B.Tech Project (6th and 7th semester)

under the guidance of

Dr. A.K. Singh

A Structural Probe for Finding Syntax in Word Representations

Yash Malik
(18075065)
B.Tech. - 7th semester

08.12.2021

Overview

Problem

Finding a syntactic structure of a sentence in the learned encodings of neural networks.

Solution Presented

- A structural probe to recover the approximate parse tree of a sentence from its word embeddings.
- A case study on ELMo and BERT.

Results

The learned encoding of a sentence to a large extent includes the information found in the parse tree structures of sentences that have been proposed by linguists.

Motivation

Human Language

- In human languages, the meaning of a sentence is constructed by composing small chunks of words together with each other.
- The order in which these chunks are combined creates a tree-structured hierarchy.

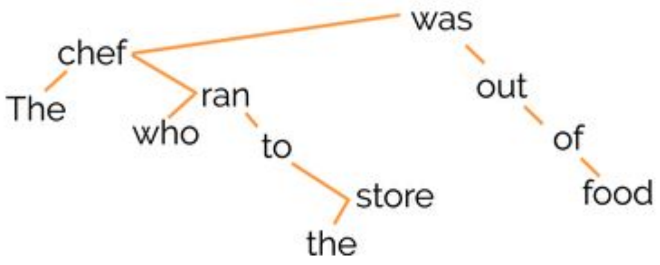
Neural Network encodings

- Neural networks used in NLP represent each word in the sentence as a real-valued vector, with no explicit representation of the parse tree
- Recent transformer based models (like the BERT), have shown great results on several language understanding tasks.

Motivation

Human Language

The chef who ran to the store was out of food.



Neural Networks

$$\begin{array}{ccccccccccc}
\text{The} & \text{chef} & \text{who} & \text{ran} & \text{to} & \text{the} & \text{store} & \text{was} & \text{out} & \text{of} & \text{food} \\
\begin{bmatrix}.4\\-.2\\.3\end{bmatrix} & \begin{bmatrix}.1\\.9\\-.2\end{bmatrix} & \begin{bmatrix}.3\\-.4\\.2\end{bmatrix} & \begin{bmatrix}.7\\-.4\\0\end{bmatrix} & \begin{bmatrix}.4\\0\\-.5\end{bmatrix} & \begin{bmatrix}.1\\-.6\\.2\end{bmatrix} & \begin{bmatrix}.3\\.1\\-.6\end{bmatrix} & \begin{bmatrix}.1\\.9\\-.8\end{bmatrix} & \begin{bmatrix}.3\\.1\\.8\end{bmatrix} & \begin{bmatrix}-.8\\.3\\-.6\end{bmatrix} & \begin{bmatrix}0\\.7\\-.9\end{bmatrix}
\end{array}$$

Instead of three-dimensional vectors, they’re more like one thousand dimensions.

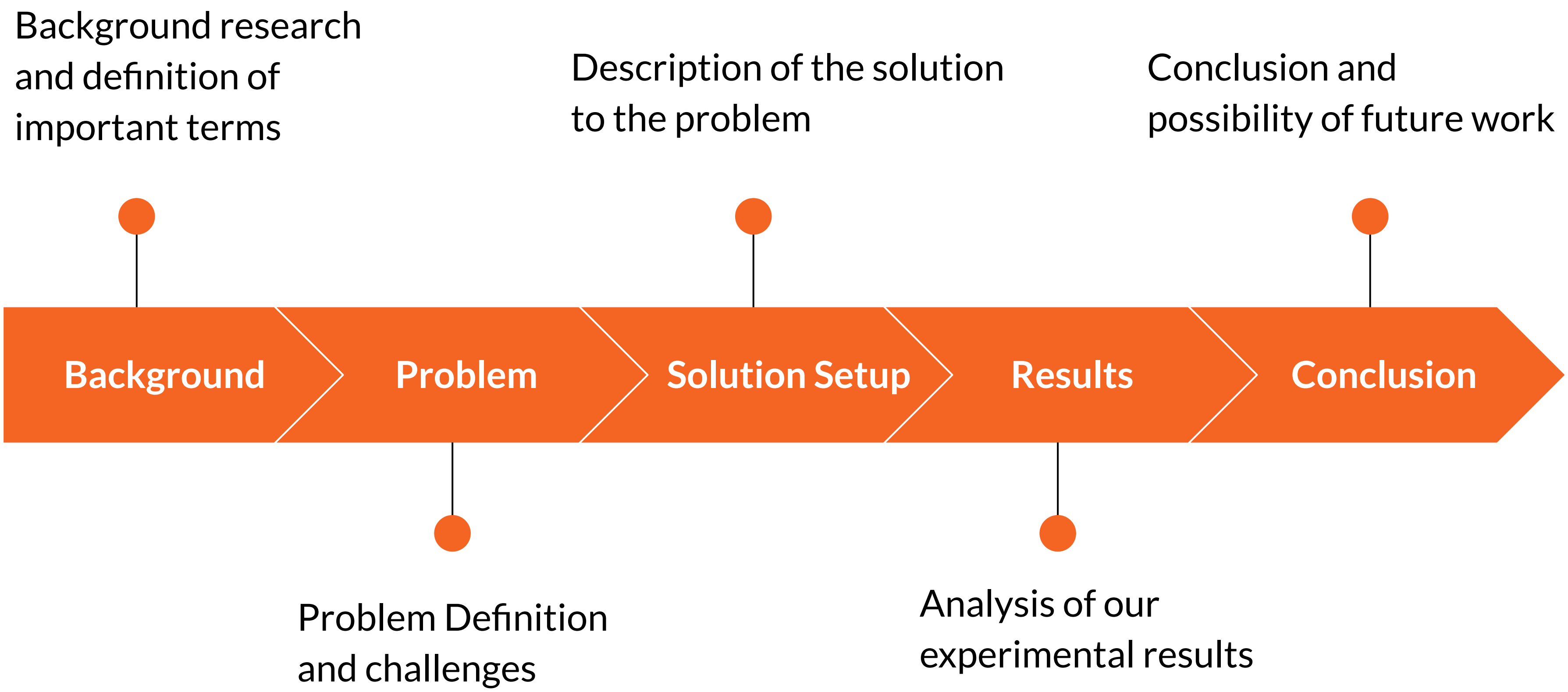
Motivation

It has been suggested that strong contextual models **implicitly**, softly perform some of the tasks we think are important for true language understanding, e.g., syntax, coreference, question answering.

Motivating Question

Human languages have tree structures, numerical machines represent it with vector representations, are these views of language reconcilable?

Structure of Presentation



Background

Background research and definition of important terms

Problem

Problem Definition and challenges

Solution Setup

Description of the solution to the problem

Results

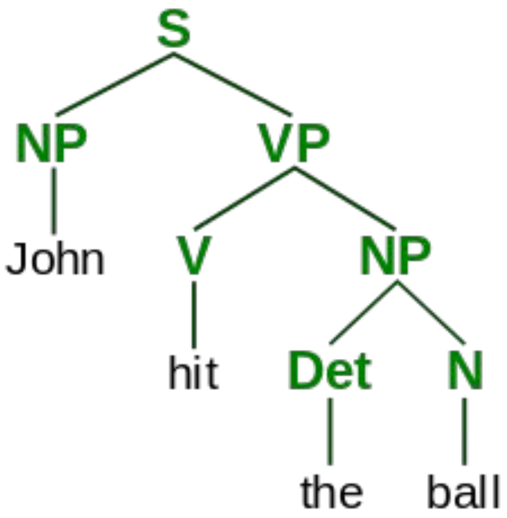
Analysis of our experimental results

Conclusion

Conclusion and possibility of future work

Background Knowledge

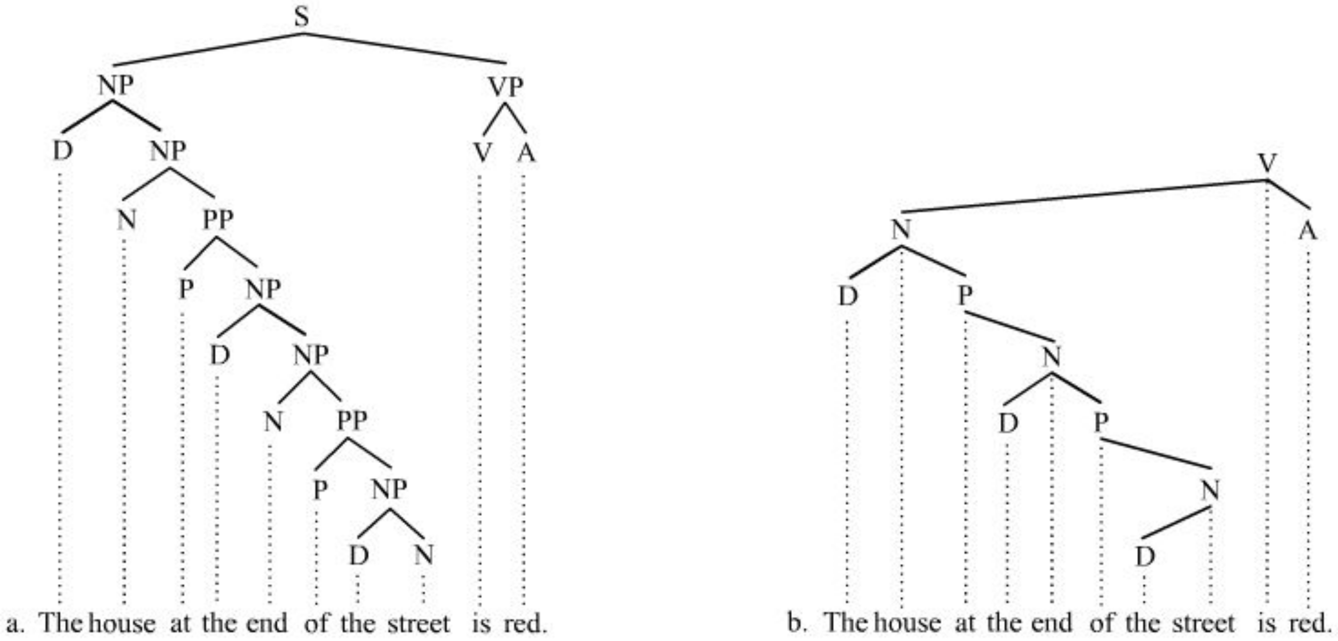
Background Knowledge



A simple parse tree (Image source Parse tree)

- **Syntax** : In linguistics, syntax is the set of rules, principles, and processes that govern the structure of sentences (sentence structure) in a given language, usually including word order.
- **Parse Tree** : A parse tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar.

