IPI PAN

Interpreting and Controlling Linguistic Features in the Neural Networks' Representation

Tomasz Limisiewicz

Charles University
Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics

Interpretability

*“**Interpretability**: the degree to which human can understand the cause of decision [of a model]”*
Miller (2019)

*“A **Black Box Model** is a system that does not reveal its internal mechanisms”*
Molnar (2020)

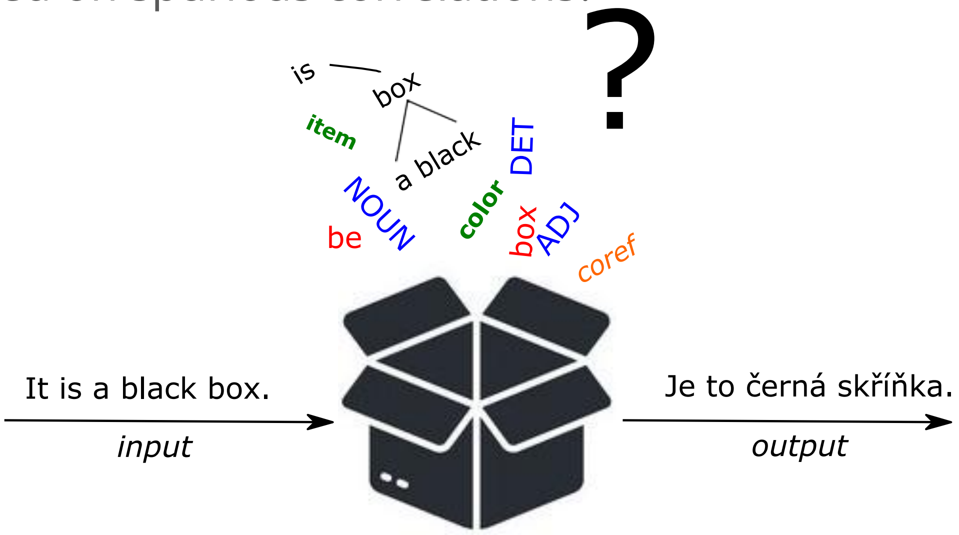
Introduction

Motivations

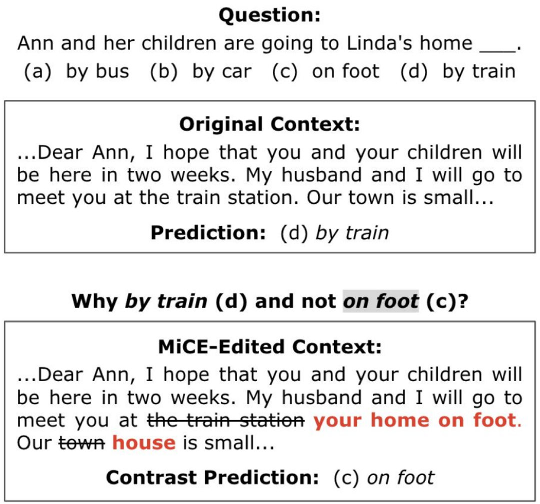
- Deep neural networks have rapidly become a central component for solving many NLP tasks.
- They learn patterns present in language corpora, and do not gain explicit knowledge of linguistic abstractions.
- Large neural models are black boxes that are very hard to interpret.

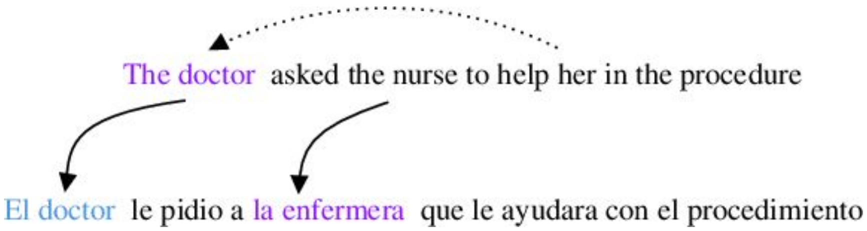
Motivations

- How do they work? What emergent abstractions can we observe in them?
- Are the emergent structures and abstractions similar to classical linguistic structures and abstractions?
- Can interpretation be useful for improving neural nets? E.g. in avoiding predictions based on spurious correlations?



Motivations





Examples: Ross et al. (2021): Minimal Contrastive Editing in Question Answering; Stanovsky et al. (2019): Gender Bias in Machine Translation; Me: Machine Translation?

Overview of the Presentation

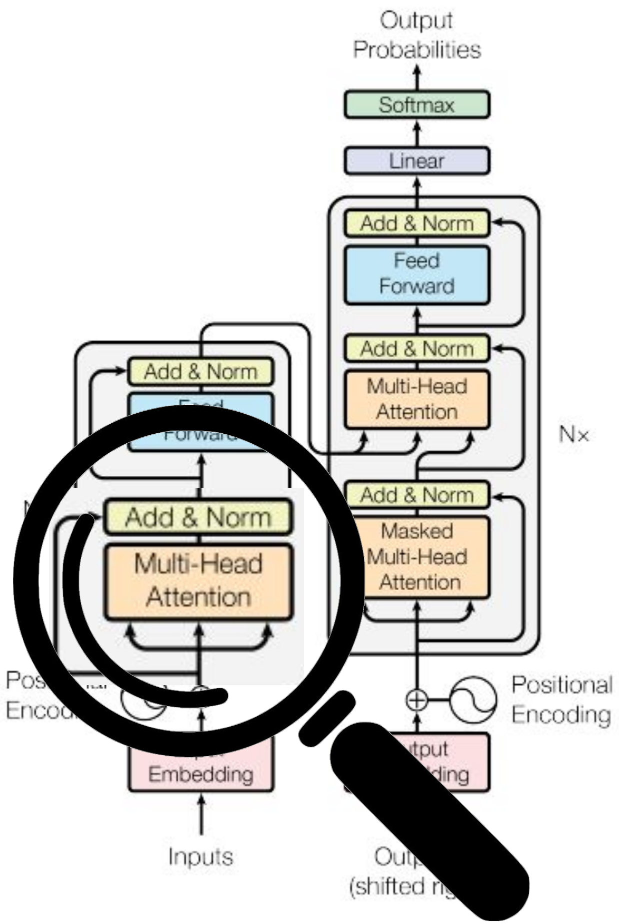
Interpretability	Probing	Orthogonal Probe	Limitations	Controlling Bias
Motivations behind interpreting neural networks.	Background works about probin neural nets for linguistic information.	Our method that allows to disentangle specific linguistic signals in the representations.	Is probing an adequate method for interpreting models?	How our approach can improve the predictions of models.

A Note on Terminology

A Neural Network / Model: We will focus on Transformer based models (mainly LLMs -> BERT)

Embeddings: vector representations of the model: numerical output or hidden states in multi-layer systems.

Representations: all of numerical representation of the data in the model. E.g. in Transformer: embeddings + attention weights



A Few Interpretation Approaches

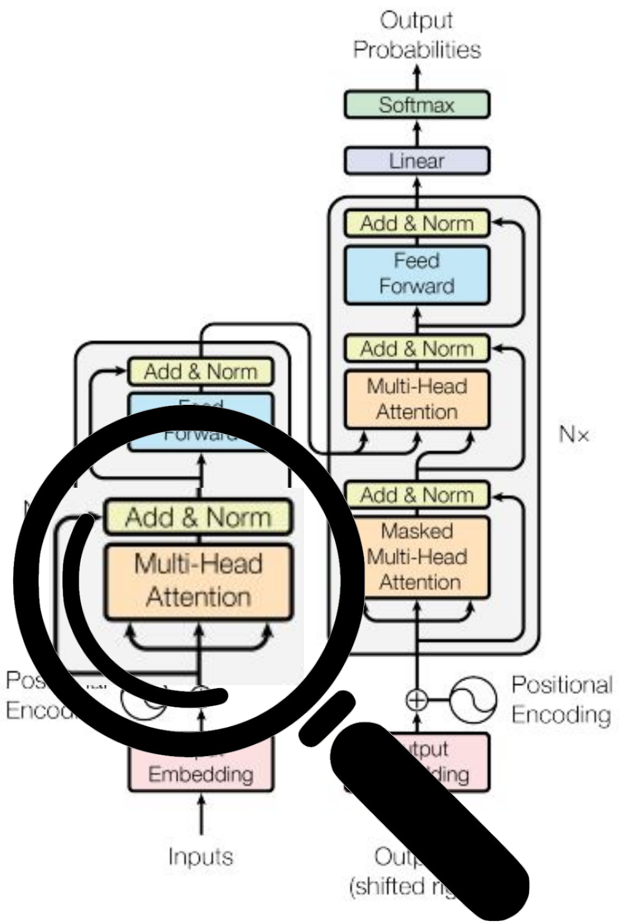
Interpretation of Attention Weights

Clustering Latent Representation

Principal Component Analysis

Probing Neural Networks

Causal Mediation



A Few Interpretation Approaches

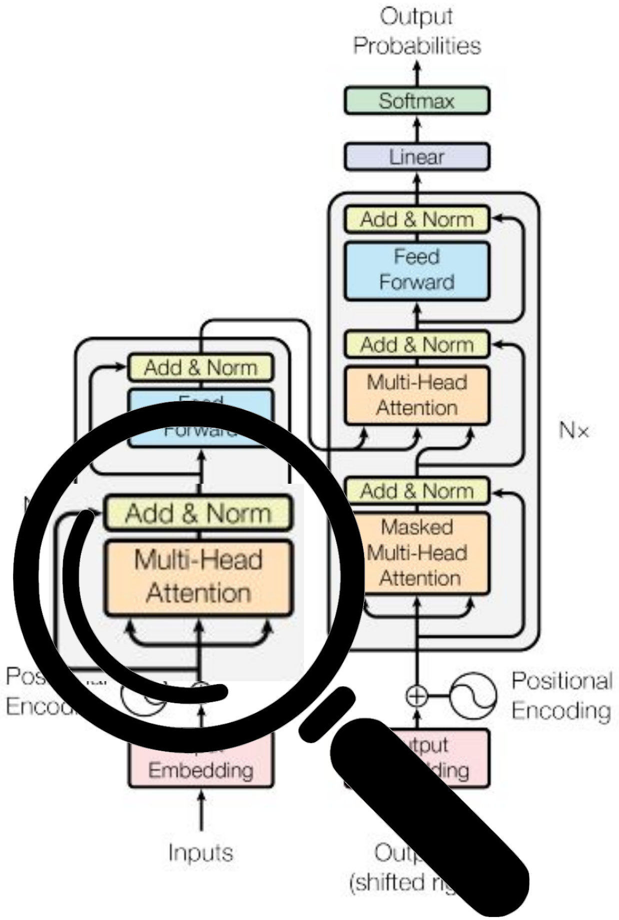
Interpretation of Attention Weights

Clustering Latent Representation

Principal Component Analysis

Probing Neural Networks

Causal Mediation



Game of Probes

Classification Probing

- Contextual neural network models is trained, e.g. for Language Modeling, Translation
- The parameters of the network are fixed (frozen). A new simple network takes is trained on top for auxiliary linguistic task, e.g. POS tags prediction.
- We assume that when probing classifier accuracy is high the networks encodes linguistic abstraction well.

