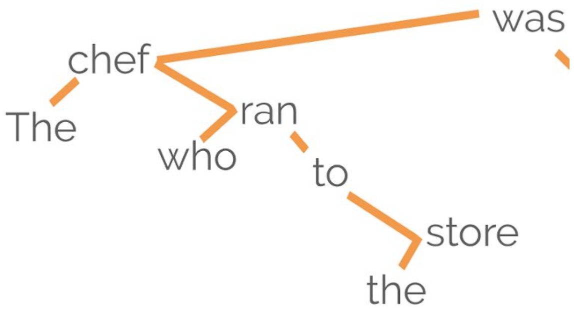
A Structural Probe for Finding Syntax in Word Representations

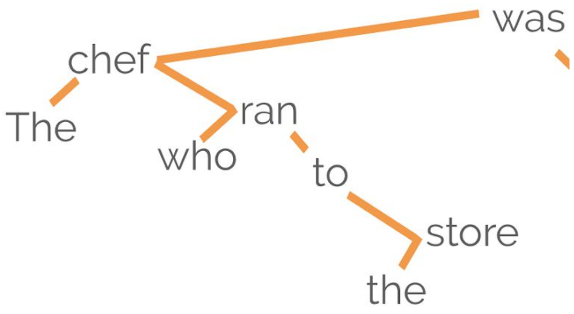
John Hewitt

Christopher Manning



Some thoughts:

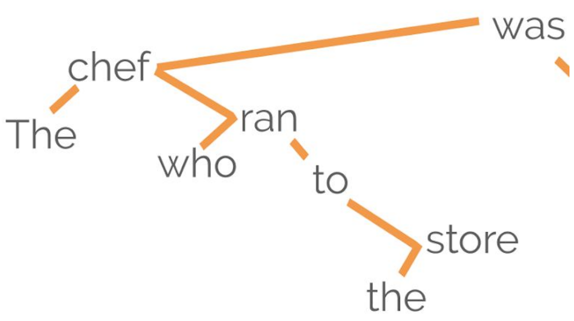
Human language has rich hierarchical structure.



Some thoughts:

Human language has rich hierarchical structure.

BERT and ELMo work really well.[citation needed]



Some thoughts:

Human language has rich hierarchical structure.

BERT and ELMo work really well.[citation needed]

...without explicit representations of hierarchy.

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.

Related work: what does my unsupervised neural network learn about language?

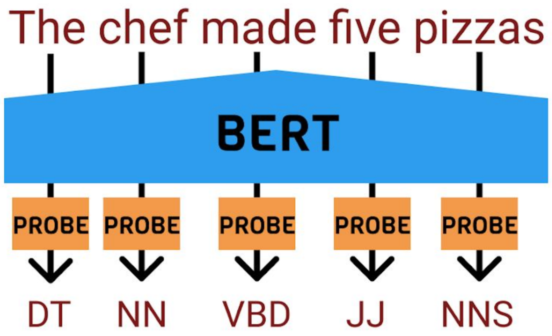
Probing: train a simple model to extract linguistic properties from vector representations.

+ Other things! [Shi et al., 2016. Peters et al., 2018. Tenney et al., 2019. Liu et al., 2019,...]

Related work: what does my unsupervised neural network learn about language?

Probing: train a simple model to extract linguistic properties from vector representations.

Part-of-speech!

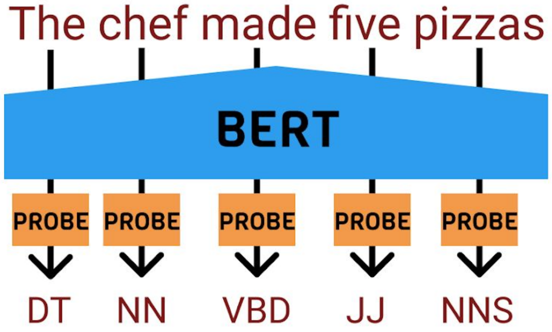


+ Other things! [Shi et al., 2016. Peters et al., 2018. Tenney et al., 2019. Liu et al., 2019,...]

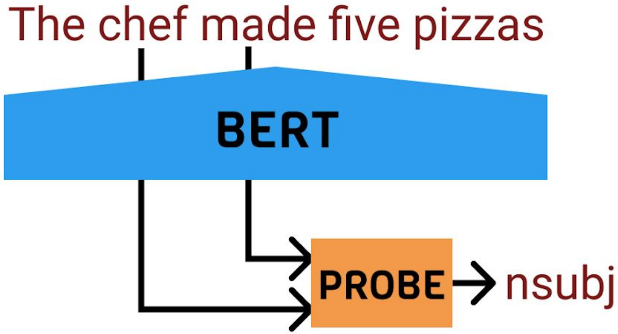
Related work: what does my unsupervised neural network learn about language?

Probing: train a simple model to extract linguistic properties from vector representations.

Part-of-speech!



Partial dependency info!

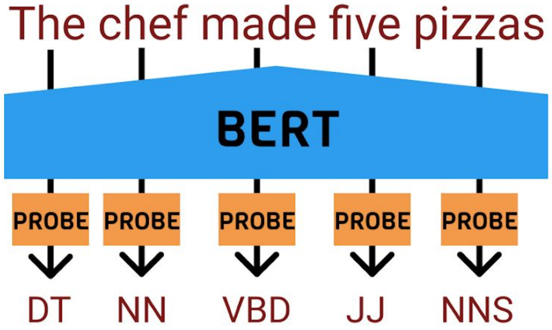


+ Other things! [Shi et al., 2016. Peters et al., 2018. Tenney et al., 2019. Liu et al., 2019,...]

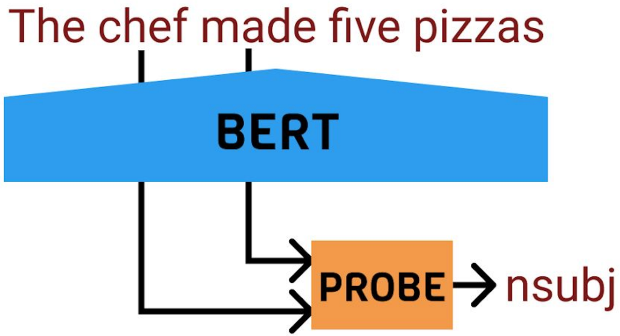
Related work: what does my unsupervised neural network learn about language?

Probing: train a simple model to extract linguistic properties from vector representations. **But hard to ask about whole trees!**

Part-of-speech!



Partial dependency info!



+ Other things! [Shi et al., 2016. Peters et al., 2018. Tenney et al., 2019. Liu et al., 2019,...]

Outline

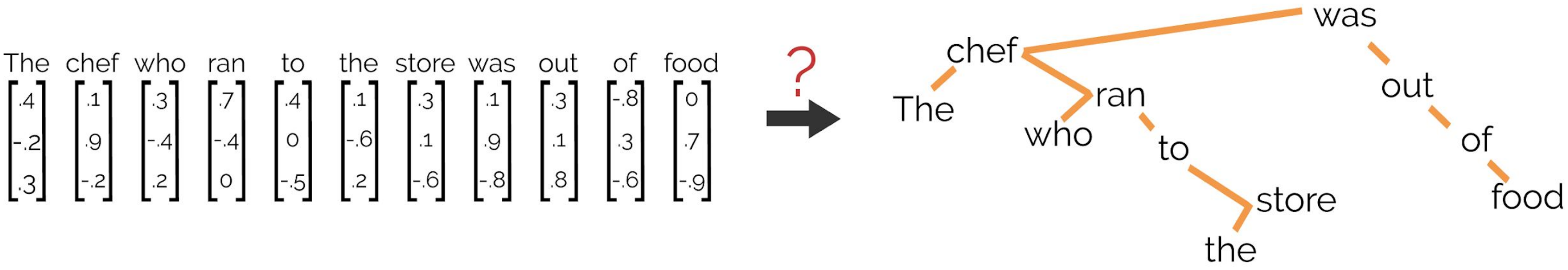
1. *connecting* **vector spaces** and **trees**

2. The **structural probe** method

3. Results and pictures and fun

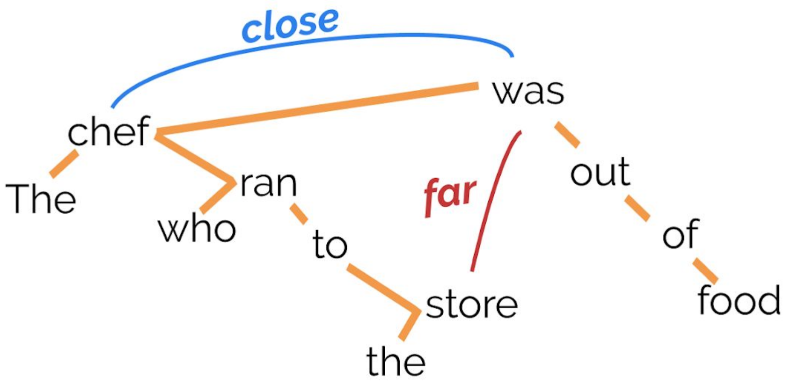
Are vector spaces and trees reconcilable?

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.

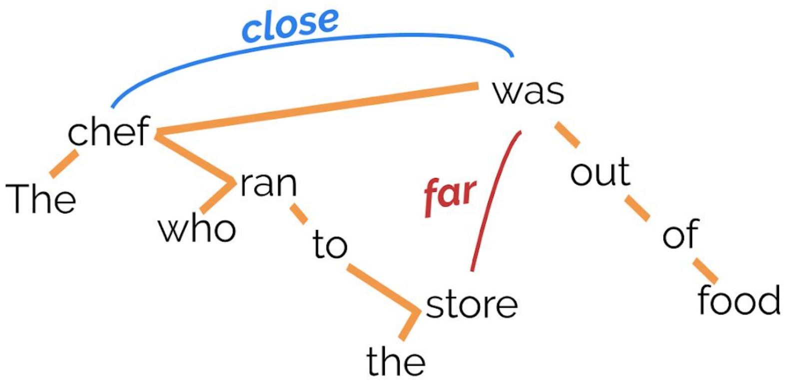


The —— chef $d_{\text{path}} = 1$

[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.