Syntactic structure in a sentence


S
NP
VP
D
NP
V
A
N
PP
P
NP
D
NP
N
PP
P
NP
D
N
a. The house at the end of the street is red.
V
N
A
D
P
N
D
P
N
D
b. The house at the end of the street is red.

Constituency structure

Dependency structure

In this project we use the latter, which is dominant in computational linguistics.

Background Research

Traditional models

- Bag of words (or n = 1, unigram)
- N-gram
- Similar variants (including/excluding stop words)

Neural Networks

- Vector of real numbers

```
John likes to watch movies.
Mary likes movies too. Mary
also likes to watch
football games.
```

```
BoW =
{"John":1,"likes":3,"to":2,"wa
tch":2,"movies":2,"Mary":2,"to
o":1,"also":1,"football":1,"ga
mes":1};
```

```
[
    "John likes",
    "likes to",
    "to watch",
    "watch movies",
    "Mary likes",
    "likes movies",
    "movies too",
]
```

Background Research

Word Embeddings

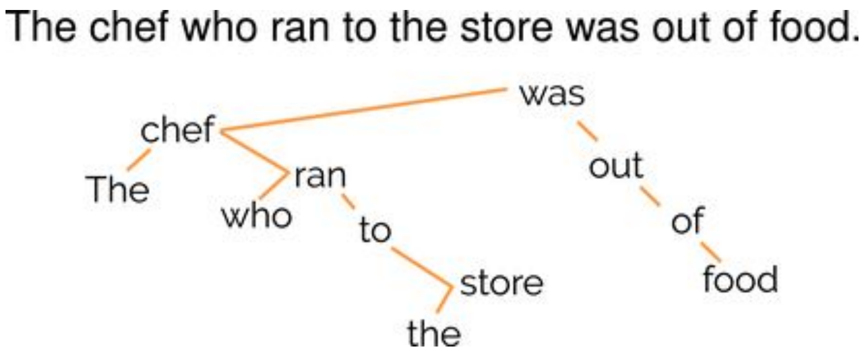
- Modern day word embeddings like in BERT and ELMo are very powerful and context-sensitive, unlike the traditional models.
- They produce different representations for words that share the same spelling but have different meanings (homonyms) such as "bank" in "river bank" and "bank balance".

out
store
the
chef
food
ran
to
of
who
was
The

Background Research

Neural Networks

- Very effective for several language processing tasks, but,
- At the cost of interpretability (opacity)
- Recent transformer based models (like BERT) have consistently performed at par or better than humans in several natural language understanding tasks.



The neural models capture the information that, the chef is out of food not store, even though linearly, a part of the sentence reads “*the store was out of food*”.

Background Research

BERT

- Bidirectional Encoder Representations from Transformers (BERT) is a Transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google.
- When BERT was published, it achieved state-of-the-art performance on a number of natural language understanding tasks.

ELMo

- Embeddings from Language Model (ELMo) is a word embedding method for representing a sequence of words as a corresponding sequence of vectors.
- Like BERT, ELMo embeddings are context-sensitive, producing different representations for words that share the same spelling but have different meanings (homonyms).

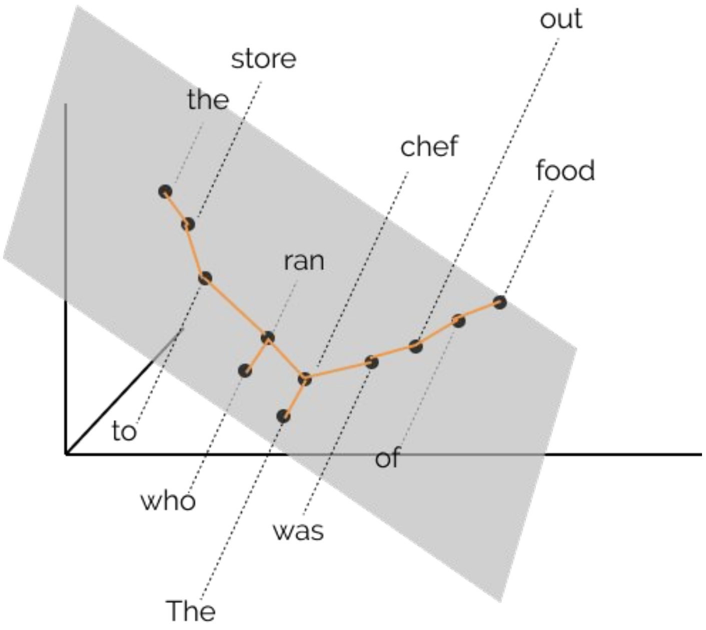
Problem Description

Problem Definition

Statement

Design a simple method for testing whether a neural network embeds each sentence's dependency parse tree in its contextual word representations – a structural hypothesis.

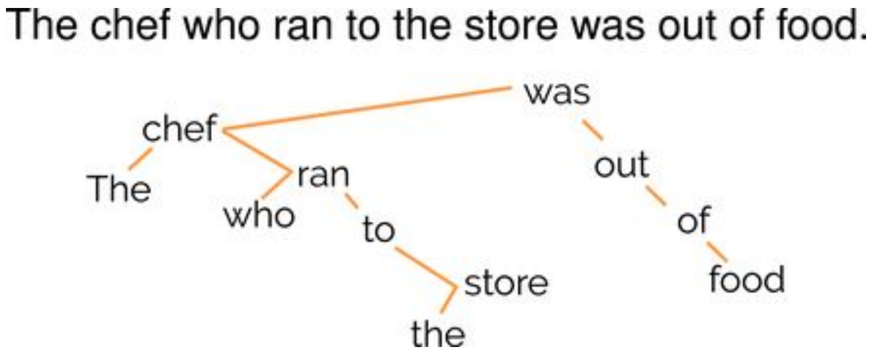
★ The probe should not simply “learn to parse” on top of any informative representation.

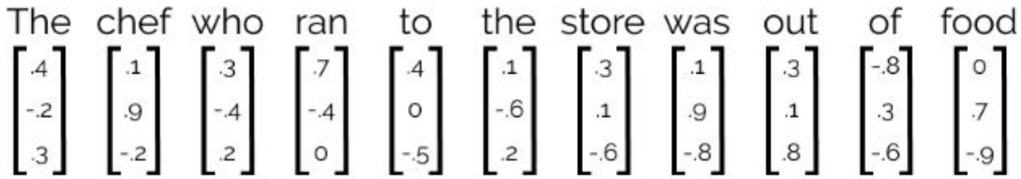


A visual intuition for what we want to achieve

Challenges associated with the task

- parse tree is a “discrete” structure
- neural networks encode the sentence as a sequence of “continuous” real valued vectors
- key difficulty is in determining whether the parse tree, a discrete structure, is encoded in the sequence of continuous vectors



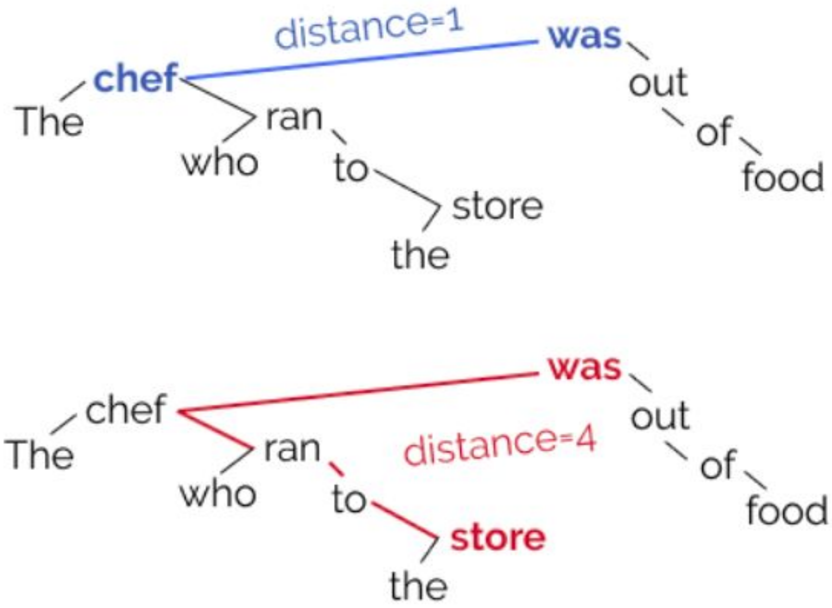


$$\begin{array}{ccccccccccc}
\text{The} & \text{chef} & \text{who} & \text{ran} & \text{to} & \text{the} & \text{store} & \text{was} & \text{out} & \text{of} & \text{food} \\
\begin{bmatrix} .4 \\ -.2 \\ .3 \end{bmatrix} & \begin{bmatrix} .1 \\ .9 \\ -.2 \end{bmatrix} & \begin{bmatrix} .3 \\ -.4 \\ .2 \end{bmatrix} & \begin{bmatrix} .7 \\ -.4 \\ 0 \end{bmatrix} & \begin{bmatrix} .4 \\ 0 \\ -.5 \end{bmatrix} & \begin{bmatrix} .1 \\ -.6 \\ .2 \end{bmatrix} & \begin{bmatrix} .3 \\ .1 \\ -.6 \end{bmatrix} & \begin{bmatrix} .1 \\ .9 \\ -.8 \end{bmatrix} & \begin{bmatrix} .3 \\ .1 \\ .8 \end{bmatrix} & \begin{bmatrix} -.8 \\ .3 \\ -.6 \end{bmatrix} & \begin{bmatrix} 0 \\ .7 \\ -.9 \end{bmatrix}
\end{array}$$

Approach to the Solution

Notable Points

- Vector spaces and graphs both have natural distance metrics.
- For a parse tree, we have the path metric, d(w_i, w_j), which is the number of edges in the path between the two words in the tree.



