A Structural Probe for Finding Syntax in Word Representations

Report submitted in fulfillment of the requirements

for the B.Tech Project (7th Semester)

Third Year B.Tech.

by

Yash Malik

18075065

Under the guidance of

Dr. A. K. Singh

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI

Varanasi 221005, India

November 2021

Dedicated to

My parents and my teachers.

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.

2. The work has not been submitted for any project.

3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi
Date: 5th December, 2021

Yash Malik
B.Tech (Part III) Student
Department of Computer Science and Engineering
Indian Institute of Technology (BHU)
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**A Structural Probe for Finding Syntax in Word Representations**" being submitted by **Yash Malik** (Roll No. **18075065**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Place: IIT (BHU) Varanasi
Date: 5th December, 2021

Dr. A. K. Singh
Associate Professor
Department of Computer Science and Engineering
Indian Institute of Technology (BHU)
Varanasi, INDIA 221005.

Acknowledgments

I would like to express my sincere gratitude to my supervising Professor Dr. A.K. Singh for his constant guidance and support during the whole project work.

Place: IIT (BHU) Varanasi
Date: 5th December **Yash Malik**

Abstract

Human language communication is via sequences of words, canonically produced as a mainly continuous speech stream (Kuhl PK, 2004). Behind this linear organization is a rich hierarchical language structure with additional links that need to be understood by a hearer (or reader).
In computational linguistics, the long-dominant way of addressing this structure induction problem has been to hand-design linguistic representations, broadly following proposals from linguistics proper (Manning et al., 2020). In recent years, however, neural models that represent such information in a real-valued vector space, have proven very effective for machine translation, but at the expense of model interpretability.

We explore the knowledge of linguistic structure learned by large artificial neural networks. A new method is introduced to shed more light on the role played by linguistic structures in the process of neural machine translation.

The report describes methods for identifying linguistic hierarchical structure emergent in artificial neural networks and demonstrate that components in these models focus on syntactic grammatical relationships. We show that a linear transformation of learned embeddings in these models captures parse tree distances to a surprising degree, allowing approximate reconstruction of the sentence tree structures normally assumed by linguists. These results help explain why these models have brought such large improvements across many language-understanding tasks.

The work done in the project is based on a recent paper by John Hewitt and Christopher D. Manning, titled the same as this project “A Structural Probe for Finding Syntax in Word Representations”, 2019.

Contents

Chapter 1

Introduction

1.1 Overview

1.1.1 Human Languages and Syntactic Structure

In human languages, the meaning of a sentence is constructed by composing small chunks of words together with each other, obtaining successively larger chunks with more complex meanings until the sentence is formed in its entirety (Hewitt et al., 2019).

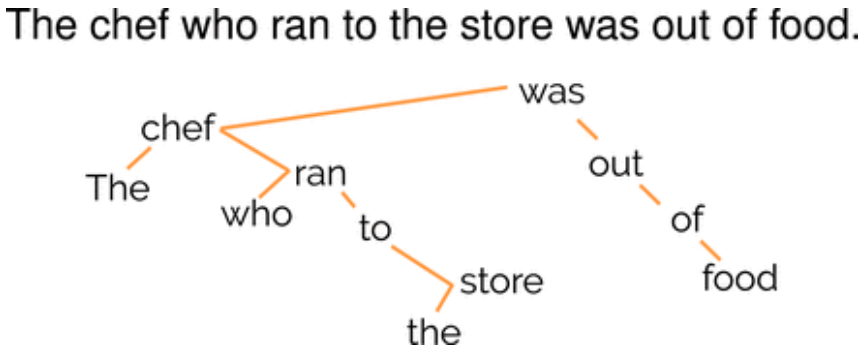


Fig 1.1 A hearer must reconstruct that the *store* is in a relative clause modifying the *chef* to know that it is the *chef* who is out of food rather than the linearly closer *store*.
(source Manning et al. 2020)

The order in which these chunks are combined creates a tree-structured hierarchy like the one in the picture above, which corresponds to the sentence

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

Note in this sentence that the *store* is combined with *chef*, which then is eventually combined with *was*, since it is the chef who was out of food, not the store (even though linearly *store* is closer than *chef*).

We refer to each sentence's tree-structured hierarchy as a **parse tree**, and the phenomenon broadly as **syntax**.

There are two main approaches to depicting a sentence's syntactic structure: phrase structure (or constituency) and dependency structure (or grammatical relations).

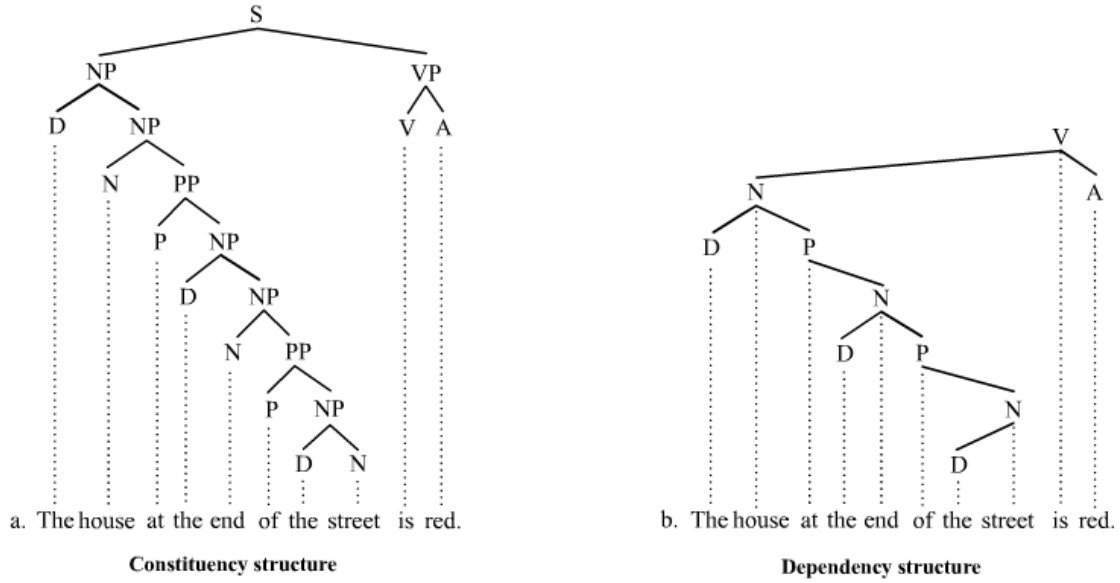


Fig. 1.2 Constituency structure and Dependency structure for a sentence.
Image source Treebank - Wikipedia

In this project, we use the latter, which is dominant in computational linguistics. Both representations capture similar, although generally not identical, information (Kaplan et al., 2010).

1.1.2 Numerical Machines

In recent years, however, neural networks used in NLP have represented each word in the sentence as a real-valued vector, with no explicit representation of the parse tree. To these networks, our example sentence looks like the image below (though instead of three-dimensional vectors, they're more like one thousand dimensions.)