The —— chef $d_{path} = 1$

...

chef —— ran $d_{path} = 1$

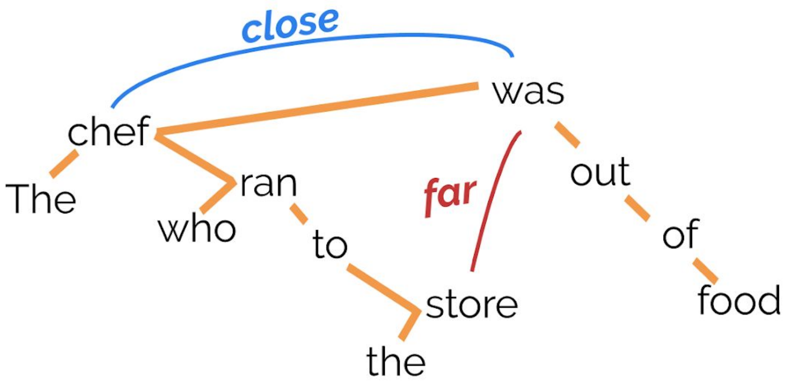
chef —— was $d_{path} = 1$

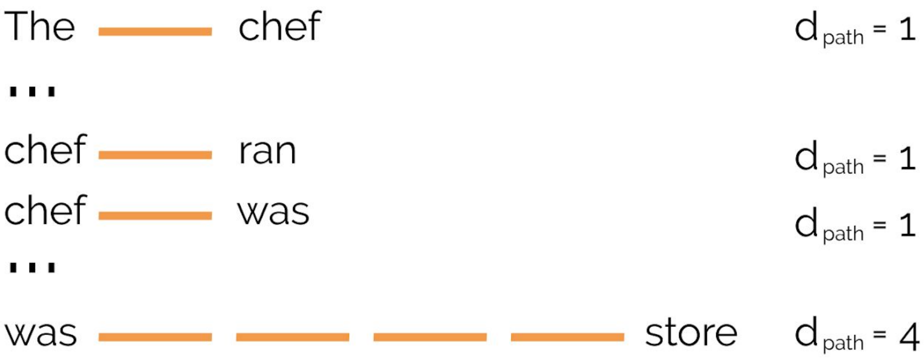
...

[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.

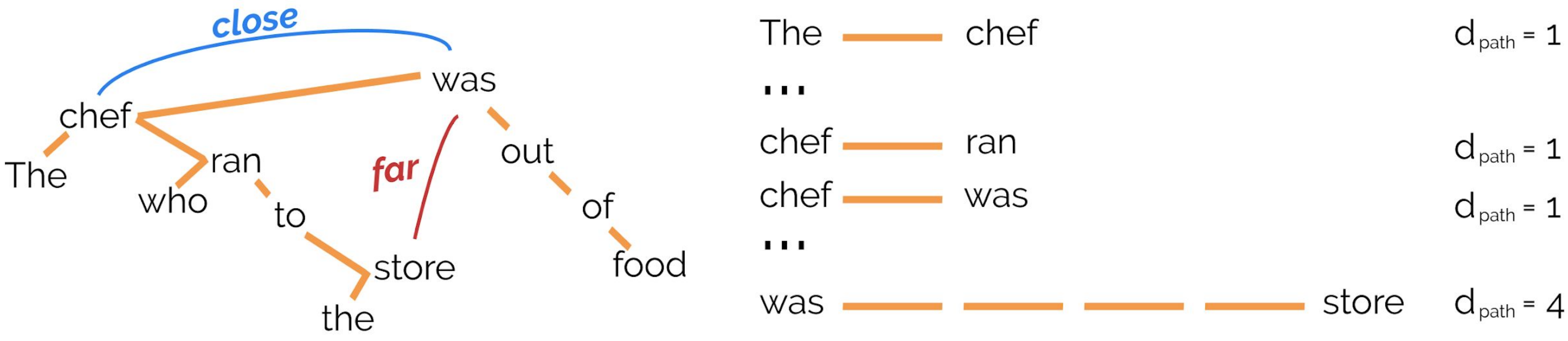




[For more fun, see Deza and Laurent. *A Geometry of Cuts and Metrics.* Springer. 2009]

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.

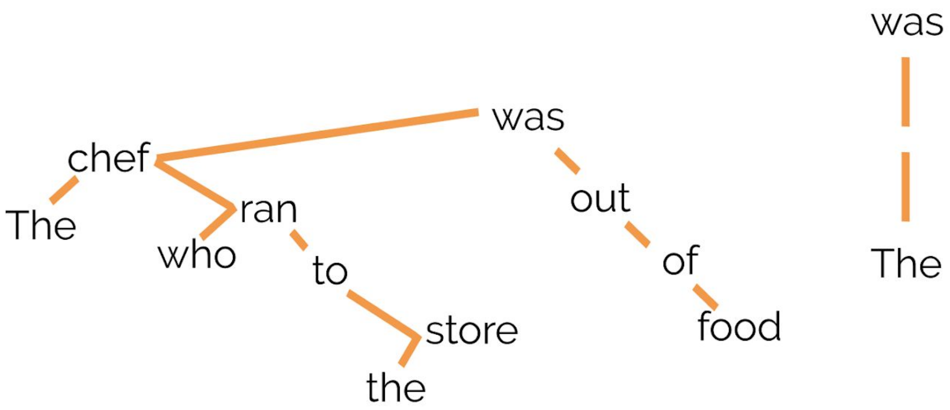


The edges of the tree can be recovered by looking at all distance=1 pairs.

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

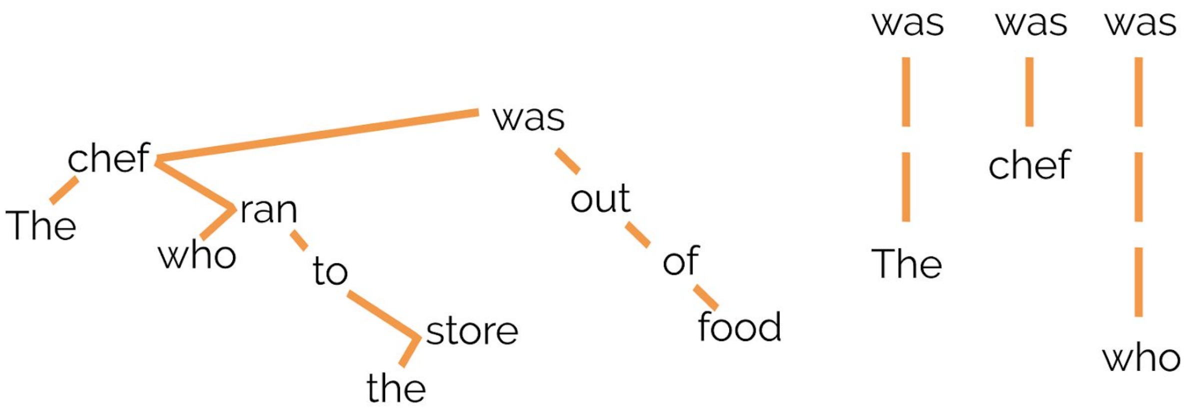
Norms unify edge directions and vectors

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



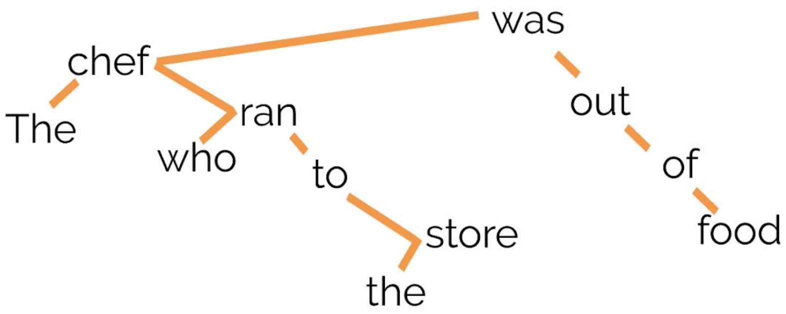
Norms unify edge directions and vectors

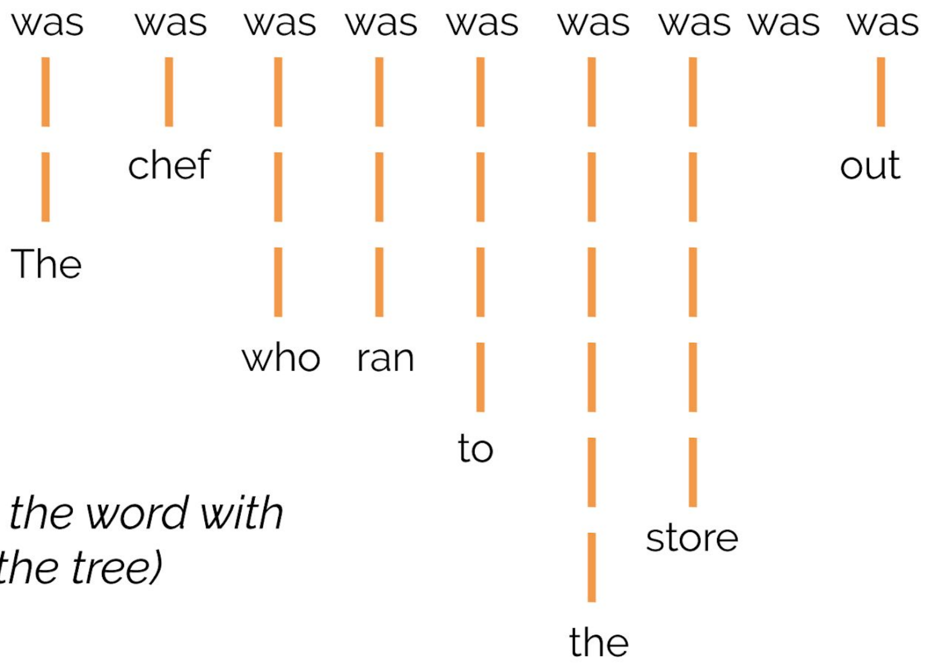
A **rooted tree** defines a **norm** on the words, the parse depth:
the number of edges from each word to ROOT.



Norms unify edge directions and vectors

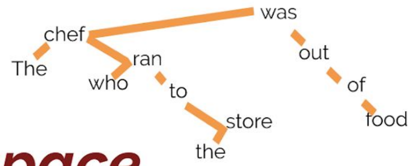
A **rooted tree** defines a **norm** on the words, the parse depth:
the number of edges from each word to ROOT.