Notable Points

- With all N^2 distances for a sentence, one can reconstruct the (undirected) parse tree simply by recognizing that all words with distance 1 are neighbors in the tree.
- This is equivalent to calculating the Minimum Spanning Tree (MST) of the completely connected graph formed by these N words.
- Thus, **embedding the tree reduces to embedding the distance metric defined by the tree**.

The Syntax Distance ≠ Distance in vector space

Intuitively, if a neural network embeds parse trees, it likely will not use its entire representation space to do so, since it needs to encode many kinds of information.

- As neural networks have to encode a lot of information in the hidden states, not just syntax, so distance on the whole vector may not make sense.
- We need to consider some **projection of the vector space** embedded by the neural network, which can account solely for the distance in the syntax parse tree.

Syntax Distance

- Distances between words before transformation by **B** aren't indicative of the tree

- After the linear transformation, however, taking a minimum spanning tree on the distances recovers the tree

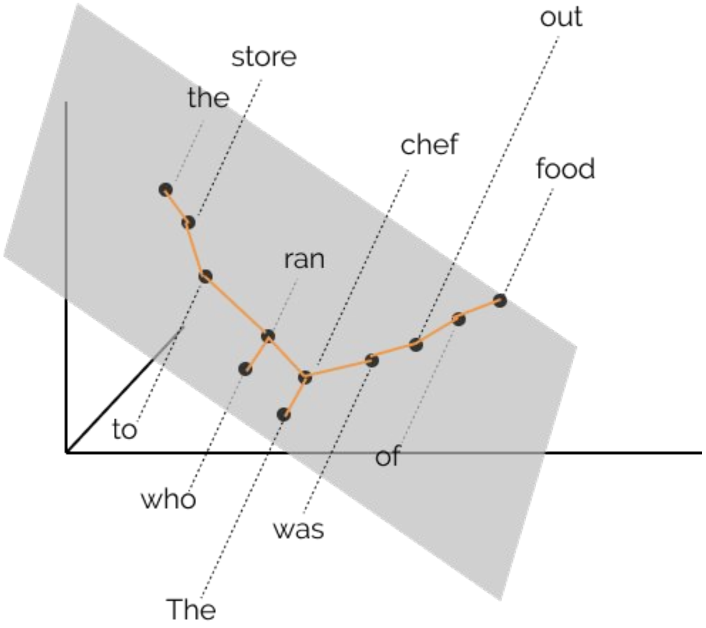


Image Source [Finding Syntax with Structural Probes]

The syntax distance hypothesis

- There exists a linear transformation **B** of the word representation space under which vector distances encodes parse trees.
- The distances we pointed out earlier between *chef*, *store* and *was*, can be visualized in a vector space as follows, where $B \in \mathbb{R}^{2\times 3}$ maps 3-dimensional word representation to a 2-dimensional space encoding syntax.

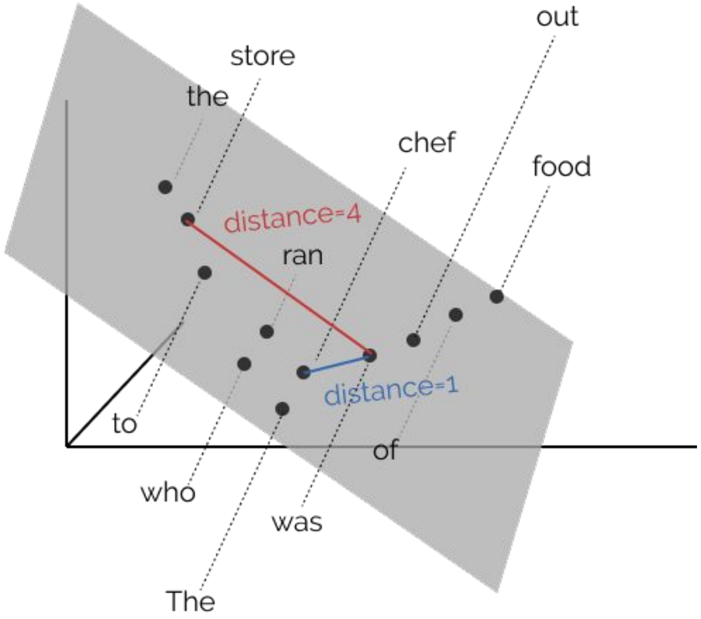


Image Source [Finding Syntax with Structural Probes]