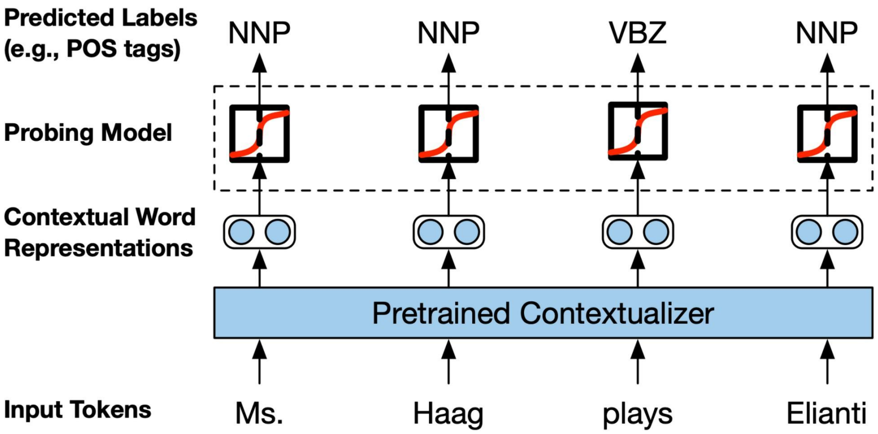


Figure: Liu et al. (2019): "Linguistic Knowledge and Transferability of Contextual Representations"

Syntax Probing

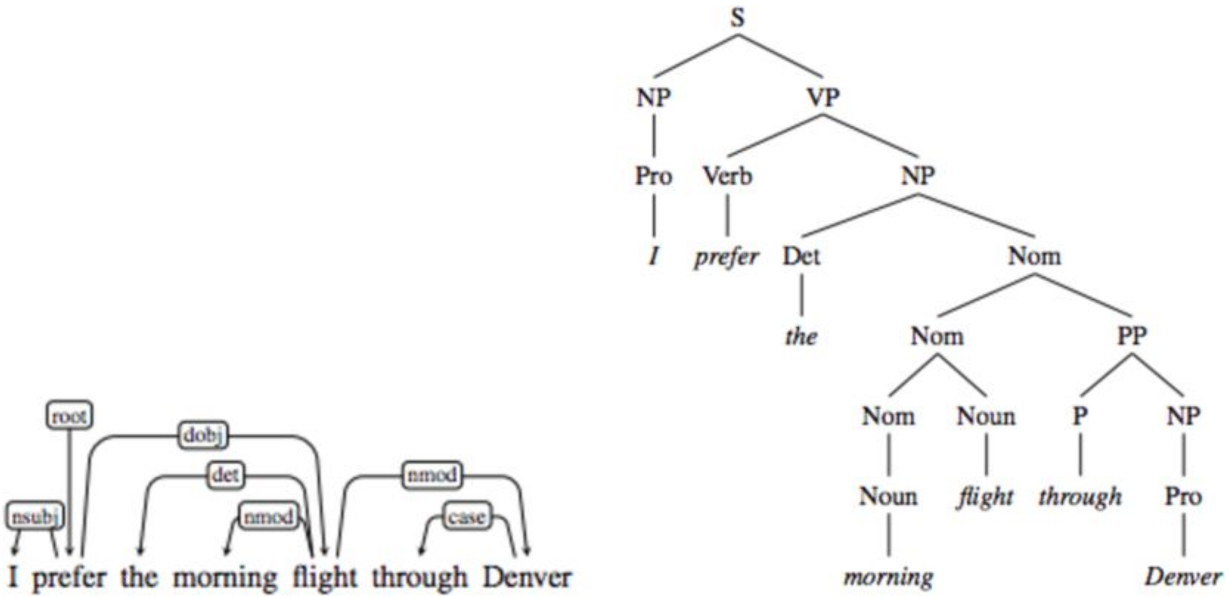


Figure: Comparison of two widely used syntactic structure types: dependency and constituency trees, from Jurafsky and Martin 2009

Syntax Probing: Background

- Blevins, Levy, and Zettlemoyer 2018 use a feed-forward classifier on top of RNN representation to predict whether a pair of tokens is connected by a dependency edge.

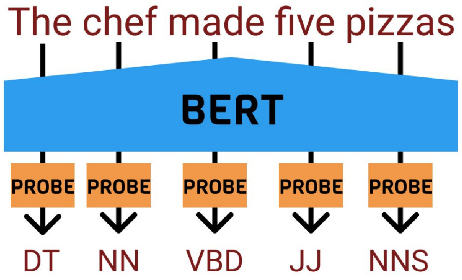
- Hewitt and Manning 2019 construct a linear to approximate syntactic tree distance between tokens by the L2 norm of the difference of the transformed vectors.

$$\min_{B} \left| (B(h_i - h_j))^T (B(h_i - h_j)) - d_T(w_i, w_j) \right|$$

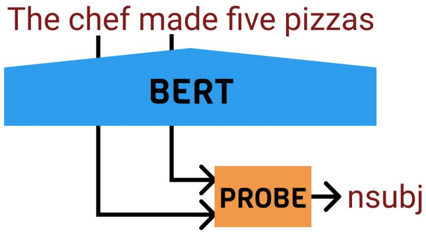
- This approach produces the approximate syntactic pairwise distances for each pair of tokens. The minimum spanning tree is used to create a dependency tree with high accuracy (82.5% UAS on Penn Treebank).

Syntax Probing

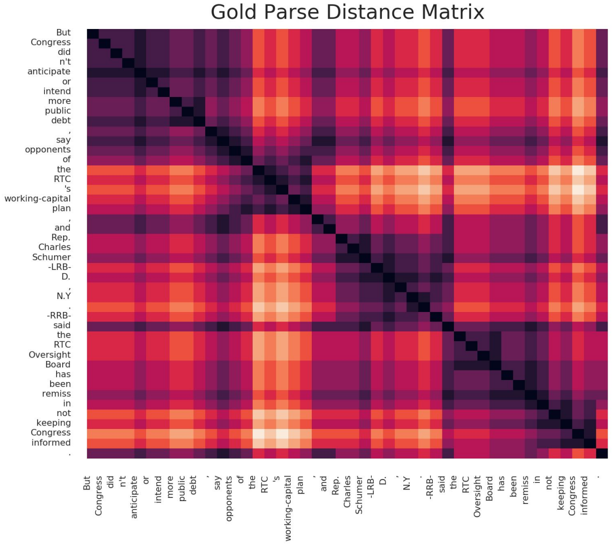
Part-of-speech!

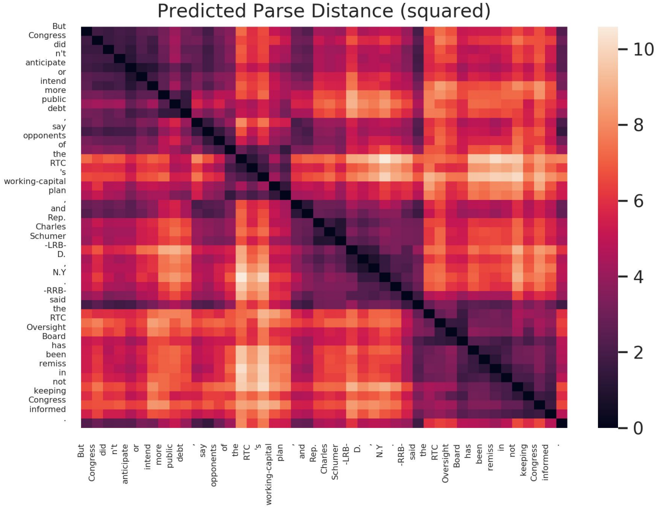


Partial dependency info!



Figures from John Hewitt's blog





What Is Encoded Where?

Tenney et al. (2019) performs probing for linguistic features encoded in BERT (POS-tagging, syntactic parsing, semantic roles parsing, coreference resolution, …). They observe that subsequent layers specializes in encoding specific types of information and make an analogy to standard* NLP-pipeline.

What Is Encoded Where?

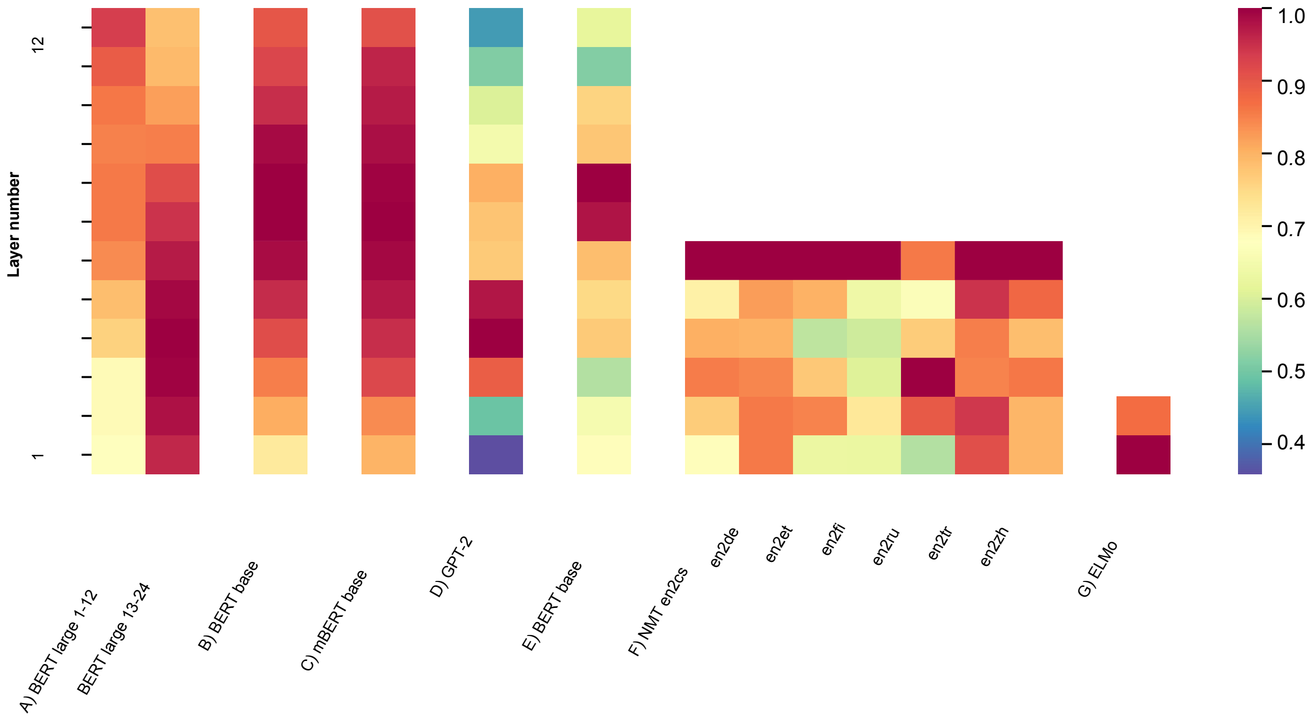


Figure: Relative syntactic information across attention models and layers

Orthogonal Probe

Orthogonal Structural Probe

Tomasz Limisiewicz and David Mareček. Introducing orthogonal constraint in structural probes.
In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, August 2021b

