *what has* to *what's* should never change the answer!



[Ribeiro et al., 2018]

# Understanding models by breaking them

Question: Are our models robust to typos or noise in their input?



[Belinkov and Bisk, 2018]

# Understanding models by breaking them

Question: Are ~~our models~~ Humans robust to typos or noise in their input?



[Belinkov and Bisk, 2018]

# Understanding models by breaking them

Question: Are ~~our models~~ *Humans* robust to typos or noise in their input?

"Aoccdrnig to a rscheearch at Cmabrigde Uinervtisy, it deosn't mttaer in waht oredr the ltteers in a wrod are, the olny iprmoetnt tihng is taht the frist and lsat ltteer be at the rghit pclae."

Just 1 data point/meme, but interpretation: humans are!



[Belinkov and Bisk, 2018]

# Understanding models by breaking them

Question: Are our models robust to typos or noise in their input?

BLEU on clean text

| | | Vanilla |
|---|---|---|
| French | charCNN | 42.54 |
| German | charCNN | 34.79 |
| | char2char | 29.97 |
| | Nematus | 34.22 |
| Czech | charCNN | 25.99 |
| | char2char | 25.71 |
| | Nematus | 29.65 |



[Belinkov and Bisk, 2018]

# Understanding models by breaking them

Question: Are our models robust to typos or noise in their input?

BLEU on clean text

BLEU on data with noise like we just saw

BLEU on data with natural noise (real misspellings, +)

| | | | Synthetic | | | | |
|---|---|---|---|---|---|---|---|
| | | Vanilla | Swap | Mid | Rand | Key | Nat |
| French | charCNN | 42.54 | 10.52 | 9.71 | 1.71 | 8.26 | 17.42 |
| German | charCNN | 34.79 | 9.25 | 8.37 | 1.02 | 6.40 | 14.02 |
| | char2char | 29.97 | 5.68 | 5.46 | 0.28 | 2.96 | 12.68 |
| | Nematus | 34.22 | 3.39 | 5.16 | 0.29 | 0.61 | 10.68 |
| Czech | charCNN | 25.99 | 6.56 | 6.67 | 1.50 | 7.13 | 10.20 |
| | char2char | 25.71 | 3.90 | 4.24 | 0.25 | 2.88 | 11.42 |
| | Nematus | 29.65 | 2.94 | 4.09 | 0.66 | 1.41 | 11.88 |



[Belinkov and Bisk, 2018]

# Lecture Outline

Lecture 20: Analysis and Interpretability of Neural NLP

1. Motivation: what are our models doing?
2. Neural networks as linguistic test subjects
3. Careful ablation studies and architecture modifications
4. Analysis of inherently interpretable architectures
5. Playing the adversary: breaking NLP models
6. **Analyzing representations using supervised methods**
7. Aggregating analysis insights across studies

# Understanding representations by probing

Hypothesis:

Neural models, especially large ones like BERT, perform well without any explicit linguistic supervision in part because they learn similar notions themselves.

Question:

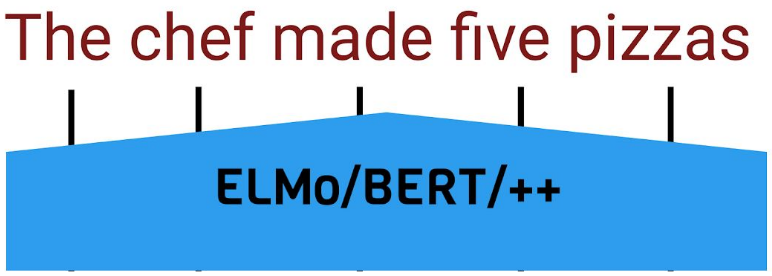
Do neural networks' internal representations encode linguistic notions of structure, like *parts-of-speech*, *dependency trees*, *named entities?*



[Lakretz et al., 2019]

# Probing: supervised analysis of representations

***Does my network make task (e.g., part-of-speech) labels accessible?***



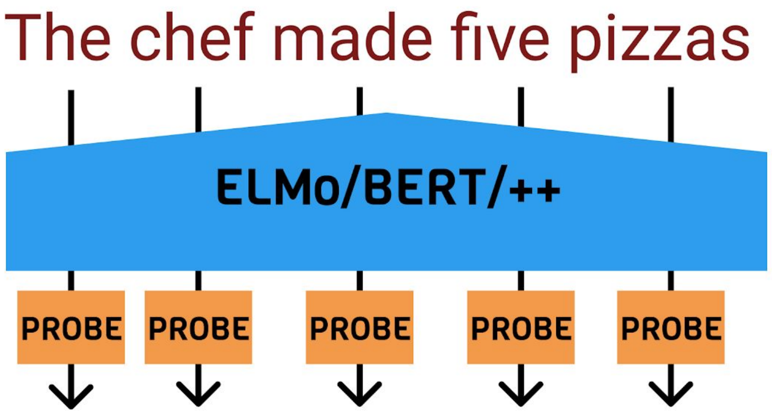



[Shi et al., 2016, Peters et al., 2018, Tenney et al., 2019, Liu et al., 2019, Hewitt and Manning 2019, Kim et al., 2019...] [Hewitt and Liang, 2019]

# Probing: supervised analysis of representations

***Does my network make task (e.g., part-of-speech) labels accessible?***

**Choose** a function family
to decode the task**.** (e.g., linear)





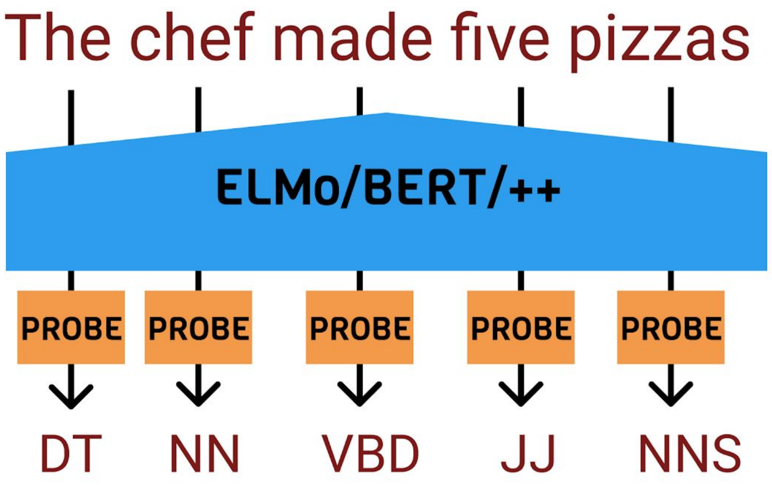

[Shi et al., 2016, Peters et al., 2018, Tenney et al., 2019, Liu et al., 2019, Hewitt and Manning 2019, Kim et al., 2019...] [Hewitt and Liang, 2019]

# Probing: supervised analysis of representations

***Does my network make task (e.g., part-of-speech) labels accessible?***

**Choose** a function family
to decode the task**.** (e.g., linear)

**Train** a function
representations --> task






[Shi et al., 2016, Peters et al., 2018, Tenney et al., 2019, Liu et al., 2019, Hewitt and Manning 2019, Kim et al., 2019...] [Hewitt and Liang, 2019]

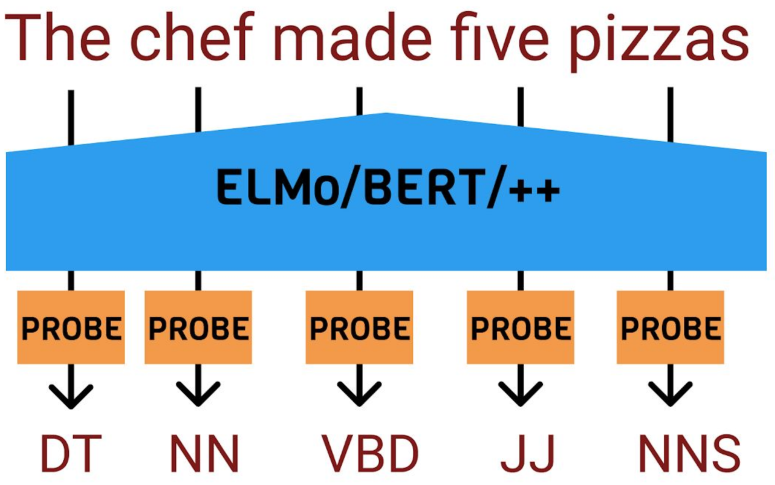
# Probing: supervised analysis of representations

***Does my network make task (e.g., part-of-speech) labels accessible?***

**Choose** a function family
to decode the task**.** (e.g., linear)

**Train** a function
representations --> task

**Interpret** accuracy
on held-out data




(Don’t fine-tune the model while doing this!)

[Shi et al., 2016, Peters et al., 2018, Tenney et al., 2019, Liu et al., 2019, Hewitt and Manning 2019, Kim et al., 2019...] [Hewitt and Liang, 2019]

# Understanding representations by probing

| Pretrained Representation | Avg. | CCG | POS PTB | POS EWT | Chunk | NER | ST | GED | Supersense ID PS-Role | Supersense ID PS-Fxn | EF |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BERT (base, cased) best layer | 84.09 | 93.67 | 96.95 | 95.21 | 92.64 | 82.71 | 93.72 | 43.30 | **79.61** | 87.94 | 75.11 |
| BERT (large, cased) best layer | **85.07** | **94.28** | 96.73 | **95.80** | **93.64** | **84.44** | 93.83 | **46.46** | 79.17 | **90.13** | **76.25** |
| GloVe (840B.300d) | 59.94 | 71.58 | 90.49 | 83.93 | 62.28 | 53.22 | 80.92 | 14.94 | 40.79 | 51.54 | 49.70 |
| Previous state of the art (without pretraining) | 83.44 | 94.7 | 97.96 | 95.82 | 95.77 | 91.38 | 95.15 | 39.83 | 66.89 | 78.29 | 77.10 |

Interpretation 1: BERT’s representations, when used as features for a linear classifier, lead to high accuracy on linguistic tasks; this is evidence that BERT makes these properties linearly accessible.

Interpretation 2: BERT-large seems to perform better than BERT-base, indicating that it may learn better representations of linguistic properties.



[Liu et al., 2019; table cropped]

# Understanding representations by probing

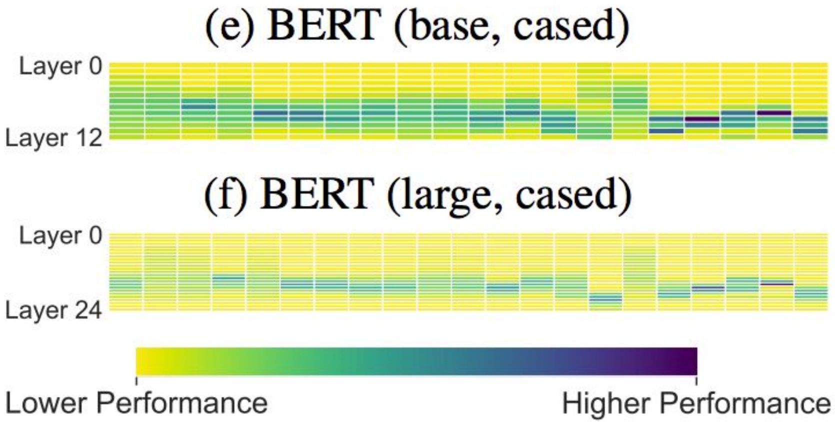



Figure 3: A visualization of layerwise patterns in task performance. Each column represents a probing task, and each row represents a contextualizer layer.

Interpretation: BERT makes linguistic properties most accessible in middle layers

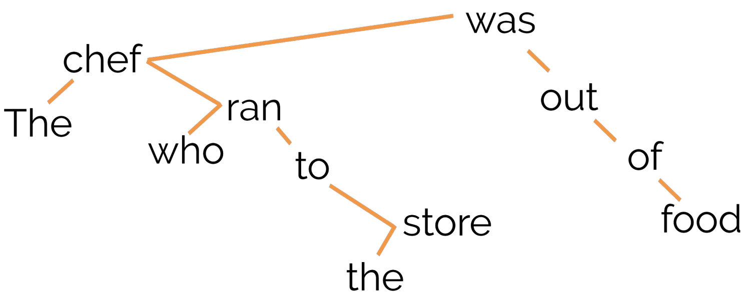


[Liu et al., 2019; figure cropped]

# Understanding representations by probing

Question: Can we ask questions about *structure* in neural representations?


*A neural (vector) representation*




*A structured linguistic representation*

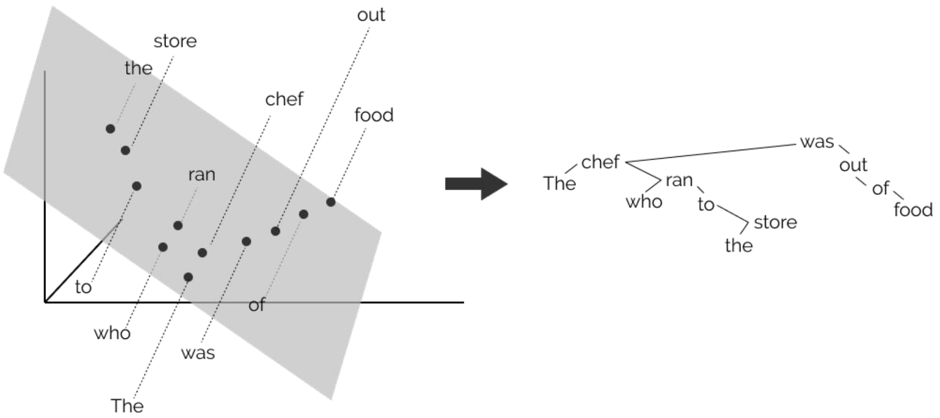


[Hewitt and Manning, 2019]

# Understanding representations by probing

Let's walk through a whole analysis paper, step-by-step

**A Structural Probe for Finding Syntax in Word Representations**






[Hewitt and Manning, 2019]

## *This work's questions!*

Do *ELMo and BERT* encode English dependency trees in their *contextual* representations?

## ***This work's questions!***

Do *ELMo and BERT* encode English dependency trees in their *contextual* representations?

How do we ask whether vector representations encode trees?

***This work's questions!***

**tl;dr answers**

Do *ELMo and BERT* encode English dependency trees in their *contextual* representations?