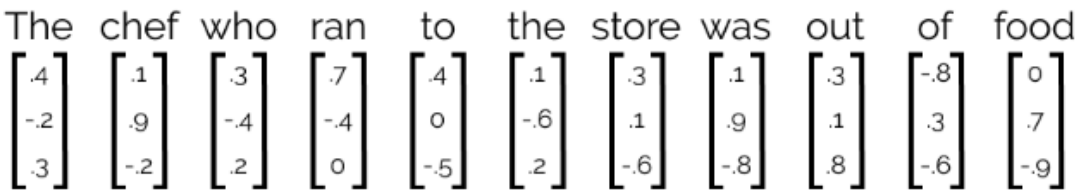


Fig. 1.3 An example (hypothetical) of word representation through real-valued vectors, which are used in neural models.
Image source [Finding Syntax with Structural Probes · John Hewitt]

And perhaps in a vector space, the sentence looks like below

Fig. 1.4 Visualization of how the vectors presented in fig. 1.3 might look in the 3-dimensional space.
Image source [Finding Syntax with Structural Probes]

1.2 Motivation

Language is made of discrete structures, yet neural networks operate on continuous data: vectors in high-dimensional space.

In the last few years, we have seen that the neural models have proven very effective for machine translation and similar tasks, but this comes at the expense of the interpretability of the model used.

Recently the transformer architectures have rapidly become the model of choice for NLP problems, replacing older recurrent neural network models such as the long short-term memory (LSTM) (Wolf et al. 2020).

For example, Bidirectional Encoder Representations from Transformers (**BERT**) which is a Transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google. When BERT was published, it achieved state-of-the-art performance on a number of natural language understanding tasks (Devlin et al. 2018).

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.

(BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding)

For example, the models understand that it is the *chef* who is out of *food* in the premise sentence, not the *store*.

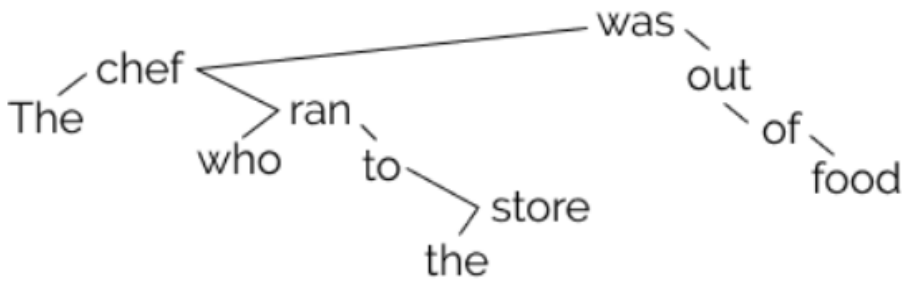


Fig. 1.5 Hierarchical structure in a sentence.

However, the string *the store was out of food* is a substring of the premise. Thus, knowing that the (*chef*, was) not the (*store* was) may be helpful.

More generally, implicitly understanding the syntax of the language may be useful in optimizing the language modeling objective. The dependency parse tree above encodes this intuition. (Hewitt et al. 2019)

Thus, transformer architectures show significant promise for natural language processing. Given that a single pre-trained model can be fine-tuned to perform well on many different tasks, these networks appear to extract generally useful linguistic features such as syntactic structure (Manning et al. 2020).

Some natural questions arise

- **How do such networks represent this information internally?**
- Human languages have tree structures, numerical machines with vector representations, **are these views of language reconcilable?**

In the project, we explore the vector representations learned by large neural networks like ELMo and BERT, which were simply trained to learn what typical English sentences are like, given a lot of English text.

1.3 Contribution

In this work, we explore a structural probe, a simple model which tests whether syntax trees are consistently embedded in a linear transformation of a neural network's word representation space.
We contribute a simple structural probe for finding syntax in word representations, and experiments providing insights into and examples of how a low-rank transformation recovers parse trees from ELMo and BERT representations.

1.4 Organization of Report

The second chapter contains some vital background information about natural language processing and definitions for the terms and tools used throughout the report.

The third chapter describes the project work, the problem is defined formally, and the approach methodology is described. The method of the structural probe is introduced and described in this chapter.

The fourth chapter describes the experiment details and lays out the visualizations and other important representations that help us to interpret the results described.

In the last chapter, we conclude by discussing the results and discussing the probable effects of the results in the field of NLP.

Chapter 2

Background Knowledge

2.1 Basic Definitions

- 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.

(Syntax)

- A **parse tree** is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar. The term parse tree itself is used primarily in computational linguistics; in theoretical syntax, the term syntax tree is more common.

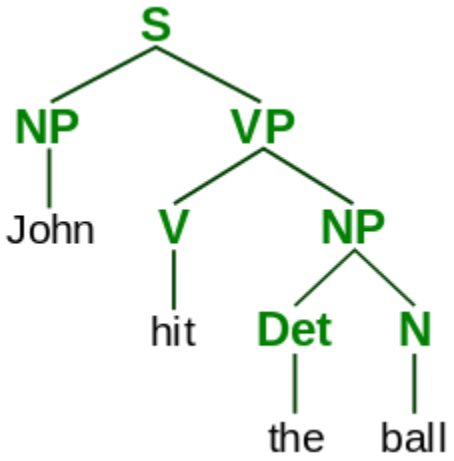


Fig. 2.1 A simple **parse tree** (constituency structure)

(Parse tree)

- An **artificial neural network** is an interconnected group of nodes, inspired by a simplification of neurons in a brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it.

  (Artificial neural network)

- **Bidirectional Encoder Representations from Transformers** (**BERT**) is a Transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. BERT was created and published in 2018 by Jacob Devlin and his colleagues from Google.

  When BERT was published, it achieved state-of-the-art performance on a number of **natural language understanding** tasks:

  - GLUE (General Language Understanding Evaluation) task set (consisting of 9 tasks)
  - SQuAD (Stanford Question Answering Dataset) v1.1 and v2.0
  - SWAG (Situations With Adversarial Generations)

  (BERT (language model))

- **Embeddings from Language Model** (**ELMo**) is a word embedding method for representing a sequence of words as a corresponding sequence of vectors. Character-level tokens are taken as the inputs to a bi-directional LSTM which produces word-level embeddings. Like BERT, ELMo embeddings are context-sensitive, producing different representations for words that share the same spelling but have different meanings (homonyms) such as "bank" in "river bank" and "bank balance".

  (ELMo)

2.2 Natural Language Processing - Evolution

2.2.1 NLP before the Deep Learning Era

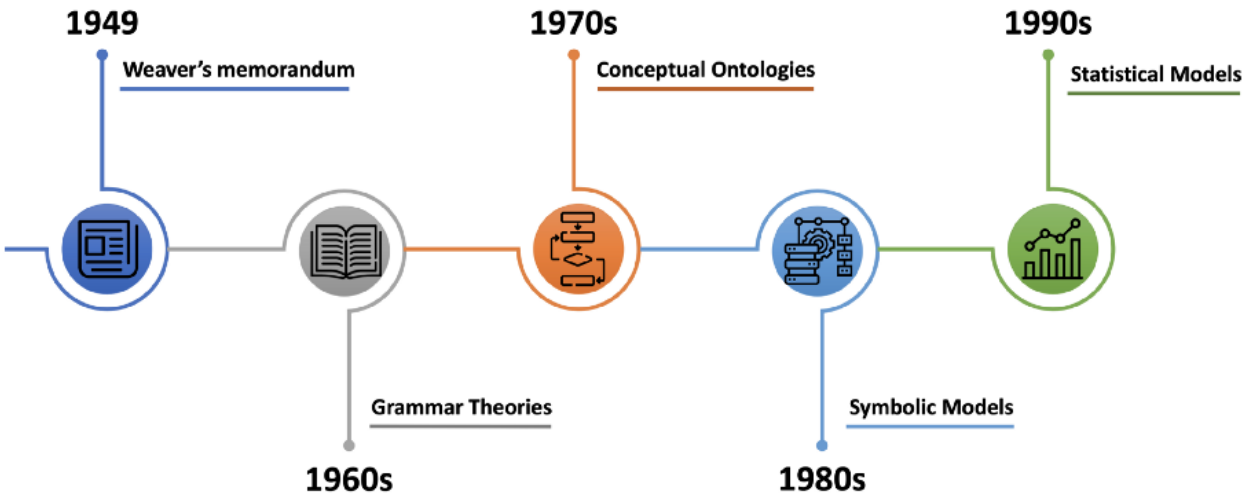


Fig. 1.3 The big stages of NLP before the deep learning era.
Image source A Brief History of Natural Language Processing