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

summary

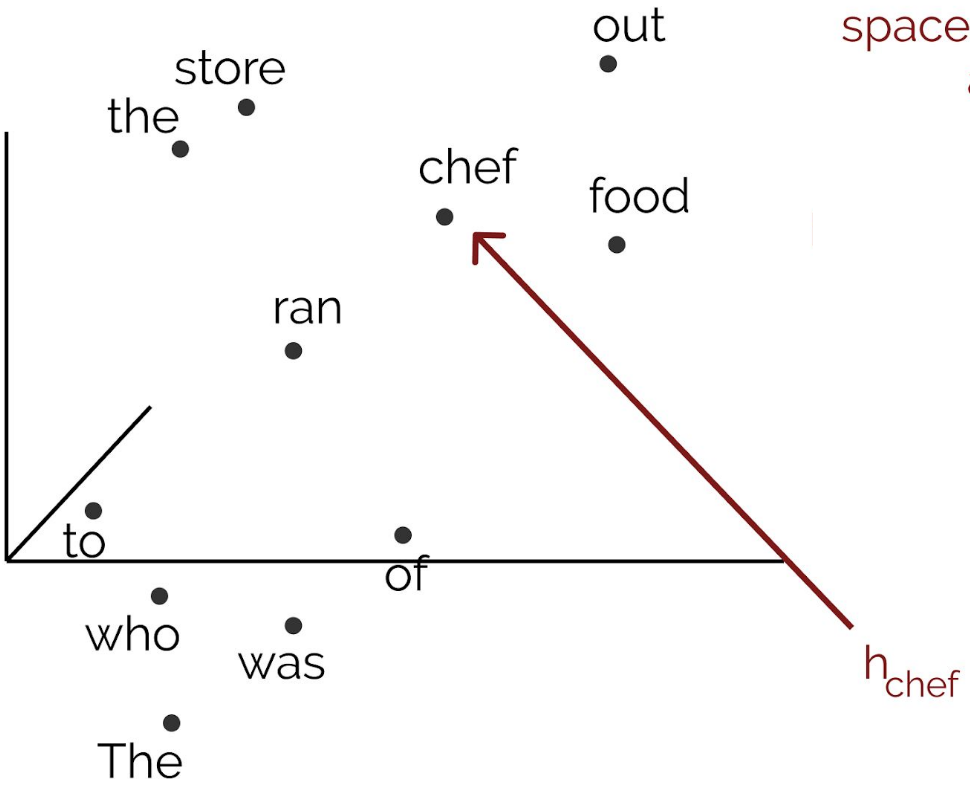


distance** unifies **undirected trees** and **vector space

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

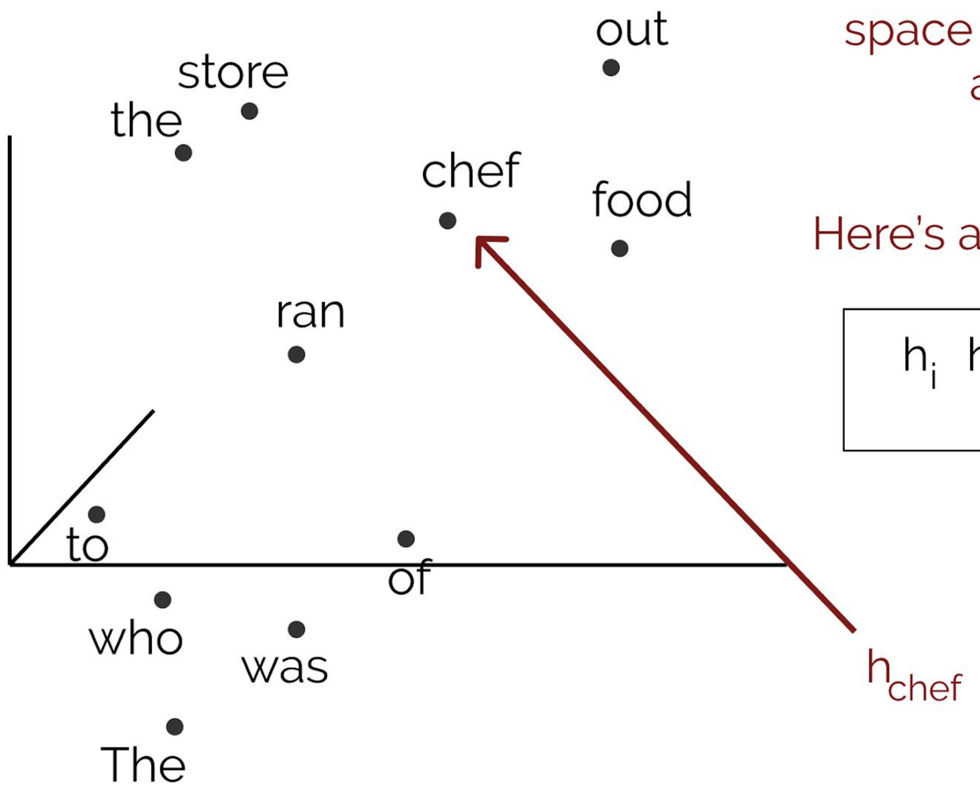
The *structural probe* method

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

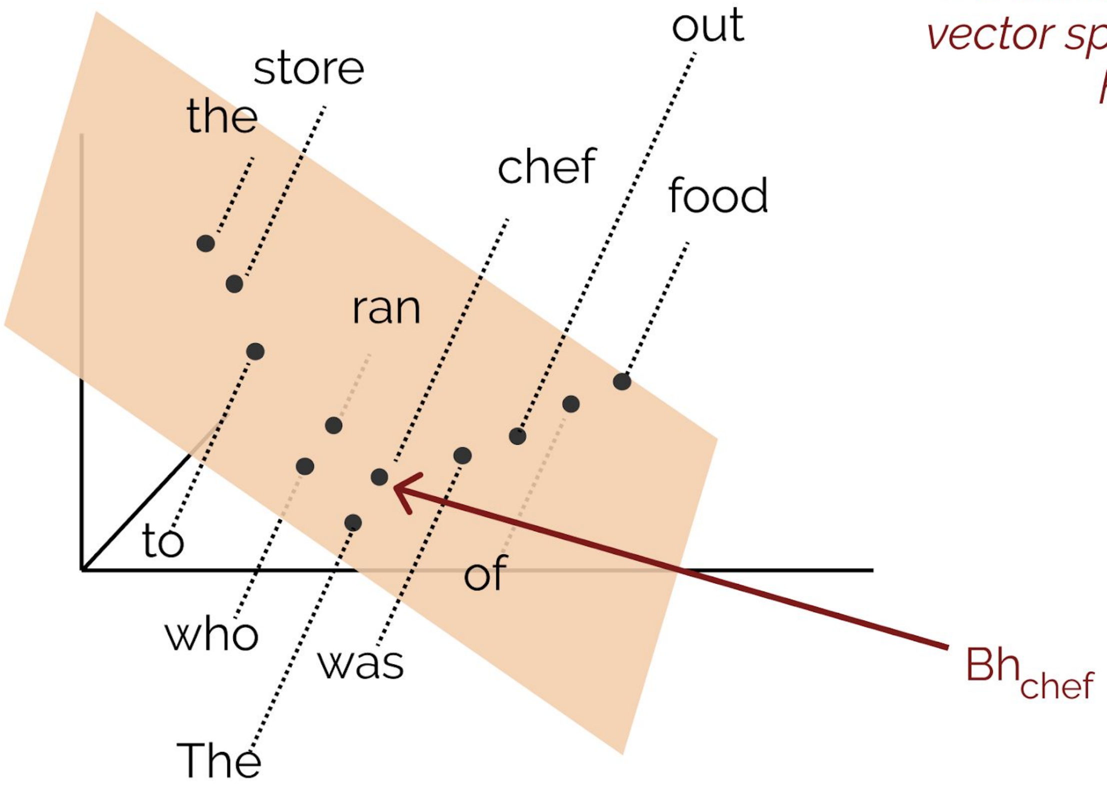


We can look for trees in the vector space by looking for their **distances** and **norms** in the space.

Here's a sentence embedded by a NN!

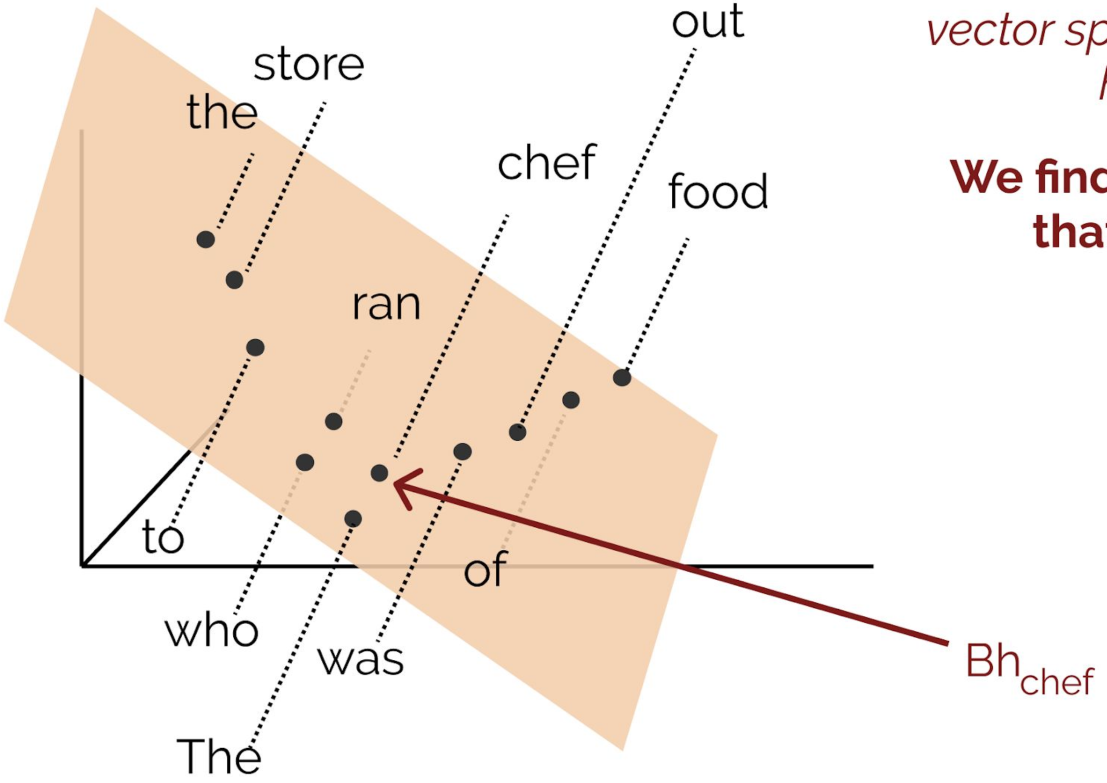
h_i h_j : vector representation of words *i* and *j*.

