Probing for Labeled Dependency Trees

Max Müller-Eberstein and **Rob van der Goot** and **Barbara Plank**
Department of Computer Science
IT University of Copenhagen, Denmark
mamy@itu.dk, robv@itu.dk, bapl@itu.dk

Abstract

Probing has become an important tool for analyzing representations in Natural Language Processing (NLP). For graphical NLP tasks such as dependency parsing, linear probes are currently limited to extracting undirected or unlabeled parse trees which do not capture the full task. This work introduces DEPPROBE, a linear probe which can extract *labeled* and *directed* dependency parse trees from embeddings while using fewer parameters and compute than prior methods. Leveraging its full task coverage and lightweight parametrization, we investigate its predictive power for selecting the best transfer language for training a full biaffine attention parser. Across 13 languages, our proposed method identifies the best source treebank 94% of the time, outperforming competitive baselines and prior work. Finally, we analyze the informativeness of task-specific subspaces in contextual embeddings as well as which benefits a full parser's non-linear parametrization provides.

1 Introduction

Pre-trained, contextualized embeddings have been found to encapsulate information relevant to various syntactic and semantic tasks out-of-the-box (Tenney et al., 2019; Hewitt and Manning, 2019). Quantifying this latent information has become the task of *probes* — models which take frozen embeddings as input and are parametrized as lightly as possible (e.g. linear transformations). Recent proposals for edge probing (Tenney et al., 2019) and structural probing (Hewitt and Manning, 2019) have enabled analyses beyond classification tasks, including graphical tasks such as dependency parsing. They are able to extract dependency graphs from embeddings, however these are either undirected (Hewitt and Manning, 2019; Hall Maudslay et al., 2020) or unlabeled (Kulmizev et al., 2020), thereby capturing only a subset of the full task.

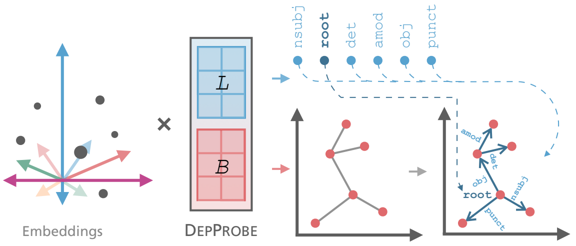


Figure 1: **DEPPROBE** extracts tree structure using transformation B, labels using L and infers directionality using `root`, based on contextualized embeddings.

In this work, we investigate whether this gap can be filled and ask: *Can we construct a lightweight probe which can produce fully directed and labeled dependency trees?* Using these trees, we further aim to study the less examined problem of transferability estimation for graphical tasks, extending recent work targeting classification and regression tasks (Nguyen et al., 2020; You et al., 2021). Specifically: *How well do our probe's predictions correlate with the transfer performance of a full parser across a diverse set of languages?*

To answer these questions, we contribute DEPPROBE (Figure 1), the first linear probe to extract directed and labeled dependency trees while using fewer parameters than prior work and three orders of magnitude fewer trainable parameters than a full parser (Section 3). As this allows us to measure labeled attachment scores (LAS), we investigate the degree to which our probe is predictive of cross-lingual transfer performance of a full parser across 13 typologically diverse languages, finding that our approach chooses the best transfer language 94% of the time, outperforming competitive baselines and prior work (Section 4). Finally, we perform an in-depth analysis of which latent information is most relevant for dependency parsing as well as which edges and relations benefit most from the expressivity of the full parser (Section 5).[1]

[1] Code available at https://personads.me/x/acl-2022-code.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7711 - 7726
May 22-27, 2022 ©2022 Association for Computational Linguistics

2 Related Work

Given the ubiquitous use of contextualized embeddings (Devlin et al., 2019; Conneau et al., 2020; Xue et al., 2021), practitioners have turned to various methods for analyzing their linguistic features (Rogers et al., 2020). Hewitt and Manning (2019) examine these intrinsic properties in greater detail for English dependency parsing using a *structural probe*, finding that *undirected* dependency graphs are recoverable from BERT by learning a linear transformation on its embeddings (Section 3.1).

Extending the structural probe of Hewitt and Manning (2019) to 12 languages, Chi et al. (2020) extract *undirected* dependency graphs from mBERT (Devlin et al., 2019), further showing that head-to-child difference vectors in the learned subspace cluster into relations from the Universal Dependencies taxonomy (de Marneffe et al., 2014).

Building on both the structural and tree depth probes (Hewitt and Manning, 2019), Kulmizev et al. (2020) extract *directed* dependency graphs from mBERT for 13 languages (Section 3.2). Further variations to structural probing include regularization of the linear transformation (Limisiewicz and Mareček, 2021) as well as alternative objective functions (Hall Maudslay et al., 2020).

None of the proposed linear probing approaches so far are able to produce full dependency parse trees (i.e. directed and labeled), however the closer a probe approximates the full task, the better it quantifies relevant information (Hall Maudslay et al., 2020). It would for example be desirable to estimate LAS for parsing a target treebank with a model trained on a different source without having to train a resource-intensive parser (e.g. Dozat and Manning, 2017) on each source candidate. Although performance prediction methods for such scenarios exist, they typically do not cover graph prediction (Nguyen et al., 2020; You et al., 2021).

In order to bridge the gap between full parsers and unlabeled probes, in addition to the gap between full fine-tuning and lightweight performance prediction, this work proposes a linear probe which can extract *labeled* and *directed* dependency parse trees while using less compute than prior methods (Section 3). We use our probe's LAS to evaluate its predictive power for full parser performance and leverage its linear nature to investigate how dependencies are represented in subspaces of contextual embeddings (Section 5).

3 Probing for Dependencies

In order to construct a directed and labeled dependency parse tree for a sentence s consisting of the words $\{w_0, \dots, w_N\}$, we require information on the presence or absence of edges between words, the directionality of these edges $(\overrightarrow{w_i, w_j})$, and the relationships $\{r_0, \dots, r_N\}$ which they represent. Using the contextualized embeddings $\{\boldsymbol{h}_0, \dots, \boldsymbol{h}_N\}$ with $\boldsymbol{h}_i \in \mathbb{R}^e$, prior probing work has focused on the first step of identifying edges (Section 3.1) and later directionality (Section 3.2). In this work, we propose a probe which completes the final relational step (Section 3.3) and simultaneously provides a more efficient method for identifying directionality (Section 3.4).

3.1 Undirected Probing

The structural probe introduced by Hewitt and Manning (2019) recovers the first piece of information (i.e. the undirected graph) remarkably well. Here, the probe is a linear transformation $B \in \mathbb{R}^{e \times b}$ with $b < e$ which maps contextual embeddings into a subspace in which the distance measure

$$d_B(\boldsymbol{h}_i, \boldsymbol{h}_j) = \sqrt{(B\boldsymbol{h}_i - B\boldsymbol{h}_j)^T (B\boldsymbol{h}_i - B\boldsymbol{h}_j)} \tag{1}$$

between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ is optimized towards the distance between two words in the dependency graph $d_P(w_i, w_j)$, i.e. the number of edges between the words. For each sentence, the loss is defined as the mean absolute difference across all word pairs:

$$\mathcal{L}_B(s) = \frac{1}{N^2} \sum_{i=0}^{N} \sum_{j=0}^{N} \left| d_P(w_i, w_j) - d_B(\boldsymbol{h}_i, \boldsymbol{h}_j) \right| . \tag{2}$$

In order to extract an undirected dependency graph, one computes the distances for a sentence's word pairs using d_B and extracts the minimum spanning tree (Jarník, 1930; Prim, 1957; MST).

3.2 Directed Probing

Apart from the structural probe B, Hewitt and Manning (2019) also probe for tree depth. Using another matrix $C \in \mathbb{R}^{e \times c}$, a subspace is learned in which the squared L_2 norm of a transformed embedding $||C\boldsymbol{h}_i||_2^2$ corresponds to a word's depth in the tree, i.e. the number of edges from the root.

Kulmizev et al. (2020) combine the structural and tree depth probe to extract directed graphs.

This directed probe (DirProbe) constructs a score matrix $M \in \mathbb{R}^{N \times N}$ for which each entry corresponds to a word pair's negative structural distance $-d_B(\boldsymbol{h}_i, \boldsymbol{h}_j)$. The shallowest node in the depth subspace C is set as root. Entries in M which correspond to an edge between w_i and w_j for which the word depths follow $\|C\boldsymbol{h}_i\|_2^2 > \|C\boldsymbol{h}_j\|_2^2$ are set to $-\infty$. A word's depth in subspace C therefore corresponds to edge directionality. The directed graph is built from M using Chu-Liu-Edmonds decoding (Chu and Liu, 1965; Edmonds, 1967).

DirProbe extracts directed dependency parse trees, however it would require additional complexity to label each edge with a relation (e.g. using an additional probe). In the following, we propose a probe which can extract both directionality and relations while using fewer parameters and no dynamic programming-based graph-decoding algorithm.