- **1949, Weaver's memorandum**: Weaver's memorandum was designed to suggest more fruitful methods than any simplistic word-for-word approach, which had grave limitations. He put forward four proposals.
    1. The problem of multiple meanings might be tackled by the examination of the immediate context. For example, the English word fast has at least two meanings which we can paraphrase as rapid or motionless.
    2. Given a set of premises, any logical conclusion could be deduced automatically by computer. Weaver hypothesized that translation could be addressed as a problem of formal logic, deducing "conclusions" in the target language from "premises" in the source language.
    3. Cryptographic methods were possibly applicable to translation. If

we want to translate, say, a Russian text into English, we can take the Russian original as an encrypted version of the English plaintext.

4. There may also be linguistic universals underlying all human languages which could be exploited to make the problem of translation more straightforward.

Source: Warren Weaver

- **1960s, Grammar Theories**: The idea of generative grammar (Chomsky, 1957) was introduced, a rule-based system of syntactic structures that brought insight into how mainstream linguistics could help machine translation.

Source: A Brief History of Natural Language Processing

- **1970s, Conceptual Ontologies**: structured real-world information into computer-understandable data. Examples are MARGIE (Schank and Abelson, 1975), TaleSpin (Meehan, 1976), QUALM (Lehnert, 1977), SAM (Cullingford, 1978), PAM (Schank and Wilensky, 1978), and Politics (Carbonell, 1979).

Source: A Brief History of Natural Language Processing

- **1980s, Symbolic Models**: Complex hard-coded rules and grammars to parse language. Practically, text was segmented into meaningless tokens (words and punctuation). Representations were then manually created by assigning meanings to these tokens and their mutual relationships through well-understood knowledge representation schemes and associated algorithms. Those representations were eventually used to perform a deep analysis of linguistic phenomena.

Source: A Brief History of Natural Language Processing

- **1990s, Statistical Models**: Statistical models came as a revolution in NLP

> (Bahl et al., 1989; Brill et al., 1990; Chitrao and Grishman, 1990; Brown et al., 1991), replacing most natural language processing systems based on complex sets of hand-written rules. Some of the earliest-used machine learning algorithms, such as decision trees (Tanaka, 1994; Allmuallim et al., 1994), produced systems similar in performance to the old school hand-written rules, statistical models broke through the complexity barrier of hand-coded rules by creating them through automatic learning, which led researchers to increasingly focus on these models. At the time, these statistical models were capable of making soft, probabilistic decisions.
>
> Source: A Brief History of Natural Language Processing

2.2.2 NLP during the Deep Learning Era

Recently, neural networks have replaced the traditional approaches and excelled in the field of natural language processing and understanding. But, neural models have a high opacity for what's actually going on inside them.

Therefore, this project aims to shed more light on the role played by linguistic structure in the process of neural machine translation.

Chapter 3

Project Work

3.1 Problem Definition

Motivating questions

- Identify if the deep learning models, that represent the sentences as real-valued vectors, encode syntax.
- Whether these deep contextual models encode entire parse trees in their word representations.
- Targeted case studies on ELMo and BERT representations.

Our goal is to 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 hypothesis being that there exists a latent parse tree on every sentence, which the neural network does not have access to. For the dependency parsing formalisms, each word in the sentence has a corresponding node in the parse tree.

The task and probe construction are designed not to test for some notion of syntactic knowledge broadly construed, but instead for an extremely strict notion

where all pairs of words know their syntactic distance, and this information is a global structural property of the vector space.

3.2 Methodology

Under a reasonable definition, to embed a graph is to learn a vector representation of each node such that geometry in the vector space - distances and norms - approximates geometry in the graph (Hamilton et al., 2017).

3.2.1 Trees as distances and norms

The key difficulty is in determining whether the parse tree, a discrete structure, is encoded in the sequence of continuous vectors.

The first intuition is that **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 (see fig 3.1).
- Distance metrics for vector spaces are quite well-known, and we use the L2 Euclidean distance metric, in our project (described later).

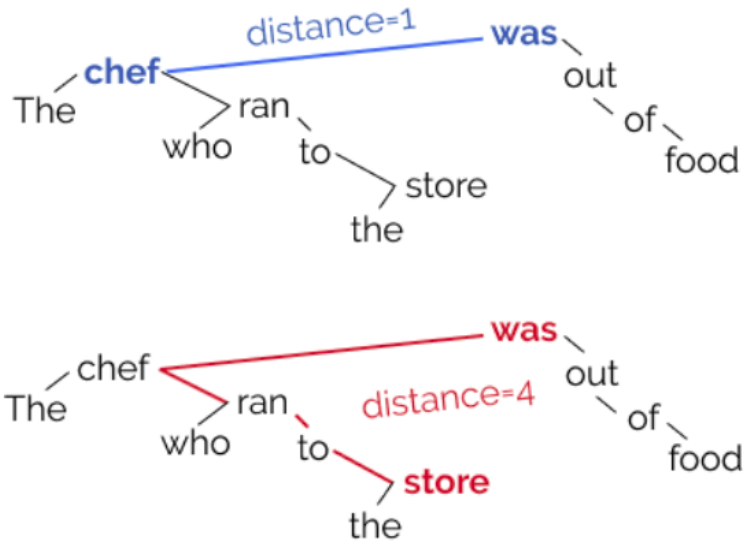


Fig. 3.1 Path metric for a tree, the number of edges in the unique path between two words in the tree. (Distance between *was* and *chef* is 1, and that between *was* and *store* is 4.)
Source [Finding Syntax with Structural Probes]

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.

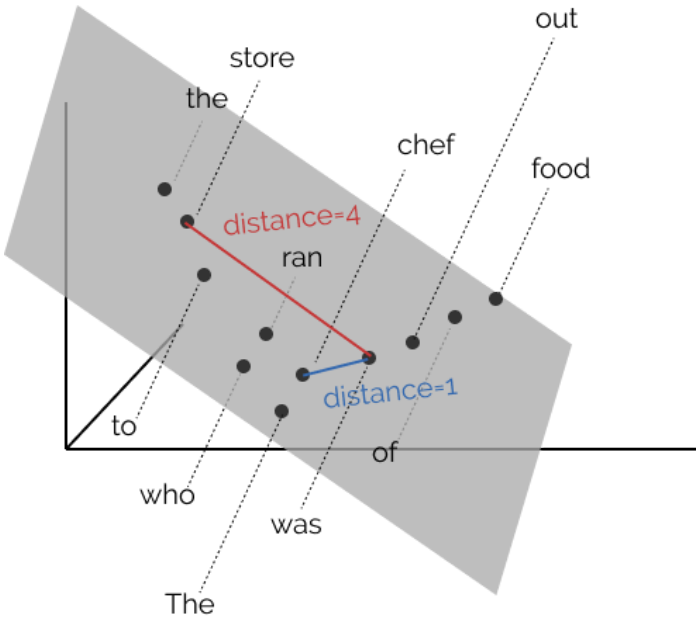
Thus, **embedding the tree reduces to embedding the distance metric defined by the tree.**