How do we ask whether vector representations encode trees?

By **structural probes**: look at the geometry! A hypothesis for syntax in word representations.

## ***This work's questions!***

## **tl;dr answers**

Do *ELMo and BERT* encode English dependency trees in their *contextual* representations?

We provide evidence for *yes, approximately!*

How do we ask whether vector representations encode trees?

By **structural probes**: look at the geometry! A hypothesis for syntax in word representations.

*Outline*

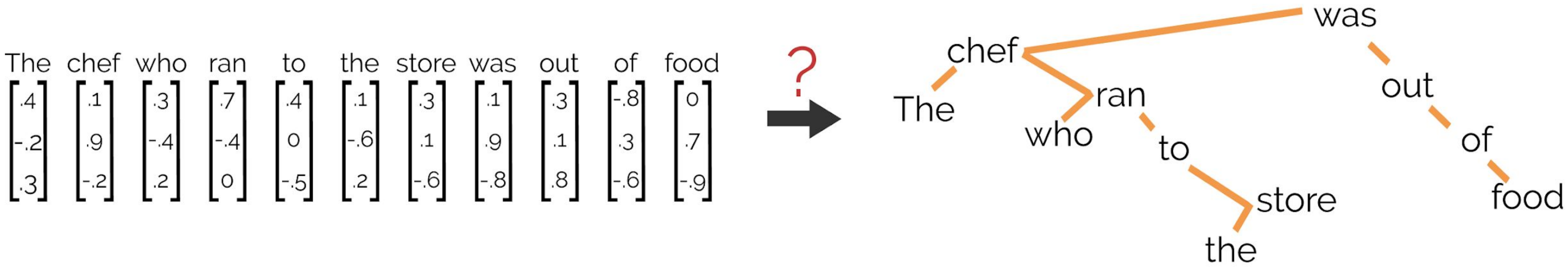
1. *connecting* **vector spaces** and **trees**

2. The **structural probe** method

3. Results and pictures and fun

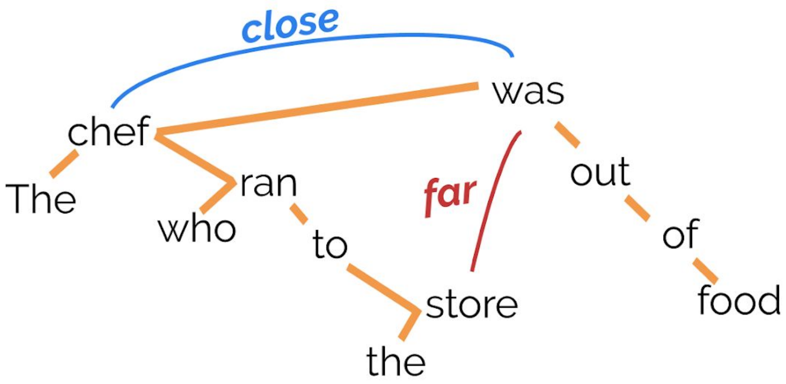
# Are vector spaces and trees reconcilable?

Are **vector space representations** in NLP reconcilable with the **discrete (syntactic) tree** structures hypothesized in 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.





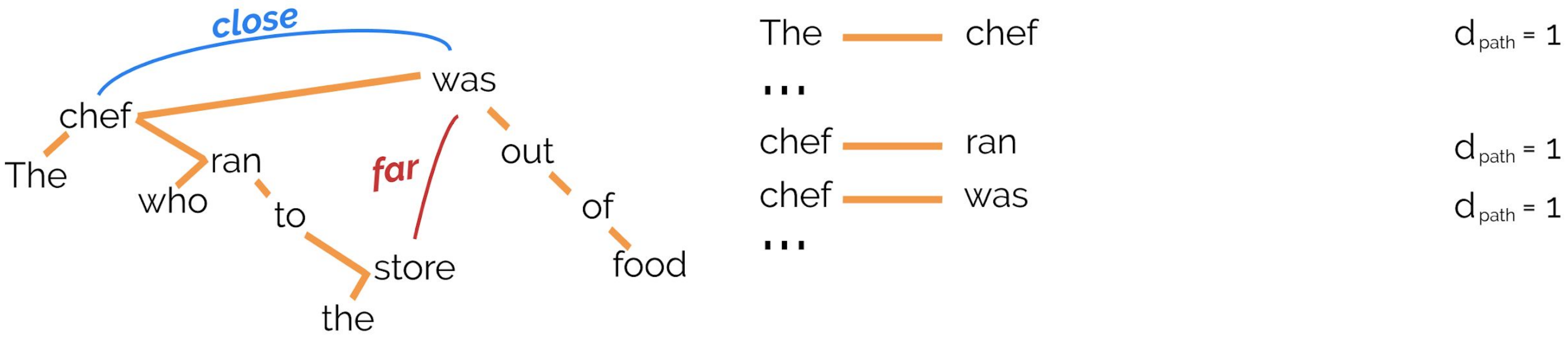
$d_{path} = 1$



[For more fun, see Deza and Laurent. *A Geometry of Cuts and Metrics.* Springer. 2009]

# 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.

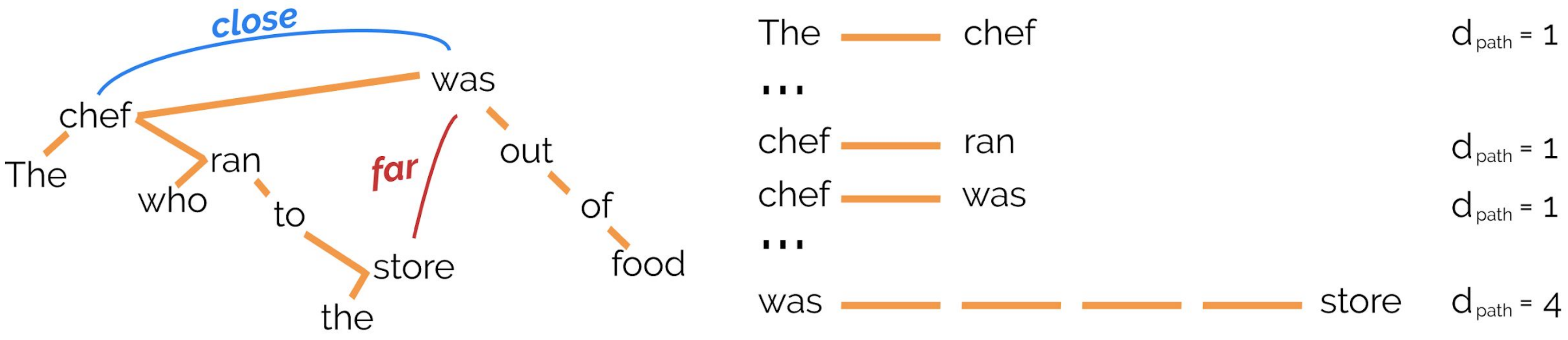





[For more fun, see Deza and Laurent. *A Geometry of Cuts and Metrics.* Springer. 2009]

# Distance metrics unify trees and vectors

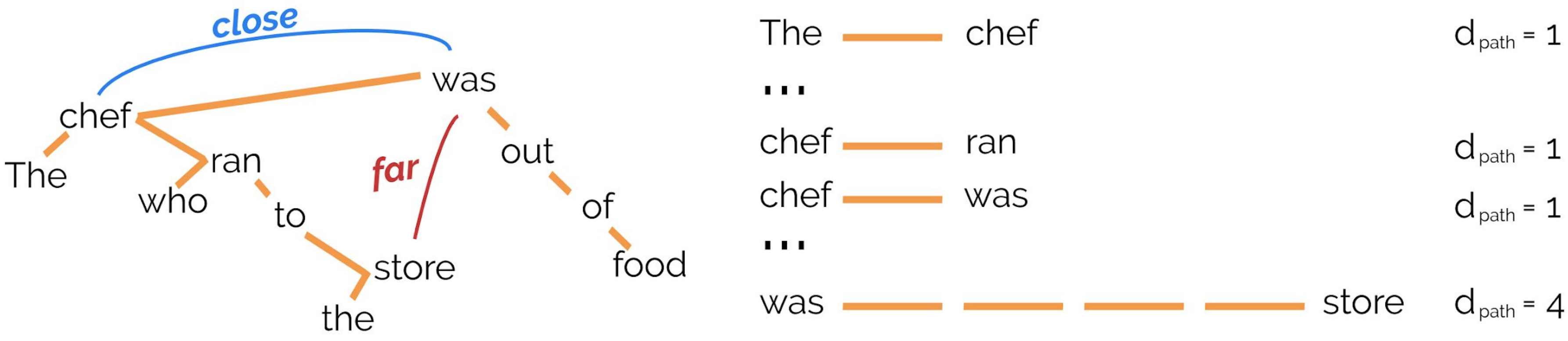
An **undirected tree** defines a **distance metric** on pairs of words, the path metric: the number of edges in the path between the words.




[For more fun, see Deza and Laurent. *A Geometry of Cuts and Metrics.* Springer. 2009]

# 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.



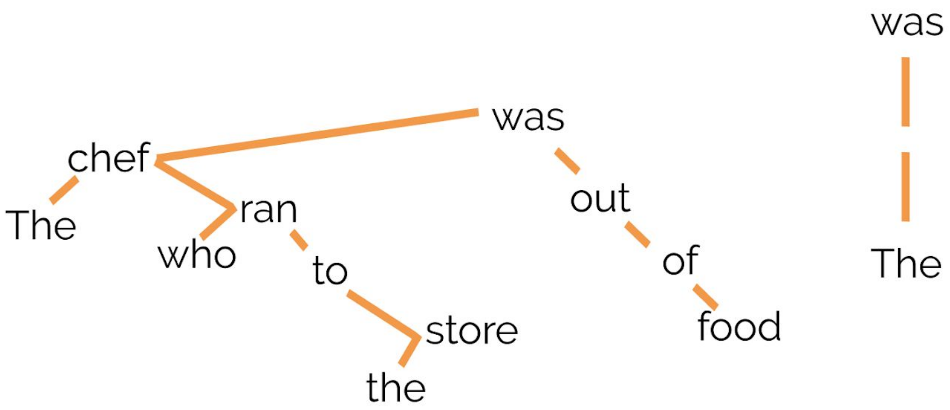

*The edges of the tree can be recovered by looking at all distance=1 pairs.*



[For more fun, see Deza and Laurent. *A Geometry of Cuts and Metrics.* Springer. 2009]

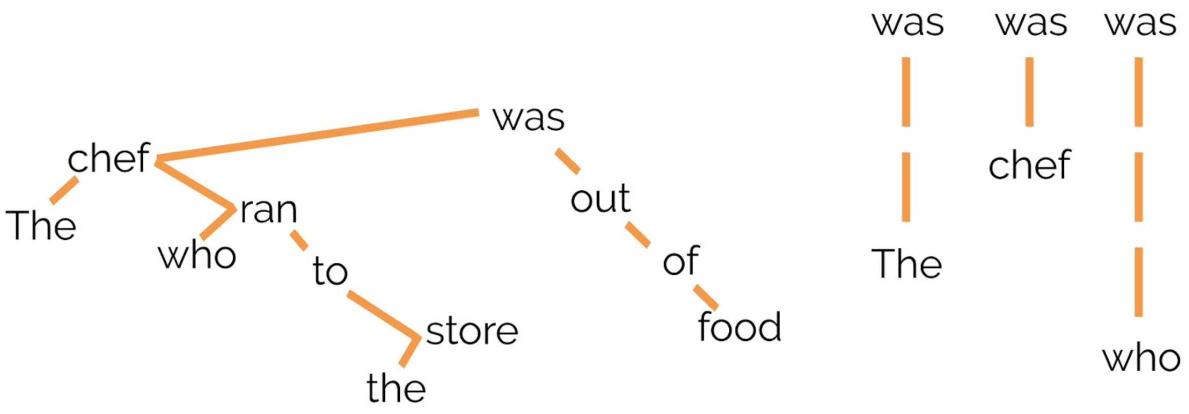
# Norms unify edge directions and vectors

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



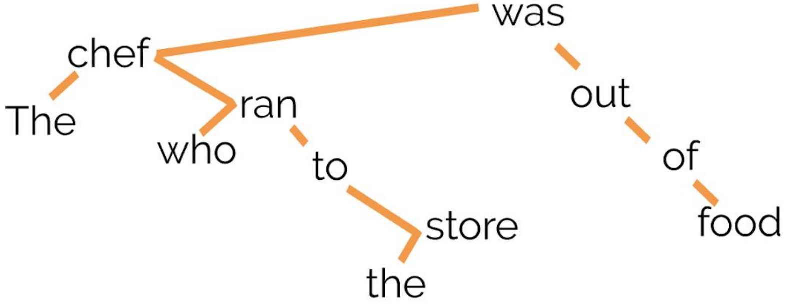
# Norms unify edge directions and vectors

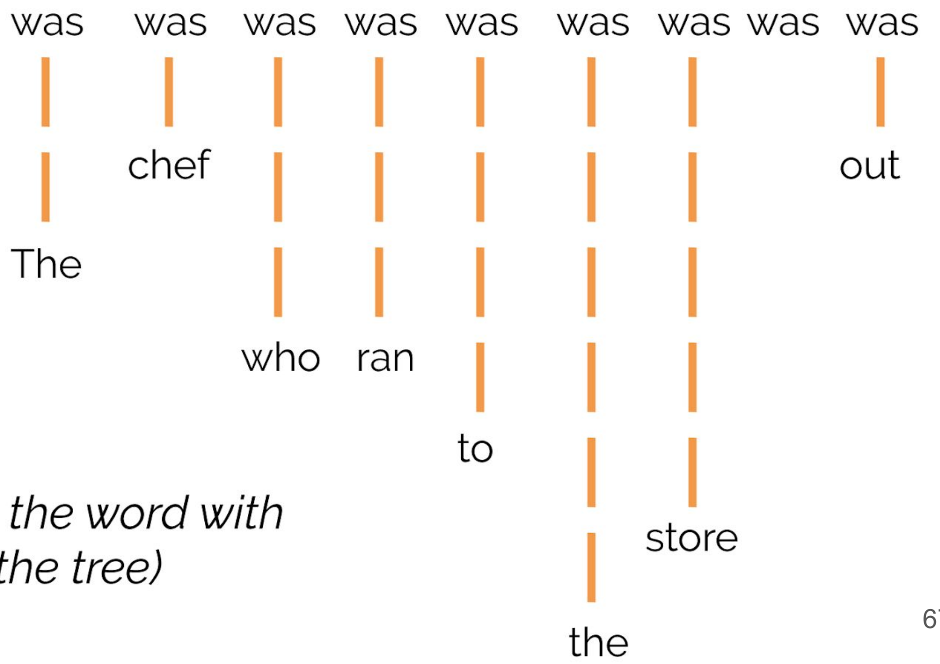
A **rooted tree** defines a **norm** on the words, the parse depth:
the number of edges from each word to ROOT.