Syntax Distance

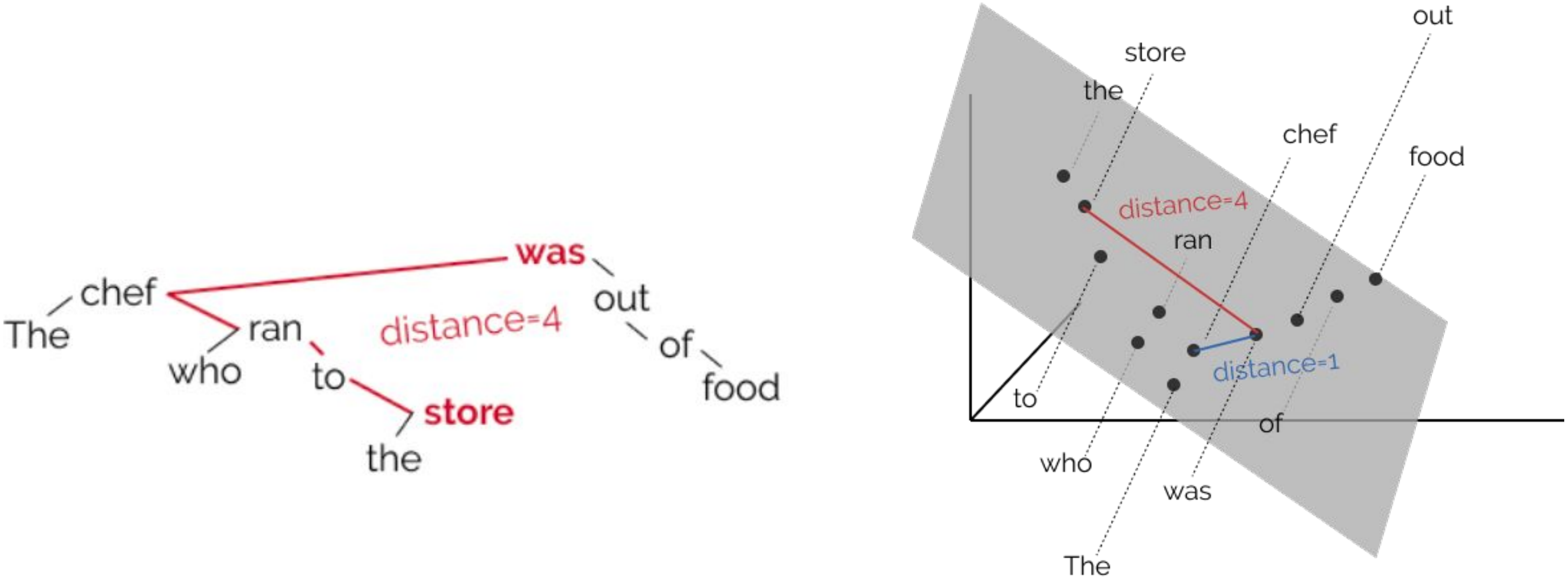


Image Source Finding Syntax with Structural Probes

Structural Probe

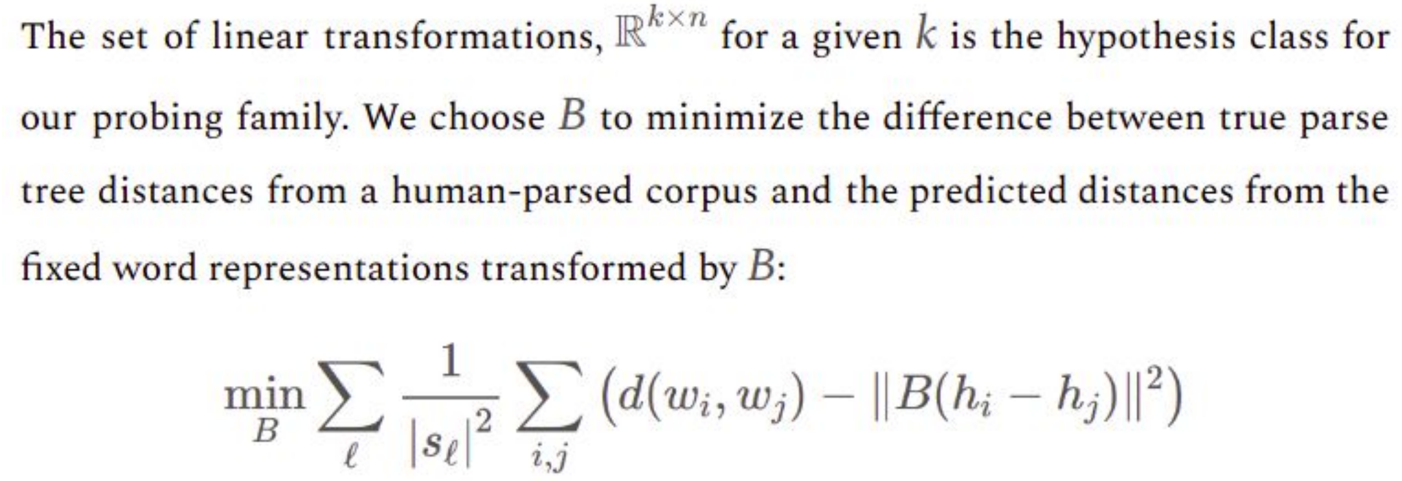
- the probe learns a linear transformation of a word representation space such that the transformed space embeds parse trees across all sentences.
- can be interpreted as finding the component of the representation space that is used to encode syntax.
- equivalently, it is finding the distance on the original space that best fits the tree metrics.

$$\|h_i - h_j\|_B^2 = (B(h_i - h_j))^T(B(h_i - h_j))$$

$$A = B^T B$$

$$\|h_i - h_j\|_A^2 = (h_i - h_j)^T A(h_i - h_j)$$

1. Finding a parse tree-encoding distance metric

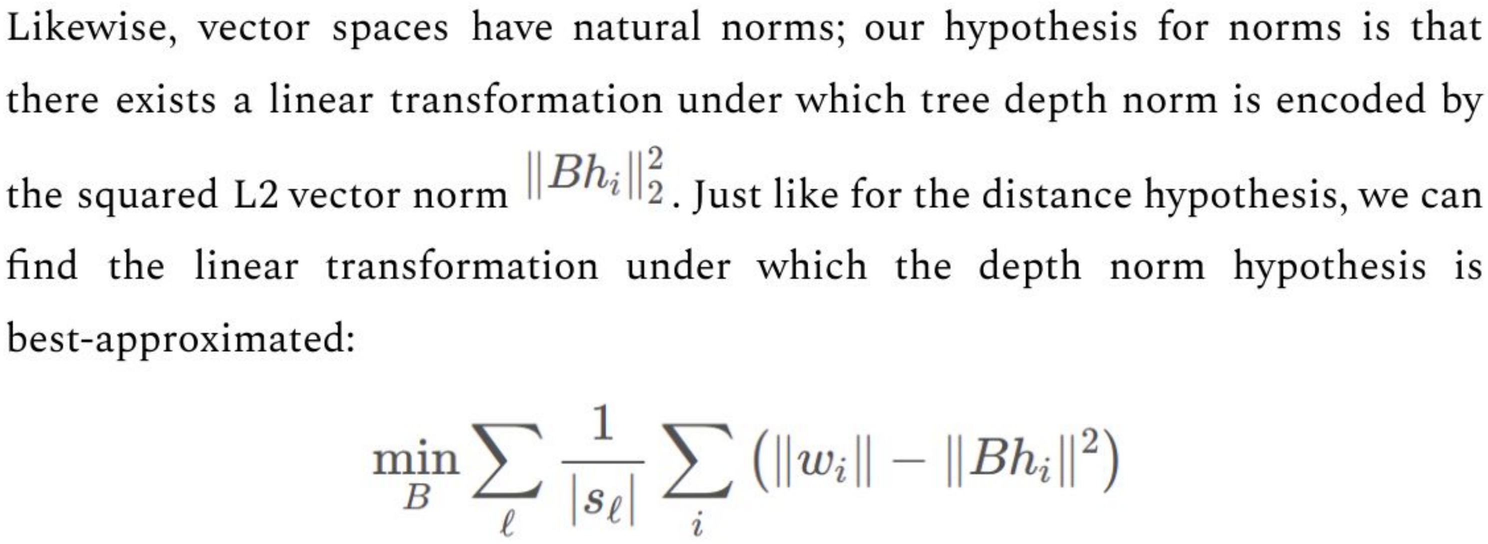


The set of linear transformations, $\mathbb{R}^{k \times n}$ for a given k is the hypothesis class for our probing family. We choose B to minimize the difference between true parse tree distances from a human-parsed corpus and the predicted distances from the fixed word representations transformed by B:

$$\min_B \sum_\ell \frac{1}{|s_\ell|^2} \sum_{i,j} \left(d(w_i, w_j) - \|B(h_i - h_j)\|^2 \right)$$

l indexes the sentences s_l in the corpus, and $\frac{1}{|s_\ell|^2}$ normalizes for the number of pairs of words in each sentence

2. Finding a parse tree depth-encoding norm



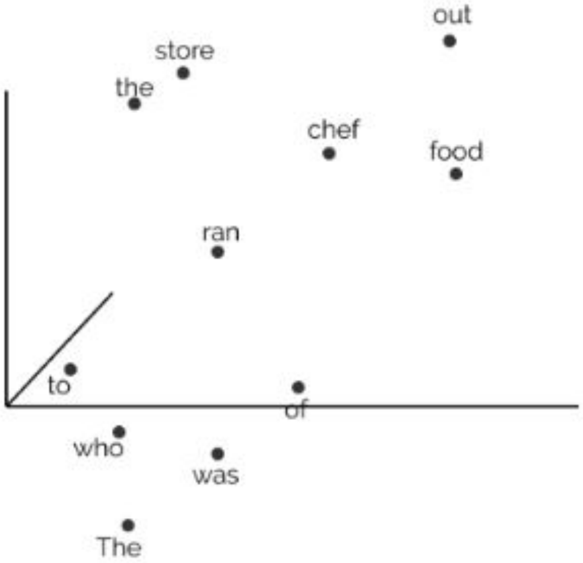
Likewise, vector spaces have natural norms; our hypothesis for norms is that there exists a linear transformation under which tree depth norm is encoded by the squared L2 vector norm $\|Bh_i\|_2^2$. Just like for the distance hypothesis, we can find the linear transformation under which the depth norm hypothesis is best-approximated:

$$\min_B \sum_\ell \frac{1}{|s_\ell|} \sum_i \left(\|w_i\| - \|Bh_i\|^2 \right)$$

Overview of the entire process in structural probe

Step - 1

Get the word embeddings from the models



Each of the words of the sentence *The chef who ran to the store was out of food* is internally represented in context as a vector.

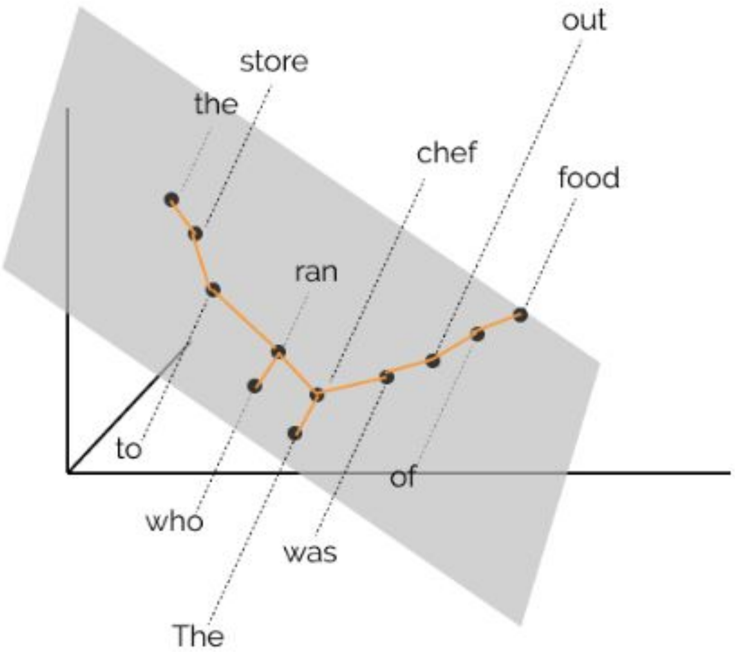
Image Source Finding Syntax with Structural Probes

Overview of the entire process in structural probe

Step - 2

Using the probes, learn the linear transformations of the space.

1. Approximates the tree distance
2. Approximates the tree depth



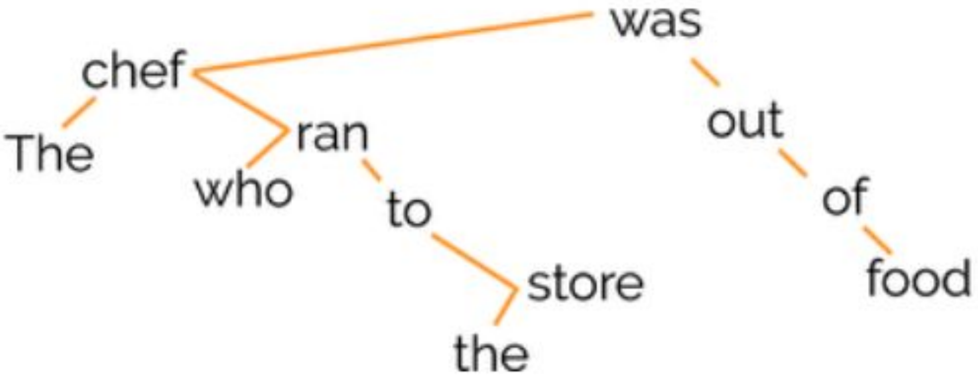
A structural probe finds a linear transform of that space under which squared L_2 distance between vectors best reconstructs tree path distance between words.