3.2.2 The Syntax Distance

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.
The structural probe relaxes this requirement somewhat; 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. From this, we get the syntax distance hypothesis:

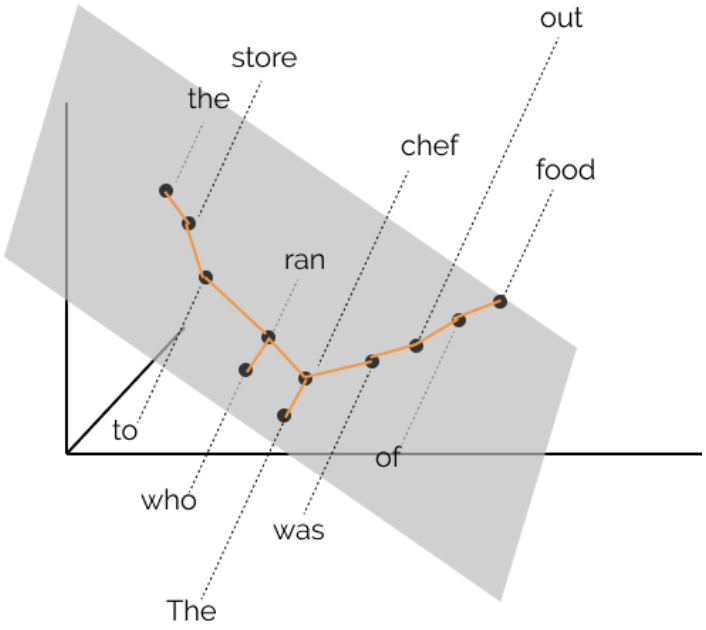
The syntax distance hypothesis: There exists a linear transformation **B** of the word representation space under which vector distance encodes parse trees. Equivalently, there exists an inner product on the word representation space such that distance under the inner product encodes parse trees. This (indefinite) inner product is specified by $\mathbf{B}^{\mathbf{T}}\mathbf{B}$. (Hewitt et al., 2019)

Now, the distances we pointed out earlier (fig 3.1) 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 representations to a 2-dimensional space encoding syntax:



Source [Finding Syntax with Structural Probes]

Note, in the image above that the 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**, as shown in the following image:



Source [Finding Syntax with Structural Probes]

3.3 The Structural Probe

The probe learns a linear transformation of a word representation space such that the transformed space embeds parse trees across all sentences.
This can be interpreted as **finding the part 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.

3.3.1 Finding a parse tree-encoding distance metric

Our potentially tree-encoding distances are parametrized by the linear transformation $B \in \mathbb{R}^{k \times n}$,

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

Where $B(h)$ is the linear transformation of the word representation; equivalently, it is the parse tree node representation. This is equivalent to finding an L2 distance on the original vector space, parametrized by the positive semi-definite matrix $A = B^T B$:

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

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:

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

where 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. Note that we do actually attempt to minimize the difference between the squared distance $\|h_i - h_j\|_B^2$ and the tree distance. This means that the actual vector distance $\|h_i - h_j\|_B$ will always be off from the true parse tree distances, but the tree information encoded is identical, and we found that optimizing with the squared distance performs considerably better in practice. (Hewitt et al., 2019)

3.3.2 Finding a parse depth-encoding norm

As a second application of our method, we note that the directions of the edges in a parse tree is determined by the depth of words in the parse tree; the deeper node in the governance relationship is the governed word. The depth in the parse tree is like a norm, or length, defining a total order on the nodes in the tree. We denote this tree depth norm $\|w_i\|$.

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:

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

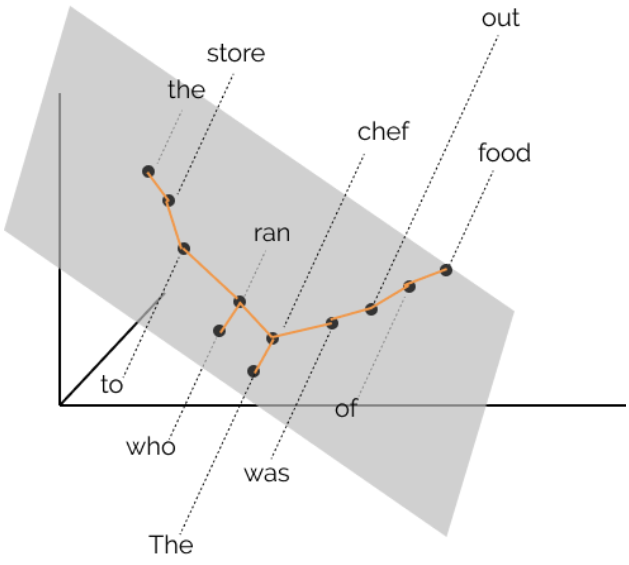
3.4 Overview

An overview of the structural probe method

A.

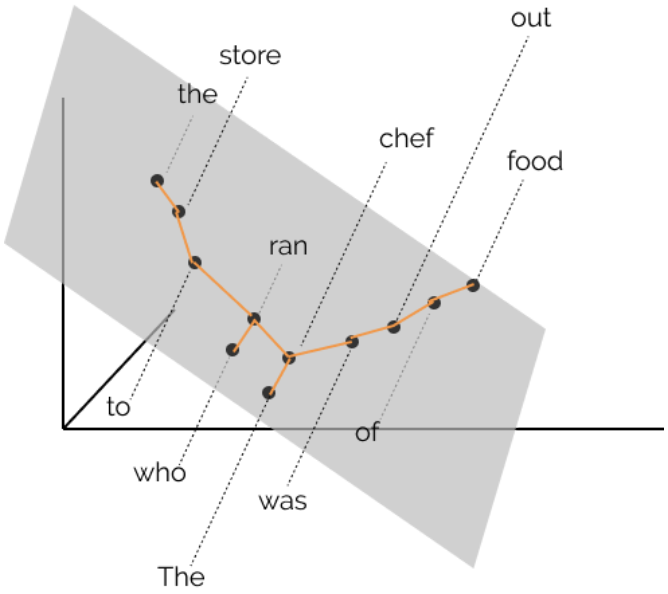
Each of the words of the sentence *The chef who ran to the store was out of food* is internally represented in context as a vector.

B.



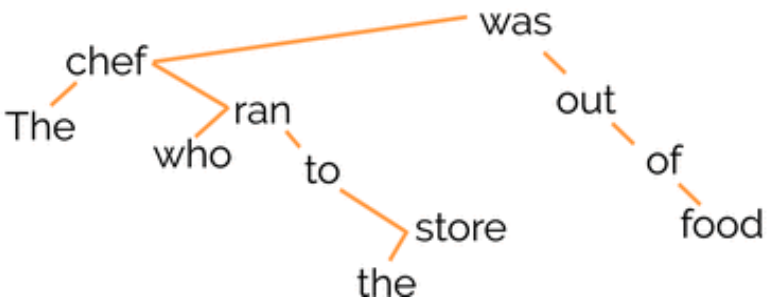
A structural probe finds a linear transform of that space under which squared L_2 distance between vectors best reconstructs tree path distance between words.