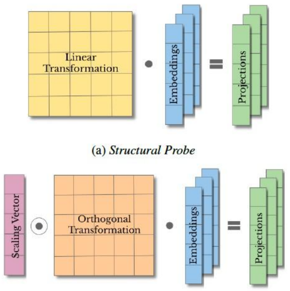
- Based on structural probing approach Hewitt and Manning (2019)
- Probe for syntactic dependency, lexical hypernymy, and non-linguistic structures
- Decompose embeddings into parts encoding specific linguistic structures

Introducing Orthogonal Constraint in Structural Probes

Tomasz Limisiewicz and **David Mareček**
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
{limisiewicz, marecek}@ufal.mff.cuni.cz

Abstract

With the recent success of pre-trained models in NLP, a significant focus was put on interpreting their representations. One of the most prominent approaches is structural probing (Hewitt and Manning, 2019), where a linear projection of word embeddings is performed in order to approximate the topology of dependency structures.In this work, we introduce a new type of structural probing, where the linear projection is decomposed into 1. isomorphic space rotation; 2. linear scaling that identifies and scales the most relevant dimensions. In addition to syntactic dependency, we evaluate our method on two novel tasks (lexical hypernymy and position in a sentence). We jointly train the probes for multiple tasks and experimentally show that lexical and syntactic information is separated in the representations. Moreover, the orthogonal constraint makes the *Structural Probes* less vulnerable to memorization.



(a) *Structural Probe*

(b) *Orthogonal Structural Probe*

Figure 1: Comparison of the *Structural Probe* of Hewitt and Manning (2019) and the *Orthogonal Structural Probe* proposed by us.

Structural Probe

Hewitt and Manning (2019)

- Approximation of the dependency tree distance:

$$\min_B \left|(B(h_i - h_j))^T (B(h_i - h_j)) - d_T(w_i, w_j)\right|$$

- Approximation of the depth in a tree:

$$\min_B \left|(Bh_i)^T (Bh_i) - \|w_i\|_T\right|$$

A Structural Probe for Finding Syntax in Word Representations

John Hewitt
Stanford University
johnhew@stanford.edu

Christopher D. Manning
Stanford University
manning@stanford.edu

Abstract

Recent work has improved our ability to detect linguistic knowledge in word representations. However, current methods for detecting syntactic knowledge do not test whether syntax trees are represented in their entirety. In this work, we propose a *structural probe*, which evaluates whether syntax trees are embedded in a linear transformation of a neural network's word representation space. The probe identifies a linear transformation under which squared L2 distance encodes the distance between words in the parse tree, and one in which squared L2 norm encodes depth in the parse tree. Using our probe, we show that such transformations exist for both ELMo and BERT but not in baselines, providing evidence that entire syntax trees are embedded implicitly in deep models' vector geometry.

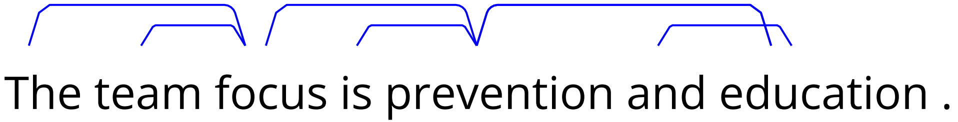
1 Introduction

As pretrained deep models that build contextualized representations of language continue to provide gains on NLP benchmarks, understanding

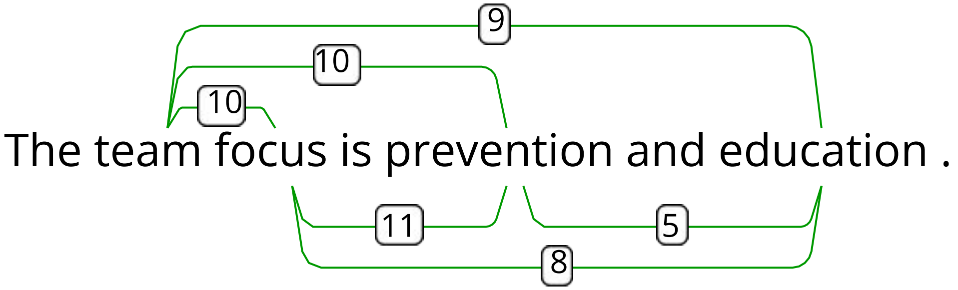
In this work, we propose a *structural probe*, a simple model which tests whether syntax trees are consistently embedded in a linear transformation of a neural network's word representation space. Tree structure is embedded if the transformed space has the property that squared L2 distance between two words' vectors corresponds to the number of edges between the words in the parse tree. To reconstruct edge directions, we hypothesize a linear transformation under which the squared L2 norm corresponds to the depth of the word in the parse tree. Our probe uses supervision to find the transformations under which these properties are best approximated for each model. If such transformations exist, they define inner products on the original space under which squared distances and norms encode syntax trees – even though the models being probed were never given trees as input or supervised to reconstruct them. This is a structural property of the word representation space, akin to vector offsets encoding word analogies (Mikolov et al., 2013). Using our probe, we conduct a targeted case study, showing that ELMo (Peters et al.,

Linguistic Structures

DEP Dependency tree from Universal Dependencies (Nivre et al., 2020)



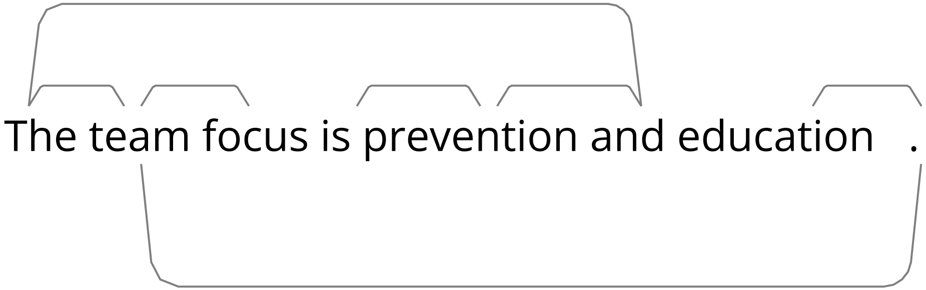
LEX Hypernymy hierarchy from WordNet (Miller, 1995)