Image Source [Finding Syntax with Structural Probes]

Overview of the entire process in structural probe

Step - 3

Generate the minimum spanning tree, and use the dept to define the hierarchy.



In fact, the tree can be approximately recovered by taking a minimum spanning tree in the latent syntax space.

Image Source [Finding Syntax with Structural Probes]

Experiment Setup

Representation Models

- ELMo, $\text{BERT}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$
- We use the 5.5B-word pre-trained ELMo weights for all

  ELMo representations.
- All ELMo and BERT-large layers are dimensionality 1024;

  BERT-base layers are dimensionality 768.

Baselines

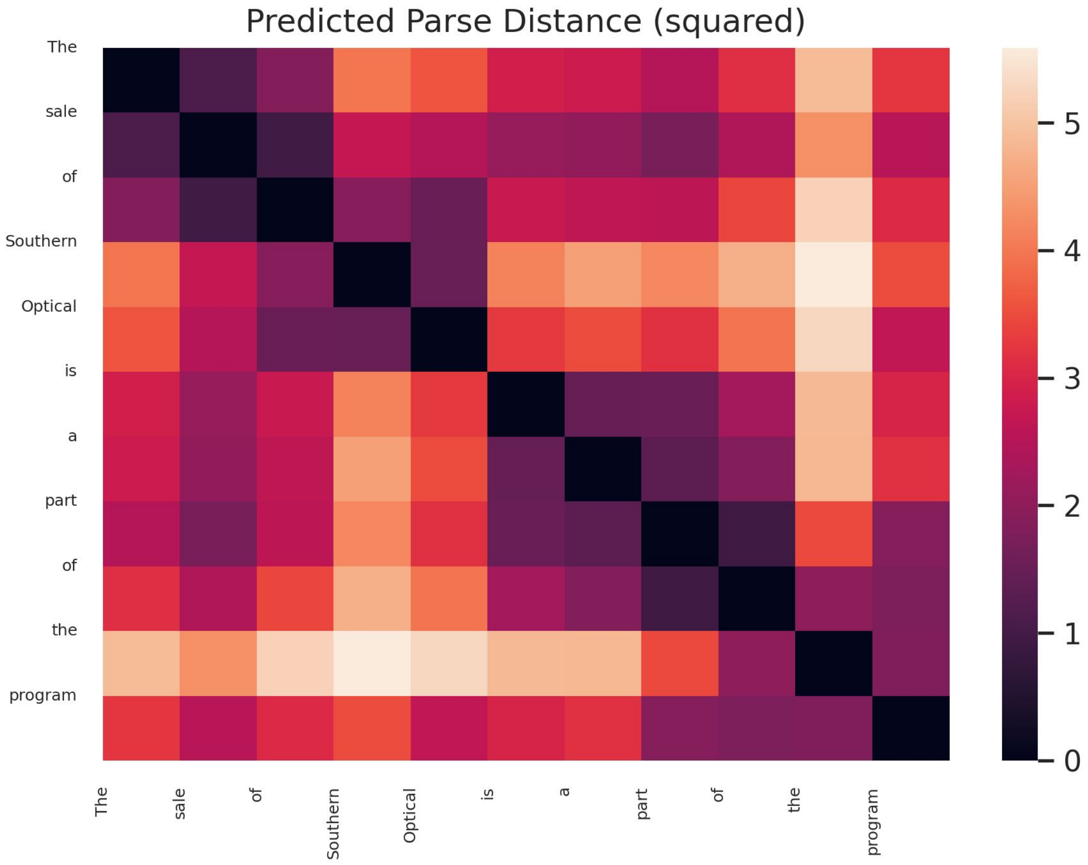
The baselines should encode features useful for training parser, but not capable of parsing themselves, to provide points of comparison against ELMo and BERT.

- **LINEAR** : The tree resulting from the assumption that English parse trees form a left-to-right chain. A model that encodes the positions of words should be able to meet this baseline.
- **ELMO0** : Strong character-level word embeddings with no contextual information. As these representations lack even position information, we should be completely unable to find syntax trees embedded.
- **DECAY0** : Assigns each word a weighted average of all ELMO0 embeddings in the sentence. The weight assigned to each word decays exponentially as $1/2^d$, where d is the linear distance between the words.
- **PROJ0** : Contextualizes the ELMO0 embeddings with a randomly initialized BiLSTM layer of dimensionality identical to ELMo (1024), a surprisingly strong baseline for contextualization (Conneau et al., 2018).

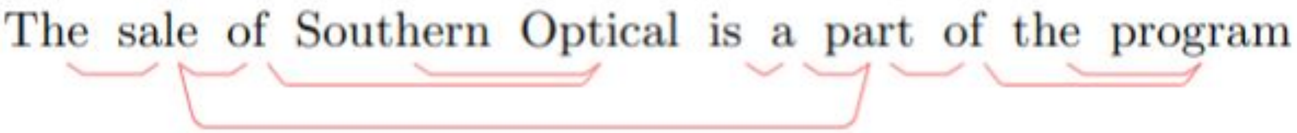
Data and Evaluation Metrics

- Penn Treebank (Marcus et al., 1993), with no pre-processing.
- Tree distance evaluation metrics
  - **Undirected unlabelled attachment score (UUAS)** - percent of undirected edges placed correctly—against the gold tree.
  - **Spearman correlation (DSpr.)** - between true and predicted distances for each word in each sentence.
- Tree depth evaluation metrics
  - **Norm Spearman (NSpr.)** - between the true depth ordering and the predicted ordering
  - **Root%** - ability to identify the root of the sentence as the least deep