# Norms unify edge directions and vectors

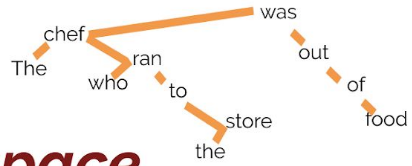
A **rooted tree** defines a **norm** on the words, the parse depth:
the number of edges from each word to ROOT.




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

# summary

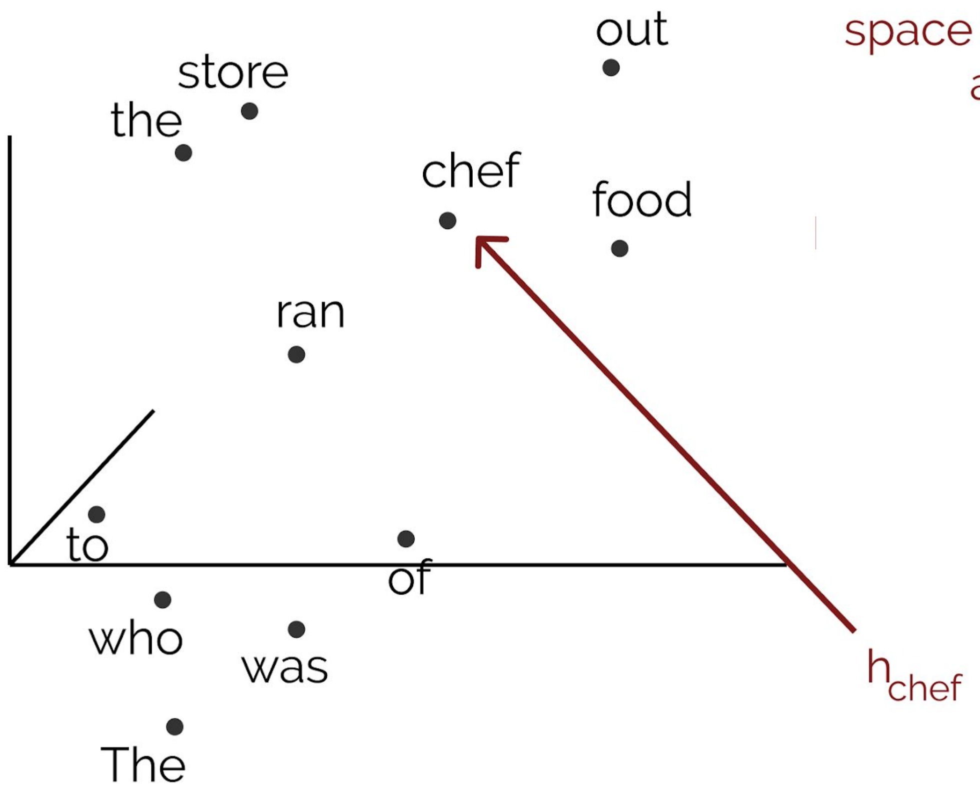



***distance*** ***unifies undirected trees and vector space***


***norm unifies edge directions and vector space***

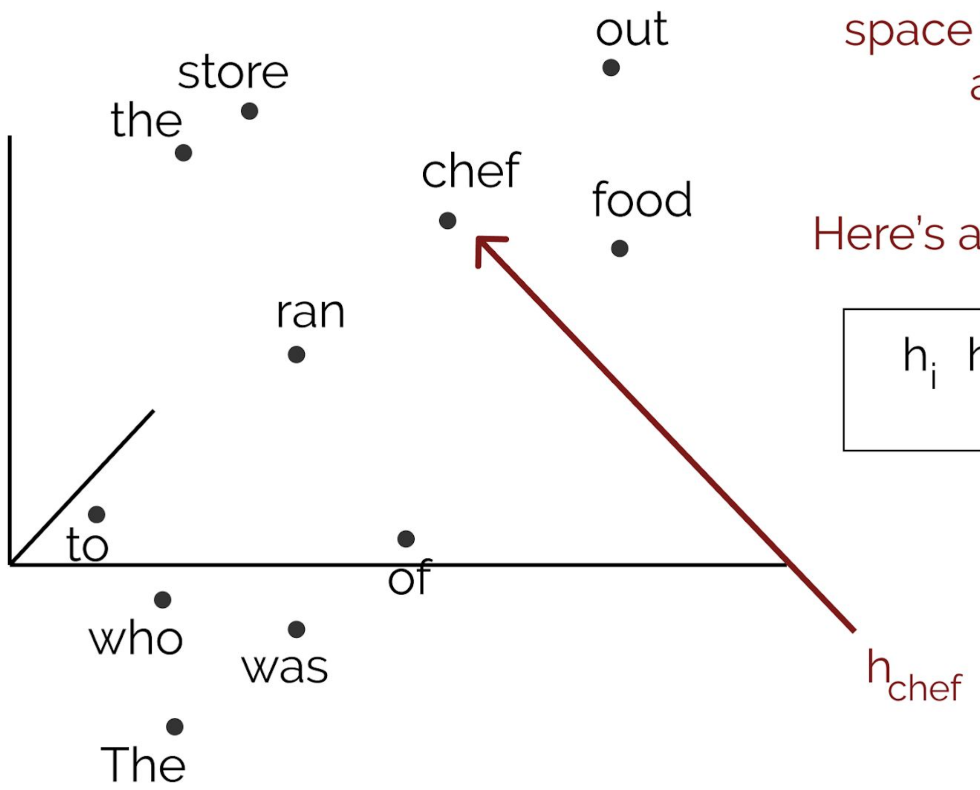
# The *structural probe* method

# Finding trees in vector spaces




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

# Finding trees in vector spaces

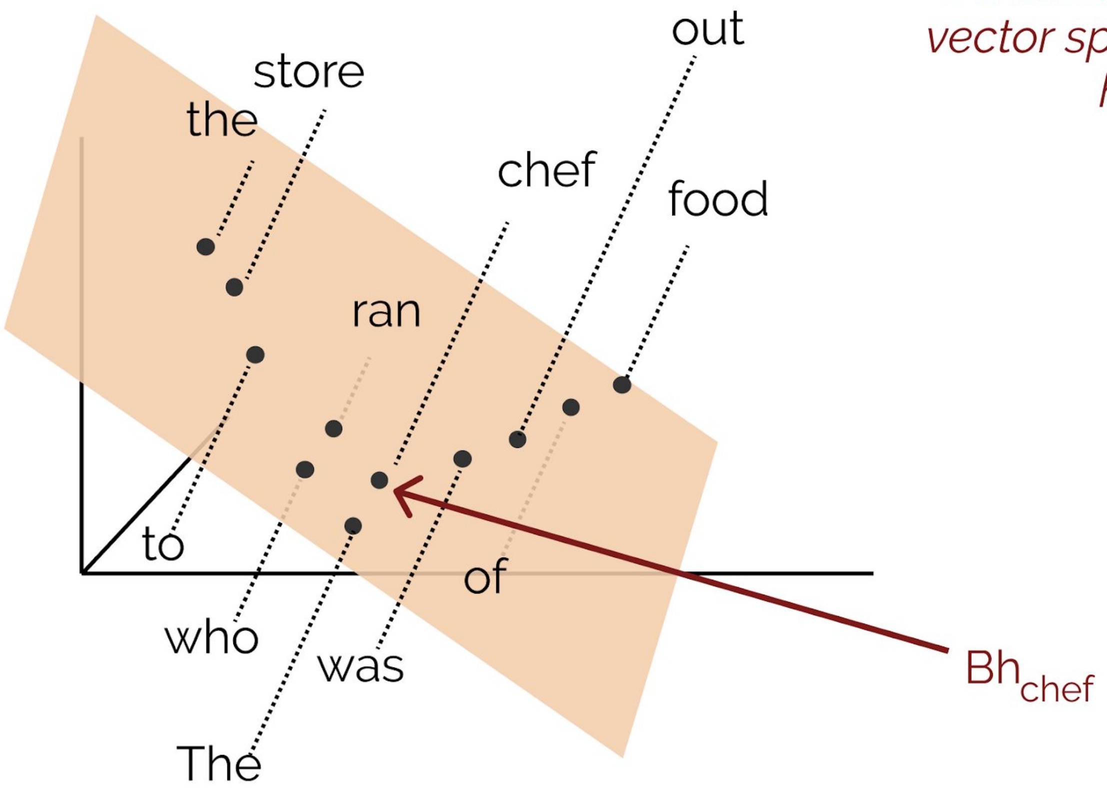



We can look for trees in the vector space by looking for their **distances** and **norms** in the space.

Here's a sentence embedded by a NN!

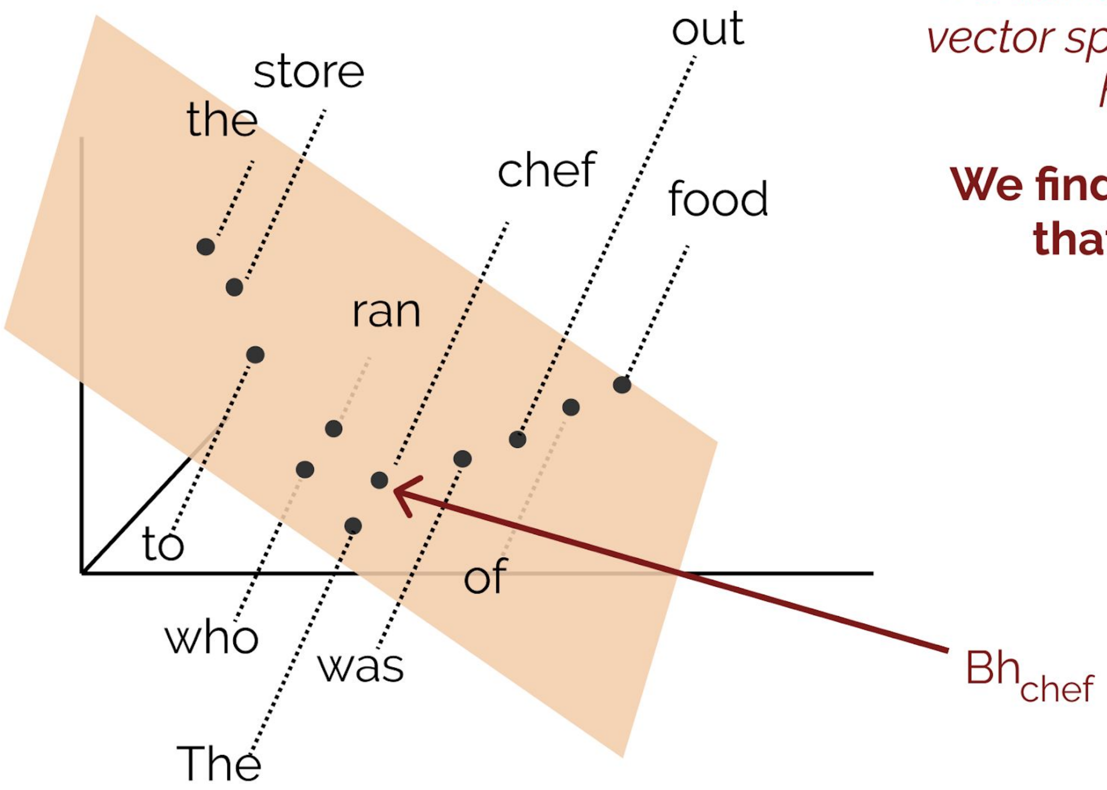
$h_i$ $h_j$ : vector representation of words *i* and *j*.