C.



Once in this latent space, the structure of the tree is globally represented by the geometry of the vector space, meaning that the words that are close in the space are close in the tree.

D.



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

Chapter 4

Experiments and Visualization

Using our probe, we evaluate whether representations from ELMo and BERT, two popular English models pre-trained on language modeling-like objectives, embed parse trees according to our structural hypothesis.

Unless otherwise specified, we permit the linear transformation B to be potentially full-rank (i.e., B is square.) Later, we explore what rank of transformation is actually necessary for encoding syntax.

4.1 Experiment

4.1.1 Representation Models

We use the 5.5B-word pre-trained ELMo weights for all ELMo representations, and both BERT-base (cased) and BERT-large (cased). The representations we evaluate are denoted ELMoK, BERTbaseK, BERTlargeK, where K indexes the

hidden layer of the corresponding model. All ELMo and BERT-large layers are dimensionality 1024; BERT-base layers are dimensionality 768.

4.1.2 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).

4.1.3 Data

We probe models for their ability to capture the Stanford Dependencies formalism (de Marneffe et al., 2006), claiming that capturing most aspects of the

formalism implies an understanding of English syntactic structure. To this end, we obtain fixed word representations for sentences of the parsing train/dev/test splits of the Penn Treebank (Marcus et al., 1993), with no pre-processing.[1]

4.1.4 Tree distance evaluation metrics

We evaluate models on how well the predicted distances between all pairs of words reconstruct gold parse trees and correlate with the parse trees' distance metrics. To evaluate tree reconstruction, **we take each test sentence's predicted parse tree distances and compute the minimum spanning tree**.

1. We evaluate the predicted tree on undirected attachment score (UAS)—the percent of undirected edges placed correctly—against the gold tree.
2. For distance correlation, we compute the Spearman correlation (DSpr.) between true and predicted distances for each word in each sentence. We average these correlations between all sentences of a fixed length, and report the macro average across sentence lengths 5–50 as the "distance Spearman (DSpr.)" metric[2].

[1] Since BERT constructs subword representations, we align subword vectors with gold Penn Treebank tokens, and assign each token the average of its subword representation. This thus represents a lower-bound on BERT's performance.

[2] The 5–50 range is chosen to avoid simple short sentences as well as sentences so long as to be rare in the test data.

4.1.5 Tree depth evaluation metrics

We evaluate models on their ability to recreate the order of words specified by their depth in the parse tree.

1. We report the Spearman correlation between the true depth ordering and the predicted ordering, averaging first between sentences of the same length, and then across sentence lengths 5–50, as the "norm Spearman (NSpr.)".
2. We also evaluate models' ability to identify the root of the sentence as the least deep, as the "root%"[3].

4.2 Visualization

Visualizing embeddings of a sentence after applying the Hewitt-Manning probe (structural probe) (see fig. 4.1). The left image is a traditional parse tree view, but the vertical length of each branch represents embedding distance. Right image: Principal Component Analysis projection of context embeddings, where color shows deviation from expected distance.

[3] In UAS and "root%" evaluations, we ignore all punctuation tokens, as is standard.

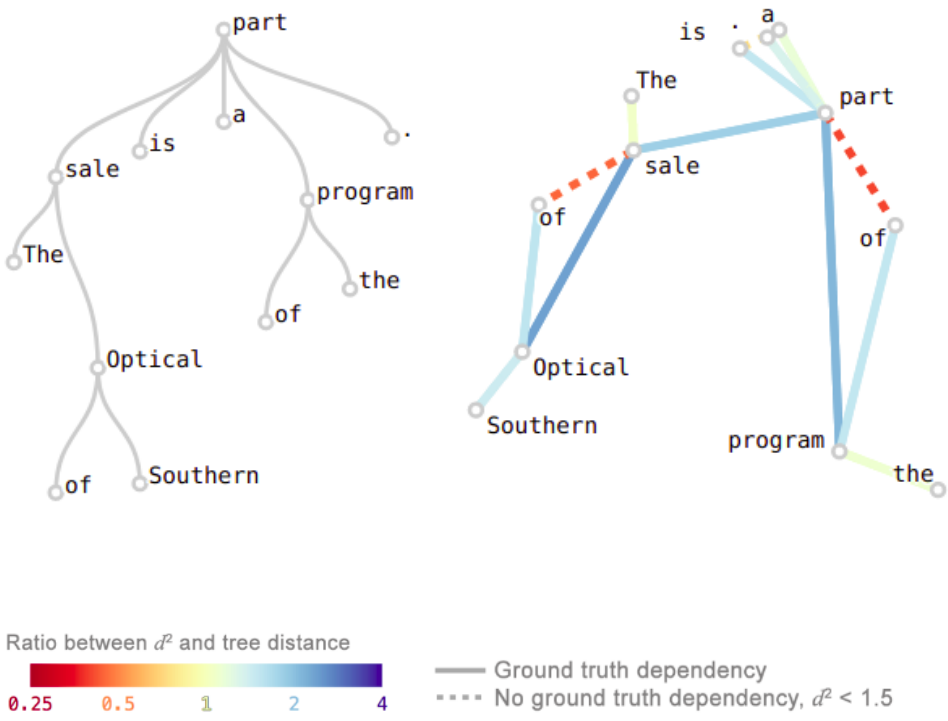


Fig. 4.1 Visualizing embeddings of a sentence after applying the Hewitt-Manning probe

Source [Language, trees, and geometry in neural networks]

4.3 Results (Reconstructed trees and depths)

The results of parse distance probes and parse depth probes are reported in table 1. We first confirm that our probe can't simply "learn to parse" on top of any informative representation, unlike parserbased probes. In particular, ELMO0 and DECAY0 fail to substantially outperform a right-branching-tree oracle that encodes the linear sequence of words. PROJ0, which has all of the

representational capacity of ELMO1 but none of the training, performs the best among the baselines.

Method	Distance		Depth	
	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**

Table 1: Results of structural probes on the PTB WSJ test set; baselines in the top half, models hypothesized to encode syntax in the bottom half. For the distance probes, we show the Undirected Unlabeled Attachment Score (UUAS) as well as the average Spearman correlation of true to predicted distances, DSpr. For the norm probes, we show the root prediction accuracy and the average Spearman correlation of true to predicted norms, NSpr.

We find **surprisingly robust syntax embedded in each of ELMo and BERT** according to our probes. Fig. 4.2, shows gold parse trees (black, above the sentences) along with the minimum spanning trees of predicted distance metrics for a sentence (blue, red, purple, below the sentence):