Results

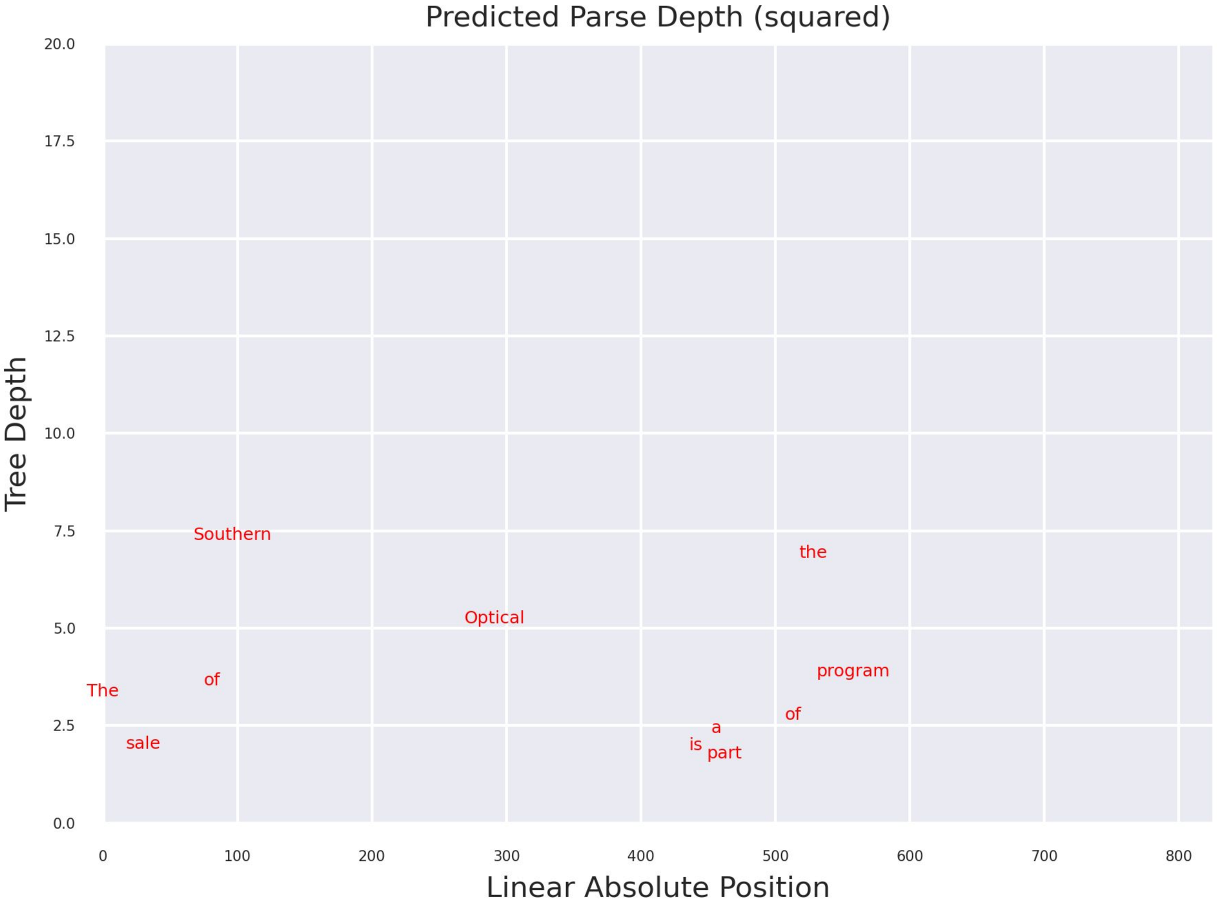


Example Results



The sale of Southern Optical is a part of the program

Example Results



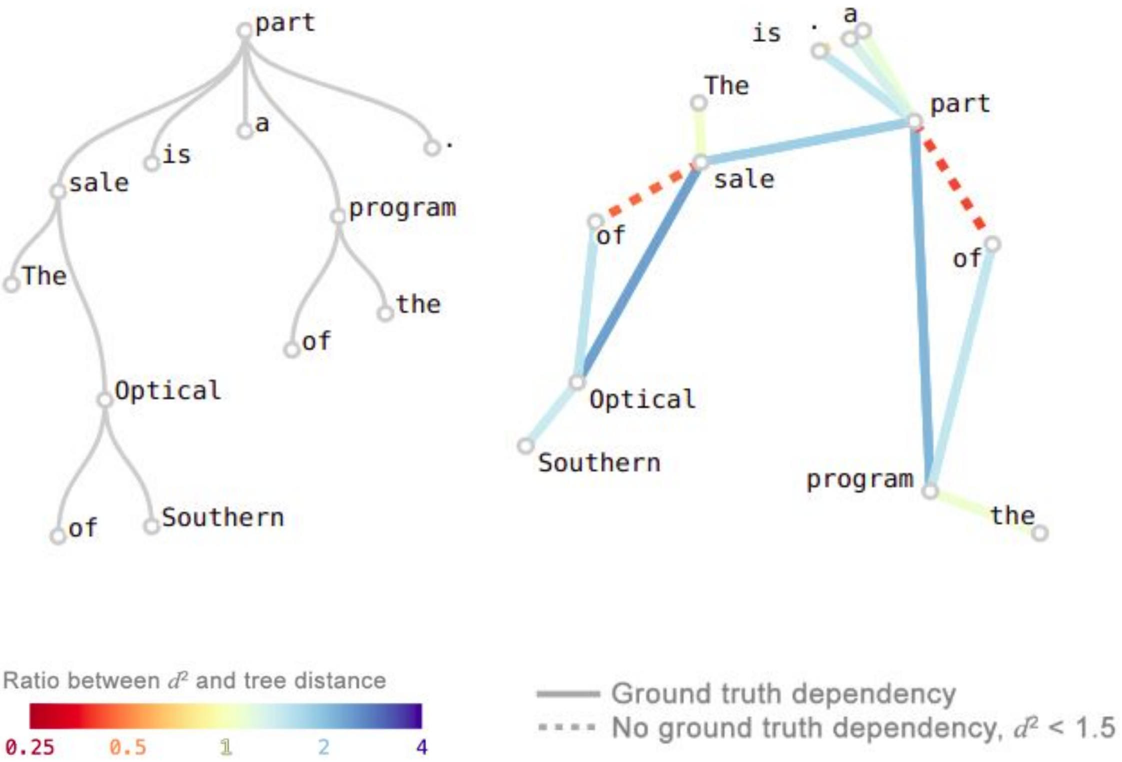
Example Results

Example Results

The left image is a traditional parse tree view, but the vertical length of each branch represents embedding distance.

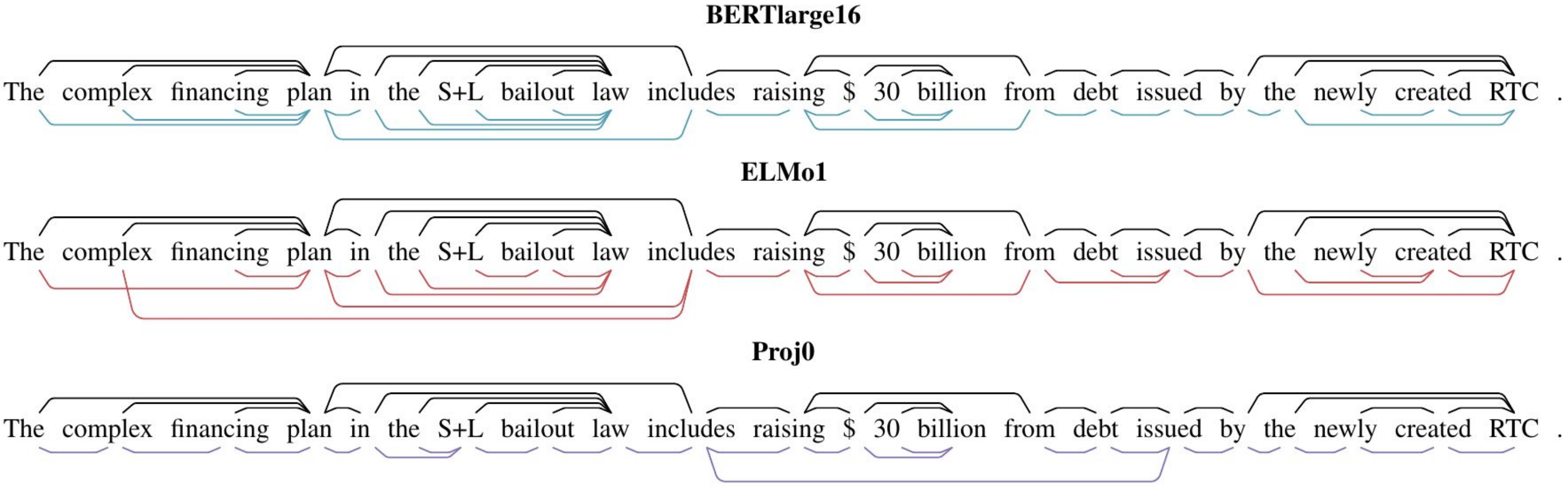
Right image: projection of context embeddings, where color shows deviation from expected distance.

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



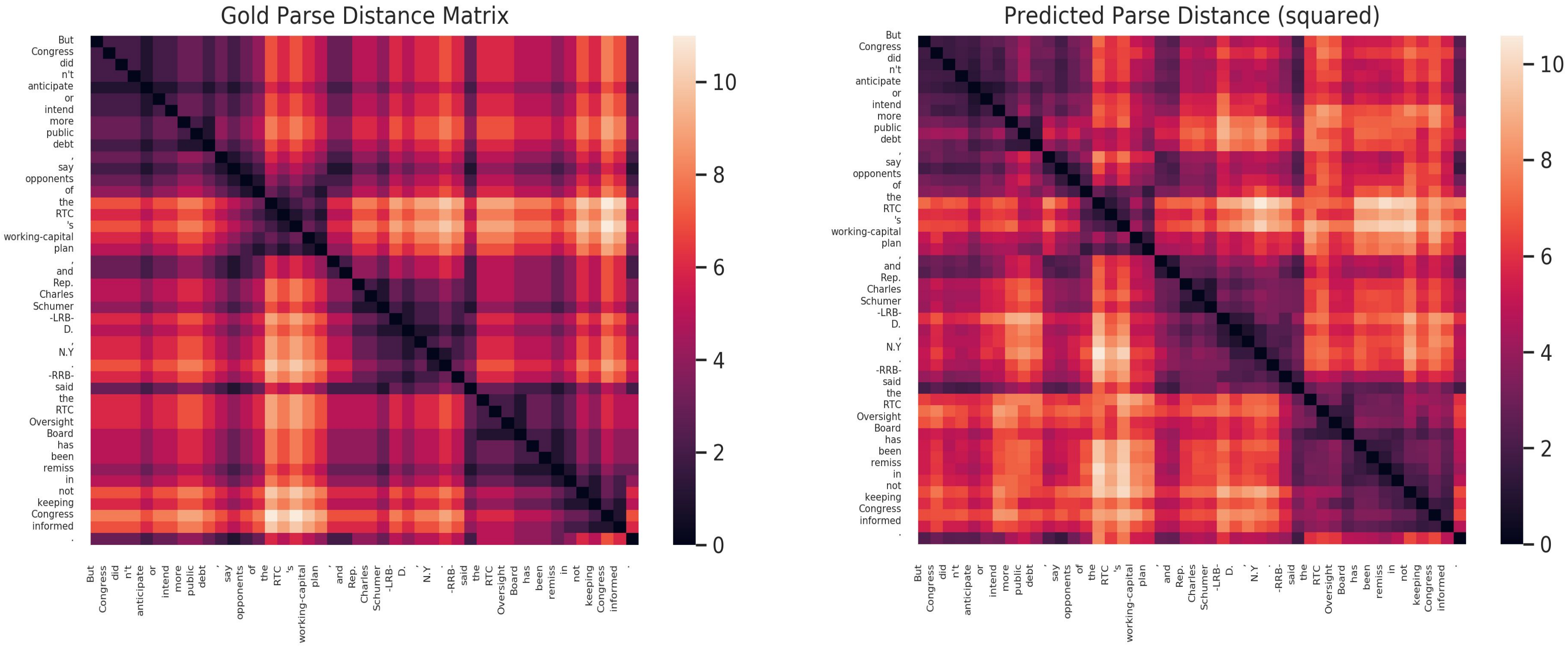
– Experimental Results

Gold parse trees (black, above the sentences) are shown along with the minimum spanning trees of predicted distance metrics.


BERTlarge16
The complex financing plan in the S+L bailout law includes raising $ 30 billion from debt issued by the newly created RTC .
ELMo1
The complex financing plan in the S+L bailout law includes raising $ 30 billion from debt issued by the newly created RTC .
Proj0
The complex financing plan in the S+L bailout law includes raising $ 30 billion from debt issued by the newly created RTC .