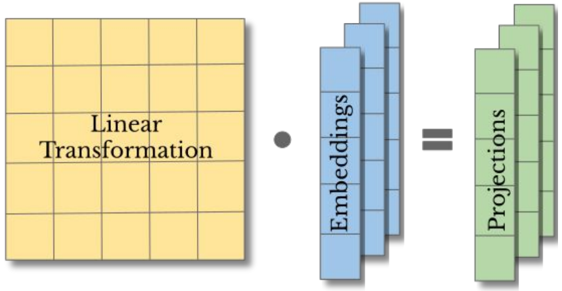
Baseline Structures

POS Right branching chain

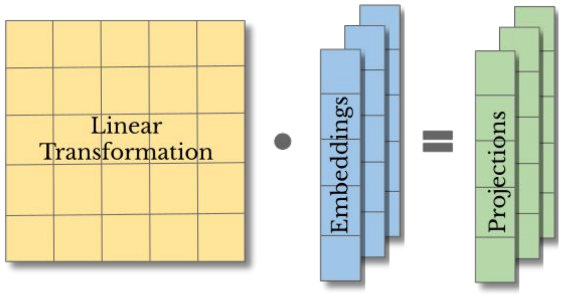
RAND Randomly generated trees

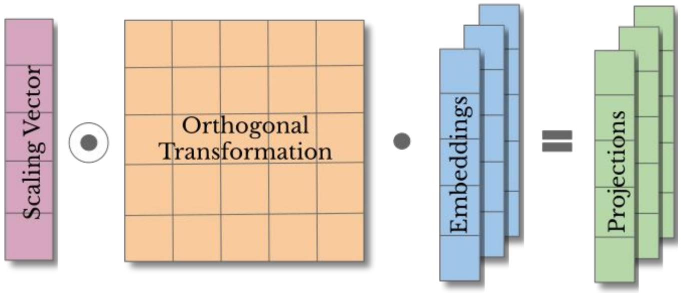


Structural Probe

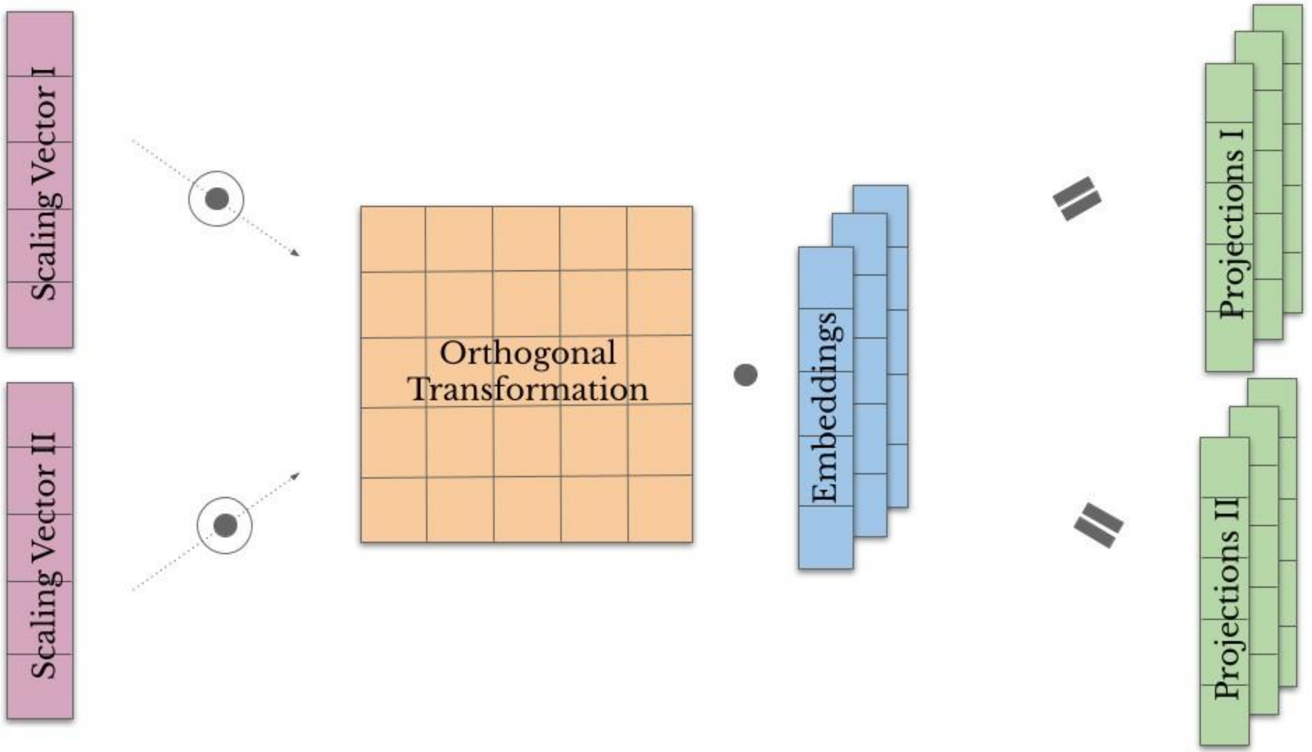


Orthogonal Structural Probe

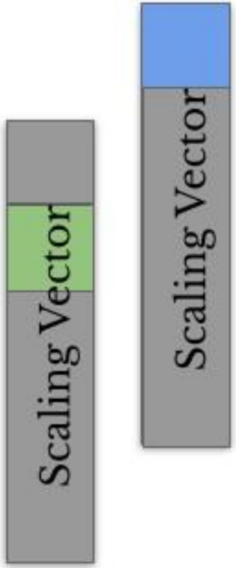




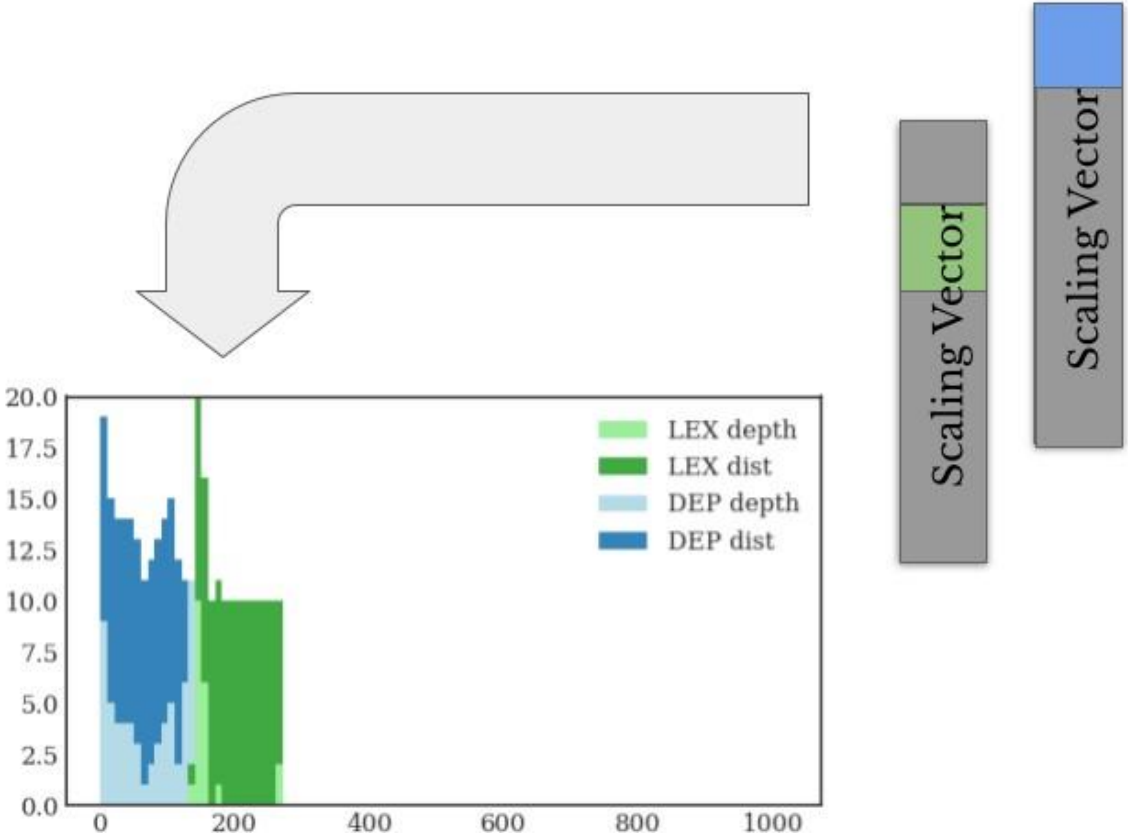
Orthogonal Structural Probe



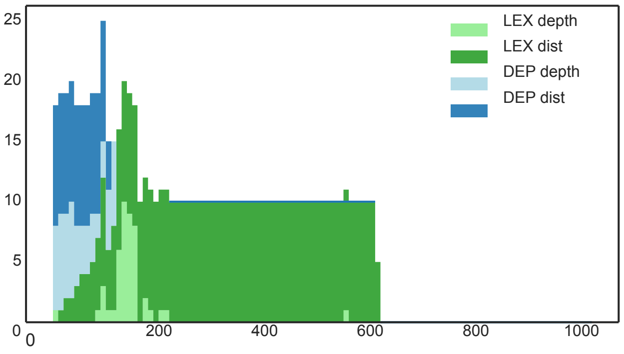
Disentanglement



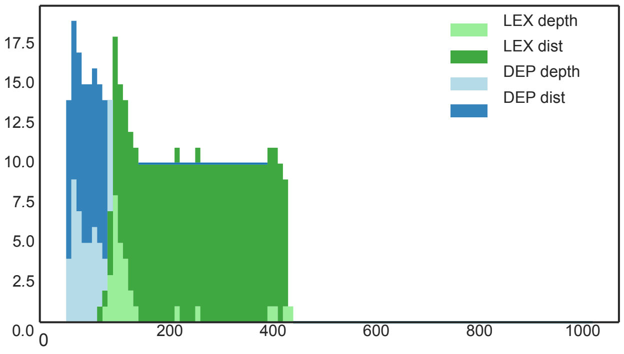
Disentanglement



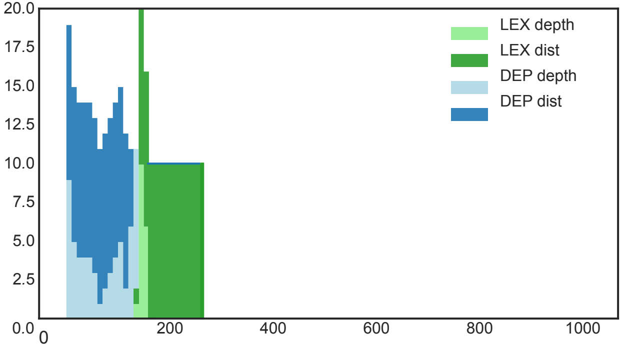
Disentanglement: Syntax and Hypernymy



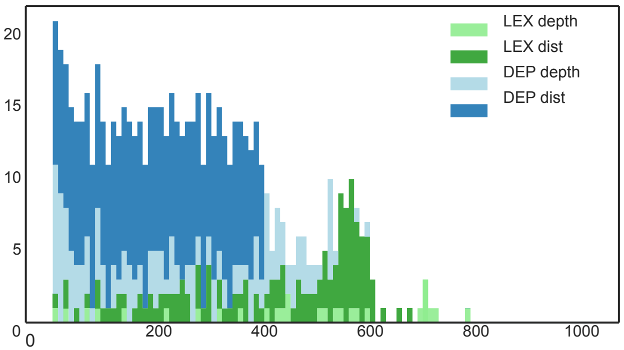
(a) Layer 1



(b) Layer 6

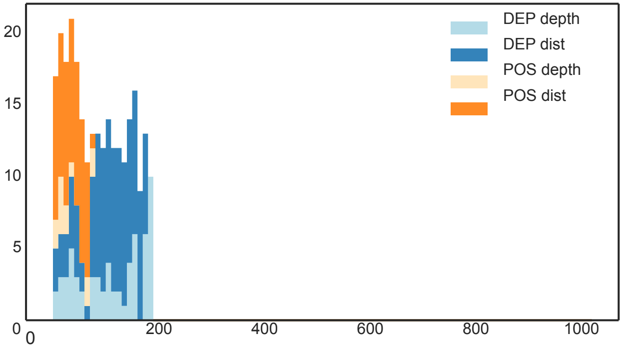


(c) Layer 16

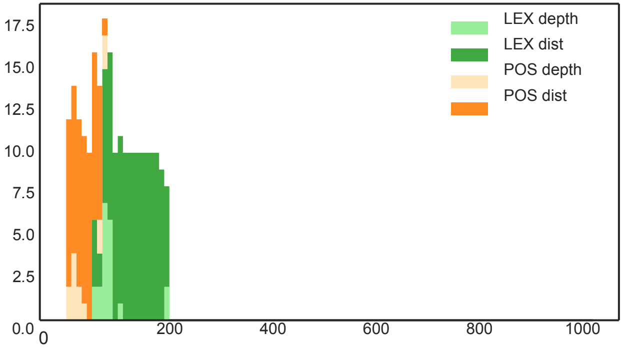


(d) Layer 24

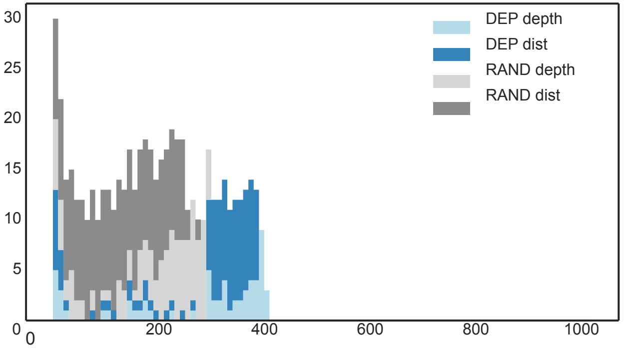
Disentanglement: Other Pairs (16th Layer)



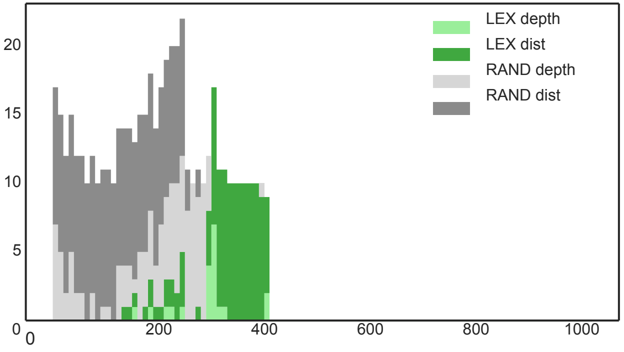
(a) DEP & POS



(b) LEX & POS



(c) DEP & RAND



(d) LEX & RAND

Disentanglement and Rank

		DEP		LEX		POS		RAND	
		Depth	Dist.	Depth	Dist.	Depth	Dist.	Depth	Dist.
DEP	Depth	62	48	0	0	10	19	23	21
	Dist.		126	0	0	9	23	25	30
LEX	Depth			20	18	0	4	1	5
	Dist.				131	0	7	5	19
POS	Depth					14	10	13	10
	Dist.						70	33	50
RAND	Depth							131	95
	Dist.								262

Table: The number of shared dimensions selected by Scaling Vector after the joint training of probe on top of the 16th layer.