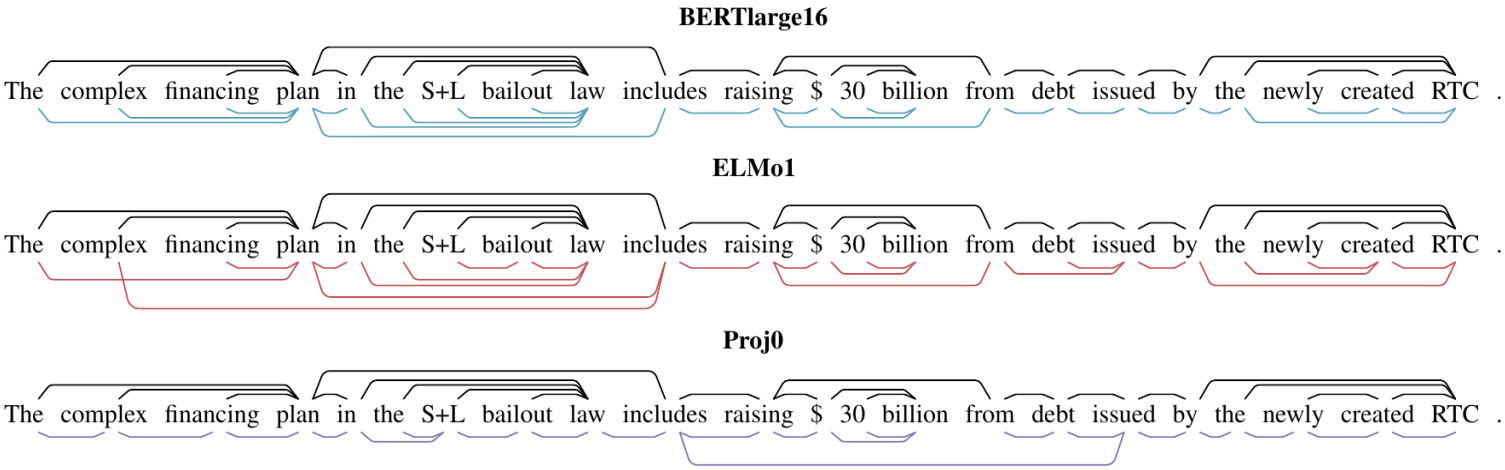


Fig. 4.2 Minimum spanning trees resulting from the structural probes on BERT, ELMo and a random control representation Proj0 compared to the human annotated parse tree. In the text sentence "S+L" refers to the American savings and loans banks and "RTC" refers to the Resolution Trust Corporation.

Source [Finding Syntax with Structural Probes]

From the figure above, it can be seen that the geometry of English parse trees is approximately discoverable in the geometry of deep models of language.

Both BERT and ELMo, are able to predict the parse structure of a sentence to a surprising extent.

Fig . 4.3, demonstrates the surprising extent to which the depth in the tree is encoded by vector norm after the probe transformation. It shows depths in the gold parse tree (grey, circle) as well as predicted (squared) parse depths according to ELMo1 (red, triangle) and BERT-large, layer 16 (blue, square).

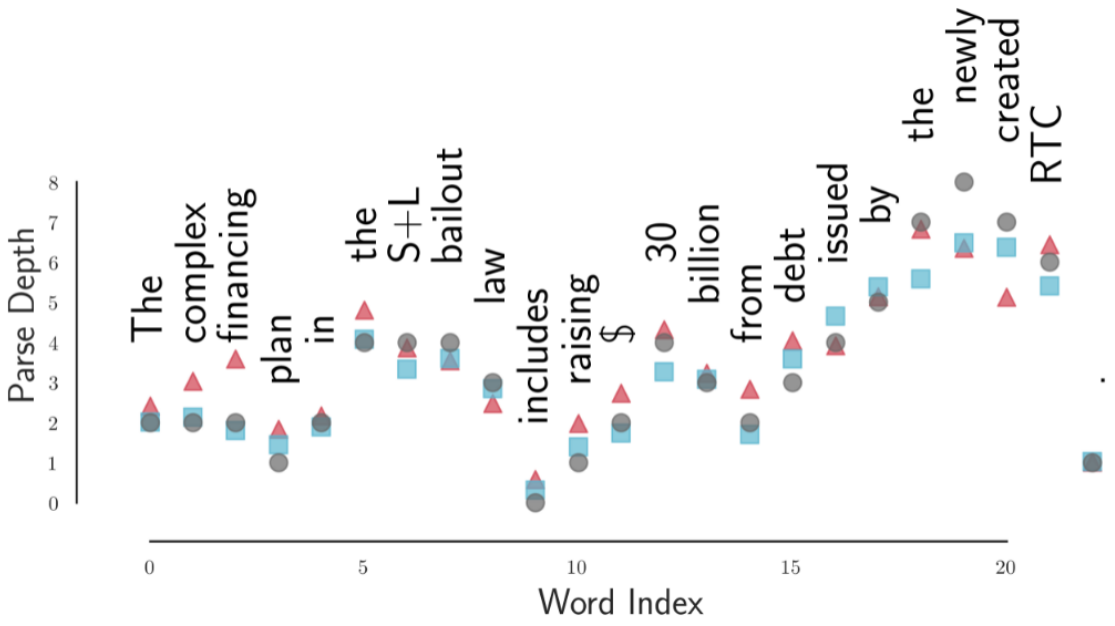


Fig. 4.3 Parse tree depth according to the gold tree (black, circle) and the norm probes (squared) on ELMo1 (red, triangle) and BERTLARGE16 (blue, square).

Finally, fig. 4.4 and 4.5 show the rich structure in a parse distance matrix, which visualizes all pairs of distances between words in a sentence. Below are both the distance defined by the actual tree, as well as the squared distance according to our probe on BERT-large, layer 16.

Long distances are lighter colors; short distances are darker colors.

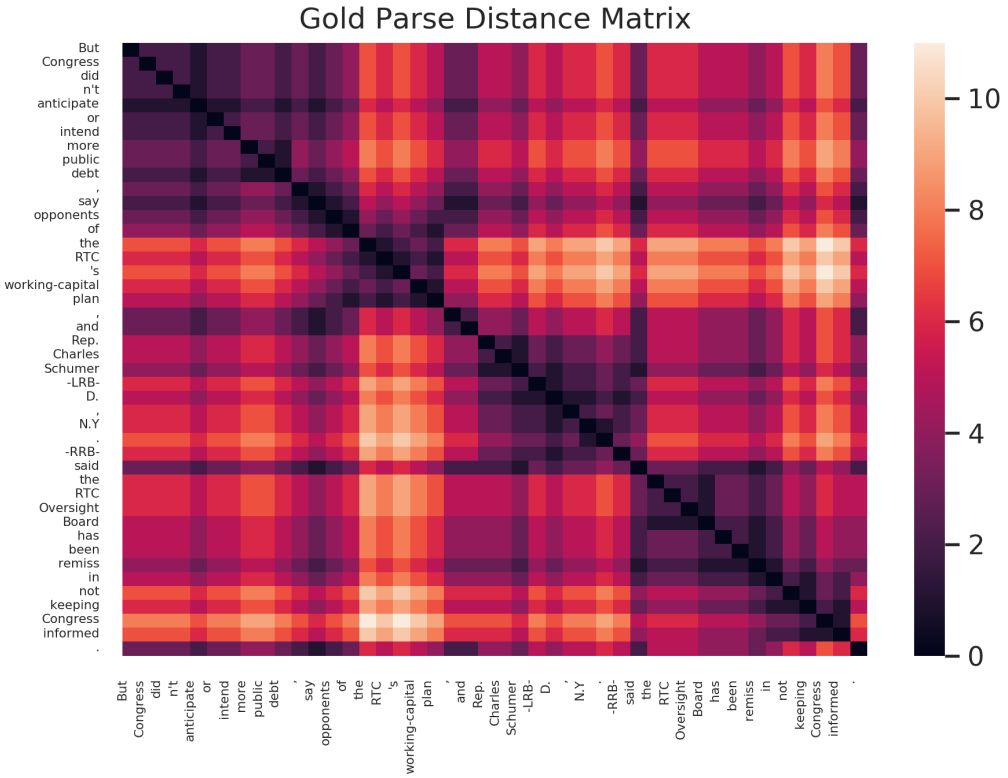

Gold Parse Distance Matrix
But
Congress
did
n't
anticipate
or
intend
more
public
debt
,
say
opponents
of
the
RTC
's
working-capital
plan
,
and
Rep.
Charles
Schumer
-LRB-
D.
,
N.Y
.
-RRB-
said
the
RTC
Oversight
Board
has
been
remiss
in
not
keeping
Congress
informed
.
10
8
6
4
2
0