# Finding trees in vector spaces




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

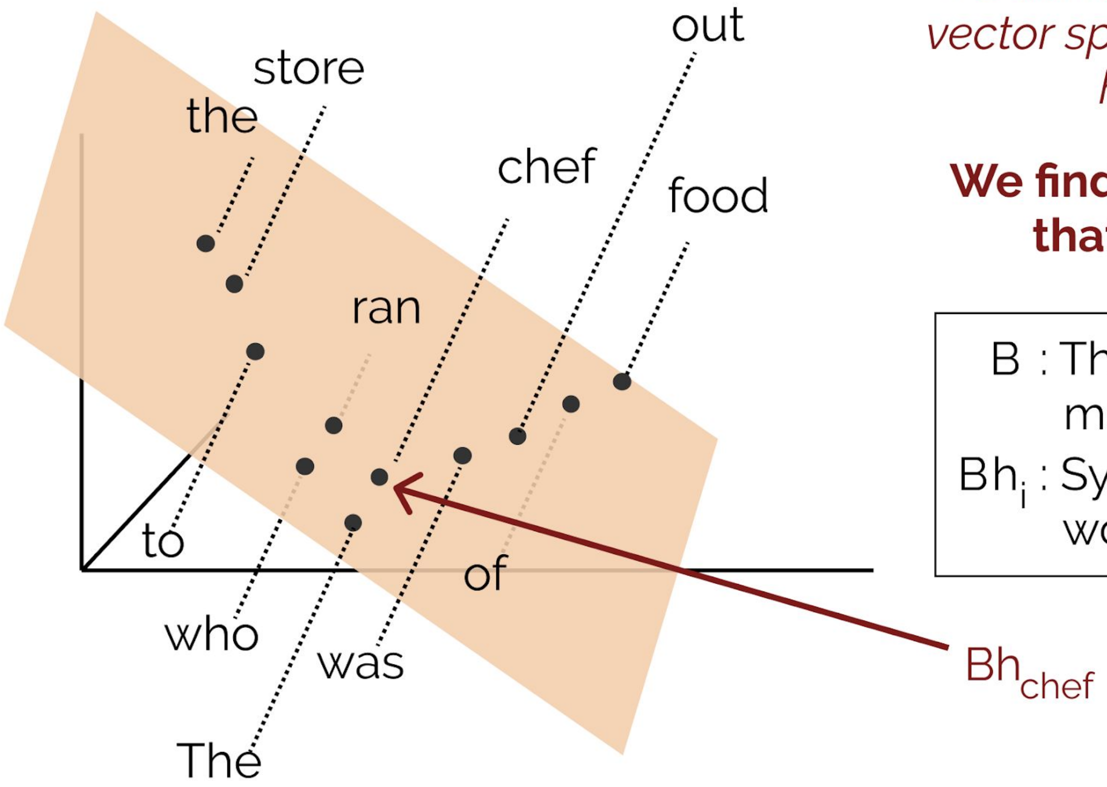
# Finding trees in vector spaces




*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.**

# Finding trees in vector spaces



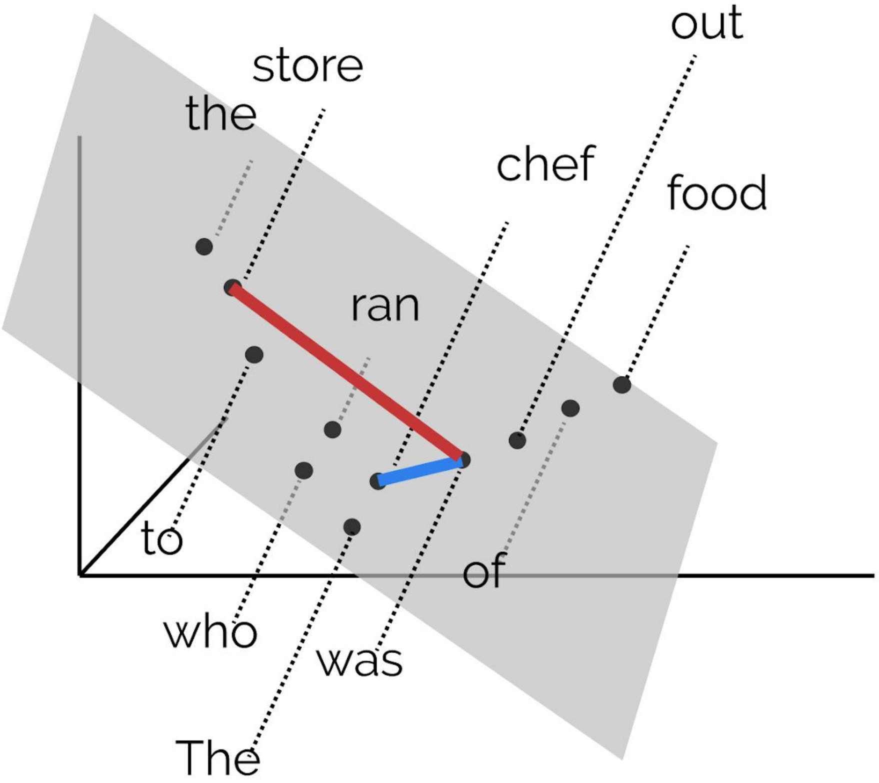

*We don't expect all dimensions of the vector space to encode syntax -- NNs have a lot to encode!*

**We find the linear transformation that encodes syntax best.**

B : The syntax transformation matrix

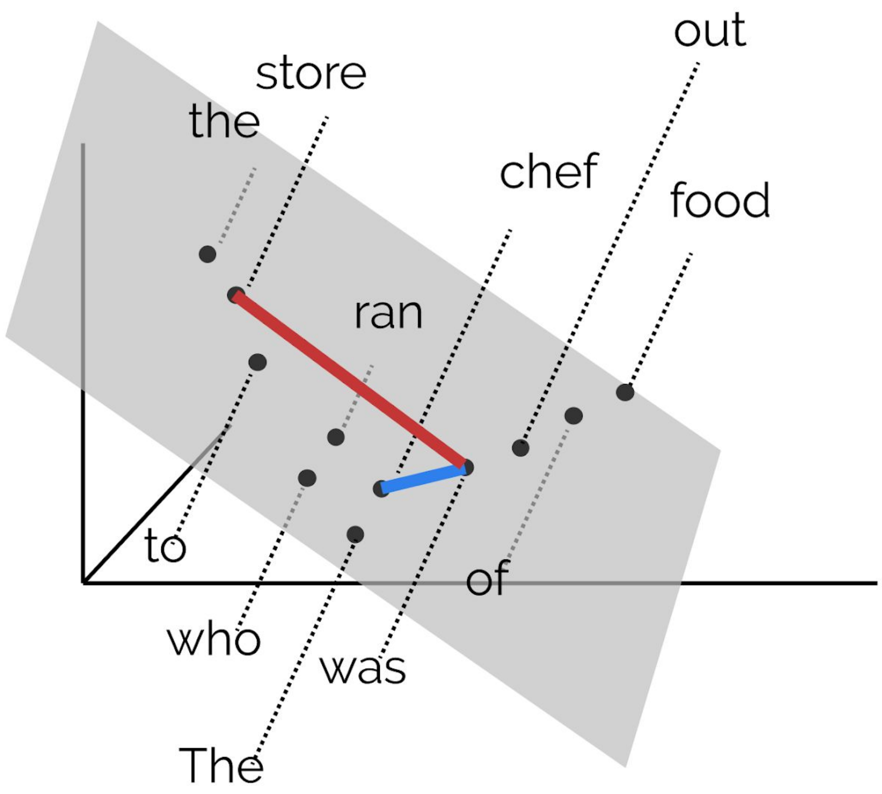
$Bh_i$ : Syntax-transformed vector word representation

# Finding trees in vector spaces




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

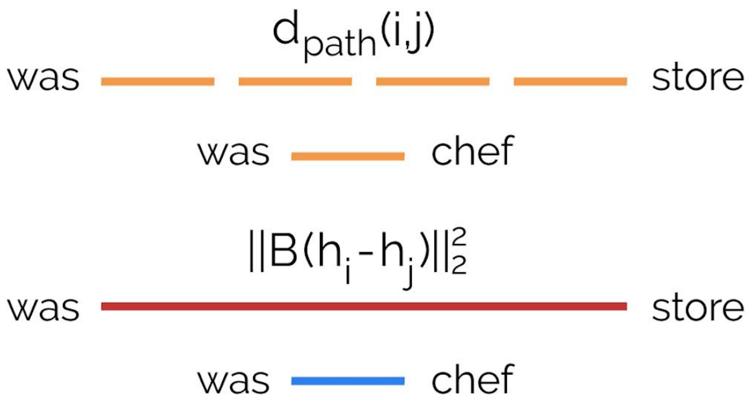
# Finding trees in vector spaces



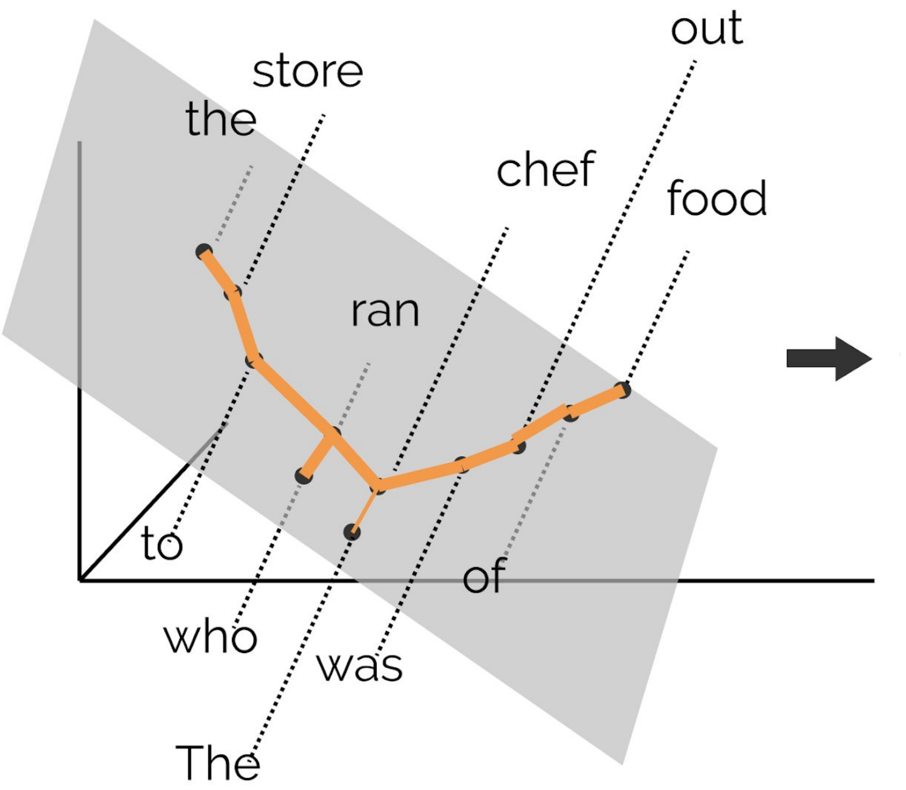

***In the transformed space, (squared) L2 distance approximates tree distance.***

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

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



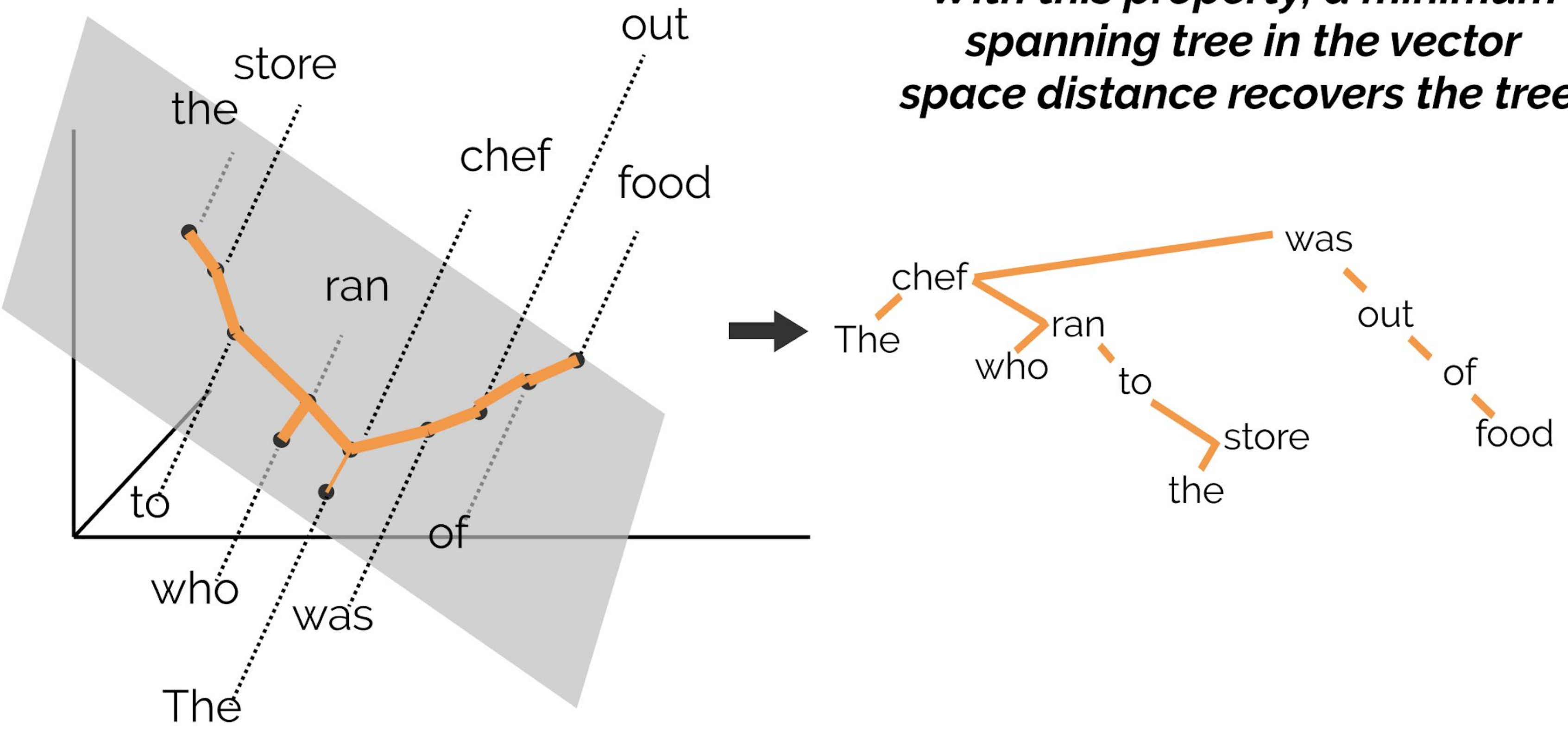
# Finding trees in vector spaces




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

# Finding trees in vector spaces

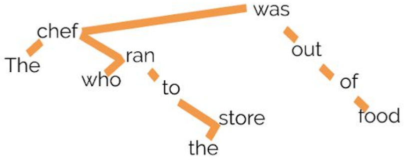
***With this property, a minimum spanning tree in the vector space distance recovers the tree.***



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

$$\arg\min_{B}$$

Find a single transformation *B*

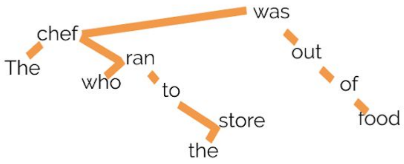


# Does BERT encode undirected parse trees -> does there exist a *distance* transformation?

$$\arg\min_{B} \sum_{\ell \in \mathrm{PTB}}$$

Find a single transformation *B*

such that over all sentences in PTB training



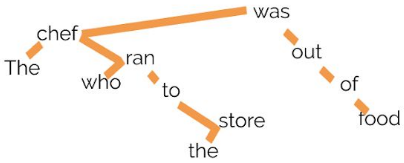
# Does BERT encode undirected parse trees -> does there exist a *distance* transformation?