Finding trees in vector spaces



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

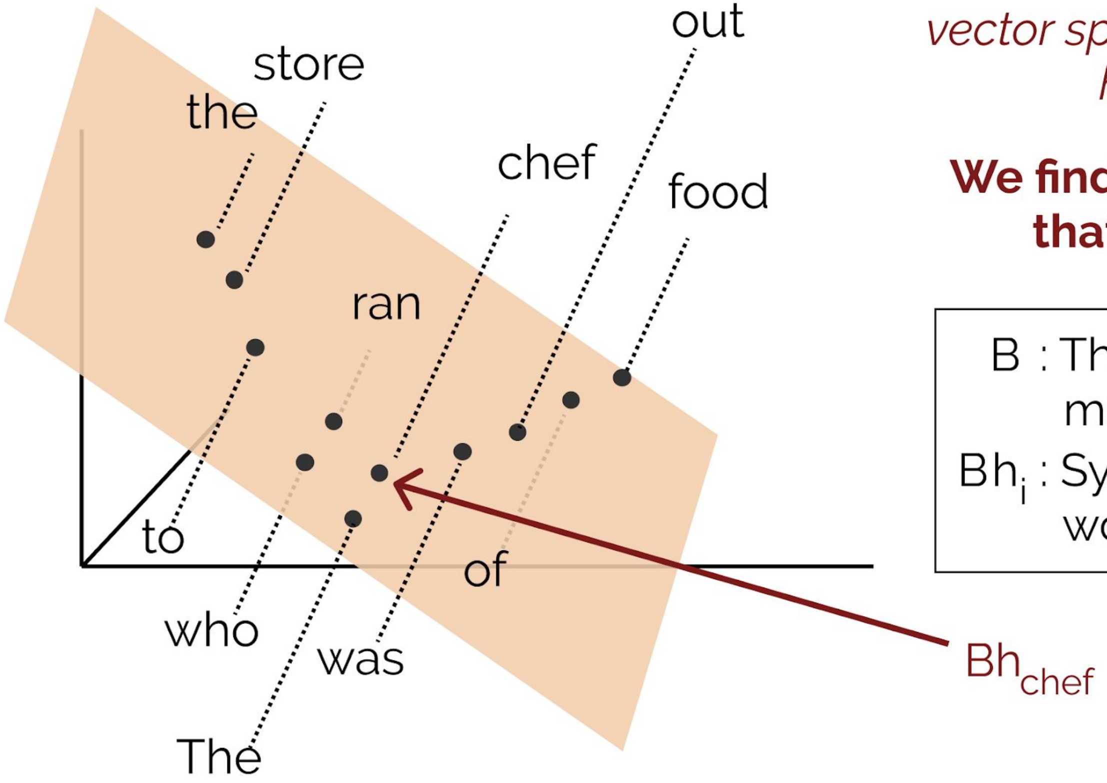
Finding trees in vector spaces



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



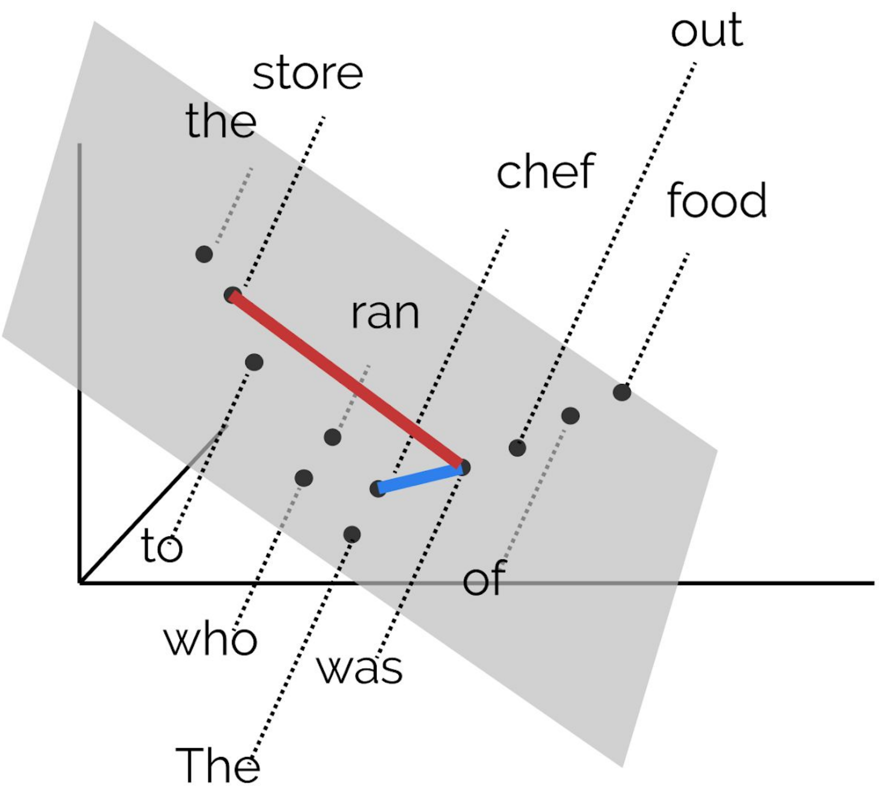
We don't expect all dimensions of the vector space to encode syntax -- NNs have a lot to encode!

We find the linear transformation that encodes syntax best.

B : The syntax transformation matrix

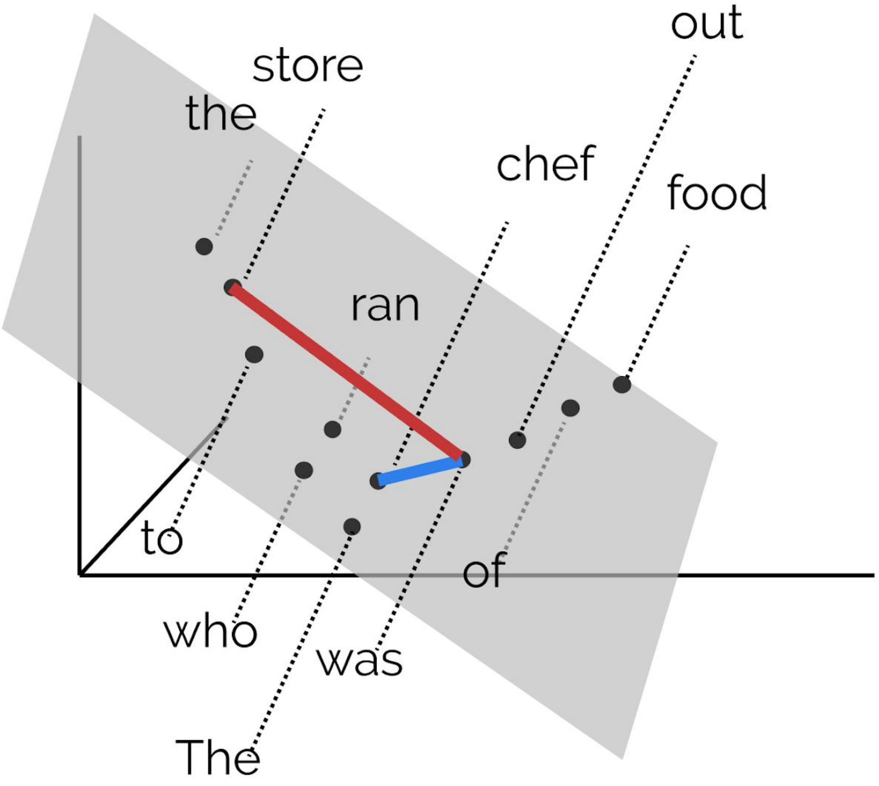
Bh_i : Syntax-transformed vector word representation

Finding trees in vector spaces



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

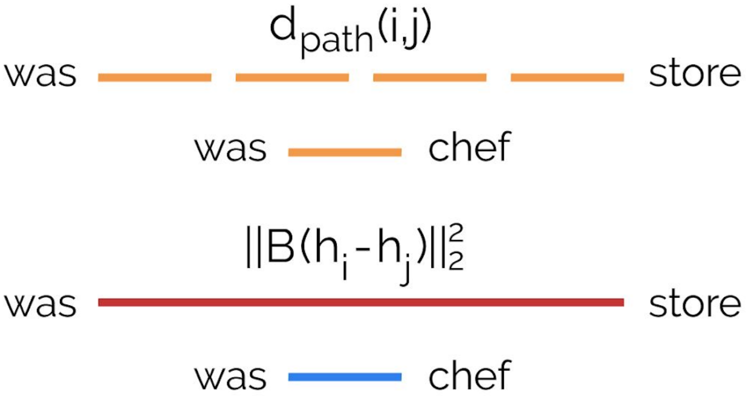
Finding trees in vector spaces



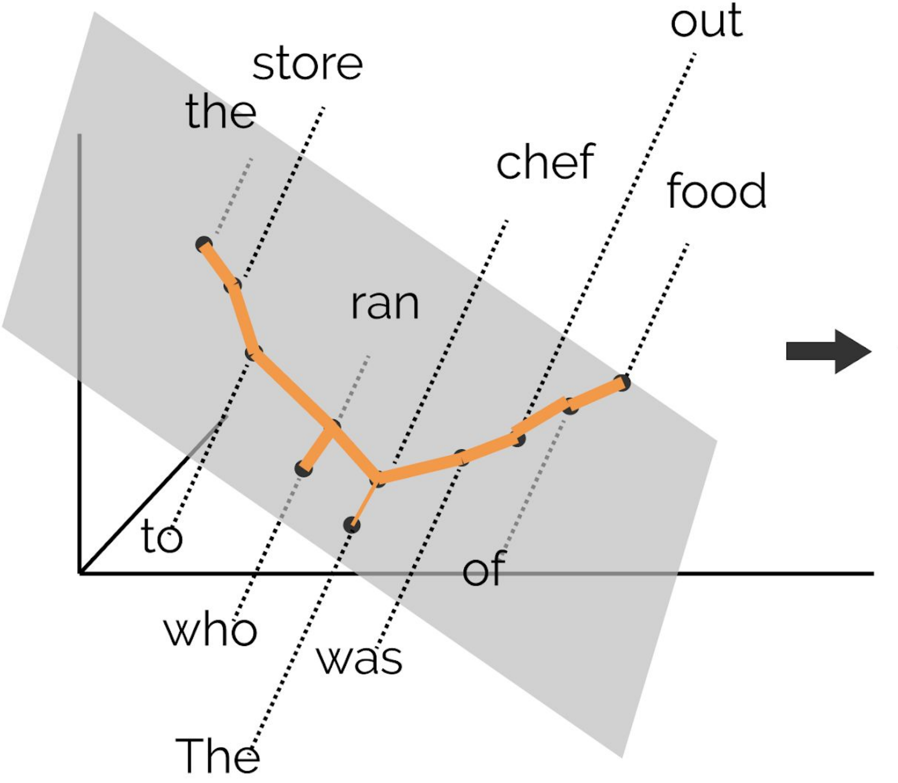
In the transformed space, (squared) L2 distance approximates tree distance.