Fig. 4.4

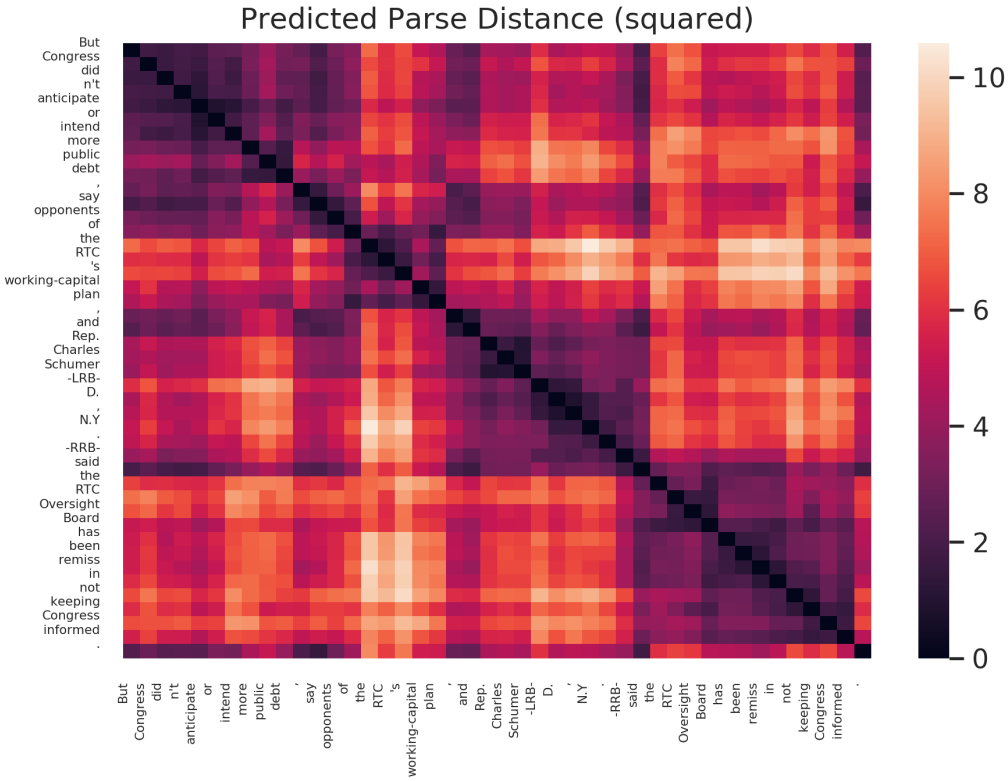

Predicted Parse Distance (squared)
But
Congress
did
n't
anticipate
or
intend
more
public
debt
,
say
opponents
of
the
RTC
's
working-capital
plan
,
and
Rep.
Charles
Schumer
-LRB-
D.
,
N.Y
.
-RRB-
said
the
RTC
Oversight
Board
has
been
remiss
in
not
keeping
Congress
informed
.
10
8
6
4
2
0

Fig. 4.5

Between models, we find consistently that $\text{BERT}_{\text{LARGE}}$ performs better than $\text{BERT}_{\text{BASE}}$, which performs better than ELMO. We also find, as in Peters et al. (2018b), a clear difference in syntactic information between layers; Figure 4.6 reports the performance of probes trained on each layer of each system.

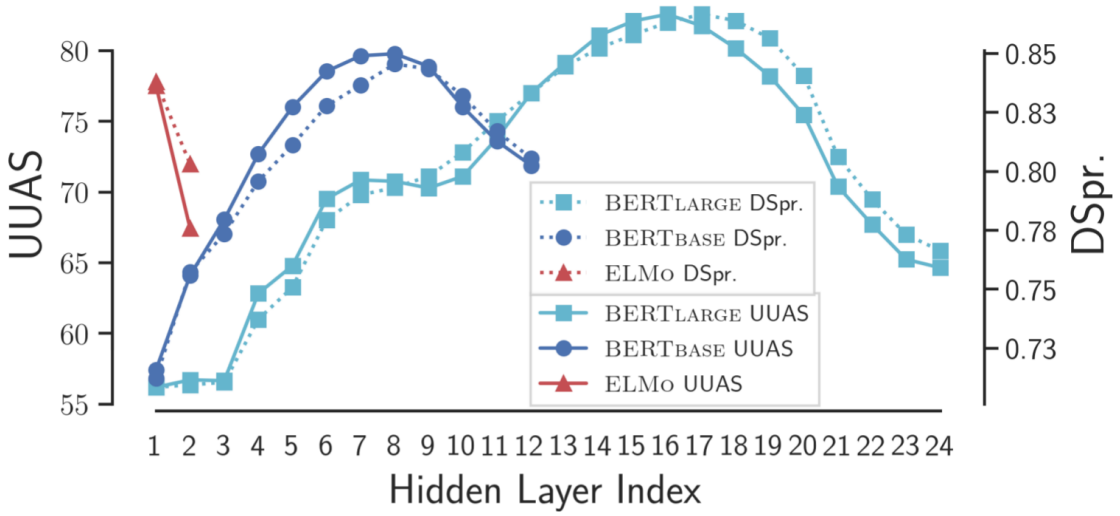


Fig 4.6 Parse distance UUAS and distance Spearman correlation across the BERT and ELMo model layers.