$$\arg\min_{B} \sum_{\ell \in \mathrm{PTB}} \sum_{i,j}$$

Find a single transformation *B*

such that over all sentences in PTB training

Over all word pairs in each sentence



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

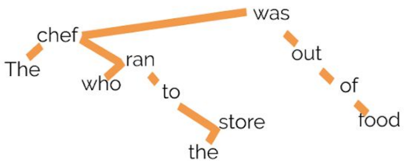
$$\arg\min_B \sum_{\ell \in \text{PTB}} \sum_{i,j} \left| d_{\text{path}}^{\ell}(i,j) - \|B(h_i^{\ell} - h_j^{\ell})\|_2^2 \right|$$

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*



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

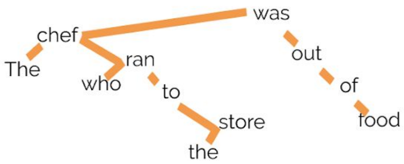
$$\arg\min_{B} \sum_{\ell \in \text{PTB}} \frac{1}{|s^\ell|^2} \sum_{i,j} \left| d^\ell_{\text{path}}(i,j) - \|B(h_i^\ell - h_j^\ell)\|_2^2 \right|$$

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*

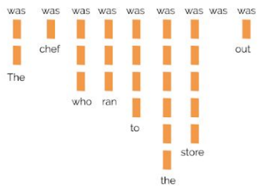


# Does BERT encode edge directions -> does there exist a *depth* transformation?

$$\arg\min_{B} \sum_{\ell \in \mathrm{PTB}} \frac{1}{|s^{\ell}|}$$

Find a single transformation *B*

such that over all sentences in PTB training



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

$$\arg\min_{B} \sum_{\ell \in \mathrm{PTB}} \frac{1}{|s^\ell|} \sum_{i}$$

Find a single transformation *B*

such that over all sentences in PTB training

Over all words in each sentence

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

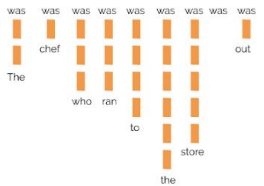
$$\arg\min_{B} \sum_{\ell \in \mathrm{PTB}} \frac{1}{|s^\ell|} \sum_i \left| \mathrm{depth}^\ell(i) - \|Bh_i^\ell\|_2^2 \right|$$

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*



# *experiments & results*

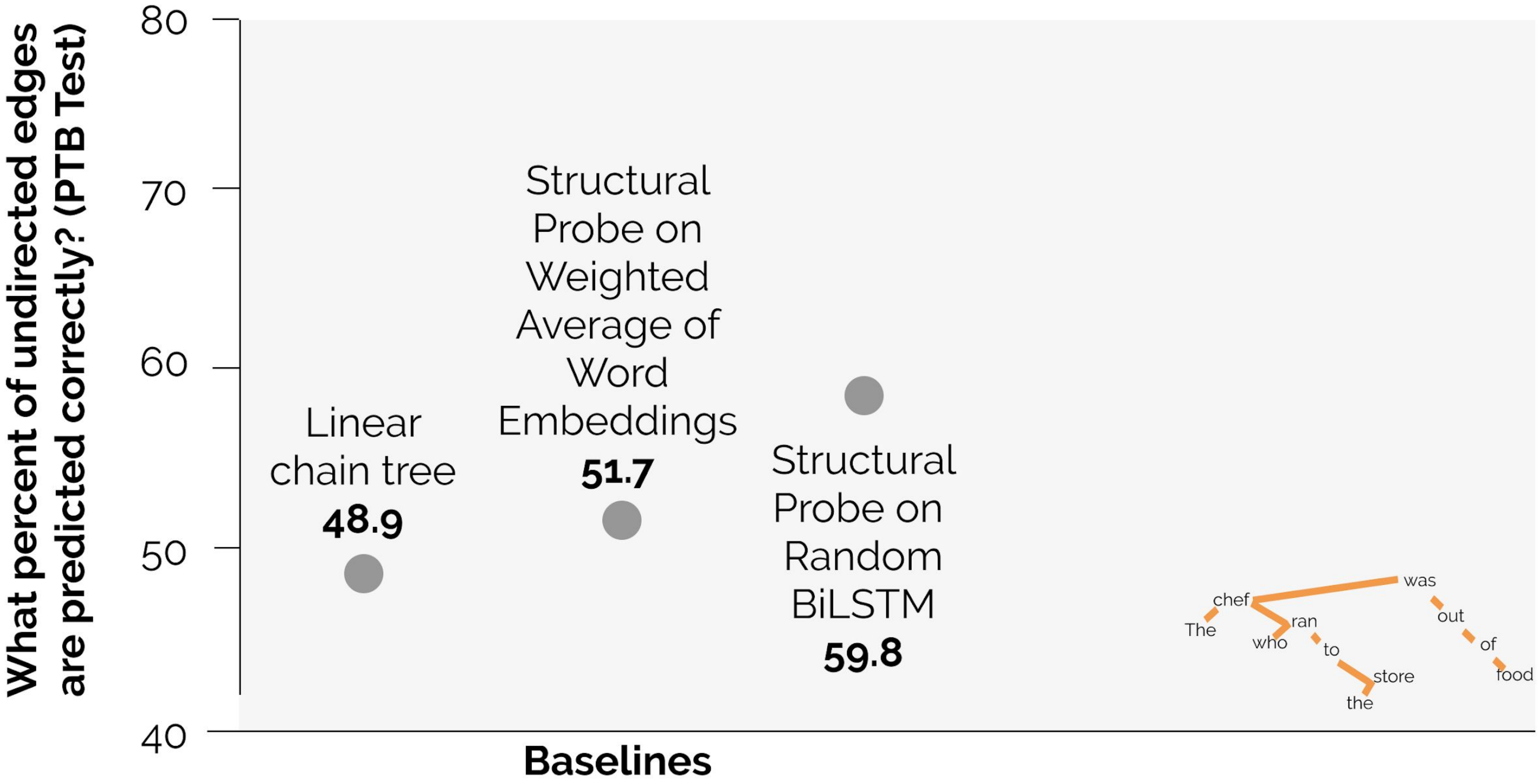
**Evaluating ELMo, BERT, and baselines**
**Training structural probes on PTB train, evaluating on test.**

**Evaluate by comparing structural probe minimum spanning trees to human-annotated parse trees.**

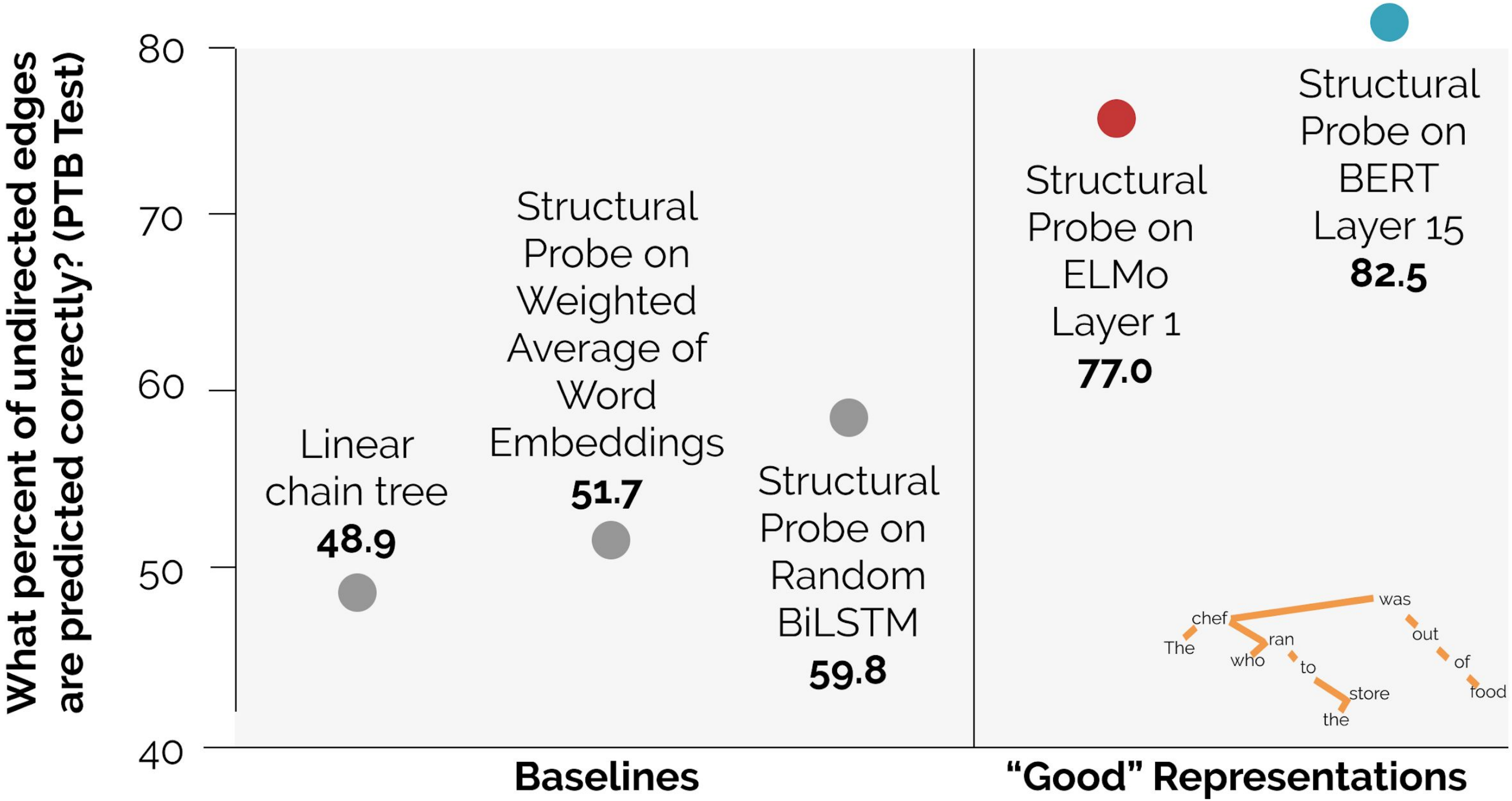
Metrics:

- **Spearman correlation:** true vs predicted distances/depths
- **UUAS**: Unlabeled Undirected Attachment Score,
  minimum spanning tree vs. gold tree

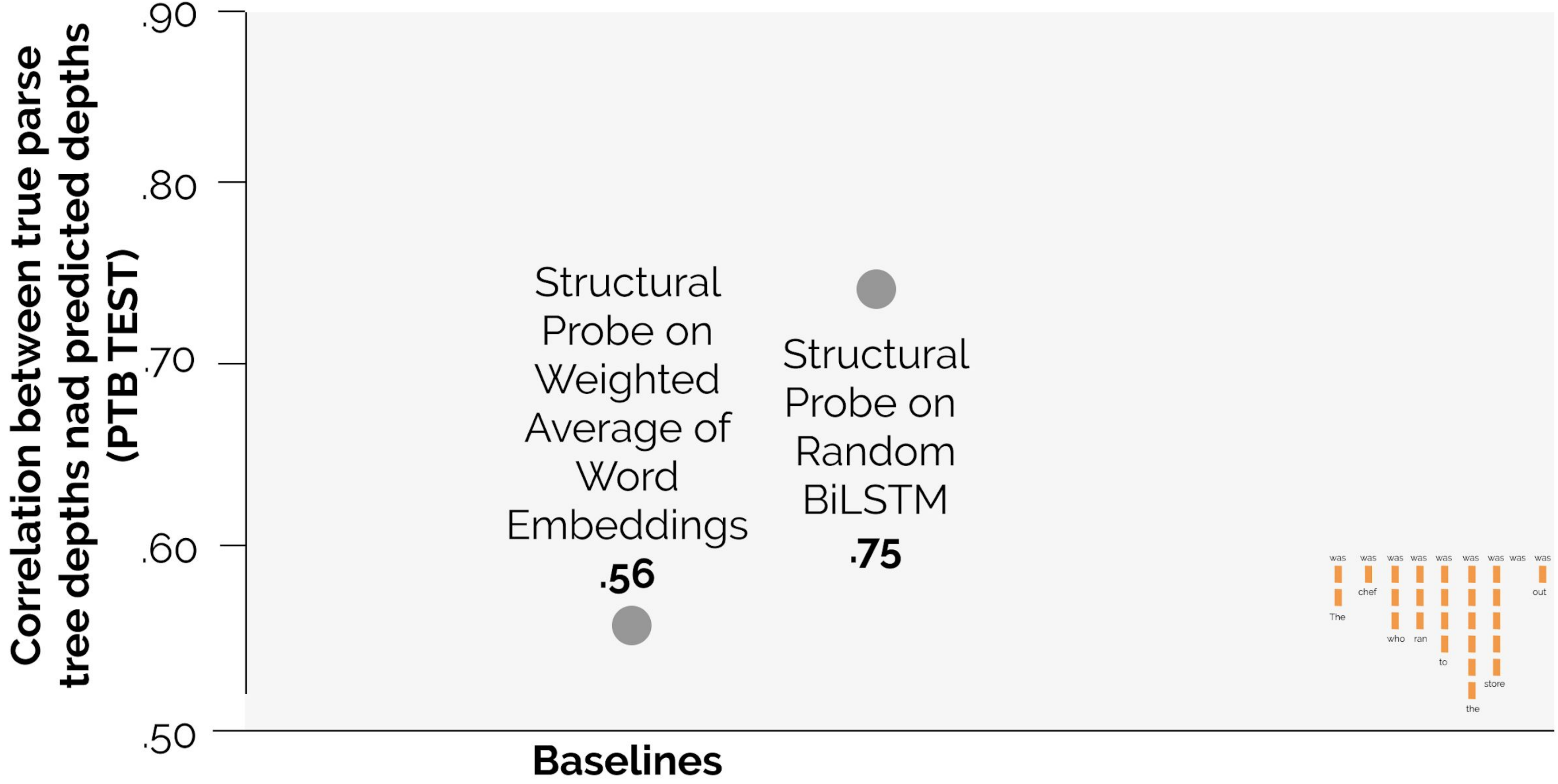
# Trees aren't well-encoded in baselines