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

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

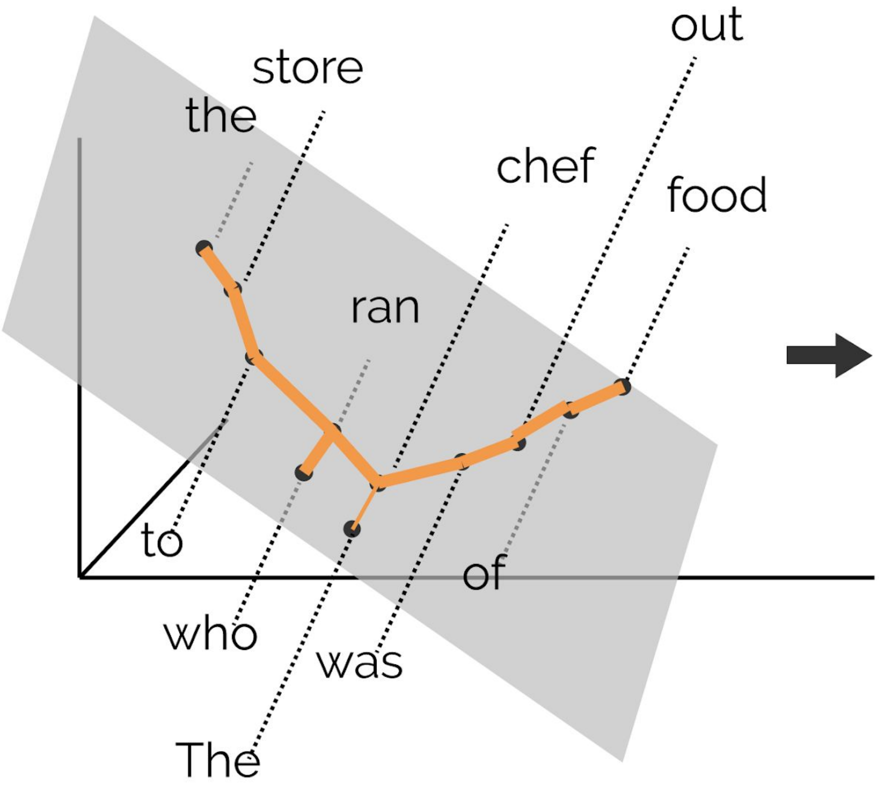


Finding trees in vector spaces

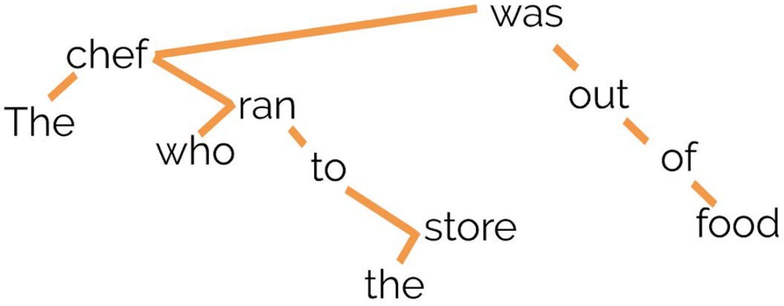


With this property, a minimum spanning tree in the vector space distance recovers the tree.

Finding trees in vector spaces



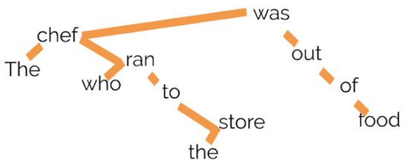
With this property, a minimum spanning tree in the vector space distance recovers the tree.



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

$$\arg\min_{B}$$

Find a single transformation *B*

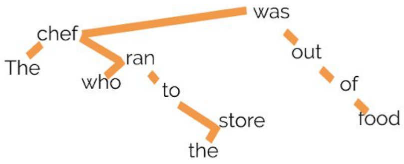


Does BERT encode undirected parse trees -> does there exist a *distance* transformation?

$$\underbrace{\arg\min_{B}}_{\text{Find a single transformation } B} \underbrace{\sum_{\ell \in \mathrm{PTB}}}_{\text{such that over all sentences in PTB training}}$$

Find a single transformation *B*

such that over all sentences in PTB training



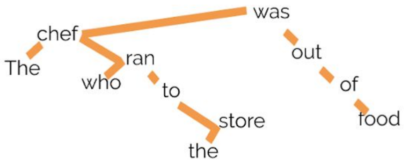
Does BERT encode undirected parse trees -> does there exist a *distance* transformation?

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

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?

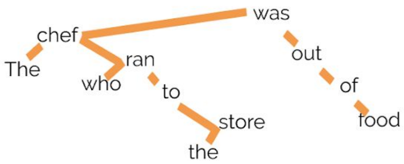
$$\underbrace{\arg\min_{B}}_{\text{Find a single transformation } B} \underbrace{\sum_{\ell \in \mathrm{PTB}}}_{\text{such that over all sentences in PTB training}} \underbrace{\sum_{i,j}}_{\text{Over all word pairs in each sentence}} \underbrace{\left| d^{\ell}_{\mathrm{path}}(i,j) - \|B(h^{\ell}_i - h^{\ell}_j)\|^2_2 \right|}_{\text{The difference between tree distance and squared vector distance is minimized}}$$

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?

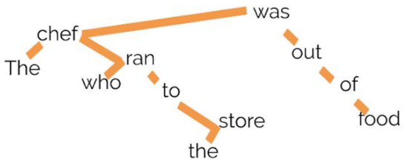
$$\arg\min_{B} \sum_{\ell \in \mathrm{PTB}} \frac{1}{|s^\ell|^2} \sum_{i,j} \left| d^\ell_{\mathrm{path}}(i,j) - \|B(h^\ell_i - h^\ell_j)\|_2^2 \right|$$

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*

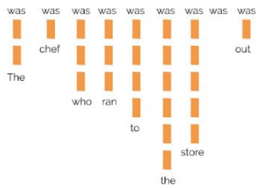


Does BERT encode edge directions
-> does there exist a *depth* transformation?

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

Find a single transformation *B*

such that over all sentences in PTB training



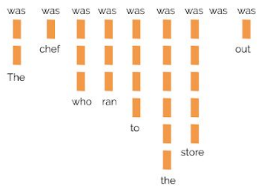
Does BERT encode edge directions -> does there exist a *depth* transformation?

$$\underset{B}{\arg\min} \sum_{\ell \in \text{PTB}} \frac{1}{|s^\ell|} \sum_i$$

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?

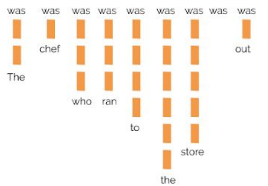
$$\underbrace{\arg\min_{B}}\ \underbrace{\sum_{\ell \in \mathrm{PTB}} \frac{1}{|s^\ell|}}\ \underbrace{\sum_{i}}\ \underbrace{\left|\mathrm{depth}^\ell(i) - \|Bh_i^\ell\|_2^2\right|}$$