Minimum spanning trees resulting from the structural probes on BERT, ELMo and a random control representation Proj0 compared to the human annotated parse tree.

– Experimental Results



Gold Parse Distance Matrix

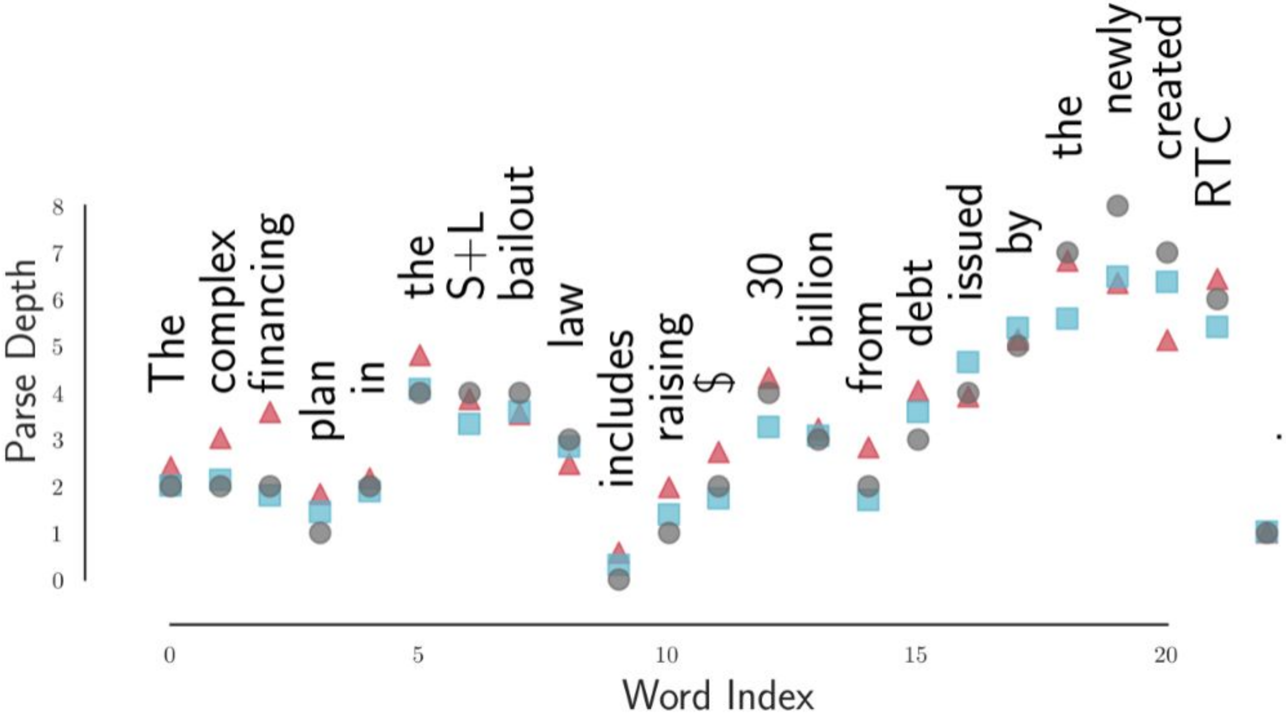
Predicted Parse Distance (squared)

Distance matrix, which visualizes all pairs of distances between words in a sentence (Gold and BERT$_{\text{LARGE}}$16)

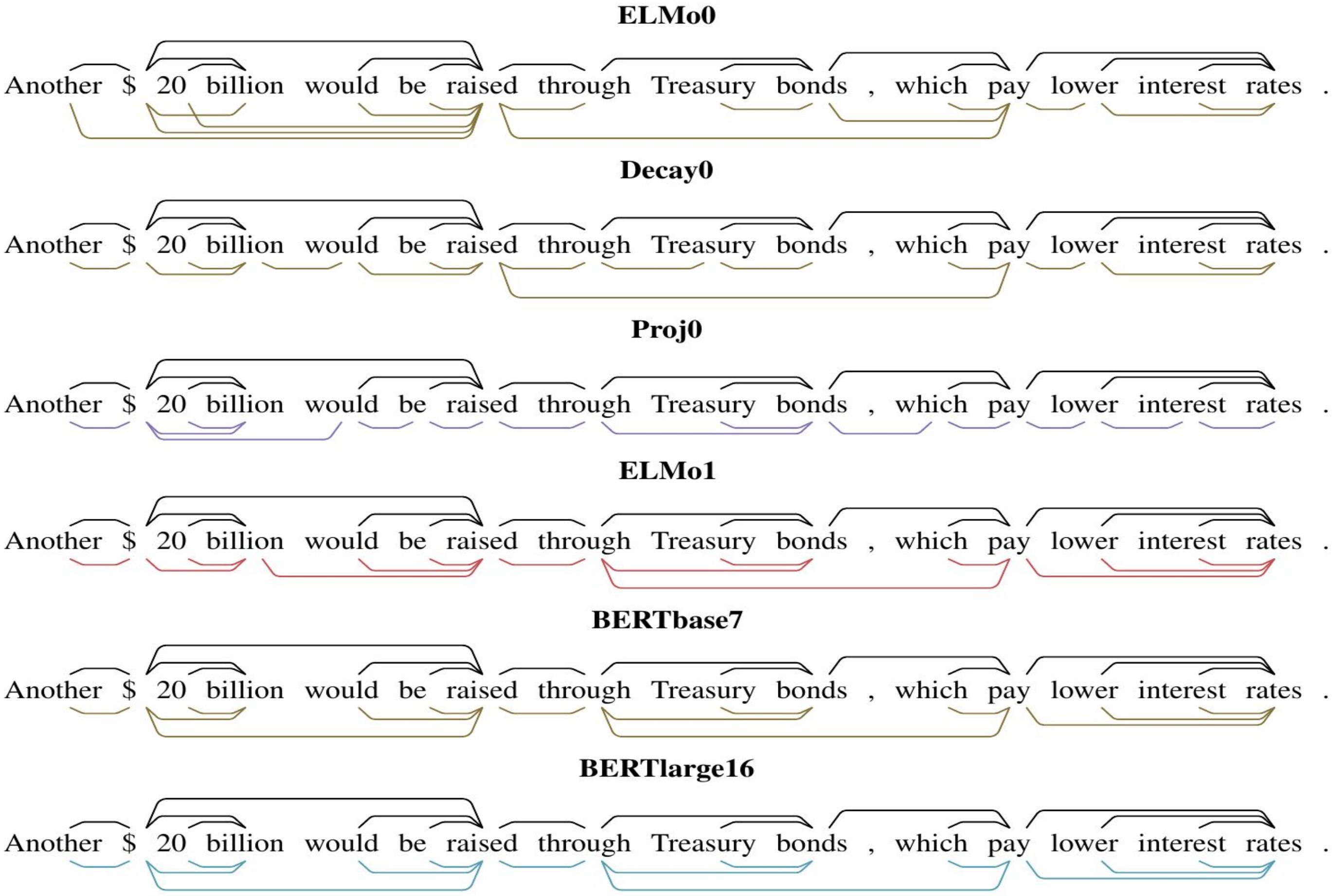
– Experimental Results

Tree depth according to the

1. gold tree (black, circle)
2. the norm probes
   a. ELMo1 (red, triangle)
   b. $\text{BERT}_{\text{LARGE}}16$ (blue, square)



– Experimental Results


ELMo0
Another $ 20 billion would be raised through Treasury bonds , which pay lower interest rates .
Decay0
Another $ 20 billion would be raised through Treasury bonds , which pay lower interest rates .
Proj0
Another $ 20 billion would be raised through Treasury bonds , which pay lower interest rates .
ELMo1
Another $ 20 billion would be raised through Treasury bonds , which pay lower interest rates .
BERTbase7
Another $ 20 billion would be raised through Treasury bonds , which pay lower interest rates .
BERTlarge16
Another $ 20 billion would be raised through Treasury bonds , which pay lower interest rates .

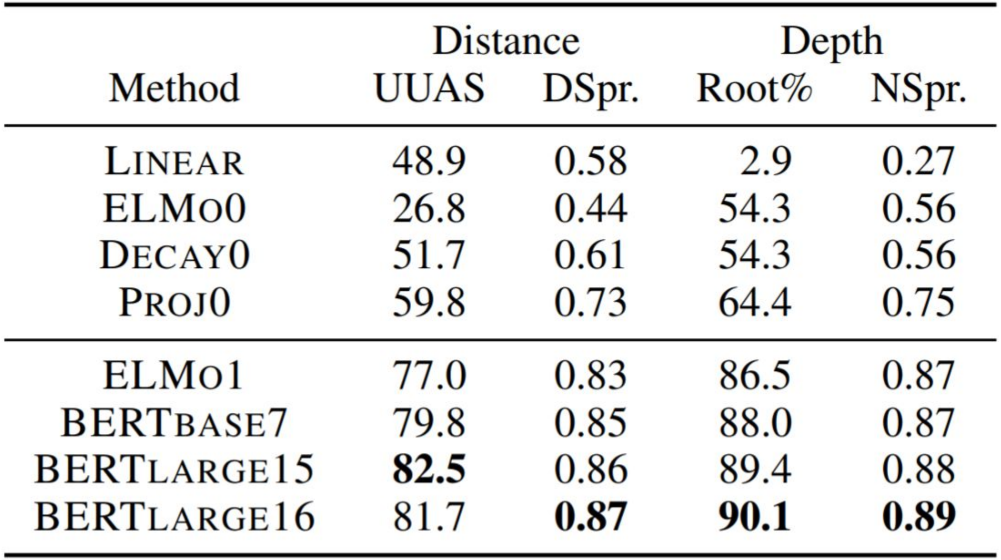
The corresponding parse structure obtained on different models. It can be seen that BERT and ELMo1, can predict these structures to a great accuracy, compared to the other baselines.