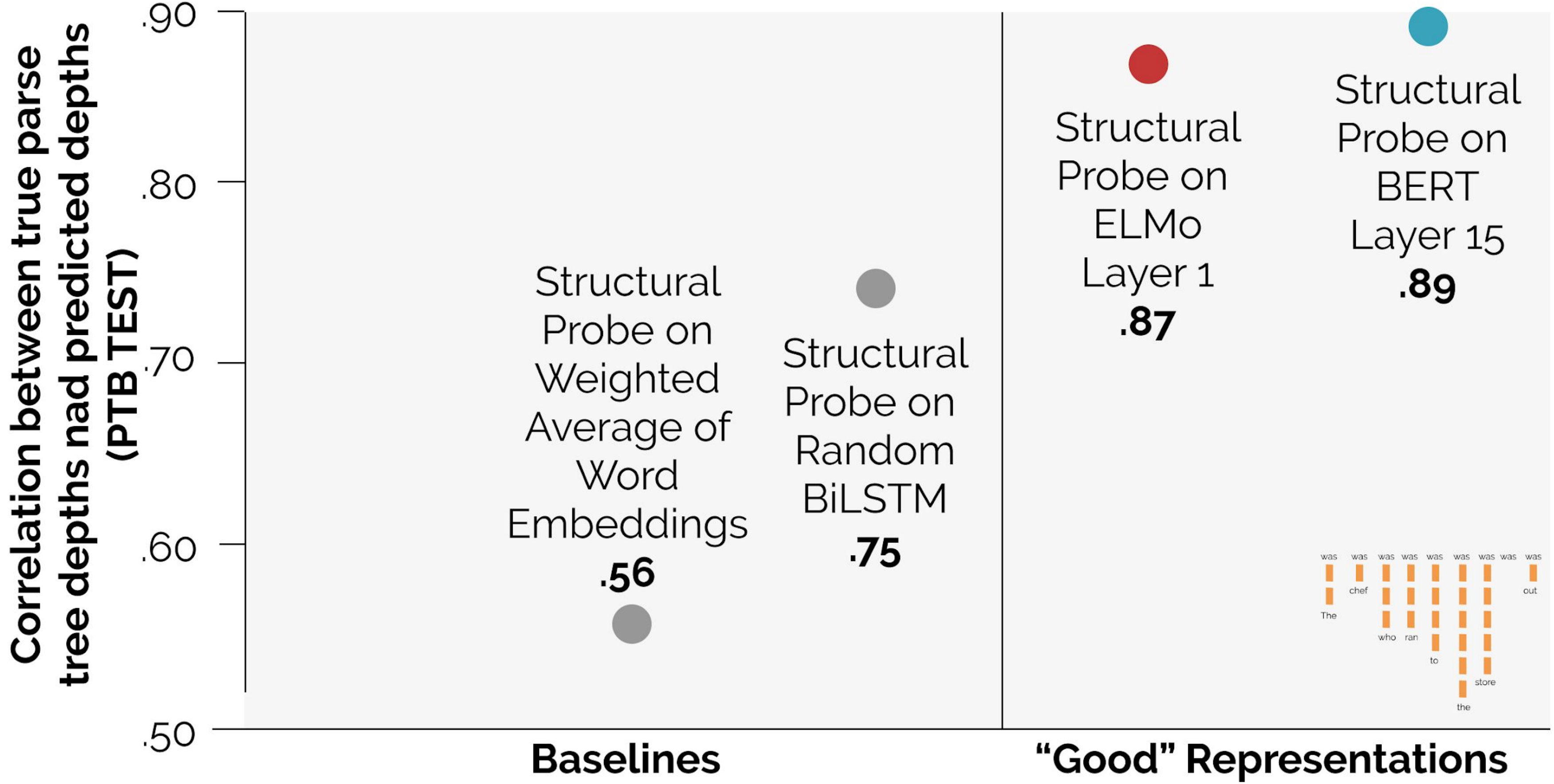
# But they are in trained representations!



# Tree depth isn't well-encoded in baselines

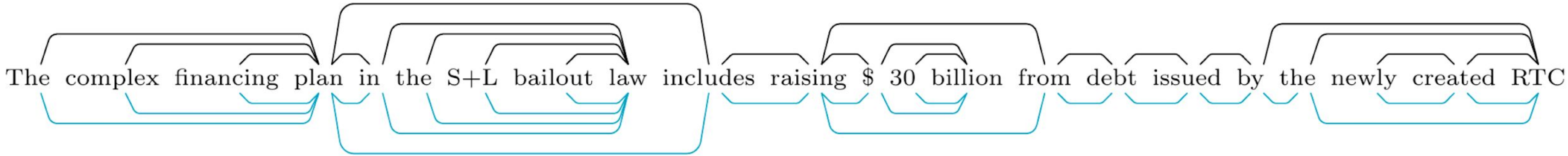


# But it is in trained representations!



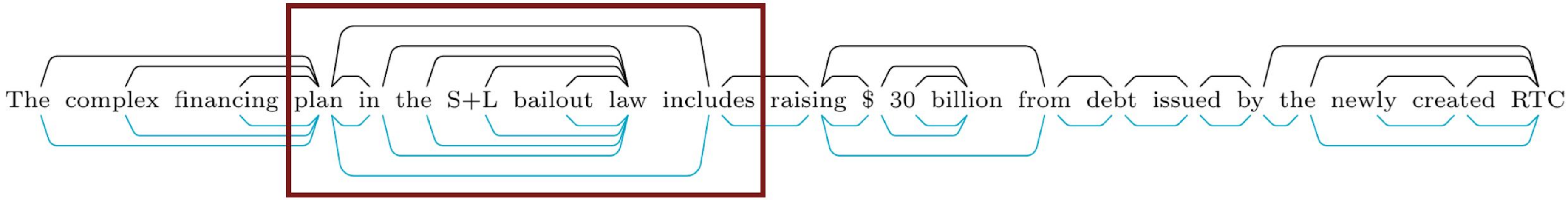
# Trees from structural probe parse distances approximate parse trees pretty well!

**Black (above sentence): Human-annotated parse tree**
**Teal (below sentence): Minimum spanning tree, structural probe on BERT**



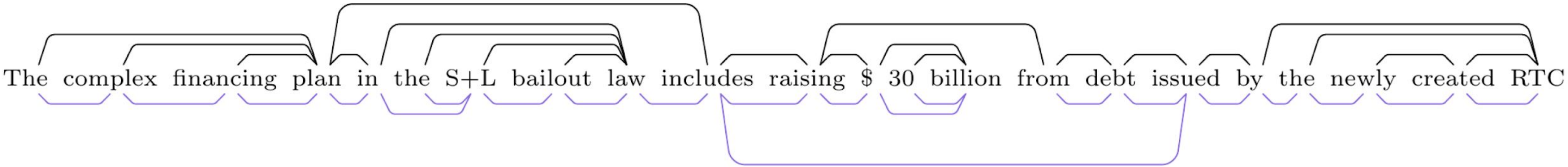
# Trees from structural probe parse distances approximate parse trees pretty well!

**Black (above sentence): Human-annotated parse tree**
**Teal (below sentence): Minimum spanning tree, structural probe on BERT**



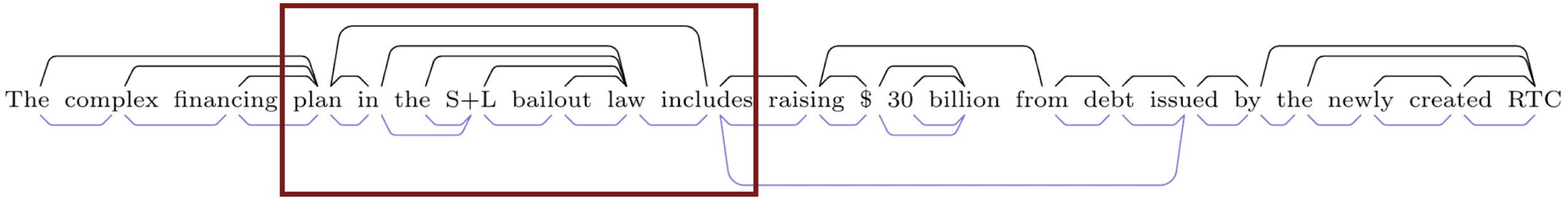
# Trees on baseline representations don't approximate gold trees well!

**Black (above sentence): Human-annotated parse tree**
**Purple (below sentence): MST, structural probe on random-weights BiLSTM**



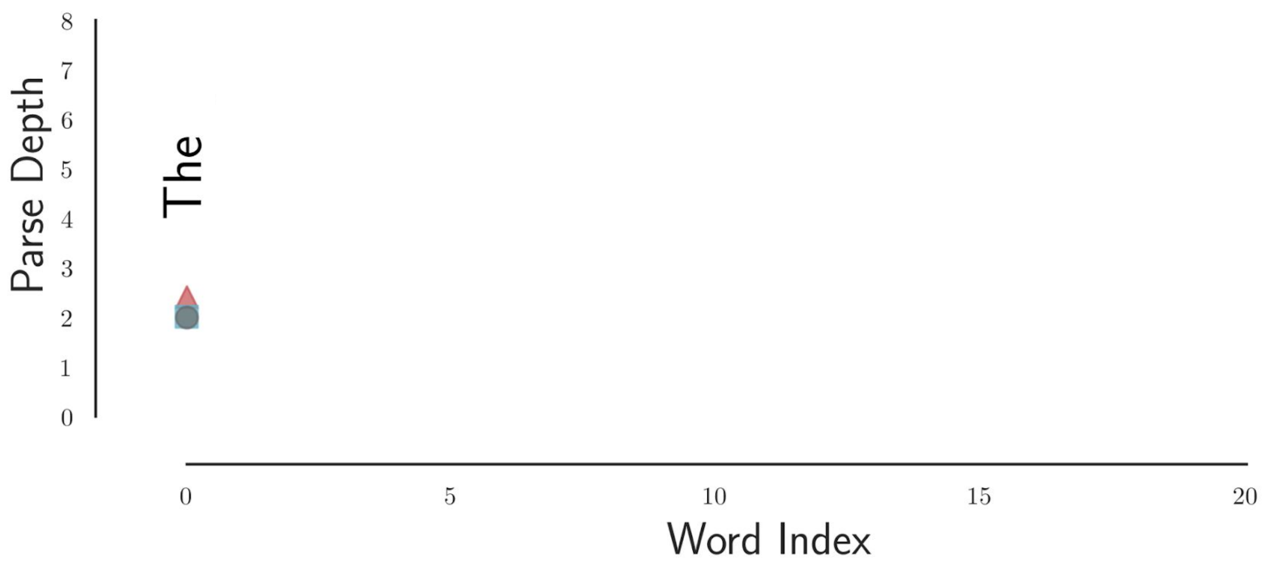
# Trees on baseline representations don't approximate gold trees well!

**Black (above sentence): Human-annotated parse tree**
**Purple (below sentence): MST, structural probe on random-weights BiLSTM**



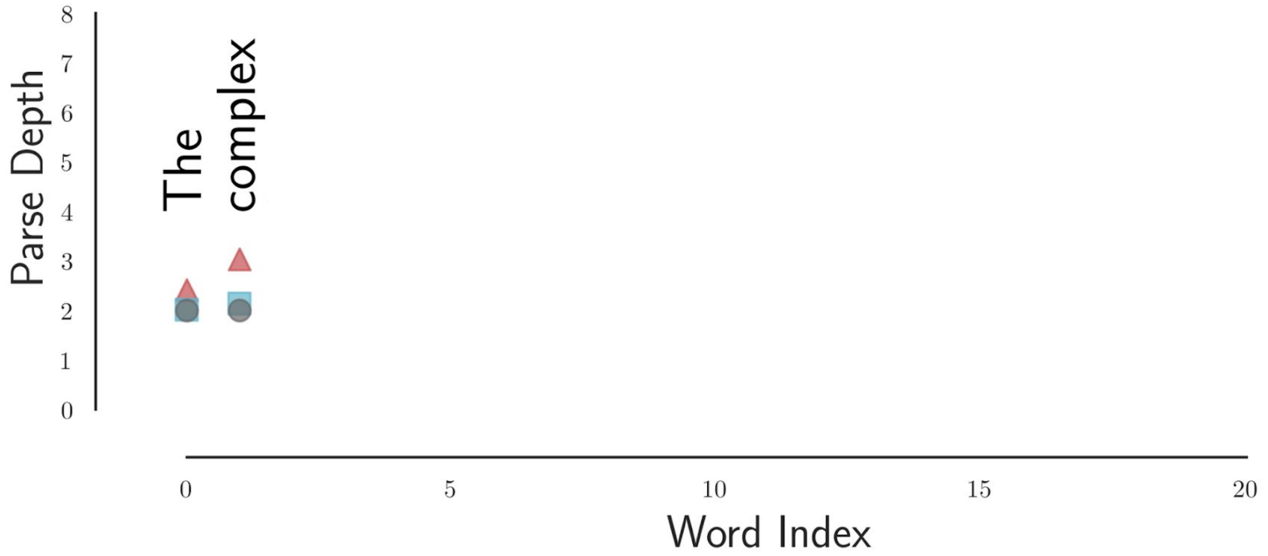
# Predicted depths on BERT + ELMo reconstruct parse depths well!

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



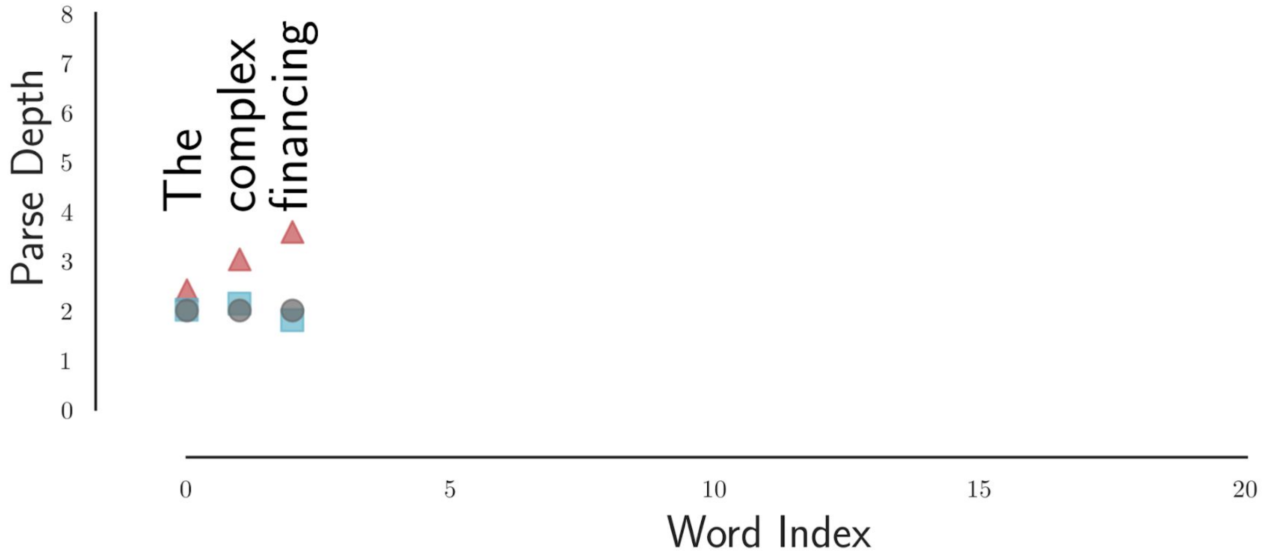
# Predicted depths on BERT + ELMo reconstruct parse depths well!