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*

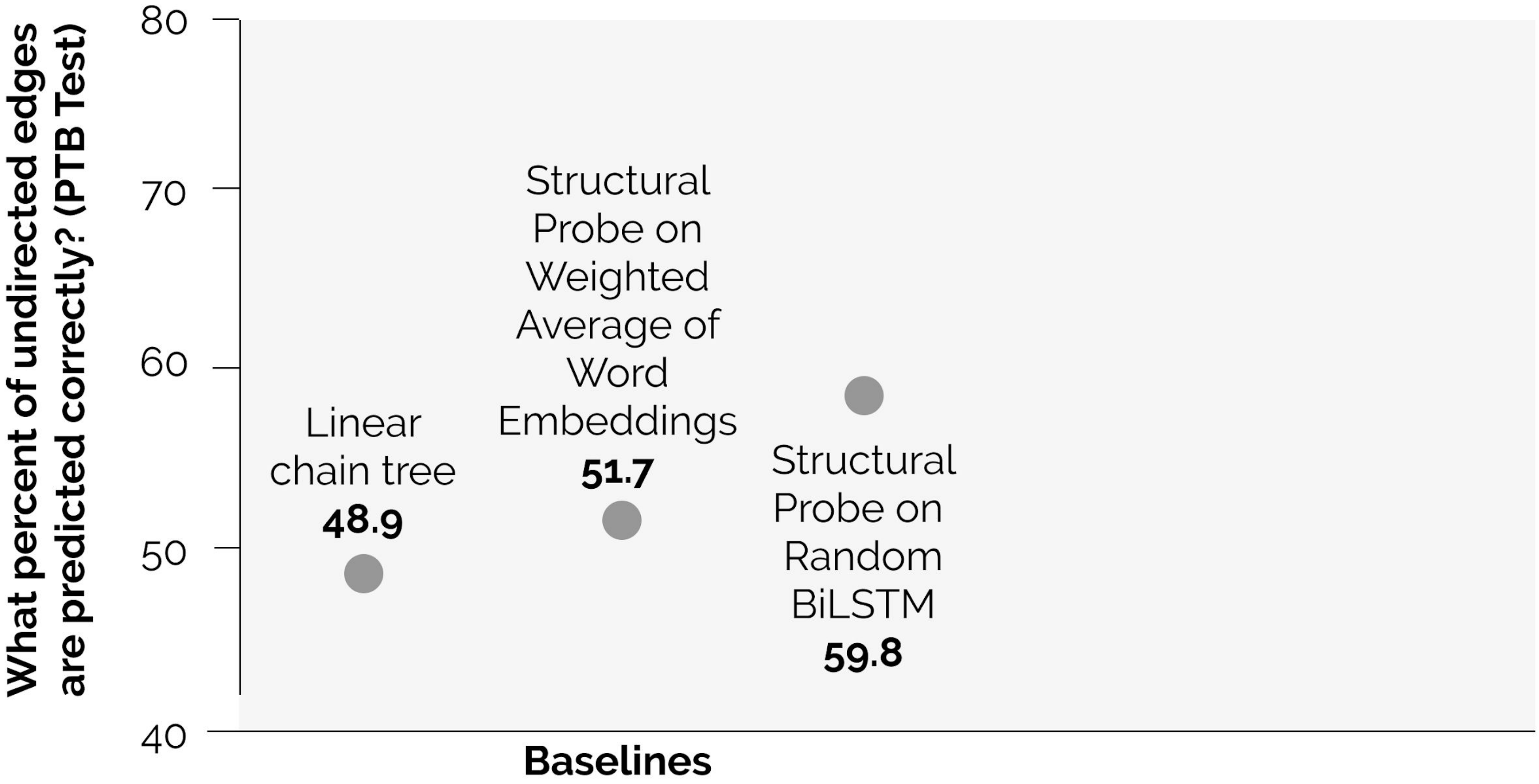


experiments & results

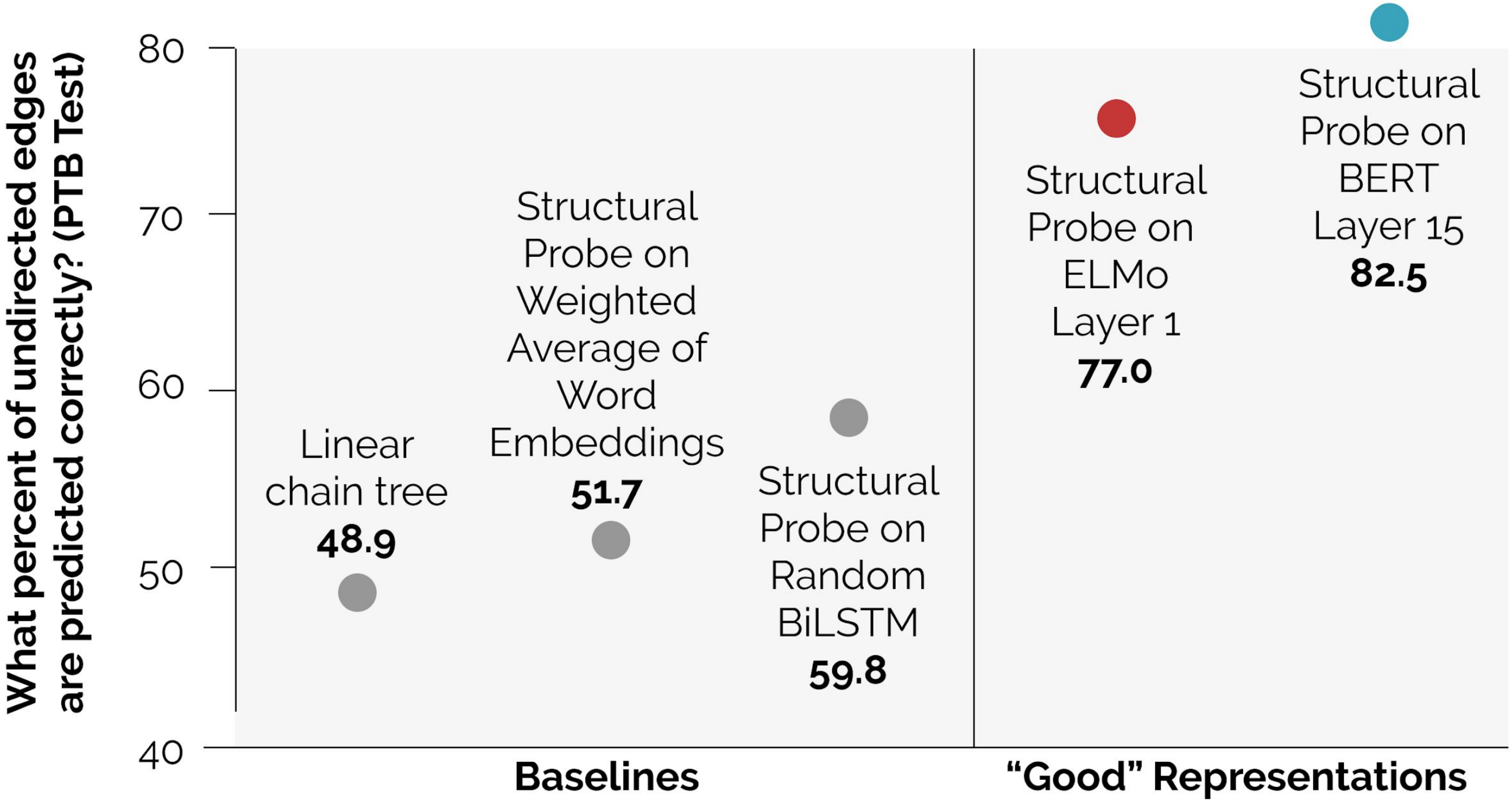
Evaluating ELMo, BERT, and baselines
Training structural probes on PTB train, evaluating on test.

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

Trees aren't well-encoded in baselines

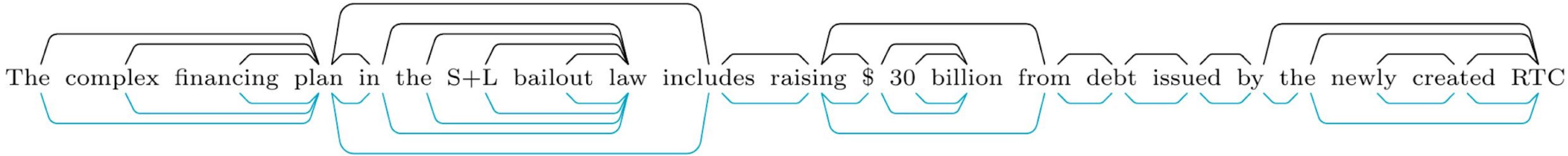


But they are in trained representations!



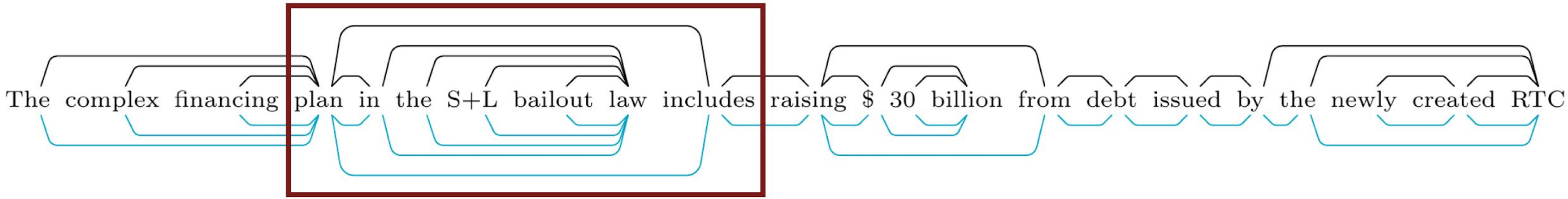
Trees from structural probe parse distances approximate parse trees pretty well!

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



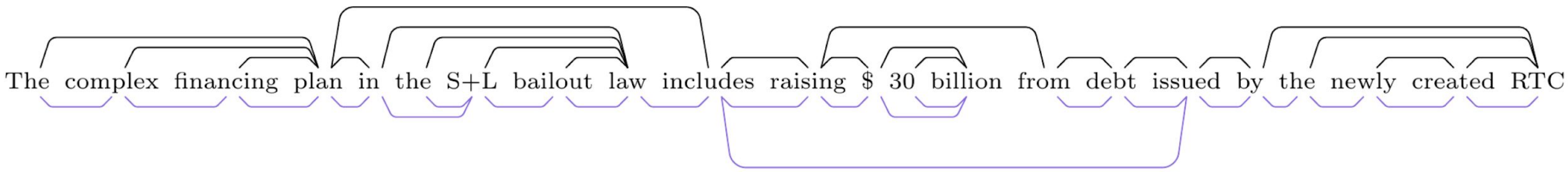
Trees from structural probe parse distances approximate parse trees pretty well!

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!

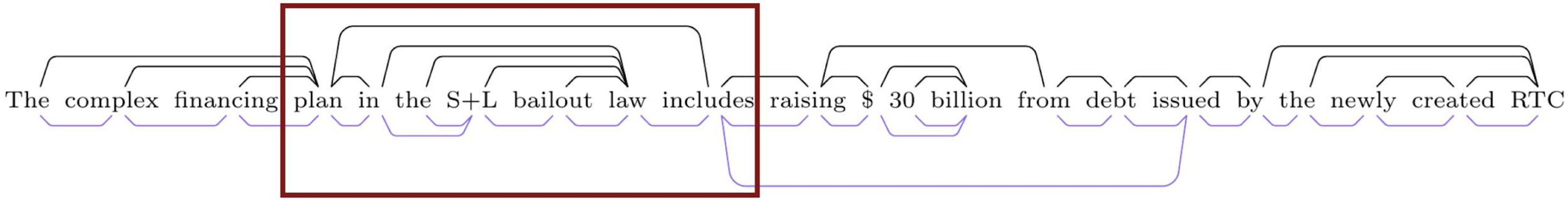
Black (above sentence): Human-annotated parse tree
Purple (below sentence): MST, structural probe on random-weights BiLSTM