Experimental Results

* UUAS score for SOTA supervised model = 97.42 (Mrini et al., 2020)

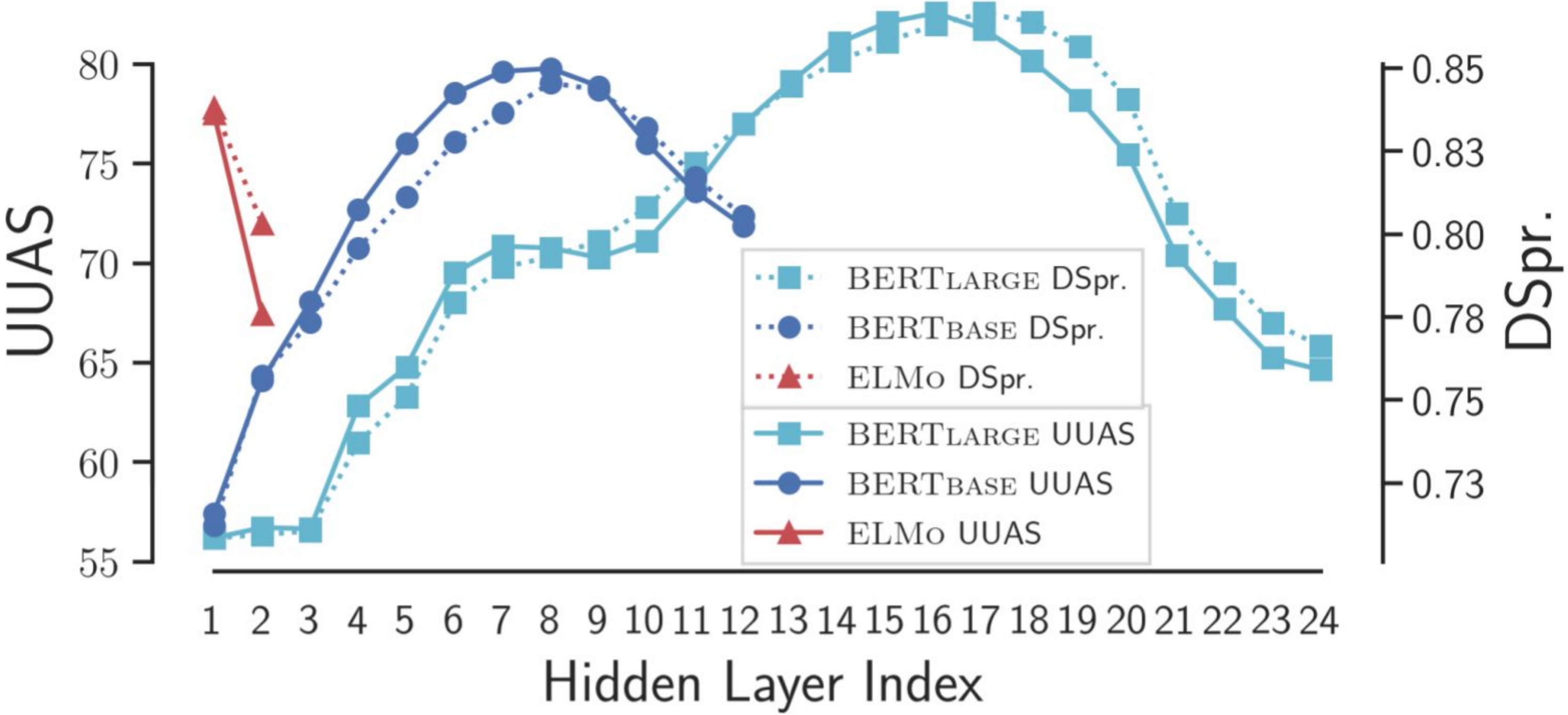
* UUAS score for SOTA unsupervised model = 66.2 (Le et al., 2015)

1. The probe doesn't simply "learn to parse" on top of any informative representation.
2. Surprisingly robust syntax embedded in each of ELMo and BERT.



	Distance		Depth	
Method	UUAS	DSpr.	Root%	NSpr.
Linear	48.9	0.58	2.9	0.27
ELMo0	26.8	0.44	54.3	0.56
Decay0	51.7	0.61	54.3	0.56
Proj0	59.8	0.73	64.4	0.75
ELMo1	77.0	0.83	86.5	0.87
BERTbase7	79.8	0.85	88.0	0.87
BERTlarge15	**82.5**	0.86	89.4	0.88
BERTlarge16	81.7	**0.87**	**90.1**	**0.89**

—

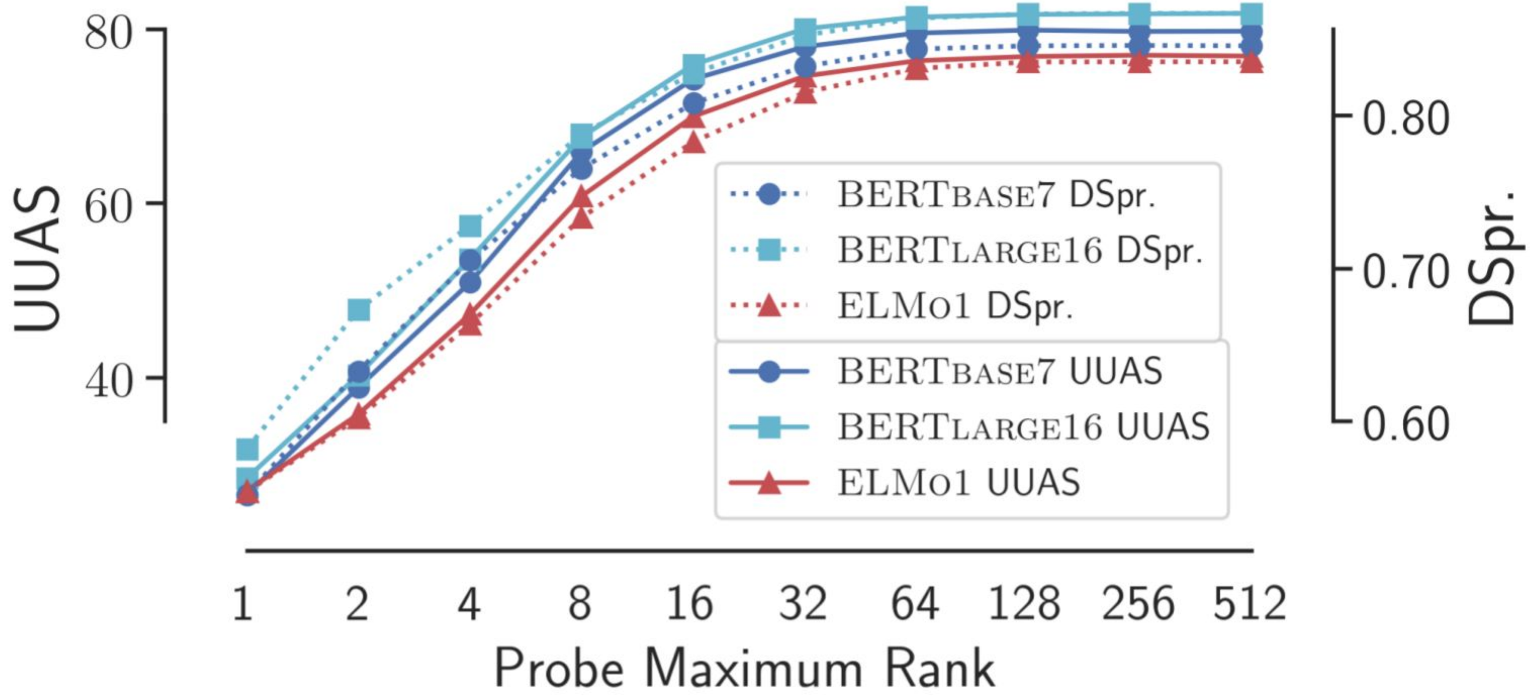


Parse distance UUAS and distance Spearman correlation across the BERT and ELMo model layers.

As in Peters et al. (2018b), we also find a clear difference in syntactic information between layers.

– Analysis of linear transformation rank

How compactly syntactic information is encoded in the vector space?



$B \in \mathbb{R}^{k \times n}$ such that the transformed vector $\mathbf{Bh}$ is in $\mathbb{R}^k$

Conclusion

1. Through the structural probes we demonstrate that the structure of syntax trees emerges through word representation spaces.

2. The effective rank of linear transformation required to represent a syntax encoding space, is surprisingly low.

Conclusion

- This is an intriguing result. Traditionally much of the emphasis in NLP has been on using labels for part of speech, syntax, etc., as an aid in other downstream tasks.
- This result suggests that large-scale hand construction of syntactically labeled training data may no longer be necessary for many tasks.

Scope for Future Work

- The probe may be useful for testing the existence of different types of graph structures on any neural representation of language.
- The curious result that the three models considered all seem to require transformations of approximately the same rank (and surprisingly low), can be further explored.

Scope for Future Work

- This allows for the possibility of developing unsupervised models on the top pre-trained models (like BERT), exploiting their learnt information to parse the syntax trees.