3.3 Relational Probing

The incoming edge of each word w_i is governed by a single relation. As such the task of dependency relation classification with l relations can be simplified to a labeling task using a linear transformation $L \in \mathbb{R}^{e \times l}$ for which the probability of a word's relation r_i being of class l_k is given by:

$$p(r_i = l_k | w_i) = \text{softmax}(L\boldsymbol{h}_i)_k \qquad (3)$$

and optimization uses standard cross-entropy loss given the gold label r_i^* for each word w_i:

$$\mathcal{L}_L(s) = -\frac{1}{N} \sum_{i=0}^{N} \ln p(r_i^* | w_i) \ . \qquad (4)$$

Should dependency relations be encoded in contextualized embeddings, each dimension of the subspace L will correspond to the prevalence of information relevant to each relation, quantifiable using relation classification accuracy (RelAcc).

3.4 Constructing Dependency Parse Trees

Combining structural probing (Section 3.1) and dependency relation probing (Section 3.3), we propose a new probe for extracting fully directed and labeled dependency trees (DepProbe). It combines undirected graphs and relational information in a computationally efficient manner, adding labels while requiring *less* parameters than prior unlabeled or multi-layer-perceptron-based approaches.

Algorithm 1: DepProbe Inference

```
1  input Distance matrix $D_B \in \mathbb{R}^{N \times N}$,
      $p(l_k|w_i)$ of relation label $l_k$ given $w_i$
2  $w_r \leftarrow \underset{w_i}{\text{argmax}}\ p(\text{root}|w_i)$
3  $\mathcal{T}_w \leftarrow \{w_r\}, \mathcal{T}_e \leftarrow \{\}$
4  while $|\mathcal{T}_w| < N$ do
5      $w_i, w_j \leftarrow \underset{w_i, w_j}{\text{argmin}}\ D_B(w_i \in \mathcal{T}_w, w_j)$
6      $r_j \leftarrow \underset{l_k}{\text{argmax}}\ p(l_k|w_j)$ with $l_k \neq$ root
7      $\mathcal{T}_w \leftarrow \mathcal{T}_w \cup \{w_j\}$
8      $\mathcal{T}_e \leftarrow \mathcal{T}_e \cup \{(\overrightarrow{w_i, w_j}, r_j)\}$
9  end
10 return $\mathcal{T}_e$
```

As outlined in Algorithm 1 and illustrated in Figure 1, DepProbe uses the distance matrix D_B derived from the structural probe B in conjunction with the relation probabilities of the relational probe L (line 1). The graph is first rooted using the word w_r for which $p(\text{root}|w_r)$ is highest (line 2). Iterating over the remaining words until all w_j are covered in $\mathcal{T}_w$, an edge is drawn to each word w_j from its head w_i based on the minimum distance in D_B. The relation r_j for an edge $(\overrightarrow{w_i, w_j}, r_j)$ is determined by taking the relation label l_k which maximizes $p(r_j = l_k|w_j)$ with $l_k \neq$ root (line 6). The edge is then added to the set of labeled tree edges $\mathcal{T}_e$. With edge directionality being inferred as simply pointing away from the root, this procedure produces a dependency graph that is both directed and labeled without the need for additional complexity, running in $\mathcal{O}(n^2)$ while dynamic programming-based decoding such as DirProbe have runtimes of up to $\mathcal{O}(n^3)$ (Stanojević and Cohen, 2021).

Constructing dependency trees from untuned embeddings requires the matrices B and L, totaling $e \cdot b + e \cdot l$ trainable parameters. Optimization can be performed using gradient descent on the sum of losses $\mathcal{L}_B + \mathcal{L}_L$. With $l = 37$ relations in UD, this constitutes a substantially reduced training effort compared to prior probing approaches (with subspace dimensionalities b and c typically set to 128) and multiple magnitudes fewer fine-tuned parameters than for a full biaffine attention parser.

4 Experiments

4.1 Setup

Parsers In our experiments, we use the deep biaffine attention parser (BAP) by Dozat and Manning (2017) as implemented in van der Goot et al. (2021) as an upper bound for MLM-based parsing performance. As it is closest to our work, we further reimplement DIRPROBE (Kulmizev et al., 2020) with $b = 128$ and $c = 128$. Note that this approach produces directed, but unlabeled dependency graphs. Finally, we compare both methods to our directed and labeled probing approach, DEPPROBE with $b = 128$ and $l = 37$.

All methods use mBERT (Devlin et al., 2019) as their encoder ($e = 768$). For BAP, training the model includes fine-tuning the encoder's parameters, while for both probes they remain fixed and only the linear transformations are adjusted. This results in 183M tuned parameters for BAP, 197k for DIRPROBE and 127k for DEPPROBE. Hyperparameters are set to the values reported by the authors,[2] while for DEPPROBE we perform an initial tuning step in Section 4.2.

Target Treebanks As targets, we use the set of 13 treebanks proposed by Kulmizev et al. (2019), using versions from Universal Dependencies v2.8 (Zeman et al., 2021). They are diverse with respect to language family, morphological complexity and script (Appendix A). This set further includes EN-EWT (Silveira et al., 2014) which has been used in prior probing work for hyperparameter tuning, allowing us to tune DEPPROBE on the same data.

Metrics We report labeled attachment scores (LAS) wherever possible (BAP, DEPPROBE) and unlabeled attachment scores (UAS) for all methods. For DEPPROBE's hyperparameters, we evaluate undirected, unlabeled attachment scores (UUAS) as well as relation classification accuracy (RelAcc). One notable difference to prior work is that we include punctuation both during training and evaluation — contrary to prior probing work which excludes all punctuation (Hewitt and Manning, 2019; Kulmizev et al., 2020; Hall Maudslay et al., 2020) — since we are interested in the full parsing task.

Training Each method is trained on each target treebank's training split and is evaluated on the test split. For cross-lingual transfer, models trained on one language are evaluated on the test splits of all other languages without any further tuning. For DEPPROBE tuning (Section 4.2) we use the development split of EN-EWT.

BAP uses the training schedule implemented in van der Goot et al. (2021) while DIRPROBE and DEPPROBE use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 10^{-3} which is reduced by a factor of 10 each time the loss plateaus (see also Hewitt and Manning, 2019).

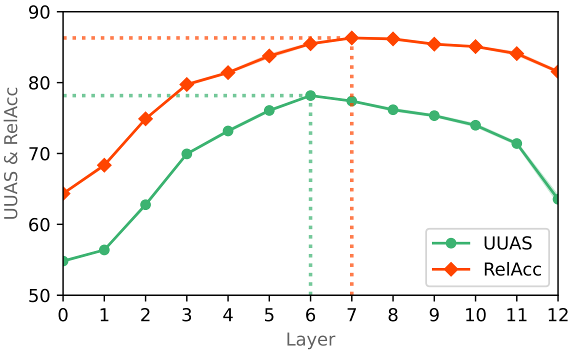


Figure 2: **Layer-wise Performance on EWT (Dev)** for DEPPROBE as measured by UUAS for the structural probe B and RelAcc for the relational probe L.

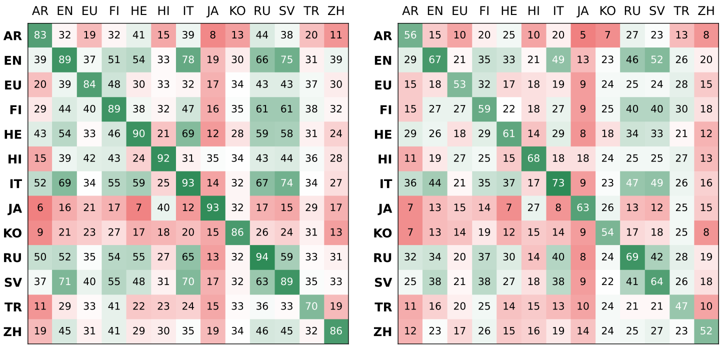
Both probing methods are implemented using PyTorch (Paszke et al., 2019) and use mBERT as implemented in the Transformers library (Wolf et al., 2020). Each model is trained with three random initializations of which we report the mean.

4.2 DEPPROBE Tuning

As prior work has repeatedly found that MLM layers encode different linguistic information, the layers which are most relevant for a probe's task are typically first identified (Tenney et al., 2019; Hewitt and Manning, 2019). Following this paradigm, we train DEPPROBE on embeddings from each layer of mBERT. Layer 0 is equivalent to the first, non-contextualized embeddings while layer 12 is the output of the last attention heads. The probe is trained on EN-EWT and evaluated on its development split using UUAS for the structural transformation B (akin to Hewitt and Manning, 2019) as well as RelAcc for the relational transformation L.

Figure 2 shows that structure is most prevalent around layer 6 at 78 UUAS, corroborating the 6–8 range identified by prior work (Tenney et al., 2019; Hewitt and Manning, 2019; Chi et al., 2020). Dependency relations are easiest to retrieve at around layer 7 with an accuracy of 86%. The standard deviation across initializations is around 0.1 in both cases. Based on these tuning results, we use layer 6 for structural probing and layer 7 for relational probing in the following experiments.