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

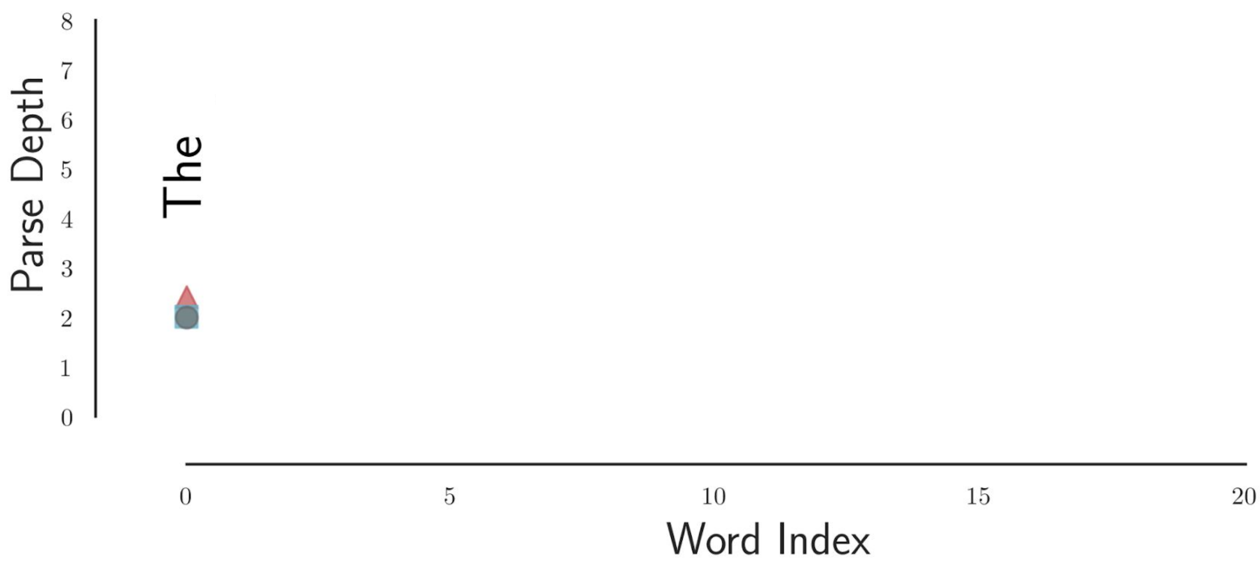
Black (above sentence): Human-annotated parse tree

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



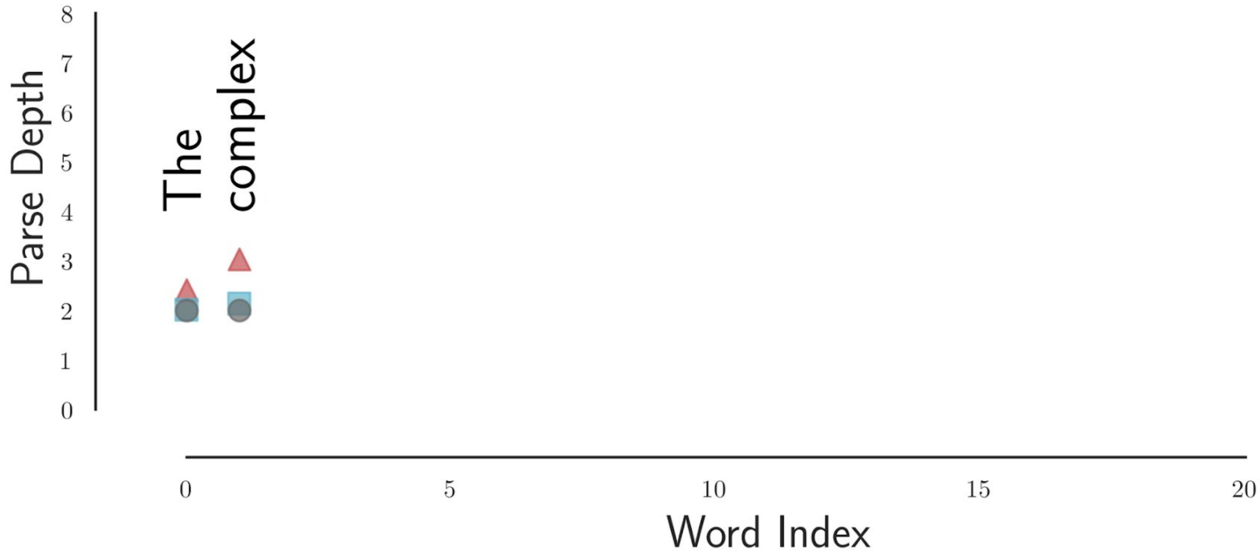
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



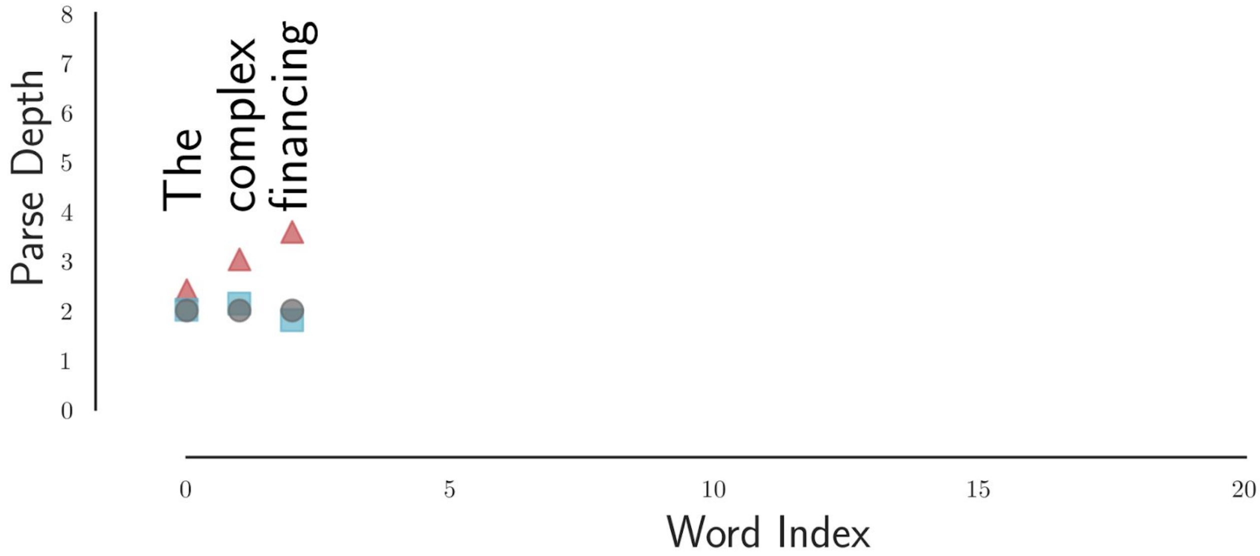
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



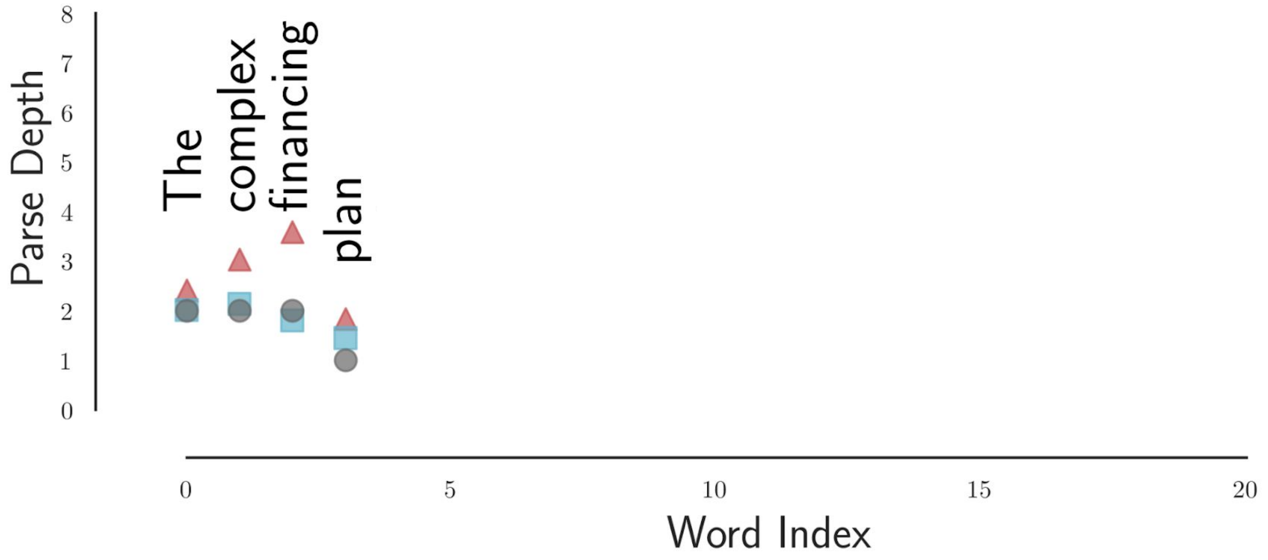
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



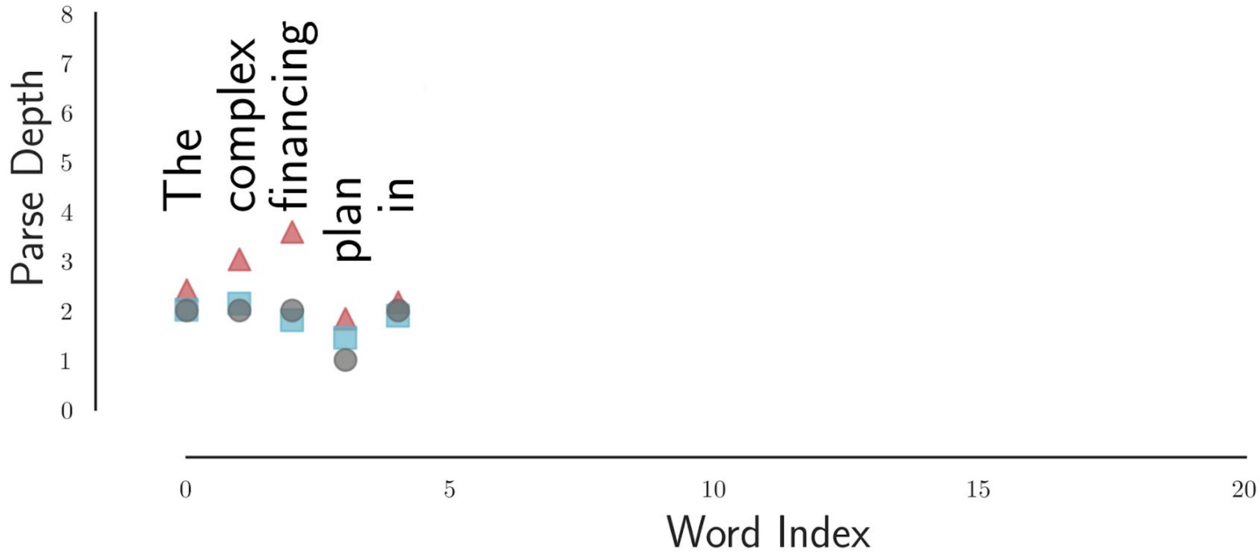
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