Disentanglement and Rank

		DEP		LEX		POS		RAND	
		Depth	Dist.	Depth	Dist.	Depth	Dist.	Depth	Dist.
DEP	Depth	62	48	0	0	10	19	23	21
	Dist.		126	0	0	9	23	25	30
LEX	Depth			20	18	0	4	1	5
	Dist.				131	0	7	5	19
POS	Depth					14	10	13	10
	Dist.						70	33	50
RAND	Depth							131	95
	Dist.								262

Table: The number of shared dimensions selected by Scaling Vector after the joint training of probe on top of the 16th layer.

Summary

- New structural objectives: lexical hypernymy, position in the sentence
- The sufficient rank for a task is self-learned by gradient optimization
- Lexical and dependency structures are encoded in the orthogonal subspaces

Multilingual Analysis

Multilingual Approach

Tomasz Limisiewicz and David Mareček. Examining cross-lingual contextual embeddings with orthogonal structural probes.
In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Association for Computational Linguistics, November 2021a

- Probing for syntactic and lexical information in multilingual representations (mBERT)
- Covers 9 diverse languages
- Motivation: How similar are the representations across languages?

Examining Cross-lingual Contextual Embeddings with Orthogonal Structural Probes

Tomasz Limisiewicz and **David Mareček**
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
{limisiewicz, marecek}@ufal.mff.cuni.cz

Abstract

State-of-the-art contextual embeddings are obtained from large language models available only for a few languages. For others, we need to learn representations using a multilingual model. There is an ongoing debate on whether multilingual embeddings can be aligned in a space shared across many languages. The novel *Orthogonal Structural Probe* (Limisiewicz and Mareček, 2021) allows us to answer this question for specific linguistic features and learn a projection based only on mono-lingual annotated datasets. We evaluate syntactic (UD) and lexical (WordNet) structural information encoded in MBERT's contextual representations for nine diverse languages.[1] We observe that for languages closely related to English, no transformation is needed. The evaluated information is encoded in a shared cross-lingual embedding space. For other languages, it is beneficial to apply orthogonal transformation learned separately for each language. We successfully apply our findings to zero-shot and few-shot cross-lingual parsing.

1 Introduction

The representation learned by language models has been successfully applied in various NLP tasks. Multilingual pre-training allows utilizing the representation for various languages, including low-resource ones. There is an open discussion about

We probe for the syntactic and lexical structures encoded in multilingual embeddings with the new *Orthogonal Structural Probes* (Limisiewicz and Mareček, 2021). Previously, Chi et al. (2020) employed *structural probing* (Hewitt and Manning, 2019) to evaluate cross-lingual syntactic information in MBERT and visualize how it is distributed across languages. Our approach's advantage is learning an orthogonal transformation that maps the embeddings across languages based on mono-lingual linguistic information: dependency syntax and lexical hypernymy. This new capability allows us to test different probing scenarios. We measure how adding assumptions of isomorphism and uniformity of the representations across languages affect probing results to answer our research questions.

2 Related Work

Probing It is a method of evaluating linguistic information encoded in pre-trained NLP models. Usually, a simple classifier for the probing task is trained on the frozen model's representation (Linzen et al., 2016; Belinkov et al., 2017; Blevins et al., 2018). The work of Hewitt and Manning (2019) introduced structural probes that linearly transform contextual embeddings to approximate the topology of dependency trees. Limisiewicz and Mareček (2021) proposed new structural tasks and introduced orthogonal constraint allowing to

How the Representation Vary Across Languages?

- To what extent embeddings are similar across languages. What can affect this similarity Vulić et al. (2020)
    - Is language signal encoded uniformly across languages?
    - Will applying orthogonal map improve cross-lingual transfer?
- We can study relations between languages based on the multilingual probes Chi et al. (2020)

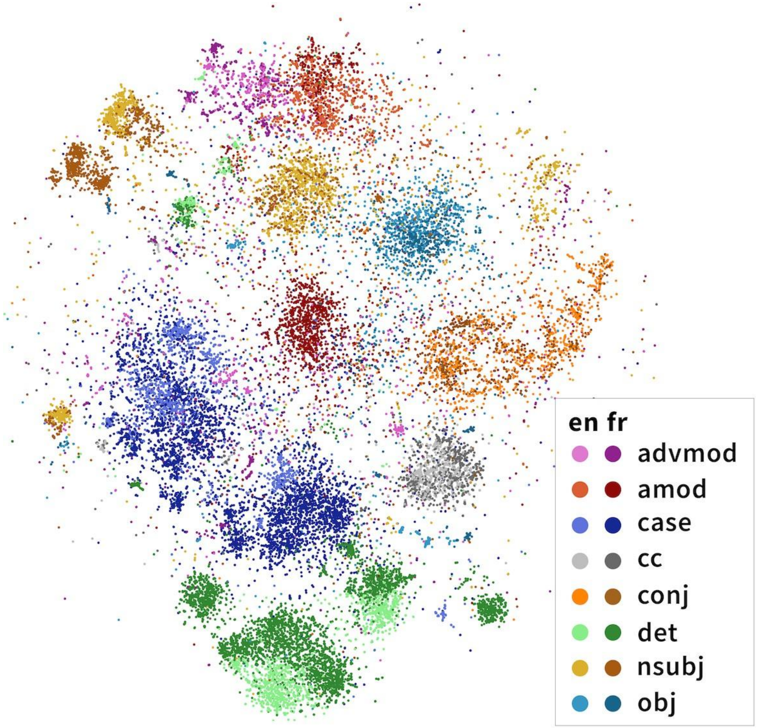
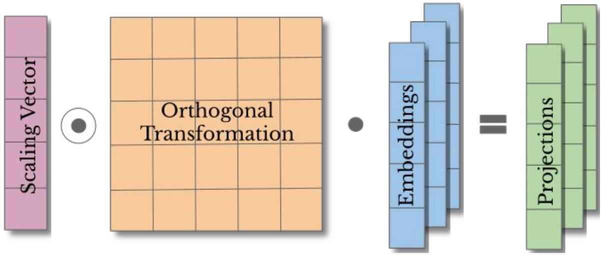


Figure: Visualization of multilingual representation (Chi et al., 2020)

Multilingual Approach

Our approaches and corresponding assumptions about the likeness of the cross-lingual embeddings:

- **In-Lang no assumption** We train a separate instance of a probe for each language.

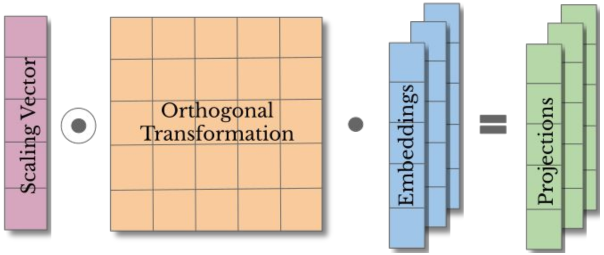


Multilingual Approach

Our approaches and corresponding assumptions about the likeness of the cross-lingual embeddings:

- **In-Lang no assumption** We train a separate instance of a probe for each language.

- **MappedLangs isomorphity assumption** We train a shared *Scaling Vector* for each probing task and a separate *Orthogonal Transformation* per language.



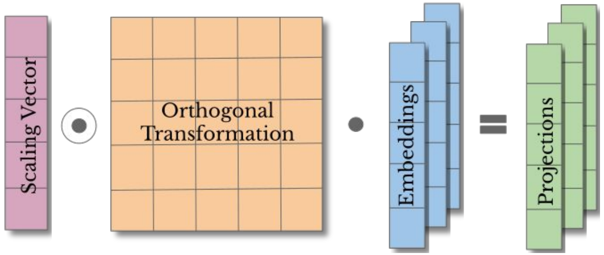
Multilingual Approach

Our approaches and corresponding assumptions about the likeness of the cross-lingual embeddings:

- **In-Lang no assumption** We train a separate instance of a probe for each language.

- **MappedLangs isomorphity assumption** We train a shared *Scaling Vector* for each probing task and a separate *Orthogonal Transformation* per language.

- **AllLangs uniformity assumption** Both the *Scaling Vector* and *Orthogonal Transformation* are shared across languages.



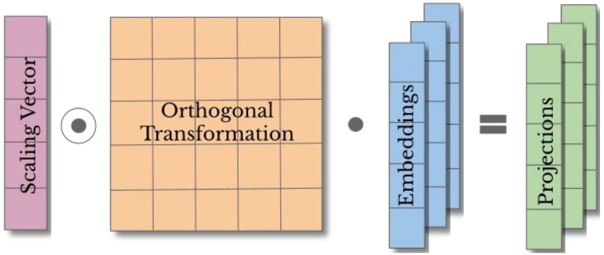
Multilingual Approach

Our approaches and corresponding assumptions about the likeness of the cross-lingual embeddings:

- **In-Lang no assumption** We train a separate instance of a probe for each language.