[2] For better comparability, we use the best single layer reported by Kulmizev et al. (2020) instead of the weighted sum over all layers.



	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	83	32	19	32	41	15	39	8	13	44	38	20	11
EN	39	89	37	51	54	33	78	19	30	66	75	31	39
EU	20	39	84	48	30	33	32	17	34	43	43	37	30
FI	29	44	40	89	38	32	47	16	35	61	61	38	32
HE	43	54	33	46	90	21	69	12	28	59	58	31	24
HI	15	39	42	43	24	92	31	35	34	43	44	36	28
IT	52	69	34	55	59	25	93	14	32	67	74	34	27
JA	6	16	21	17	7	40	12	93	32	17	15	29	17
KO	9	21	23	27	17	18	20	15	86	26	24	31	13
RU	50	52	35	54	55	27	65	13	32	94	59	33	31
SV	37	71	40	55	48	31	70	17	32	63	89	35	33
TR	11	29	33	41	22	23	24	15	33	36	33	70	19
ZH	19	45	31	41	29	30	35	19	34	46	45	32	86

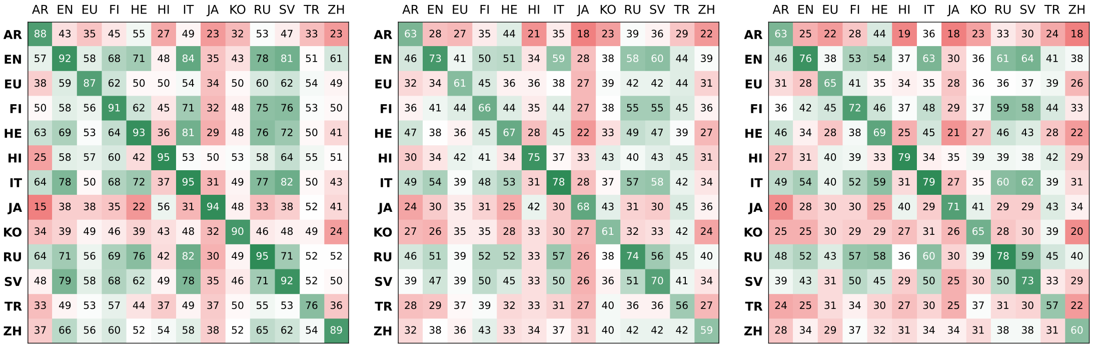
(a) BAP (LAS)

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	56	15	10	20	25	10	20	5	7	27	23	13	8
EN	29	67	21	35	33	21	49	13	23	46	52	26	20
EU	15	18	53	32	17	18	19	9	24	25	24	28	15
FI	15	27	27	59	22	18	27	9	25	40	40	30	18
HE	29	26	18	29	61	14	29	8	18	34	33	21	12
HI	11	19	27	25	15	68	18	18	24	25	25	27	13
IT	36	44	21	35	37	17	73	9	23	47	49	26	16
JA	7	13	15	14	7	27	8	63	26	13	12	25	15
KO	7	13	14	19	12	15	14	9	54	17	18	25	8
RU	32	34	20	37	30	14	40	8	24	69	42	28	19
SV	25	38	21	38	27	18	38	9	22	41	64	26	18
TR	11	16	20	25	14	15	13	10	24	21	21	47	10
ZH	12	23	17	26	15	16	19	14	24	25	27	23	52

(b) DEPPROBE (LAS)

MODEL	BAP	DEP	DIR
LAS=L	88 ±6.4	60 ±7.8	—
LAS¬L	35 ±15.7	22 ±9.9	—
UAS=L	91 ±5.0	67 ±6.7	70 ±7.8
UAS¬L	52 ±14.5	38 ±8.8	36 ±10.4

(c) Mean in-language (=L) and transfer (¬L) UAS/LAS (± stddev).



	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	88	43	35	45	55	27	49	23	32	53	47	33	23
EN	57	92	58	68	71	48	84	35	43	78	81	51	61
EU	38	59	87	62	50	50	54	34	50	60	62	54	49
FI	50	58	56	91	62	45	71	32	48	75	76	53	50
HE	63	69	53	64	93	36	81	29	48	76	72	50	41
HI	25	58	57	60	42	95	53	50	53	58	64	55	51
IT	64	78	50	68	72	37	95	31	49	77	82	50	43
JA	15	38	38	35	22	56	31	94	48	33	38	52	41
KO	34	39	49	46	39	43	48	32	90	46	48	49	24
RU	64	71	56	69	76	42	82	30	49	95	71	52	52
SV	48	79	58	68	62	49	78	35	46	71	92	52	50
TR	33	49	53	57	44	37	49	37	50	55	53	76	36
ZH	37	66	56	60	52	54	58	38	52	65	62	54	89

(d) BAP (UAS)

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	63	28	27	35	44	21	35	18	23	39	36	29	22
EN	46	73	41	50	51	34	59	28	38	58	60	44	39
EU	32	34	61	45	36	36	38	27	39	42	42	44	31
FI	36	41	44	66	44	35	44	27	38	55	55	45	36
HE	47	38	36	45	67	28	45	22	33	49	47	39	27
HI	30	34	42	41	34	75	37	33	43	40	43	45	31
IT	49	54	39	48	53	31	78	28	37	57	58	42	34
JA	24	30	35	31	25	42	30	68	43	31	30	45	36
KO	27	26	35	35	28	33	30	27	61	32	33	42	24
RU	46	51	39	52	52	33	57	26	38	74	56	45	40
SV	39	47	39	50	45	33	50	26	36	51	70	41	34
TR	28	29	37	39	32	33	31	27	40	36	36	56	27
ZH	32	38	36	43	33	34	37	31	40	42	42	42	59

(e) DEPPROBE (UAS)

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	63	25	22	28	44	19	36	18	23	33	30	24	18
EN	46	76	38	53	54	37	63	30	36	61	64	41	38
EU	31	28	65	41	35	34	35	28	36	36	37	39	26
FI	36	42	45	72	46	37	48	29	37	59	58	44	33
HE	46	34	28	38	69	25	45	21	27	46	43	28	22
HI	27	31	40	39	33	79	34	35	39	39	38	42	29
IT	49	54	40	52	59	31	79	27	35	60	62	39	31
JA	20	28	30	30	25	40	29	71	41	29	29	43	34
KO	25	25	30	29	29	27	31	26	65	28	30	39	20
RU	48	52	43	57	58	36	60	30	39	78	59	45	40
SV	39	43	31	50	45	29	50	25	30	50	73	33	29
TR	24	25	31	34	30	27	30	25	37	31	30	57	22
ZH	28	34	29	37	32	31	34	34	31	38	38	31	60

(f) DIRPROBE (UAS)

Figure 3: **In-language and Cross-lingual Transfer Performance** for 13 target treebanks (**train** → test) in UAS for BAP (fully tuned parser), DEPPROBE, DIRPROBE and LAS for BAP, DEPPROBE (DIRPROBE is unlabeled).

4.3 Parsing Performance

Figure 3 lists UAS for all methods and LAS for BAP and DEPPROBE both on target-language test data (=L) and zero-shot transfer targets (¬L). Table 3c further shows the mean results for each setting.

Unsurprisingly, the full parametrization of BAP performs best, with in-language scores of 88 LAS and 91 UAS. For zero-shot transfer, these scores drop to 35 LAS and 52 UAS, with some language pairs seeing differences of up to 85 points: e.g. JA → JA (93 LAS) versus AR → JA (8 LAS) in Figure 3a. This again confirms the importance of selecting appropriate source data for any given target.

Both probes, with their limited parametrization, fall short of the full parser's performance, but still reach up to 73 LAS and 79 UAS. DIRPROBE has a mean in-language UAS which is 3 points higher than for DEPPROBE, attributable to the more complex decoder. Due to DIRPROBE's output structures being unlabeled, we cannot compare LAS.

DEPPROBE reaches a competitive 67 UAS despite its much simpler decoding procedure and appears to be more stable for zero-shot transfer as it outperforms DIRPROBE by around 2 UAS while maintaining a lower standard deviation. Most importantly, it produces directed and *labeled* parses such that we can fully compare it to BAP. Considering that DEPPROBE has more than three orders of magnitude fewer tunable parameters, a mean in-language LAS of 60 is considerable and highlights the large degree of latent dependency information in untuned, contextual embeddings. For zero-shot transfer, the performance gap to BAP narrows to 13 LAS and 14 UAS.

4.4 Transfer Prediction

Given that DEPPROBE provides a highly parameter-efficient method for producing directed, labeled parse trees, we next investigate whether its performance patterns are indicative of the full parser's performance and could aid in selecting an appropriate source treebank for a given target without having to train the 183 million parameters of BAP.