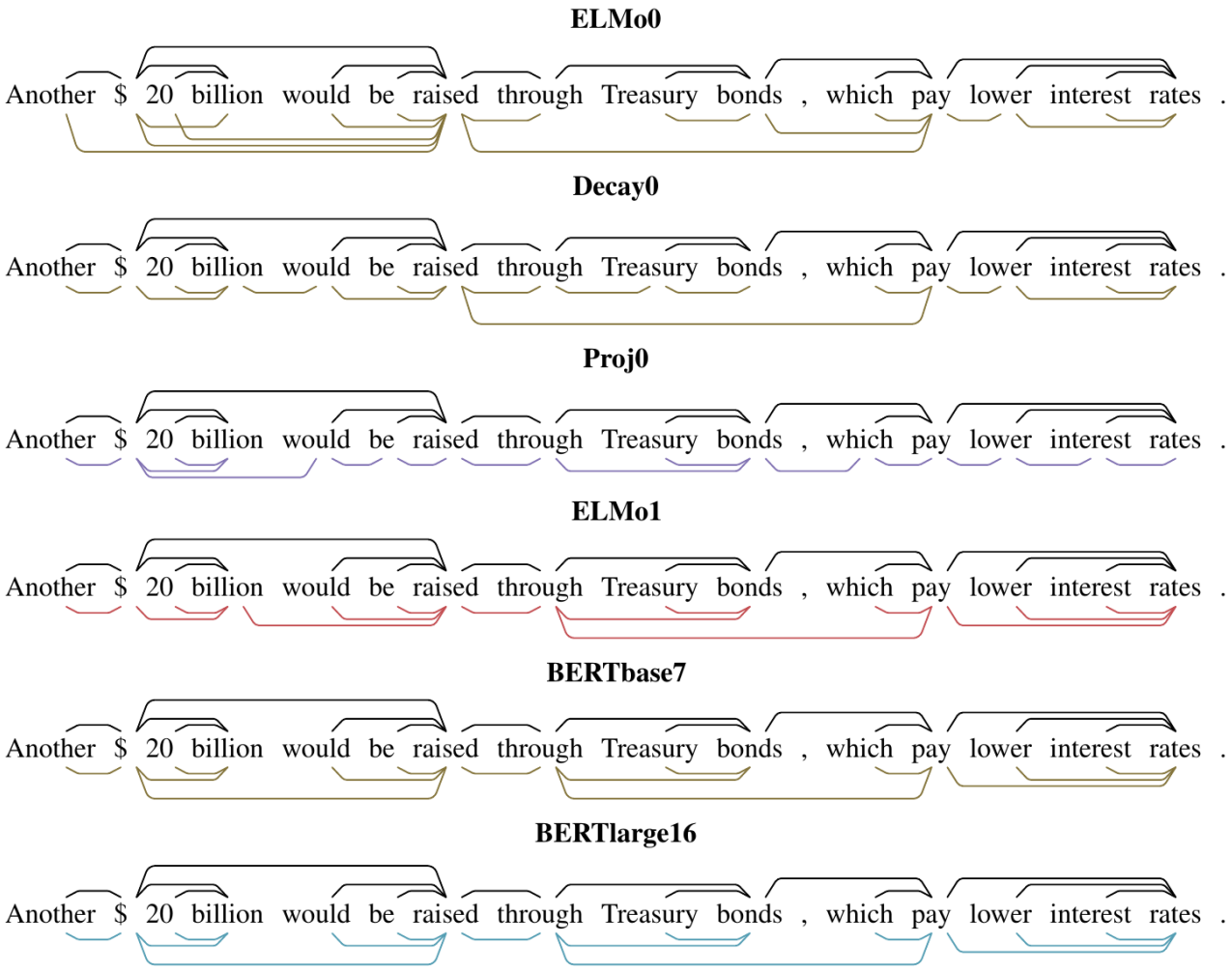
Chapter 5

Conclusion and Discussion

5.1 Result

We examine syntactic structure in models and develop simple probes to show that these models know about the syntactic linguistic information. Indeed, 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.

Fig. 5.1 provides the trees for another example sentence, and the corresponding parse structure obtained on different models. It can be seen that BERT and ELMo1, can predict these structures to a great accuracy, as compared to the other baselines.



This result has been demonstrated through structural probes on internal vector representations, showing that the hierarchical tree structures of language emerge in BERT/ELMo vector space. The fact that such rich information emerges is surprising and exciting, with intriguing implications for both NLP research and the logical problem of language acquisition.

5.2 Analysis of linear transformation rank

With the result that there exists syntax-encoding vector structure in both ELMo and BERT, it is natural to ask how compactly syntactic information is encoded in the vector space. We find that in both models, the effective rank of linear transformation required is surprisingly low. We train structural probes of varying k, that is, specifying a matrix $B \in \mathbb{R}^{k \times n}$ such that the transformed vector $\mathbf{Bh}$ is in $\mathbb{R}^k$.

As shown in Figure 5.2, increasing k beyond 64 or 128 leads to no further gains in parsing accuracy. Intuitively, larger k means a more expressive probing model, and a larger fraction of the representational capacity of the model being devoted to syntax. We also note with curiosity that the three models we consider all seem to require transformations of approximately the same rank; we leave exploration of this to exciting future work.

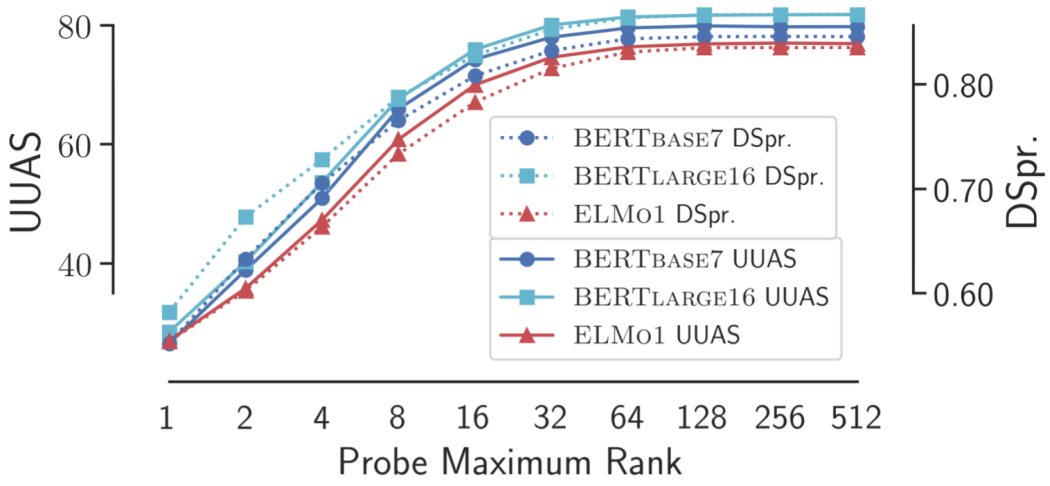


Fig. 5.2 Parse distance tree reconstruction accuracy when the linear transformation is constrained to varying maximum dimensionality.

5.2 Conclusion

In summary, through the structural probes we demonstrate that the structure of syntax trees emerges through properly defined distances and norms on two deep models' word representation spaces.

This is a startling and 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.

Bibliography

Kuhl PK. Early language acquisition: cracking the speech code. Nat Rev Neurosci. 2004 Nov;5(11):831-43. doi: 10.1038/nrn1533. PMID: 15496861.

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, Omer Levy Emergent linguistic structure in artificial neural networks trained by self-supervision Proceedings of the National Academy of Sciences Dec 2020, 117 (48) 30046-30054; DOI: 10.1073/pnas.1907367117

J. Hewitt, C. D. Manning, "A structural probe for finding syntax in word representations" in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, J. Burstein, C. Doran, T. Solorio, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2019), pp. 4129–4138.

R. Kaplan, J. Burstein, M. Harper, G. Penn O. Rambow, "The simple truth about dependency and phrase structure representations: An opinion piece" in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, R. Kaplan, J. Burstein, M. Harper, G. Penn, Eds. (Association for Computational Linguistics, Stroudsburg, PA, 2010), pp. 337–340

Wolf, Thomas; Debut, Lysandre; Sanh, Victor; Chaumond, Julien; Delangue, Clement; Moi, Anthony; Cistac, Pierric; Rault, Tim; Louf, Remi; Funtowicz, Morgan; Davison, Joe; Shleifer, Sam; von Platen, Patrick; Ma, Clara; Jernite,

Yacine; Plu, Julien; Xu, Canwen; Le Scao, Teven; Gugger, Sylvain; Drame, Mariama; Lhoest, Quentin; Rush, Alexander (2020). "Transformers: State-of-the-Art Natural Language Processing". Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). Association for Computational Linguistics.

Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina (11 October 2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"

Peters, W.t. (2018). Dissecting Contextual Word Embeddings: Architecture and Representation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 1499–1509). Association for Computational Linguistics.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In LREC.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn Treebank. Computational linguistics, 19(2):313–330.

Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018b. Dissecting contextual word embeddings: Architecture and representation. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 1499–1509.