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



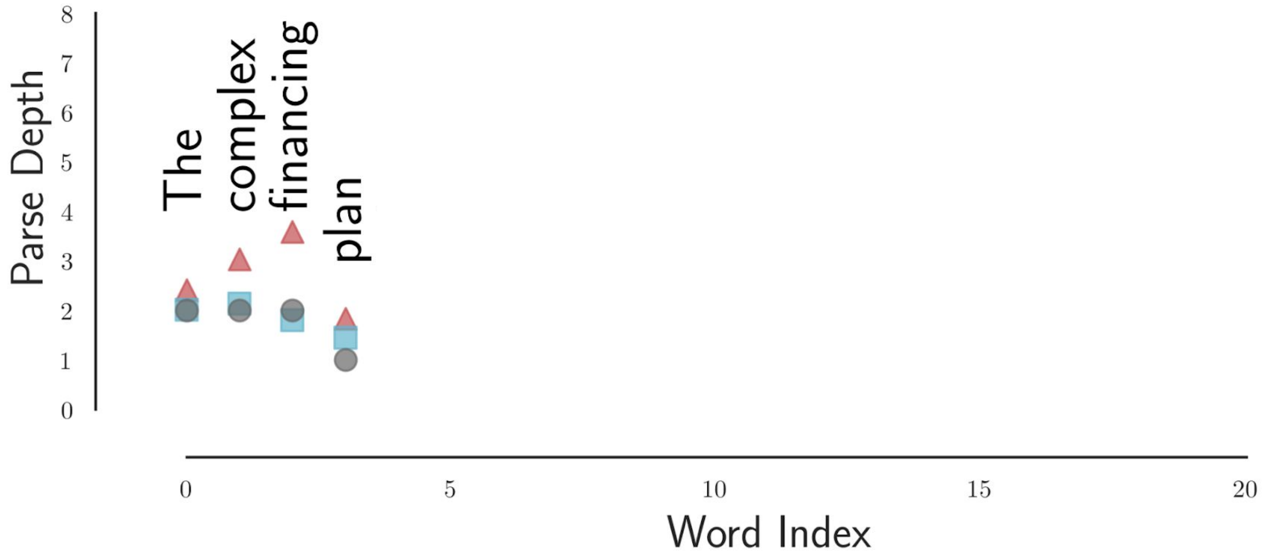
# Predicted depths on BERT + ELMo reconstruct parse depths well!

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



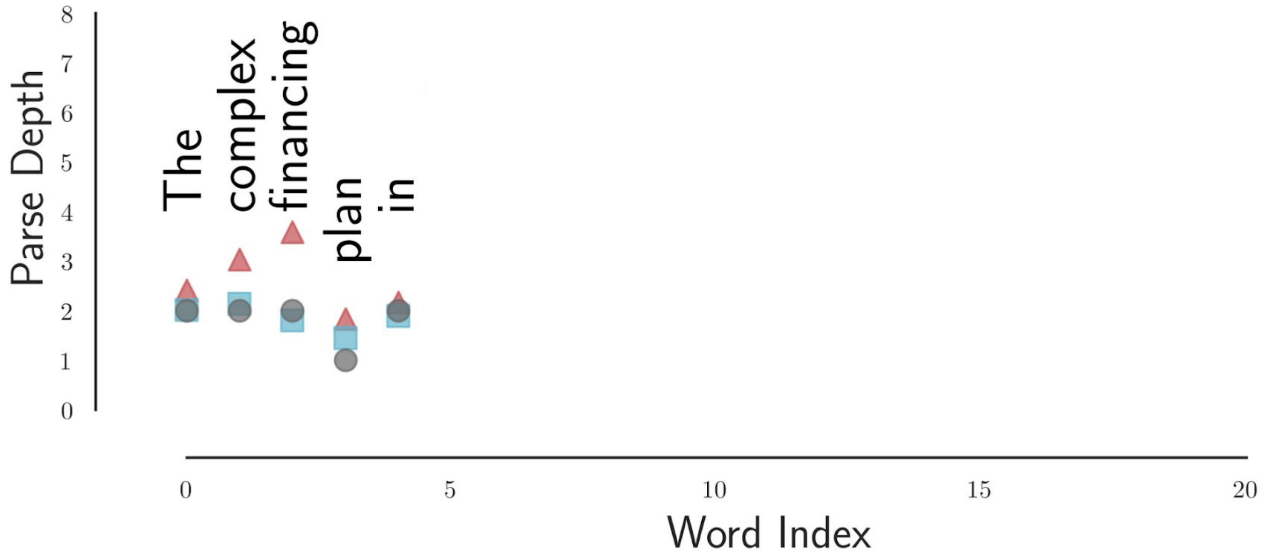
# Predicted depths on BERT + ELMo reconstruct parse depths well!

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



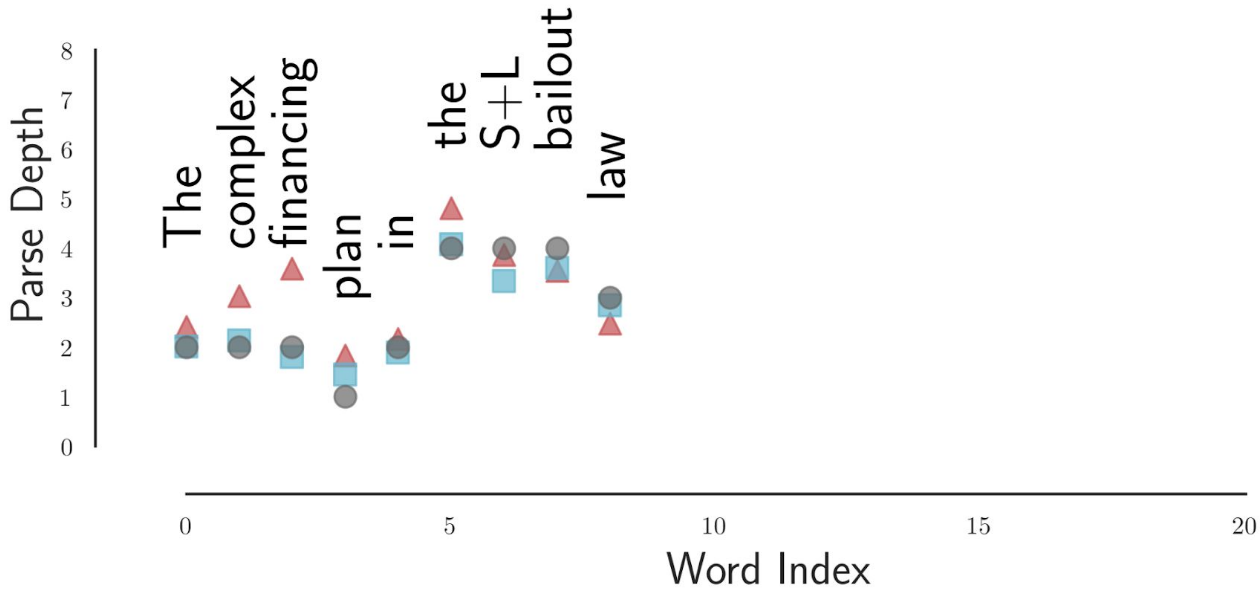
# Predicted depths on BERT + ELMo reconstruct parse depths well!

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



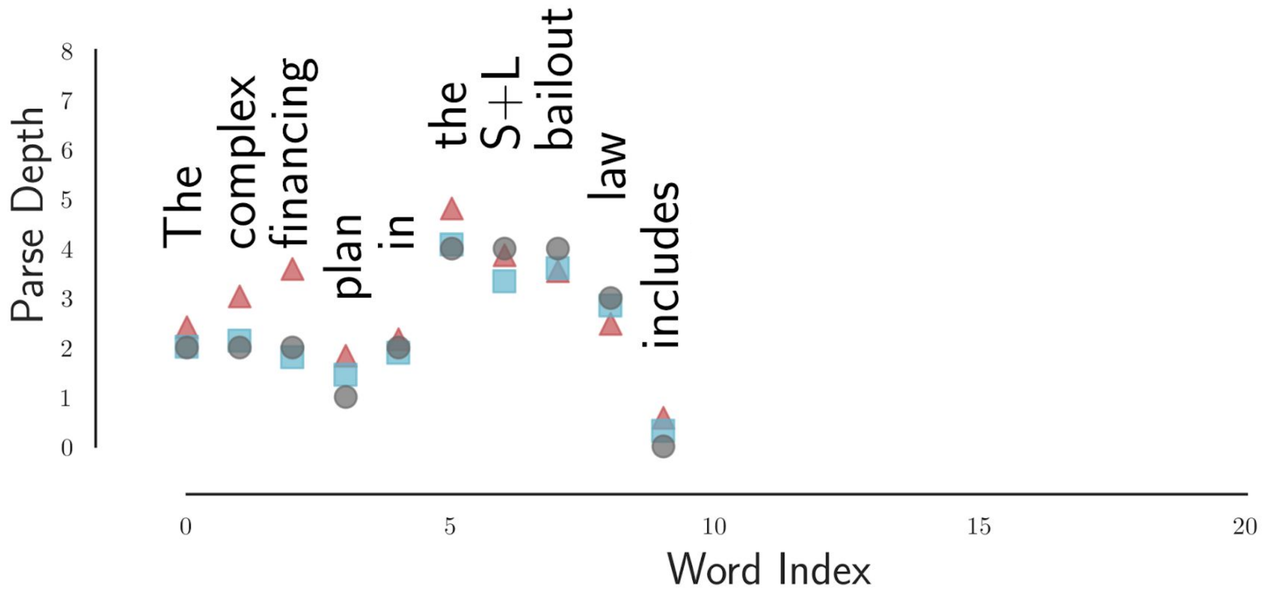
# Predicted depths on BERT + ELMo reconstruct parse depths well!

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



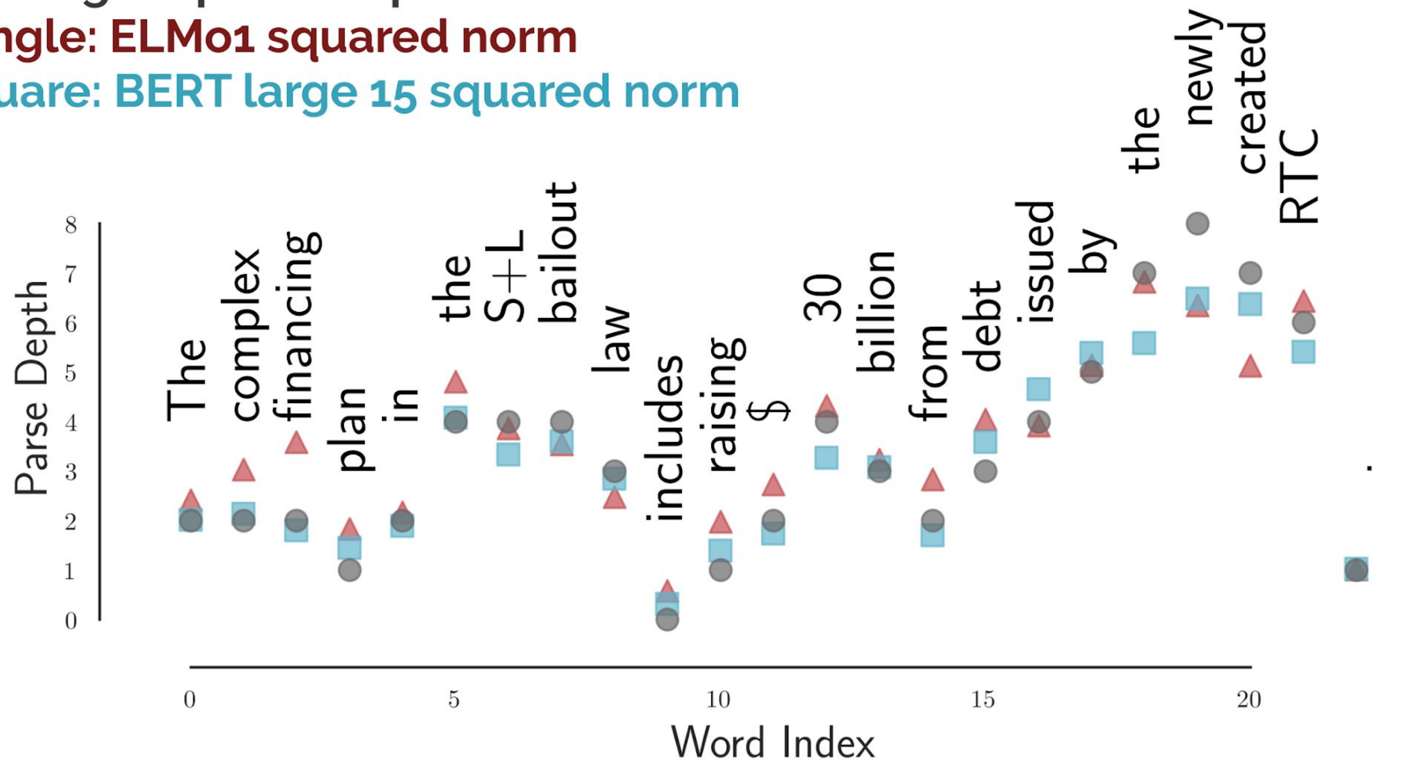
# Predicted depths on BERT + ELMo reconstruct parse depths well!