Setup Comparing UAS and LAS of BAP with respective scores of DEPPROBE and DIRPROBE, we compute the Pearson correlation coefficient ρ and the weighted Kendall's τ_w (Vigna, 2015). The latter can be interpreted as corresponding to a cor-

MODEL	LAS		UAS	
	ρ	τ_w	ρ	τ_w
L2V	.86	.72	.80	.70
DIRPROBE	—	—	.91	.81
DEPPROBE	**.97**	**.88**	.94	.85

Table 1: **Transfer Correlation with BAP.** Pearson ρ and weighted Kendall's τ_w for BAP's LAS and UAS with respect to DIRPROBE's UAS, DEPPROBE's UAS and LAS as well as lang2vec cosine similarity (L2V).

MODEL	LAS		UAS	
	ρ	τ_w	ρ	τ_w
SSA-STRUCT	.68	.42	.60	.43
SSA-DEPTH	.62	.34	.53	.35
SSA-REL	**.73**	**.55**	.65	.53

Table 2: **SSA Correlation with BAP.** Pearson ρ and weighted Kendall's τ_w for BAP's LAS and UAS with respect to subspace angles between structural (STRUCT), depth (DEPTH) and relation probes (REL).

relation in $[-1, 1]$, and that given a probe ranking one source treebank over another, the probability of this higher rank corresponding to higher performance in the full parser is $\frac{\tau_w+1}{2}$. All reported correlations are significant at $p < 0.001$. Similarly, differences between correlation coefficients are also significant at $p < 0.001$ as measured using a standard Z-test. In addition to the probes, we also compare against a method commonly employed by practitioners by using the cosine similarity of typological features from the URIEL database as represented in lang2vec (Littell et al., 2017; L2V) between our 13 targets (details in Appendix A).

Results Table 1 shows that the L2V baseline correlates with final parser performance, but that actual dependency parses yield significantly higher correlation and predictive power. For UAS, we find that despite having similar attachment scores, DEPPROBE performance correlates higher with BAP than that of DIRPROBE, both with respect to predicting the ability to parse any particular language as well as ranking the best source to transfer from. Using the labeled parse trees of DEPPROBE results in almost perfect correlation with BAP's LAS at $\rho = .97$ as well as a τ_w of .88, highlighting the importance of modeling the full task and including dependency relation information. Using Kendall's τ_w with respect to LAS, we can estimate that selecting the highest performing source treebank from DEPPROBE to train the full parser will be the best choice 94% of the time for any treebank pair.

5 Analysis

5.1 Tree Depth versus Relations

Why does DEPPROBE predict transfer performance more accurately than DIRPROBE despite its simpler architecture? As each probe consists only of two matrices optimized to extract tree structural, depth or relational information, we can directly compare the similarity of all task-relevant parameters across languages against the full BAP's cross-lingual performance.

In order to measure the similarity of probe matrices from different languages, we use mean subspace angles (Knyazev and Argentati, 2002; SSA), similarly to prior probing work (Chi et al., 2020). Intuitively, SSA quantifies the energy required to transform one matrix to another by converting the singular values of the transformation into angles between 0°and 90°. SSAs are computed for the structural probe (SSA-STRUCT) which is equivalent in both methods, DIRPROBE's depth probe (SSA-DEPTH) and DEPPROBE's relational probe (SSA-REL). We use Pearson ρ and the weighted Kendall's τ_w to measure the correlation between cross-lingual probe SSAs and BAP performance. This allows us to investigate which type of information is most important for final parsing performance.

From Table 2, we can observe that SSAs between probes of different languages correlate less with transfer performance than UAS or LAS (Table 1), underlining the importance of extracting full parses. Among the different types of dependency information, we observe that SSAs between the *relational* probes used by DEPPROBE correlate highest with final performance at .73 for LAS and .65 for UAS. Structural probing correlates significantly both with BAP's LAS and UAS at .68 and .60 respectively, but to a lesser degree. Probes for tree depth have the lowest correlation at .62 for LAS and .53 for UAS. Despite tree depth being a distinctive syntactic feature for language pairs such as the agglutinative Turkish and the more function word-based English, depth is either not as relevant for BAP or may be represented less consistently in embeddings across languages, leading to lower correlation between SSAs and final performance.

5.2 Full Parser versus Probe

In the following analysis we investigate performance differences between the full BAP and DEPPROBE across all 13 targets in order to identify finer-grained limitations of the linear approach and also which kinds of dependencies benefit from full parameter tuning and non-linear decoding.

Edge Length Figure 5 shows offsets between gold and predicted head positions. The majority of heads are predicted correctly with a ratio of 92.1% for BAP and 69.7% for DEPPROBE. Both methods are less accurate in predicting long-distance edges with length 150–250, resulting in offsets of ca. 100 (aggregated into $<$ and $>$ in Figure 5). Most likely, this is due to these edges' overall sparsity in the data (only 6.7% of edges cover a distance of more than 10 tokens) as well as their higher overall subjective difficulty. Nonetheless, BAP is able to capture such dependencies more accurately as shown by its lower error rates for long edges compared to those of DEPPROBE.

In addition to very distant head nodes, BAP also seems to recover more of the nuanced edges in the $[-5, 5]$ interval. This range is particularly impactful for downstream performance as the edges in our target treebanks have a median length of 2 (mean length 3.62 with $\sigma = 5.70$). The structural probing loss (Equation 2) and the simple linear parametrization of the probe are able to capture a large number of these edges as evidenced by overall low error rates, but lack the necessary expressivity in order to accurately capture all cases.

Relations Looking at RelAcc for each category in the UD taxonomy (de Marneffe et al., 2014) in Figure 4 allows us to identify where higher parametrization and more complex decoding are required for high parsing performance. While we again observe that performance on all relations is higher for BAP than for DEPPROBE, a large subset of the relations is characterized by comparable or equivalent performance. These include simple punctuation (`punct`), but also the majority of function word relations such as `aux`, `case`, `clf`, `det` and `mark` as well as coordination (e.g. `cc`, `conj`). We attribute the high performance of DEPPROBE on these relations to the fact that the words used to express them typically stem from closed classes and consequently similar embeddings: e.g., determiners "the/a/an" (EN), case markers "di/da" (IT).

Interestingly, some relations expressed through open class words are also captured by the linear probe. These include the modifiers `advmod`, `amod` and `discourse` as well as some nominal relations such as `expl`, `nmod`, `nsubj` and `nummod`. As prior work has identified PoS information in untuned embeddings (Tenney et al., 2019), the modifiers are likely benefiting from the same embedding features. The fact that DEPPROBE nonetheless identifies syntax-specific relations such as `nsubj`, and to a lesser degree `obj` and `obl`, indicates the presence of context-dependent syntactic information in addition to PoS.

The larger the set of possible words for a relation, the more difficult it is to capture with the probe. The functional `cop` (copula) relation provides an informative example: In English (and related languages), it is almost exclusively assigned to the verb "be" resulting in 85% RelAcc, while in non-European languages such as Japanese it can be ascribed to a larger set which often overlaps with other relations (e.g. `aux`) resulting in 65% RelAcc. BAP adapts to each language by tuning all parameters while DEPPROBE, using fixed embeddings, reaches competitive scores on European languages, but performs worse in non-European settings (details in Appendix B).

Besides capturing larger variation in surface forms, BAP also appears to benefit from higher expressivity when labeling clausal relations such as `ccomp`, `csubj`. These relations are often characterized not only by surface form variation, but also by PoS variation of head/child words and overlap with other relation types (e.g. clausal subjects stem from verbs or adjectives), making them difficult to distinguish in untuned embeddings. Simultaneously, they often span longer edges compared to determiners or other function words.

Another relation of particular importance is `root` as it determines the direction of all edges predicted by DEPPROBE. An analysis of the 14% RelAcc difference to BAP reveals that both methods most frequently confuse `root` with relations that fit the word's PoS, e.g. `NOUN` roots with `nsubj` or `nmod`. For the majority PoS `VERB` (70% of all `root`), we further observe that DEPPROBE predicts twice as many `xcomp` and `parataxis` confusions compared to BAP, likely attributable to their `root`-similar function in subclauses. Since their distinction hinges on context, the full parser, which also tunes the contextual encoder, is better equipped to differentiate between them.

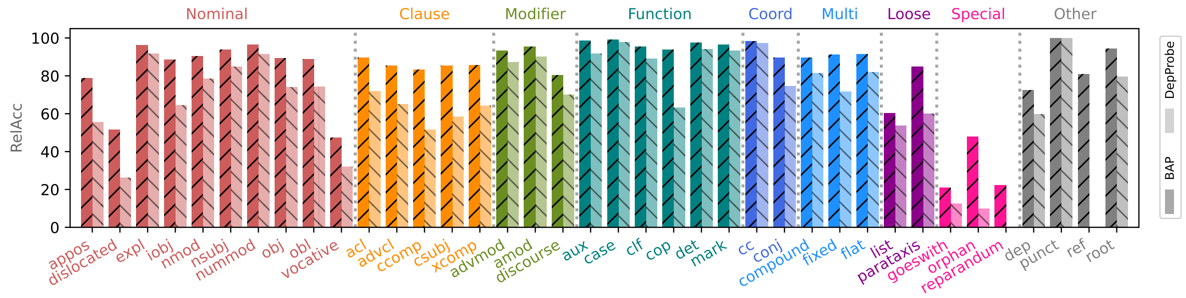


Figure 4: **Relation Accuracy of BAP and DepProbe** compared for all 13 in-language targets, grouped according to the Universal Dependencies taxonomy (de Marneffe et al., 2014).