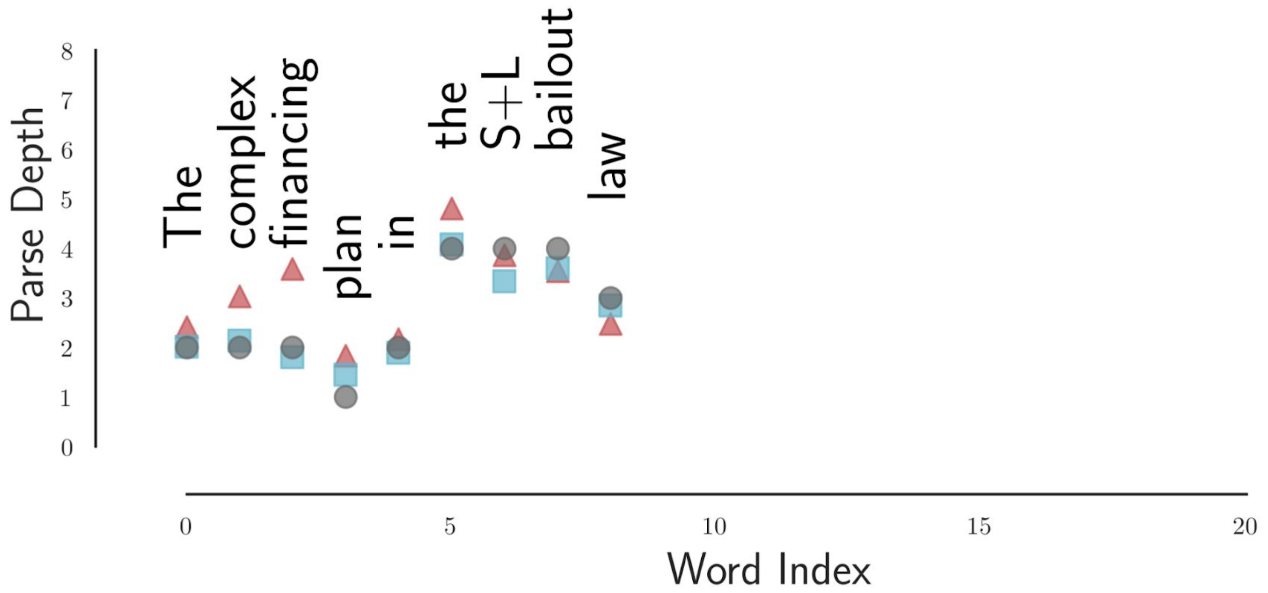
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



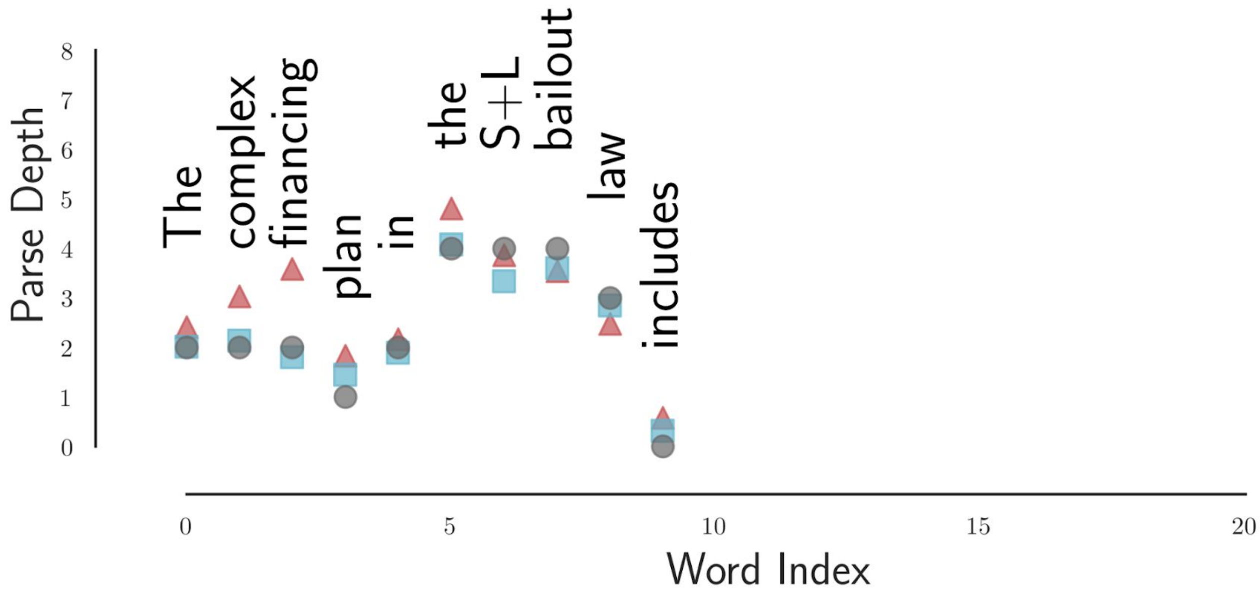
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



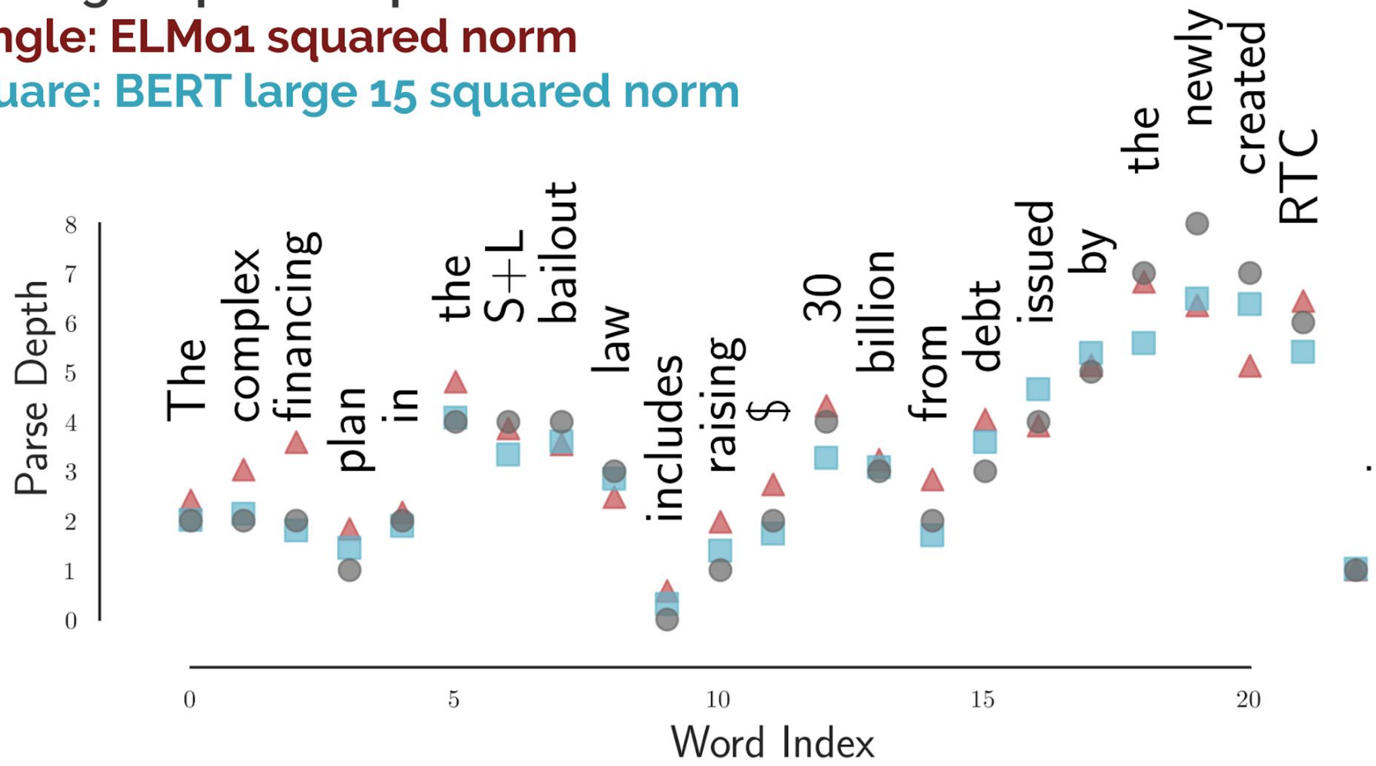
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



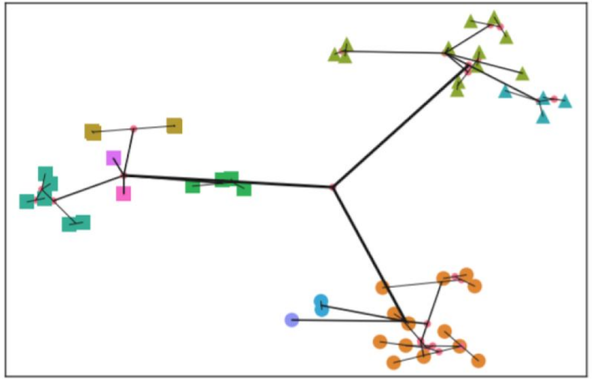
Predicted depths on BERT + ELMo reconstruct parse depths well!

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



Not just for language

The structural probe method has since been used to find **evolutionary trees** in **unsupervised representations of proteins**!



Transformer (trained)

Nodes are representations of protein families; distances are evolutionary history tree distances

Have a continuous space and wondering if discrete structures are embedded in it?

Try finding their distance metrics via a structural probe!

[Rives et al., 2019]

Summary, Musings, & Limitations

Structural probes show ELMo and BERT encode a surprising amount of syntax!

Structural probes give us intuitions about the geometric properties of contextual word representations, like we've had for word2vec and GloVe.

All probes use **supervision**, and we should be careful what fine-grained syntactic conclusions we make!

See *Saphra and Lopez (2019)* and *Lakretz et al. (2019)* for complementary methods!

The code is *super ready for you to jump in!*

https://github.com/john-hewitt/structural-probes