- **MappedLangs isomorphity assumption** We train a shared *Scaling Vector* for each probing task and a separate *Orthogonal Transformation* per language.
  COMPARABLE PERFORMANCE

- **AllLangs uniformity assumption** Both the *Scaling Vector* and *Orthogonal Transformation* are shared across languages.
  PERFORMANCE DEPENDS ON
  TYPOLOGICAL DIFFERENCES



Results for Dependency Probes

Approach	EN	ES	SL	ID	ZH	FI	AR	FR	EU	AVERAGE Indo-Eur	AVERAGE Other
Dependency Distance Spearman's Correlation											
IN-LANG	.812	.858	.857	.841	.830	.788	.838	.856	.769	.846	.813
Δ MAPPEDL	.000	-.001	.001	-.003	.000	.001	-.001	-.002	.001	-.001	.000
Δ ALLL	.000	-.007	-.006	-.013	-.039	.000	-.027	-.006	-.032	-.005	-.022
Dependency Depth Spearman's Correlation											
IN-LANG	.843	.868	.867	.855	.844	.822	.865	.877	.797	.864	.837
Δ MAPPEDL	-.004	-.003	-.002	-.002	.000	-.002	.001	-.002	-.001	-.002	-.001
Δ ALLL	-.006	-.007	-.008	-.011	-.035	-.005	-.031	-.010	-.031	-.008	-.023

Results for Lexical Probes

Approach	EN	ES	SL	ID	ZH	FI	AR	FR	EU	AVERAGE Indo-Eur	AVERAGE Other
Lexical Distance Spearman's Correlation											
In-Lang	.756	.841	.639	.719	.800	.657	.733	.794	.679	.757	.717
Δ MappedL	-.003	.005	-.011	-.001	.010	.001	.042	.001	-.008	-.002	.009
Δ AllL	-.038	-.025	-.042	-.051	-.014	-.043	.025	-.013	-.063	-.030	-.029
Lexical Depth Spearman's Correlation											
In-Lang	.853	.881	.779	.852	.875	.784	.906	.844	.842	.839	.850
Δ MappedL	.004	-.005	.013	-.011	.006	.023	-.024	.007	.021	.004	.005
Δ AllL	-.027	-.048	-.040	-.124	-.068	-.006	-.305	-.032	-.020	-.037	-.103

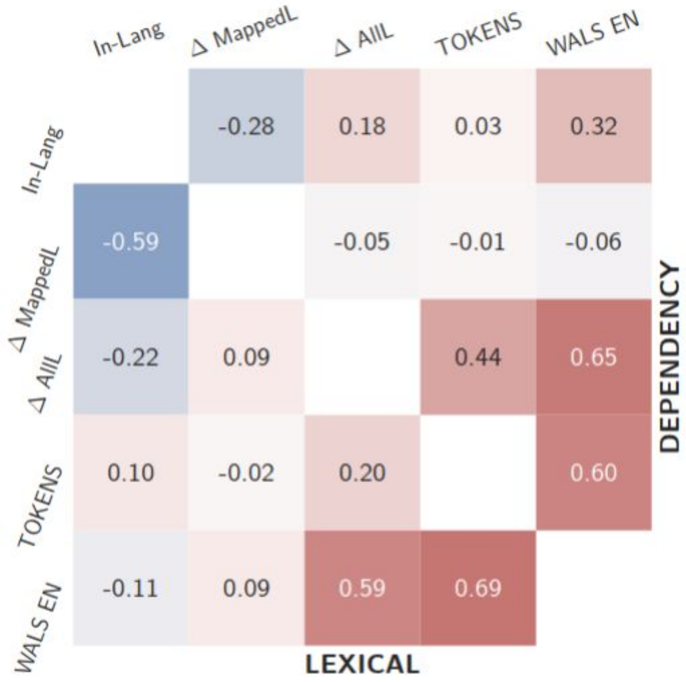
LANGUAGE SPECIFIC:

TOKENS number of tokens used in mBERT pre-training for a language

WALS EN Hamming (string) similarity between features in WALS

PROBING RESULTS:

In-Lang (no assumption)
MappedLangs (isomorphity assumption)
AllLangs (uniformity assumption)



Trends

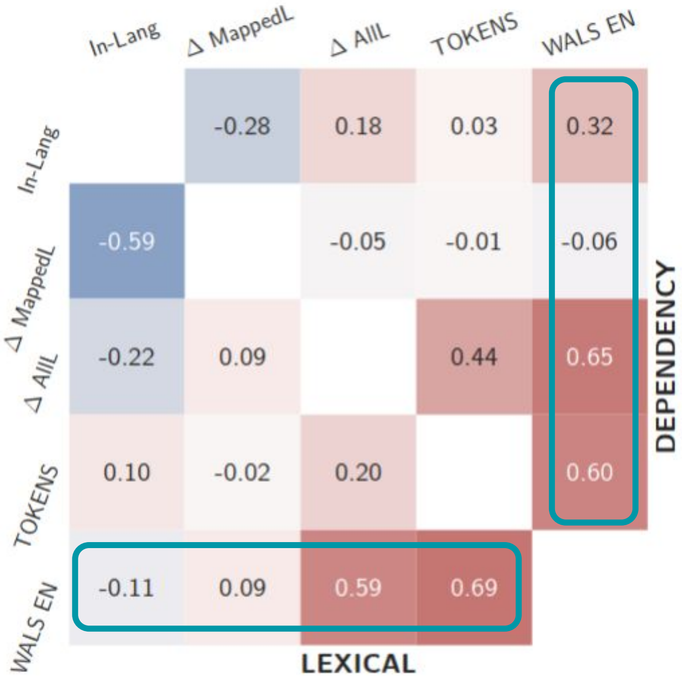
LANGUAGE SPECIFIC:

TOKENS number of tokens used in mBERT pre-training for a language

WALS EN Hamming (string) similarity between features in WALS

PROBING RESULTS:

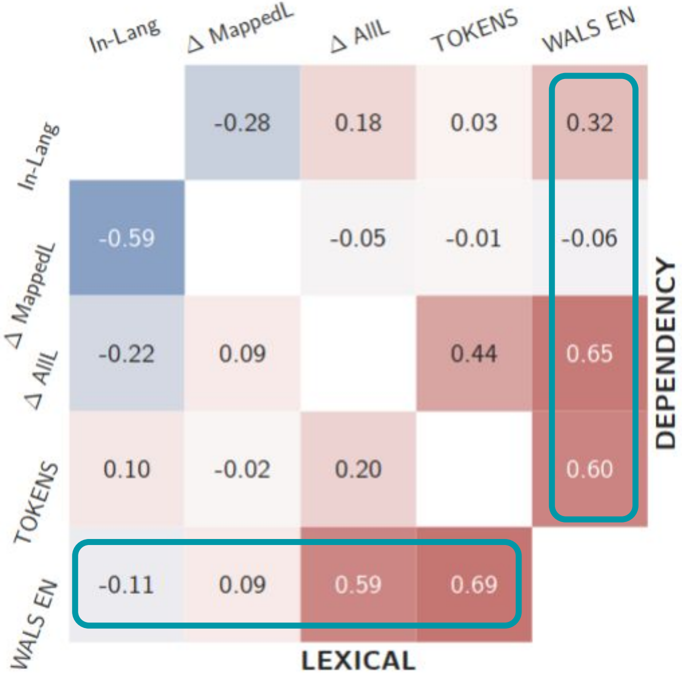
In-Lang (no assumption)
MappedLangs (isomorphity assumption)
AllLangs (uniformity assumption)



Trends

Syntactic and lexical information is **uniformly encoded** across **mBERT**'s representations of languages similar to English.

For other languages, the **orthogonal mapping** can improve results.



Criticism of Probing

Over-Fitting to Data

Hewitt and Liang (2019), optimize the probe to classify artificially assigned tags **(control task)**. The tags are assigned by random but have the same distribution as POS tags.

They define **selectivity** as the difference of accuracy on a control and a linguistic tasks.