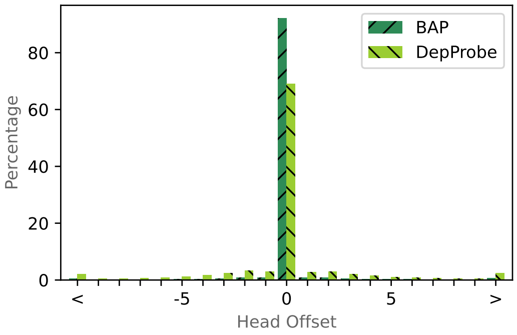


Figure 5: **Ratio of Offsets between Gold and Predicted Heads** for BAP and DepProbe (i.e. 0 is correct) across all 13 targets.

The last category in which BAP outperforms DepProbe includes rare, treebank-specific relations such as `reparandum` (reference from a corrected word to an erroneous one). Again, the larger number of tunable parameters in addition to the non-linear decoding procedure of the full parser enable it to capture more edge cases while DepProbe's linear approach can only approximate a local optimum for any relations which are represented non-linearly.

Efficiency When using a probe for performance prediction, it is important to consider its computational efficiency over the full parser's fine-tuning procedure. In terms of tunable parameters, DepProbe has 36% fewer parameters than DirProbe and three orders of magnitude fewer parameters than BAP. In practice, this translates to training times in the order of minutes instead of hours.

Despite its simple $\mathcal{O}(n^2)$ decoding procedure compared to dynamic programming-based graph-decoding algorithms ($\mathcal{O}(n^3)$), DepProbe is able to extract full dependency trees which correlate highly with downstream performance while maintaining high efficiency (Section 4.4).

6 Conclusion

With DepProbe, we have introduced a novel probing procedure to extract fully labeled and directed dependency trees from untuned, contextualized embeddings. Compared to prior approaches which extract structures lacking labels, edge directionality or both, our method retains a simple linear parametrization which is in fact more lightweight and does not require complex decoders (Section 3).

To the best of our knowledge, this is the first linear probe which can be used to estimate LAS from untuned embeddings. Using this property, we evaluated the predictive power of DepProbe on cross-lingual parsing with respect to the transfer performance of a fully fine-tuned biaffine attention parser. Across the considered 169 language pairs, DepProbe is surprisingly effective: Its LAS correlates significantly ($p < 0.001$) and most highly compared with unlabeled probes or competitive language feature baselines, choosing the best source treebank in 94% of all cases (Section 4).

Leveraging the linearity of the probe to analyze structural and relational subspaces in mBERT embeddings, we find that dependency *relation* information is particularly important for parsing performance and cross-lingual transferability, compared to both tree depth and structure. DepProbe, which models structure and relations, is able to recover many functional and syntactic relations with competitive accuracy to the full BAP (Section 5).

Finally, the substantially higher efficiency of DepProbe with respect to time and compute make it suitable for accurate parsing performance prediction. As contemporary performance prediction methods lack formulations for graphical tasks and handcrafted features such as lang2vec are not available in all transfer settings (e.g. document domains, MLM encoder choice), we see linear approaches such as DepProbe as a valuable alternative.

Acknowledgements

We would like to thank the NLPnorth group for insightful discussions on this work — in particular Elisa Bassignana and Mike Zhang. Additional thanks to ITU's High-performance Computing Cluster team. Finally, we thank the anonymous reviewers for their helpful feedback. This research is supported by the Independent Research Fund Denmark (Danmarks Frie Forskningsfond; DFF) grant number 9063-00077B.

References

Maria Jesus Aranzabe, Aitziber Atutxa, Kepa Bengoetxea, Arantza Diaz de Ilarraza, Iakes Goenaga, Koldo Gojenola, and Larraitz Uria. 2015. Automatic conversion of the Basque dependency treebank to universal dependencies. In *Proceedings of the fourteenth international workshop on treebanks an linguistic theories (TLT14)*, pages 233–241.

Masayuki Asahara, Hiroshi Kanayama, Takaaki Tanaka, Yusuke Miyao, Sumire Uematsu, Shinsuke Mori, Yuji Matsumoto, Mai Omura, and Yugo Murawaki. 2018. Universal Dependencies version 2 for Japanese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Cristina Bosco, Felice Dell'Orletta, Simonetta Montemagni, Manuela Sanguinetti, and Maria Simi. 2014. The EVALITA 2014 dependency parsing task. In *EVALITA 2014 Evaluation of NLP and Speech Tools for Italian*, pages 1–8. Pisa University Press.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Yoeng-Jin Chu and Tseng-Hong Liu. 1965. On the shortest arborescence of a directed graph. *Scientia Sinica*, 14:1396–1400.

Jayeol Chun, Na-Rae Han, Jena D. Hwang, and Jinho D. Choi. 2018. Building Universal Dependency treebanks in Korean. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marie-Catherine de Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D. Manning. 2014. Universal Stanford dependencies: A cross-linguistic typology. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 4585–4592, Reykjavik, Iceland. European Language Resources Association (ELRA).

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 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Kira Droganova, Olga Lyashevskaya, and Daniel Zeman. 2018. Data conversion and consistency of monolingual corpora: Russian UD treebanks. In *Proceedings of the 17th international workshop on treebanks and linguistic theories (tlt 2018)*, 155, pages 53–66.

Jack Edmonds. 1967. Optimum branchings. *Journal of Research of the National Bureau of Standards, B*, 71:233–240.

Jan Hajič, Otakar Smrž, Petr Zemánek, Petr Pajas, Jan Šnaidauf, Emanuel Beška, Jakub Krácmar, and Kamila Hassanová. 2009. Prague Arabic dependency treebank 1.0.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. 2020. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online. Association for Computational Linguistics.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Vojtěch Jarník. 1930. O jistém problému minimálním.(z dopisu panu o. boruvkovi). *Práce moravské přírodovědecké společnosti*, 6(4):57–63.

Andrew V Knyazev and Merico E Argentati. 2002. Principal angles between subspaces in an a-based scalar product: algorithms and perturbation estimates. *SIAM Journal on Scientific Computing*, 23(6):2008–2040.

Artur Kulmizev, Miryam de Lhoneux, Johannes Gontrum, Elena Fano, and Joakim Nivre. 2019. Deep contextualized word embeddings in transition-based and graph-based dependency parsing - a tale of two parsers revisited. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2755–2768, Hong Kong, China. Association for Computational Linguistics.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. 2020. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4077–4091, Online. Association for Computational Linguistics.

Tomasz Limisiewicz and David Mareček. 2021. Introducing orthogonal constraint in structural probes. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 428–442, Online. Association for Computational Linguistics.

Patrick Littell, David R. Mortensen, Ke Lin, Katherine Kairis, Carlisle Turner, and Lori Levin. 2017. URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 8–14, Valencia, Spain. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. Association for Computational Linguistics.

Cuong Nguyen, Tal Hassner, Matthias W. Seeger, and Cédric Archambeau. 2020. LEEP: A new measure to evaluate transferability of learned representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7294–7305. PMLR.

Martha Palmer, Rajesh Bhatt, Bhuvana Narasimhan, Owen Rambow, Dipti Misra Sharma, and Fei Xia. 2009. Hindi syntax: Annotating dependency, lexical predicate-argument structure, and phrase structure. In *The 7th International Conference on Natural Language Processing*, pages 14–17.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Robert Clay Prim. 1957. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401.

Sampo Pyysalo, Jenna Kanerva, Anna Missilä, Veronika Laippala, and Filip Ginter. 2015. Universal Dependencies for Finnish. In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, pages 163–172, Vilnius, Lithuania. Linköping University Electronic Press, Sweden.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Mo Shen, Ryan McDonald, Daniel Zeman, and Peng Qi. 2016. UD_Chinese-GSD. `https://github.com/UniversalDependencies/UD_Chinese-GSD`.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language*

Resources and Evaluation (LREC'14), pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Miloš Stanojević and Shay B. Cohen. 2021. A root of a problem: Optimizing single-root dependency parsing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10540–10557, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Umut Sulubacak, Memduh Gokirmak, Francis Tyers, Çağrı Çöltekin, Joakim Nivre, and Gülşen Eryiğit. 2016. Universal Dependencies for Turkish. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3444–3454, Osaka, Japan. The COLING 2016 Organizing Committee.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Sebastiano Vigna. 2015. A weighted correlation index for rankings with ties. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*, pages 1166–1176. ACM.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. 2021. Logme: Practical assessment of pre-trained models for transfer learning. In *International Conference on Machine Learning*, pages 12133–12143. PMLR.

Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Hamid Aghaei, Željko Agić, Amir Ahmadi, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Ika Alfina, Lene Antonsen, Katya Aplonova, Angelina Aquino, Carolina Aragon, Maria Jesus Aranzabe, Bilge Nas Arıcan, H͡órunn Arnardóttir, Gashaw Arutie, Jessica Naraiswari Arwidarasti, Masayuki Asahara, Deniz Baran Aslan, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Keerthana Balasubramani, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Starkaður Barkarson, Victoria Basmov, Colin Batchelor, John Bauer, Seyyit Talha Bedir, Kepa Bengoetxea, Gözde Berk, Yevgeni Berzak, Irshad Ahmad Bhat, Riyaz Ahmad Bhat, Erica Biagetti, Eckhard Bick, Agnė Bielinskienė, Kristín Bjarnadóttir, Rogier Blokland, Victoria Bobicev, Loïc Boizou, Emanuel Borges Völker, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Adriane Boyd, Anouck Braggaar, Kristina Brokaitė, Aljoscha Burchardt, Marie Candito, Bernard Caron, Gauthier Caron, Lauren Cassidy, Tatiana Cavalcanti, Gülşen Cebiroğlu Eryiğit, Flavio Massimiliano Cecchini, Giuseppe G. A. Celano, Slavomír Čéplö, Neslihan Cesur, Savas Cetin, Özlem Çetinoğlu, Fabricio Chalub, Shweta Chauhan, Ethan Chi, Taishi Chika, Yongseok Cho, Jinho Choi, Jayeol Chun, Alessandra T. Cignarella, Silvie Cinková, Aurélie Collomb, Çağrı Çöltekin, Miriam Connor, Marine Courtin, Mihaela Cristescu, Philemon. Daniel, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Mehmet Oguz Derin, Elvis de Souza, Arantza Diaz de Ilarraza, Carly Dickerson, Arawinda Dinakaramani, Elisa Di Nuovo, Bamba Dione, Peter Dirix, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Hanne Eckhoff, Sandra Eiche, Marhaba Eli, Ali Elkahky, Binyam Ephrem, Olga Erina, Tomaž Erjavec, Aline Etienne, Wograine Evelyn, Sidney Facundes, Richárd Farkas, Marília Fernanda, Hector Fernandez Alcalde, Jennifer Foster, Cláudia Freitas, Kazunori Fujita, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Fabrício Ferraz Gerardi, Kim Gerdes, Filip Ginter, Gustavo Godoy, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta González Saavedra, Bernadeta Griciūtė, Matias Grioni, Loïc Grobol, Normunds Grūzītis, Bruno Guillaume, Céline Guillot-Barbance, Tunga Güngör, Nizar Habash, Hinrik Hafsteinsson, Jan Hajič, Jan Hajič jr., Mika Hämäläinen, Linh Hà Mỹ, Na-Rae Han, Muhammad Yudistira Hanifmuti, Sam Hardwick, Kim Harris, Dag Haug, Johannes Heinecke, Oliver Hellwig, Felix Hennig, Barbora Hladká, Jaroslava Hlaváčová, Florinel Hociung, Petter Hohle, Eva Huber, Jena Hwang, Takumi Ikeda, Anton Karl Ingason, Radu Ion, Elena Irimia, Ọlájídé Ishola, Kaoru Ito, Tomáš Jelínek, Apoorva Jha, Anders Johannsen, Hildur Jónsdóttir, Fredrik Jørgensen, Markus Juutinen, Sarveswaran K, Hüner Kaşıkara, Andre Kaasen, Nadezhda Kabaeva, Sylvain Kahane, Hiroshi Kanayama, Jenna Kanerva, Neslihan Kara, Boris Katz, Tolga Kayadelen, Jessica Kenney, Václava Kettnerová, Jesse Kirchner, Elena Klementieva, Arne Köhn, Abdullatif Köksal, Kamil Kopacewicz, Timo Korkiakangas, Natalia Kotsyba, Jolanta Kovalevskaitė, Simon Krek, Parameswari Krishnamurthy, Oğuzhan Kuyrukçu, Aslı Kuzgun, Sookyoung Kwak, Veronika Laippala, Lucia Lam, Lorenzo Lambertino, Tatiana Lando, Septina Dian Larasati, Alexei Lavrentiev, John Lee, Phuong Lê Hồng, Alessandro Lenci, Saran Lertpradit, Herman Leung, Maria Levina, Cheuk Ying Li, Josie Li, Keying Li, Yuan Li, Kyung-Tae Lim, Bruna Lima Padovani, Krister Lindén, Nikola Ljubešić, Olga Loginova, Andry Luthfi, Mikko Luukko, Olga Lyashevskaya, Teresa Lynn, Vivien Macketanz, Aibek Makazhanov, Michael Mandl, Christopher Manning, Ruli Manurung, Büşra Marşan, Cătălina Mărănduc, David Mareček, Katrin Marheinecke, Héctor Martínez Alonso, André Martins, Jan Mašek, Hiroshi Matsuda, Yuji Matsumoto, Alessandro Mazzei, Ryan McDonald, Sarah McGuinness, Gustavo Mendonça, Niko Miekka, Karina Mischenkova, Margarita Misirpashayeva, Anna Missilä, Cătălin Mititelu, Maria Mitrofan, Yusuke Miyao, AmirHossein Mojiri Foroushani, Judit Molnár, Amirsaeid Moloodi, Simonetta Montemagni, Amir More, Laura Moreno Romero, Giovanni Moretti, Keiko Sophie Mori, Shinsuke Mori, Tomohiko Morioka, Shigeki Moro, Bjartur Mortensen, Bohdan Moskalevskyi, Kadri Muischnek, Robert Munro, Yugo Murawaki, Kaili Müürisep, Pinkey Nainwani, Mariam Nakhlé, Juan Ignacio Navarro Horñiacek, Anna Nedoluzhko, Gunta Nešpore-Bērzkalne, Manuela Nevaci, Luong Nguyễn Thị, Huyền Nguyễn Thị Minh, Yoshihiro Nikaido, Vitaly Nikolaev, Rattima Nitisaroj, Alireza Nourian, Hanna Nurmi, Stina Ojala, Atul Kr. Ojha, Adédayọ Olúòkun, Mai Omura, Emeka Onwuegbuzia, Petya Osenova, Robert Östling, Lilja Øvrelid, Şaziye Betül Özateş, Merve Özçelik, Arzucan Özgür, Balkız Öztürk Başaran, Hyunji Hayley Park, Niko Partanen, Elena Pascual, Marco Passarotti, Agnieszka Patejuk, Guilherme Paulino-Passos, Angelika Peljak-Łapińska, Siyao Peng, Cenel-Augusto Perez, Natalia Perkova, Guy Perrier, Slav Petrov, Daria Petrova, Jason Phelan, Jussi Piitulainen, Tommi A Pirinen, Emily Pitler, Barbara Plank, Thierry Poibeau, Larisa Ponomareva, Martin Popel, Lauma Pretkalniņa, Sophie Prévost, Prokopis Prokopidis, Adam Przepiórkowski, Tiina Puolakainen, Sampo Pyysalo, Peng Qi, Andriela Rääbis, Alexandre Rademaker, Taraka Rama, Loganathan Ramasamy, Carlos Ramisch, Fam Rashel, Mohammad Sadegh Rasooli, Vinit Ravishankar, Livy Real, Petru Rebeja, Siva Reddy, Georg Rehm, Ivan Riabov, Michael Rießler, Erika Rimkutė, Larissa Rinaldi, Laura Rituma, Luisa Rocha, Eiríkur Rögnvaldsson, Mykhailo Romanenko, Rudolf Rosa, Valentin Roșca, Davide Rovati, Olga Rudina, Jack Rueter, Kristján Rúnarsson, Shoval Sadde, Pegah Safari, Benoît Sagot, Aleksi Sahala, Shadi Saleh, Alessio Salomoni, Tanja Samardžić, Stephanie Samson, Manuela Sanguinetti, Ezgi Sanıyar, Dage Särg, Baiba Saulīte, Yanin Sawanakunanon, Shefali Saxena, Kevin Scannell, Salvatore Scarlata, Nathan Schneider, Sebastian Schuster, Lane Schwartz, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Mo Shen, Atsuko Shimada, Hiroyuki Shirasu, Yana Shishkina, Muh Shohibussirri, Dmitry Sichinava, Janine Siewert, Einar Freyr Sigurðsson, Aline Silveira, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Maria Skachedubova, Aaron Smith, Isabela Soares-Bastos, Carolyn Spadine, Rachele Sprugnoli, Steinh͡ór Steingrímsson, Antonio Stella, Milan Straka, Emmett Strickland, Jana Strnadová, Alane Suhr, Yogi Lesmana Sulestio, Umut Sulubacak, Shingo Suzuki, Zsolt Szántó, Dima Taji, Yuta Takahashi, Fabio Tamburini, Mary Ann C. Tan, Takaaki Tanaka, Samson Tella, Isabelle Tellier, Marinella Testori, Guillaume Thomas, Liisi Torga, Marsida Toska, Trond Trosterud, Anna Trukhina, Reut Tsarfaty, Utku Türk, Francis Tyers, Sumire Uematsu, Roman Untilov, Zdeňka Urešová, Larraitz Uria, Hans Uszkoreit, Andrius Utka, Sowmya Vajjala, Rob van der Goot, Martine Vanhove, Daniel van Niekerk, Gertjan van Noord, Viktor Varga, Eric Villemonte de la Clergerie, Veronika Vincze, Natalia Vlasova, Aya Wakasa, Joel C. Wallenberg, Lars Wallin, Abigail Walsh, Jing Xian Wang, Jonathan North Washington, Maximilan Wendt, Paul Widmer, Seyi Williams, Mats Wirén, Christian Wittern, Tsegay Woldemariam, Tak-sum Wong, Alina Wróblewska, Mary Yako, Kayo Yamashita, Naoki Yamazaki, Chunxiao Yan, Koichi Yasuoka, Marat M. Yavrumyan, Arife Betül Yenice, Olcay Taner Yıldız, Zhuoran Yu, Zdeněk Žabokrtský, Shorouq Zahra, Amir Zeldes, Hanzhi Zhu, Anna Zhuravleva, and Rayan Ziane. 2021. Universal dependencies 2.8.1. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Appendix