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

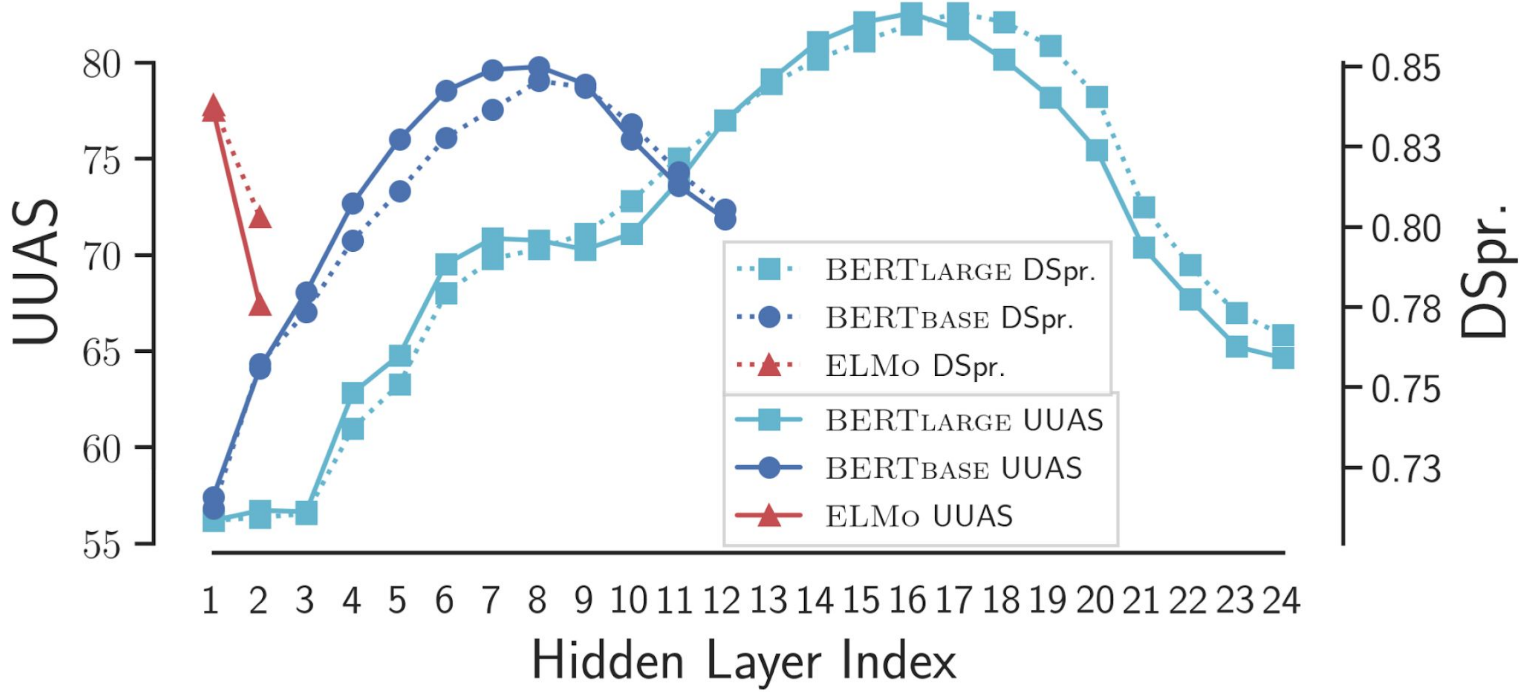


# Predicted depths on BERT + ELMo reconstruct parse depths well!

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



# Syntax geometry differs between layers

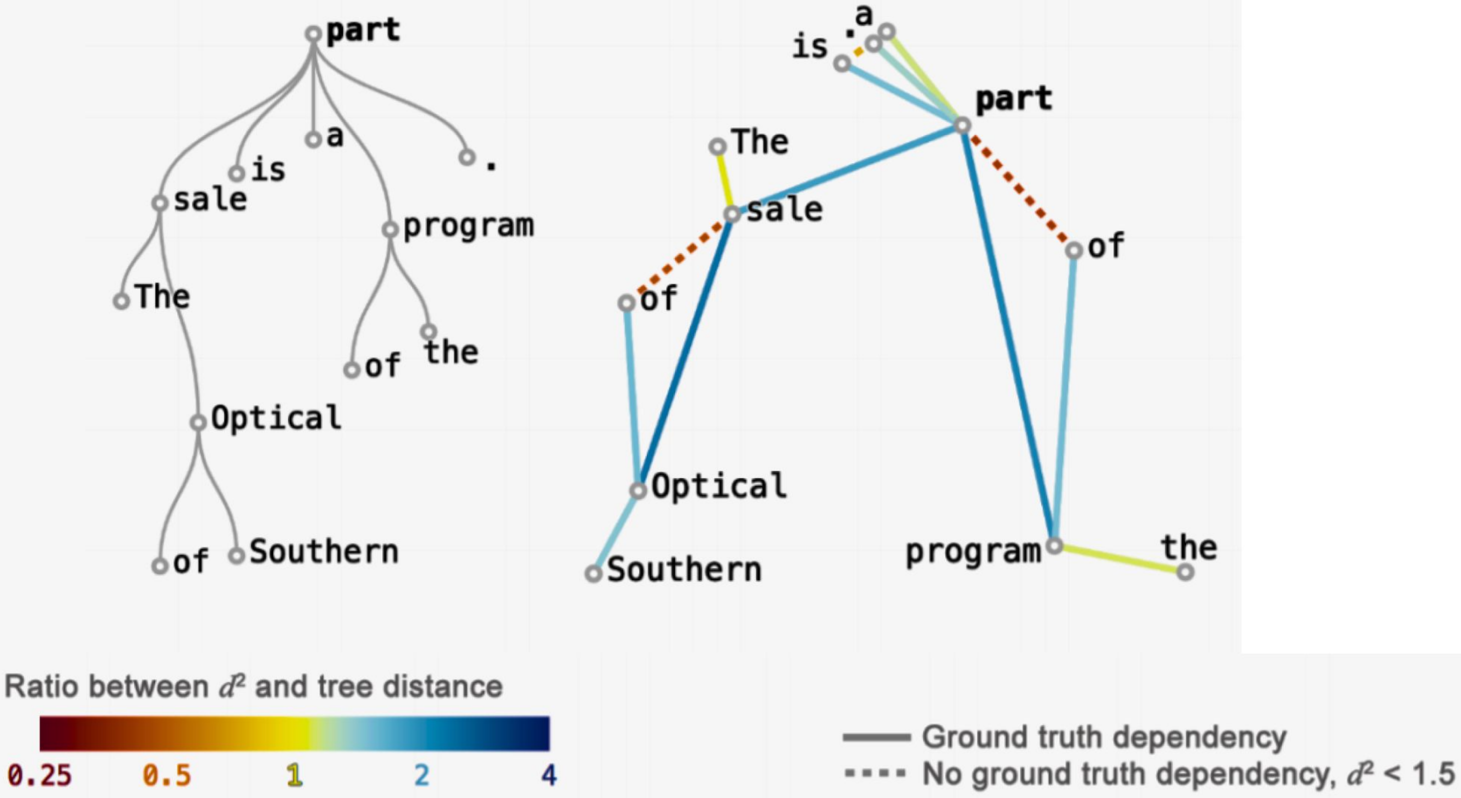


# More visualizations of structural probe

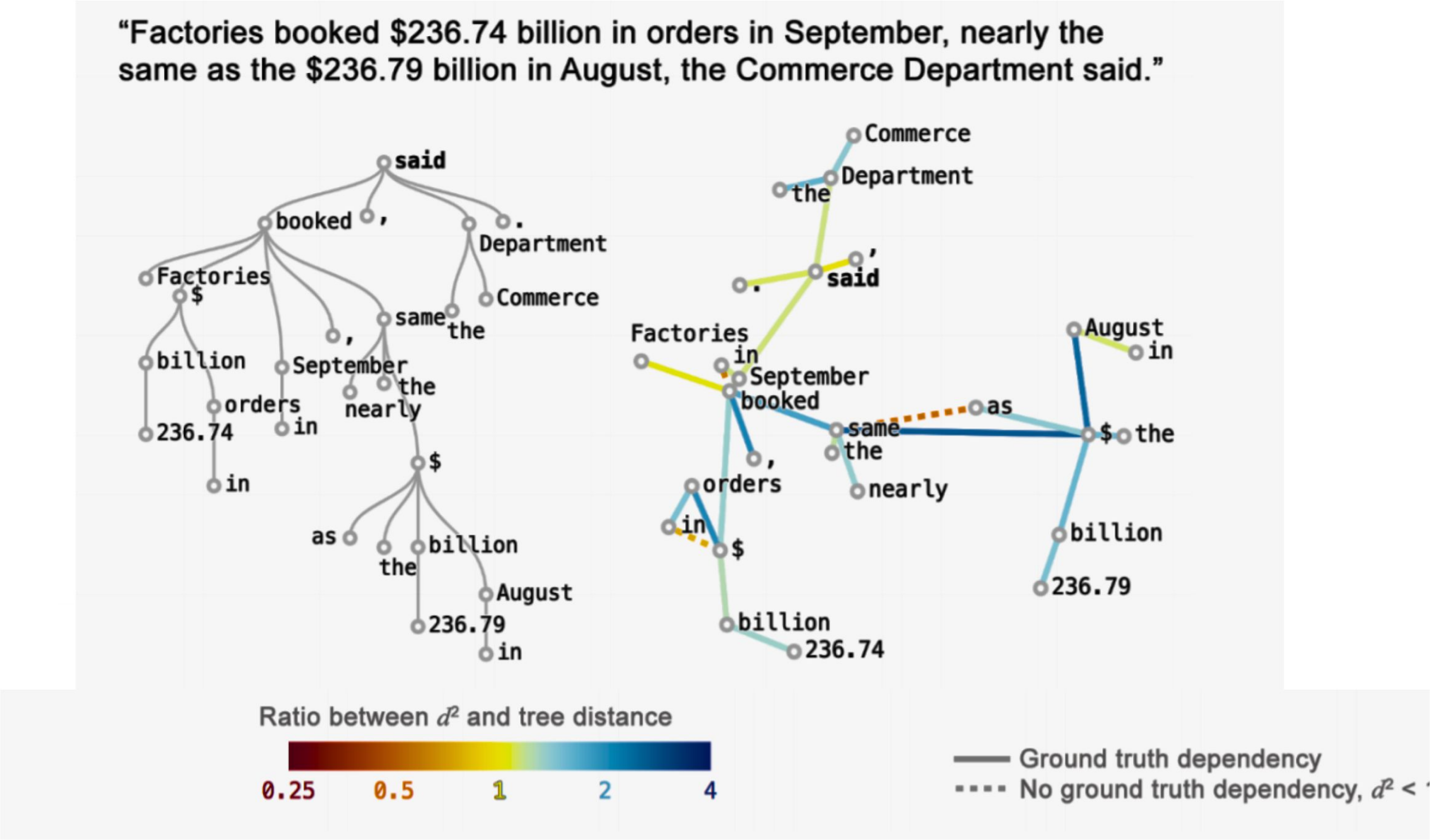
## Visualizing and Measuring the Geometry of BERT

**Andy Coenen,* Emily Reif,* Ann Yuan***
**Been Kim, Adam Pearce, Fernanda Viégas, Martin Wattenberg**
Google Brain
Cambridge, MA
{andycoenen,ereif,annyuan,beenkim,adampearce,viegas,wattenberg}@google.com

“The sale of Southern Optical is a part of the program.”




[Reif et al., 2019]




[Reif et al., 2019]

# Probing results can be hard to interpret

Supervised classifiers are powerful even when simple, and it can be unclear what you're learning about the representation itself.

You can learn good classifiers on top of lots of representations.
How do we know what a probing accuracy means?

[Hewitt and Liang, 2019]

# Lecture Outline

Lecture 20: Analysis and Interpretability of Neural NLP

1. Motivation: what are our models doing?
2. Neural networks as linguistic test subjects
3. Careful ablation studies and architecture modifications
4. Analysis of inherently interpretable architectures
5. Playing the adversary: breaking NLP models
6. Analyzing representations using supervised methods
7. **Aggregating analysis insights across studies**

# Aggregating analyses in surveys and toolkits

Each analysis paper asks a very specific question.

How do we ask, *what does the field currently know about BERT?*

Answer: meta-studies compiling results

**Analysis Methods in Neural Language Processing: A Survey**

**Yonatan Belinkov**[1,2] and **James Glass**[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory
[2]Harvard School of Engineering and Applied Sciences
Cambridge, MA, USA
{belinkov, glass}@mit.edu

**A Primer in BERTology: What we know about how BERT works**

**Anna Rogers, Olga Kovaleva, Anna Rumshisky**
Department of Computer Science, University of Massachusetts Lowell
Lowell, MA 01854
{arogers, okovalev, arum}@cs.uml.edu

Stanford | NLP

# Aggregating analyses in surveys and toolkits

How do we ask, *what can I easily find out about **my** model?*

Answer: interpretability toolkits!

**AllenNLP Interpret:**
**A Framework for Explaining Predictions of NLP Models**

**Eric Wallace**[1] **Jens Tuyls**[2] **Junlin Wang**[2] **Sanjay Subramanian**[1]
**Matt Gardner**[1] **Sameer Singh**[2]
[1]Allen Institute for Artificial Intelligence [2]University of California, Irvine
ericw@allenai.org, sameer@uci.edu

**Input Reduction**

**Input Reduction** removes as many words from the input as possible without changing the model's prediction.

**Original Premise:** Two women are wandering along the shore drinking iced tea.

**Original Hypothesis:** ~~Two women are sitting on a blanket near some rocks talking about~~ politics

**Reduced Hypothesis:** politics

# Takeaways

Neural models are complex, fascinating objects that we don't currently understand, but we're making strides to understand them better!

A wide variety of analysis methods have been developed, for:

- Understanding a model's behavior on specific phenomena
- Understanding what a model learns about a topic or task
- Understanding what seemingly innocuous input changes make a model fail
- Many other things, with more coming every day!

These methods can be integrated into your future NLP projects!