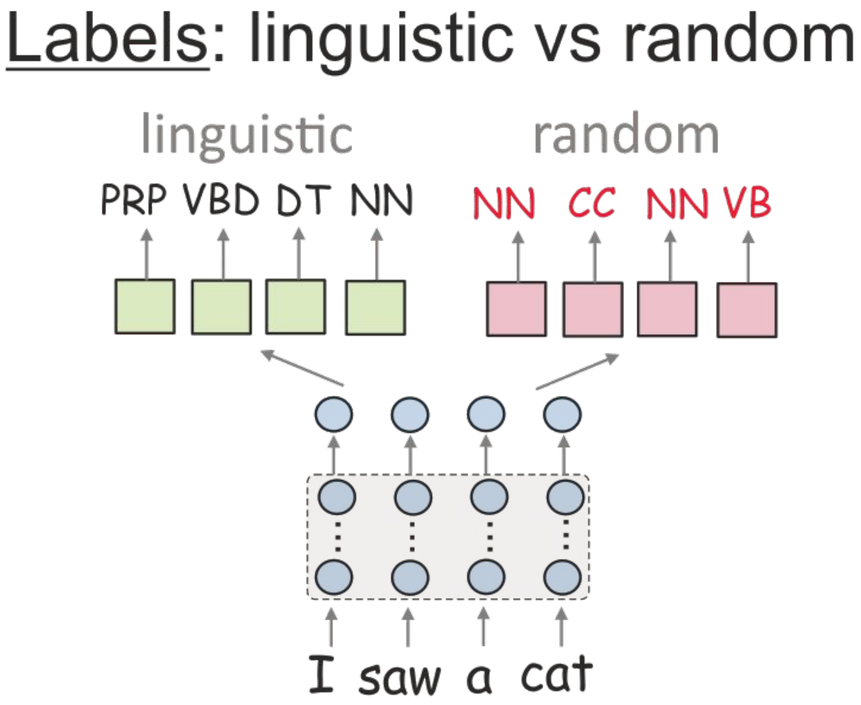


Figure from Lena Voita's blog

Over-Fitting to Data

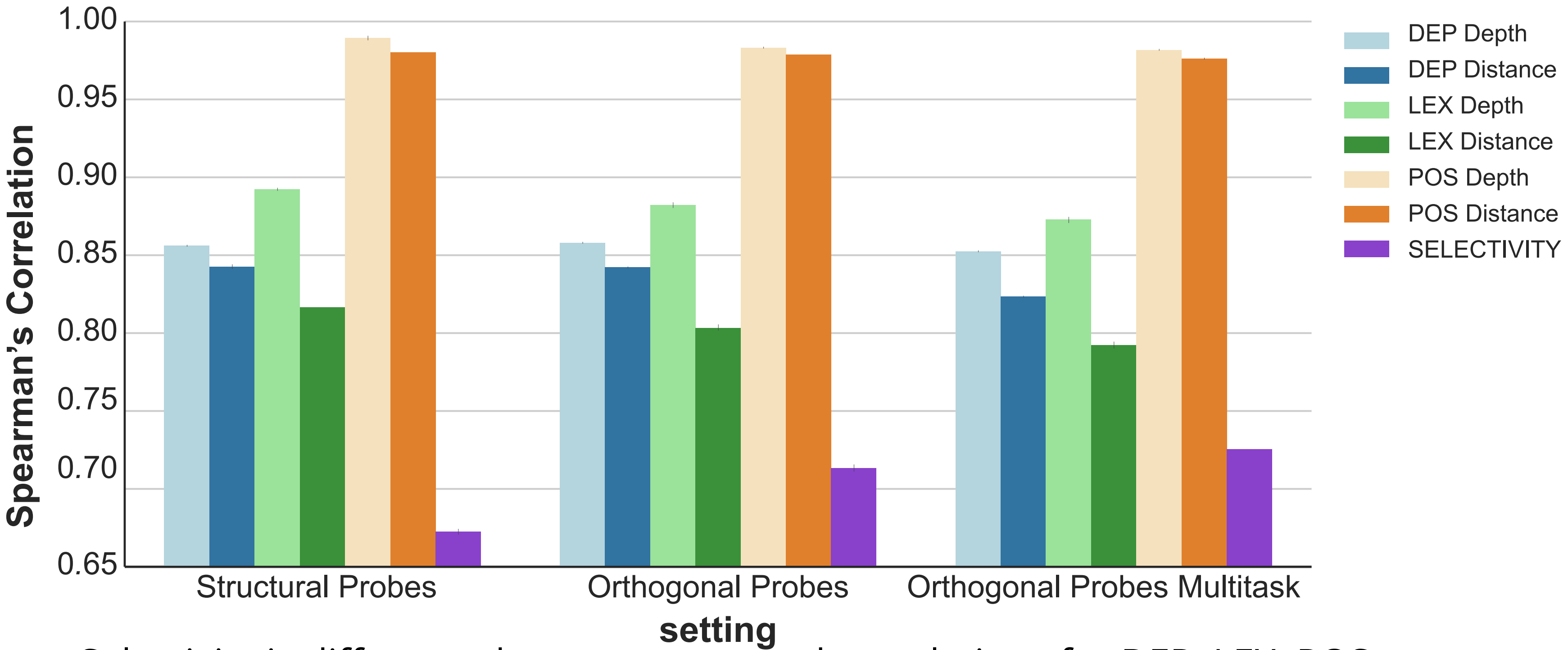
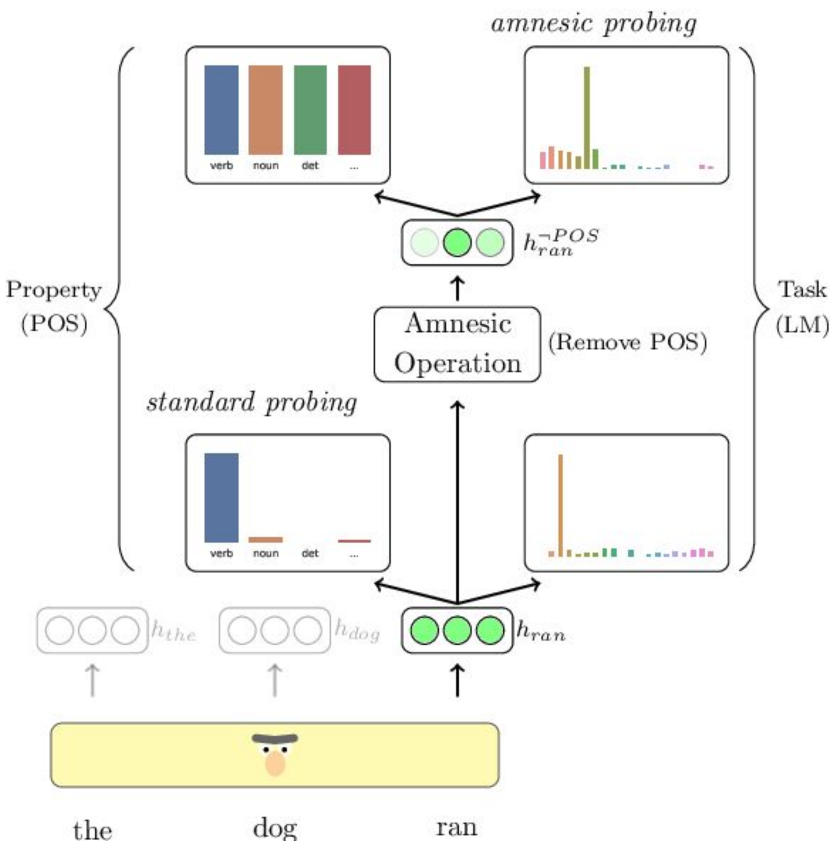


Figure: Selectivity is difference between averaged correlations for DEP, LEX, POS structures and RAND

Is Probed Information Really Useful for LM

Elazar et al. (2021) argue that to explain the model's behaviour we should identify the information that is used rather than the information that is encoded by the model.

They propose **Amnesic Probing**: selectively remove information encoded in the representation and observe the change in the performance on the main task (language modeling).



Controlling Bias with Probes

Bias in the Model

Understanding how knowledge is encoded in neural networks can help combat unwanted behaviors, such as predictions based on spurious correlations ~ **bias**

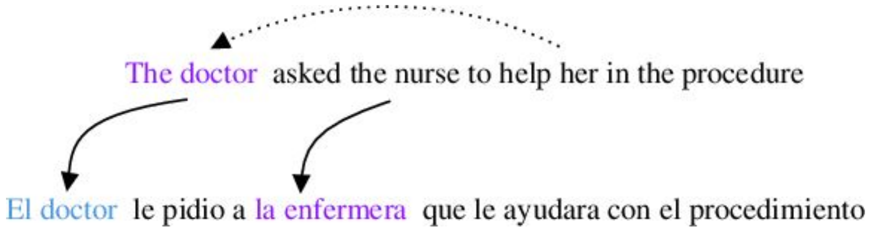
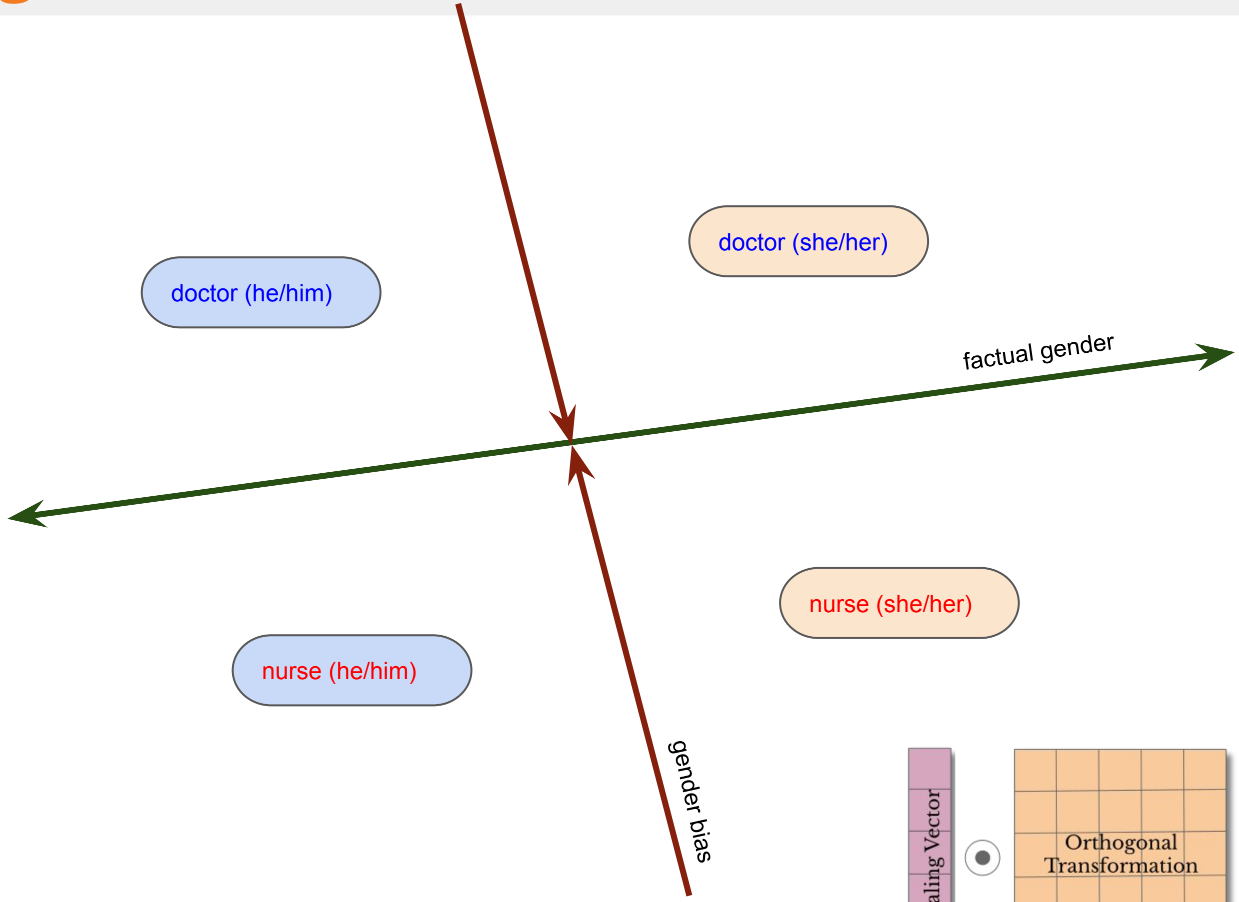
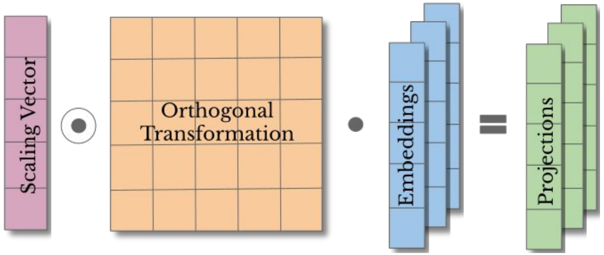


Figure : Probable manifestation of **gender bias** in Machine Translation Stanovsky et al. (2019)