A Experimental Setup

TARGET	LANG	FAMILY	SIZE
AR-PADT	Arabic	Afro-Asiatic	7.6k
EN-EWT	English	Indo-European	16.6k
EU-BDT	Basque	Basque	9.0k
FI-TDT	Finnish	Uralic	15.1k
HE-HTB	Hebrew	Afro-Asiatic	6.2k
HI-HDTB	Hindi	Indo-European	16.6k
IT-ISDT	Italian	Indo-European	14.1k
JA-GSD	Japanese	Japanese	8.1k
KO-GSD	Korean	Korean	6.3k
RU-SynTagRus	Russian	Indo-European	61.9k
SV-Talbanken	Swedish	Indo-European	6.0k
TR-IMST	Turkish	Turkic	5.6k
ZH-GSD	Chinese	Sino-Tibetan	5.0k

Table 3: **Target Treebanks** based on Kulmizev et al. (2019) with language family (FAMILY) and total number of sentences (SIZE).

Target Treebanks Table 3 lists the 13 target treebanks based on the set by Kulmizev et al. (2019): AR-PADT (Hajič et al., 2009), EN-EWT (Silveira et al., 2014), EU-BDT (Aranzabe et al., 2015), FI-TDT (Pyysalo et al., 2015), HE-HTB (McDonald et al., 2013), HI-HDTB (Palmer et al., 2009), IT-ISDT (Bosco et al., 2014), JA-GSD (Asahara et al., 2018), KO-GSD (Chun et al., 2018), RU-SynTagRus (Droganova et al., 2018), SV-Talbanken (McDonald et al., 2013), TR-IMST (Sulubacak et al., 2016), ZH-GSD (Shen et al., 2016). In our experiments, we use these treebanks as provided in Universal Dependencies version 2.8 (Zeman et al., 2021). Each method (BAP, DEPPROBE, DIRPROBE) is trained on each target's respective training split and evaluated on each test split both in the in-language and cross-lingual setting without further fine-tuning. For the layer-hyperparameter of DEPPROBE, we use the development split of EN-EWT as in prior probing work (Hewitt and Manning, 2019).

Implementation BAP (Dozat and Manning, 2017) uses the implementation in the MaChAmp toolkit v0.2 (van der Goot et al., 2021) with the default training schedule and hyperparameters. DIRPROBE (Kulmizev et al., 2020) is reimplemented based on the authors' algorithm description and uses their reported hyperparameters. Both it and DEPPROBE (this work) are implemented in PyTorch v1.9.0 (Paszke et al., 2019) and use mBERT (`bert-base-multilingual-cased`) from the Transformers library v4.8.2 (Wolf et al., 2020). Following prior probing work, each token which is split by mBERT into multiple subwords is mean-pooled (Hewitt and Manning, 2019). For lang2vec (Littell et al., 2017), we use its `syntax_knn`, `phonology_knn` and `inventory_knn` features from v1.1.2. For our analyses (Section 5), we use numpy v1.21.0 (Harris et al., 2020), SciPy v1.7.0 (Virtanen et al., 2020) and Matplotlib v3.4.3 (Hunter, 2007).

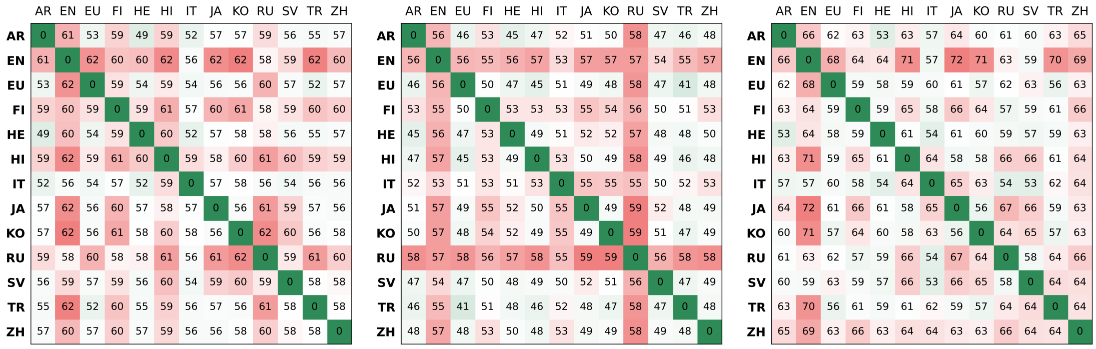
Training Details Each model is trained on an NVIDIA A100 GPU with 40GBs of VRAM and an AMD Epyc 7662 CPU. Mean training time for BAP is ca. 2 h (± 30 min). DIRPROBE requires around 20 min (± 5 min). DEPPROBE can be trained the fastest in around 15 min (± 5 min) with the embedding forward operation consuming most of the time. The models use batches of size 64 and both probes have an early stopping patience of 3 (max. 30 epochs) on each target's dev data. All models are initialized thrice using the random seeds 41, 42 and 43.

Reproducibility In order to ensure reproducibility for future work, we release the code for our methods and reimplementations in addition to token-level predictions (e.g. for significance testing) at https://personads.me/x/acl-2022-code.

B Additional Results

Subspace Angles (SSA) are used in Section 5.1 in order to identify which types of dependency information are most relevant to final parsing performance. Figure 6 lists all cross-lingual SSAs for the structural (Figure 6a), depth (Figure 6b) and relational probes (Figure 6c). SSA values are converted from radians to degrees $\in [0, 90]$ for improved readability. Correlation in Table 2 is calculated based on negative SSA (Chi et al., 2020).

Relation Accuracy (RelAcc) is used in Section 5.2 to analyze dependency relations which benefit from the full parametrization of BAP compared to the linear DEPPROBE. Figures 7–19 show RelAcc per language in addition to the aggregated scores in Figure 4. As noted in Section 5.2, some relations such as `cop` differ substantially across languages with respect to their realization (e.g. surface form variation). Furthermore, the set of relations represented in each target treebank may differ, especially for specializied categories.



(a) SSA-Struct

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	0	61	53	59	49	59	52	57	57	59	56	55	57
EN	61	0	62	60	60	62	56	62	62	58	59	62	60
EU	53	62	0	59	54	59	54	56	56	60	57	52	57
FI	59	60	59	0	59	61	57	60	61	58	59	60	60
HE	49	60	54	59	0	60	52	57	58	58	56	55	57
HI	59	62	59	61	60	0	59	58	60	61	60	59	59
IT	52	56	54	57	52	59	0	57	58	56	54	56	56
JA	57	62	56	60	57	58	57	0	56	61	59	57	56
KO	57	62	56	61	58	60	58	56	0	62	60	56	58
RU	59	58	60	58	58	61	56	61	62	0	59	61	60
SV	56	59	57	59	56	60	54	59	60	59	0	58	58
TR	55	62	52	60	55	59	56	57	56	61	58	0	58
ZH	57	60	57	60	57	59	56	56	58	60	58	58	0

(b) SSA-Depth

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	0	56	46	53	45	47	52	51	50	58	47	46	48
EN	56	0	56	55	56	57	53	57	57	57	54	55	57
EU	46	56	0	50	47	45	51	49	48	58	47	41	48
FI	53	55	50	0	53	53	53	55	54	56	50	51	53
HE	45	56	47	53	0	49	51	52	52	57	48	48	50
HI	47	57	45	53	49	0	53	50	49	58	49	46	48
IT	52	53	51	53	51	53	0	55	55	55	50	52	53
JA	51	57	49	55	52	50	55	0	49	59	52	48	49
KO	50	57	48	54	52	49	55	49	0	59	51	47	49
RU	58	57	58	56	57	58	55	59	59	0	56	58	58
SV	47	54	47	50	48	49	50	52	51	56	0	47	49
TR	46	55	41	51	48	46	52	48	47	58	47	0	48
ZH	48	57	48	53	50	48	53	49	49	58	49	48	0