Controlling Bias with Probes





Interpreting Attention

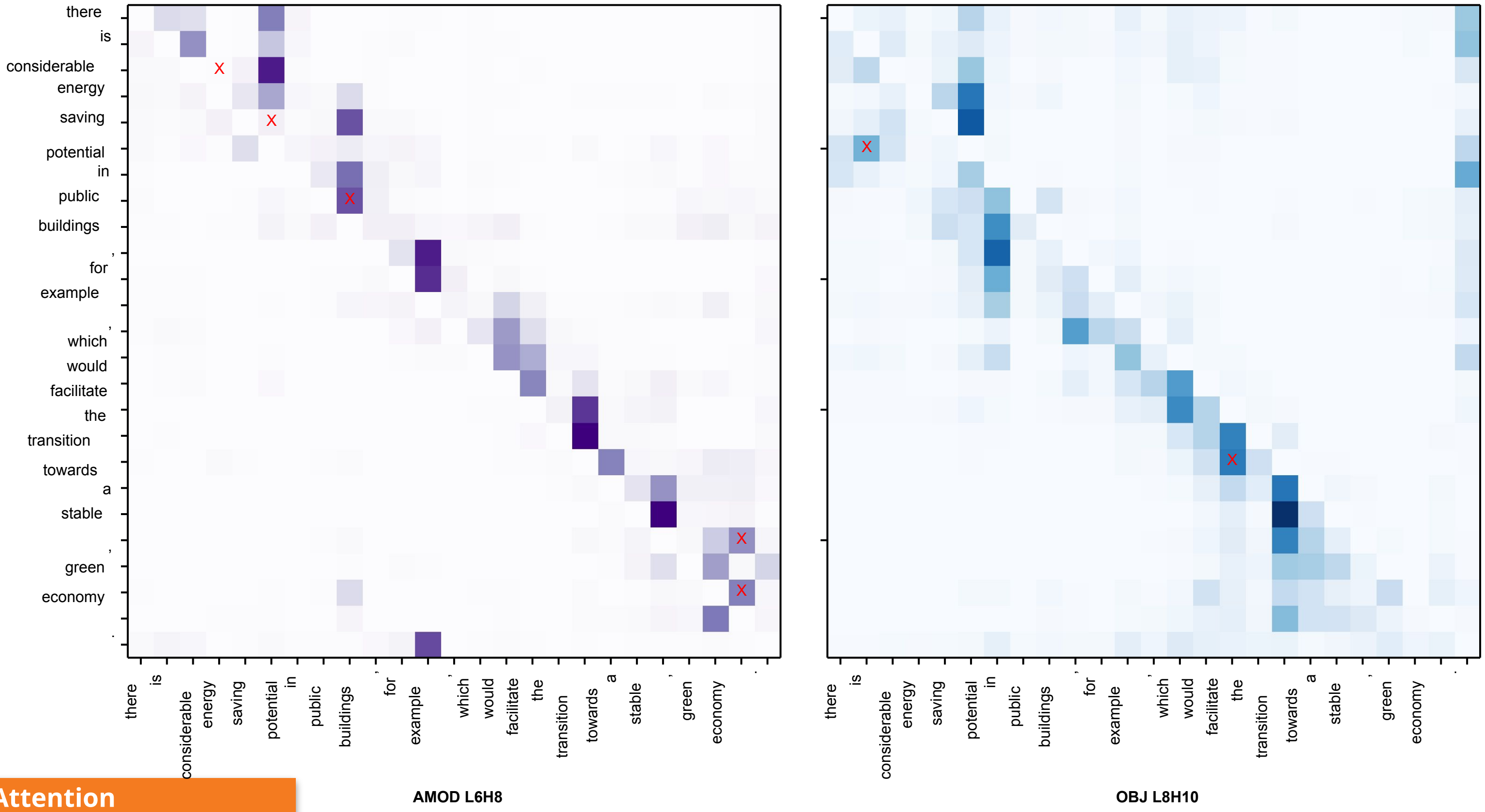
Interpreting Attention: Background

Past works:

- Vig and Belinkov 2019 showed that in some language model (GPT-2) heads attention is higher for pairs of tokens that are in a specific dependency relation.
- Raganato and Tiedemann 2018 induce dependency trees from each self-attention matrix of Transformer with maximum spanning tree algorithm. They obtain the trees which are on pair with right-branching chains.
- Clark et al. 2019 uses weighted average of all heads of language model (BERT) to induce dependency tree. This method gives much better results than using each single head.

BERT and Dependency Relations

Self-attention in a particular heads of a language model aligns with dependency relations



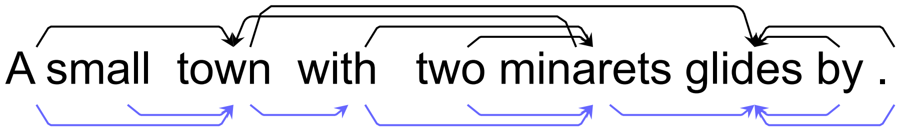
BERT and Dependency Relations

Previous works showed that individual BERT attention heads tend to encode particular dependency relations.

We identify:

- Abstract heads (encode dependency of multiple labels)
- Specific heads (separate one relation type into multiple subtypes)

We show a method how to extract labeled dependency trees (52% UAS, 22% LAS on English UD).



Closing Remarks

ÚFAL at Charles University

Institute of Formal and Applied Linguistics (ÚFAL)

- Established in 1990 (beginnings in the 60s)
- 20 Academic Staff and 29 Researchers
- 41 Ph.D. Students
- Research cluster: >2000 CPUs; >100 GPUs

Picture by Ondra Dušek

My Collaborators

David Mareček

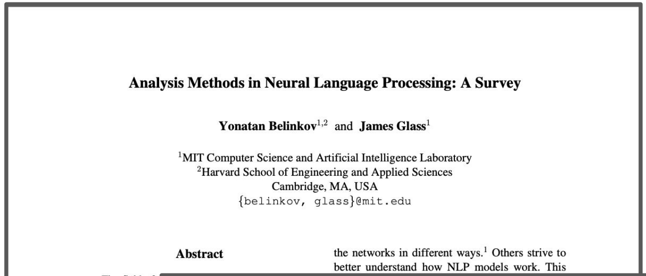
Jindřich Libovický

Rudolf Rosa

Tomáš Musil

Tomasz Limisiewicz

References



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

Abstract

the networks in different ways.[1] Others strive to better understand how NLP models work. This

A Primer in BERTology: What We Know About How BERT Works

Anna Rogers
Center for Social Data Science
University of Copenhagen
arogers@sodas.ku.dk

Olga Kovaleva
Dept. of Computer Science
University of Massachusetts Lowell
okovalev@cs.uml.edu

Anna Rumshisky
Dept. of Computer Science
University of Massachusetts Lowell
arum@cs.uml.edu

Abstract

Transformer-based models have pushed state of the art in many areas of NLP, but our understanding of what is behind their success is still limited. This paper is the first survey of over 150 studies of the popular BERT model. We review the current state of knowledge about how BERT works, what kind of information it learns and how it is represented, common modifications to its training objectives and architecture, the overparameterization issue, and approaches to compression. We then outline directions for future research.

1 Introduction

Since their introduction in 2017, Transformers (Vaswani et al., 2017) have taken NLP by storm, offering enhanced parallelization and better modeling of long-range dependencies. The best known Transformer-based model is BERT (Devlin et al., 2019); it obtained state-of-the-art results in numerous benchmarks and is still a must-have baseline.

Although it is clear that BERT works remarkably well, it is less clear *why*, which limits further hypothesis-driven improvement of the architecture. Unlike CNNs, the Transformers have little cognitive motivation, and the size of these models limits our ability to experiment with pre-training and perform ablation studies. This explains a large

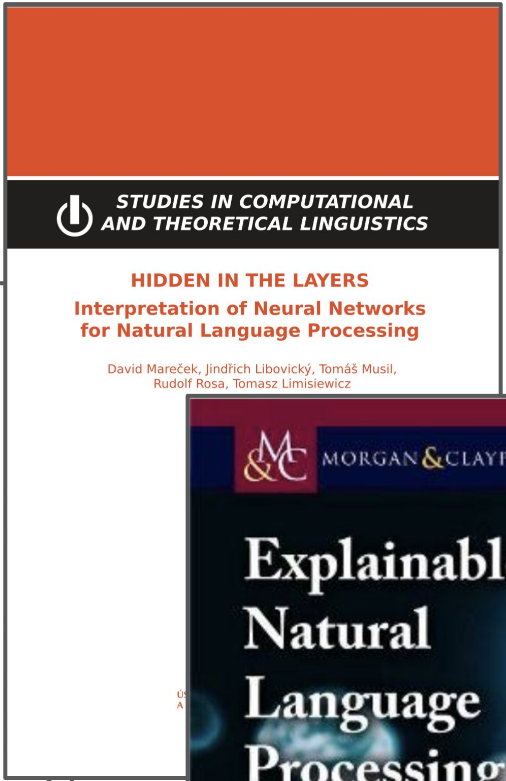
and provide an overview of the current proposals to improve BERT's architecture, pre-training, and fine-tuning. We conclude by discussing the issue of overparameterization, the approaches to compressing BERT, and the nascent area of pruning as a model analysis technique.

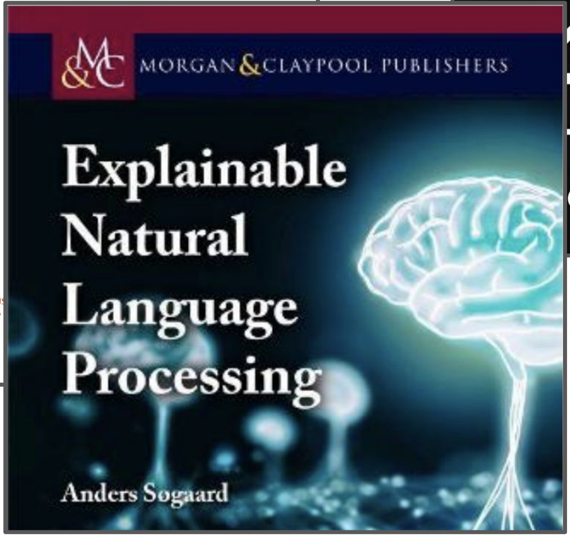
2 Overview of BERT Architecture

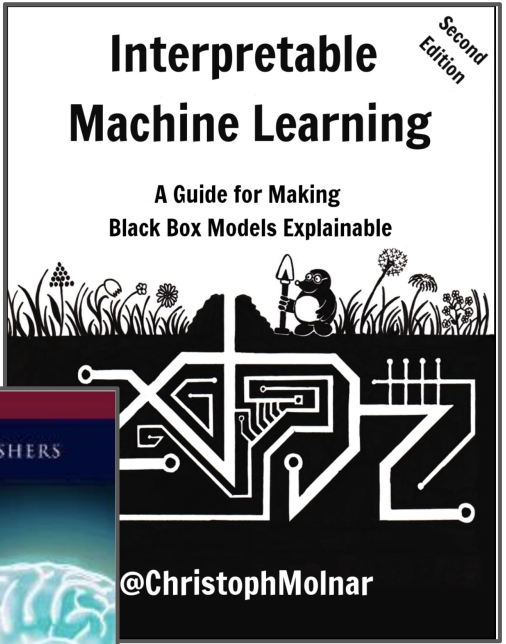
Fundamentally, BERT is a stack of Transformer encoder layers (Vaswani et al., 2017) that consist of multiple self-attention ''heads''. For every input token in a sequence, each head computes key, value, and query vectors, used to create a weighted representation. The outputs of all heads in the same layer are combined and run through a fully connected layer. Each layer is wrapped with a skip connection and followed by layer normalization.

The conventional workflow for BERT consists of two stages: pre-training and fine-tuning. Pre-training uses two self-supervised tasks: masked language modeling (MLM, prediction of randomly masked input tokens) and next sentence prediction (NSP, predicting if two input sentences are adjacent to each other). In fine-tuning for downstream applications, one or more fully connected layers are typically added on top of the final encoder layer.

The input representations are computed as follows: Each word in the input is first tokenized







Thank you!

🐦 @TomLimi

✉️ limisiewicz@ufal.mff.cuni.cz