(c) SSA-Rel

	AR	EN	EU	FI	HE	HI	IT	JA	KO	RU	SV	TR	ZH
AR	0	66	62	63	53	63	57	64	60	61	60	63	65
EN	66	0	68	64	64	71	57	72	71	63	59	70	69
EU	62	68	0	59	58	59	60	61	57	62	63	56	63
FI	63	64	59	0	59	65	58	66	64	57	59	61	66
HE	53	64	58	59	0	61	54	61	60	59	57	59	63
HI	63	71	59	65	61	0	64	58	58	66	66	61	64
IT	57	57	60	58	54	64	0	65	63	54	53	62	64
JA	64	72	61	66	61	58	65	0	56	67	66	59	63
KO	60	71	57	64	60	58	63	56	0	64	65	57	63
RU	61	63	62	57	59	66	54	67	64	0	58	64	66
SV	60	59	63	59	57	66	53	66	65	58	0	64	64
TR	63	70	56	61	59	61	62	59	57	64	64	0	64
ZH	65	69	63	66	63	64	64	63	63	66	64	64	0

Figure 6: **SSA of Probe Transformations** in degrees across 13 target treebanks for the structural (SSA-Struct), depth (SSA-Depth) and relational probes (SSA-Rel).

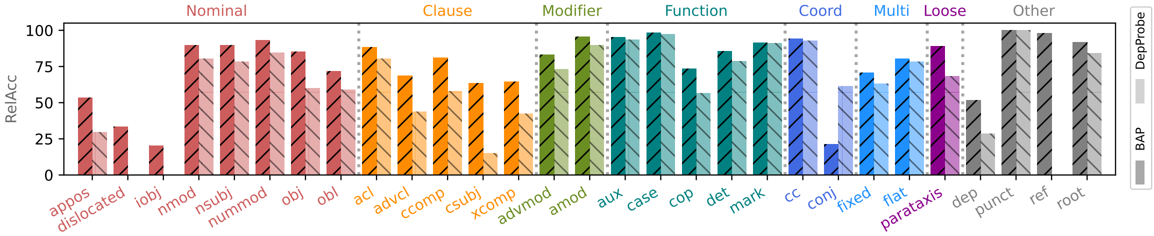


Figure 7: **RelAcc of BAP and DepProbe on AR-PADT (Test)** grouped according to UD taxonomy.

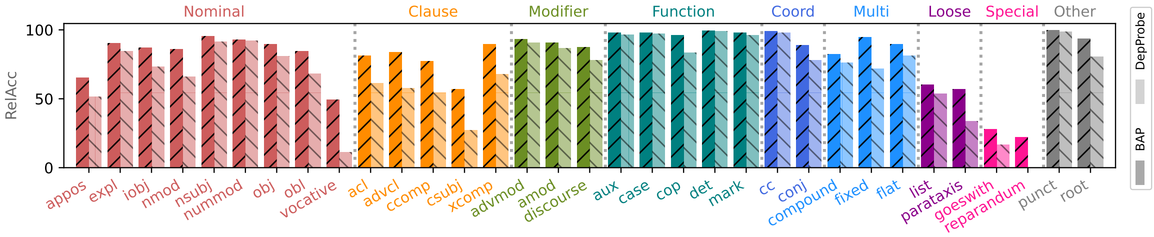


Figure 8: **RelAcc of BAP and DepProbe on EN-EWT (Test)** grouped according to UD taxonomy.

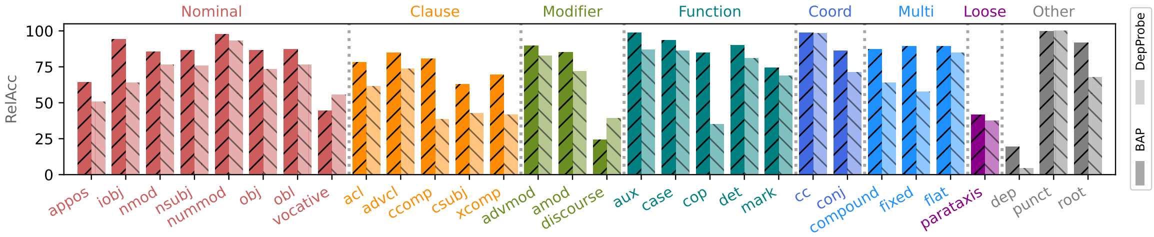


Figure 9: **RelAcc of BAP and DepProbe on EU-BDT (Test)** grouped according to UD taxonomy.

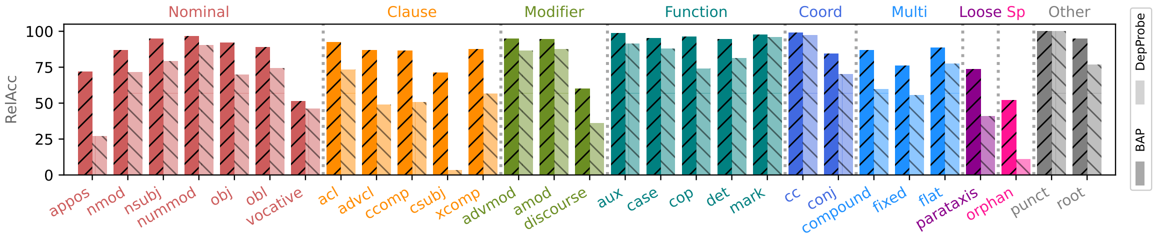


Figure 10: **RelAcc of BAP and DepProbe on FI-TDT (Test)** grouped according to UD taxonomy.

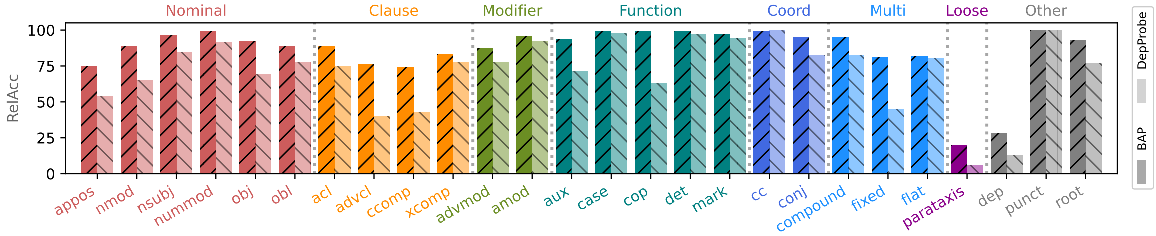


Figure 11: **RelAcc of BAP and DEPPROBE on HE-HTB (Test)** grouped according to UD taxonomy.

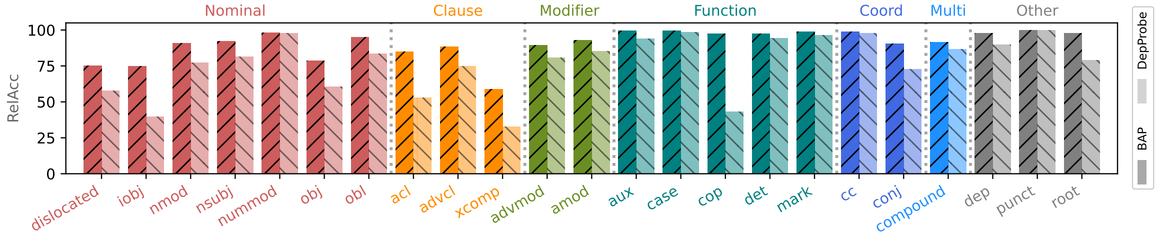


Figure 12: **RelAcc of BAP and DEPPROBE on HI-HDTB (Test)** grouped according to UD taxonomy.

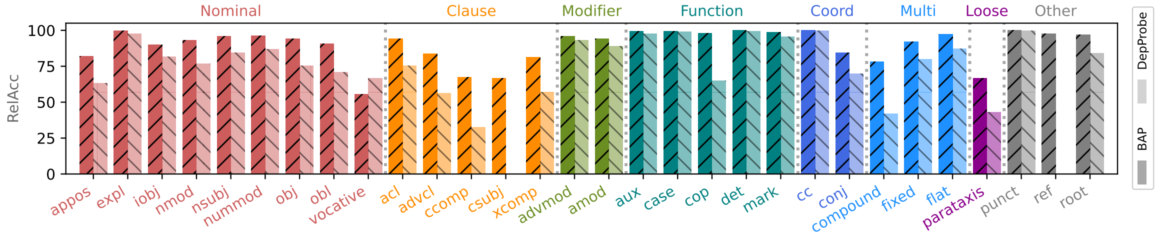


Figure 13: **RelAcc of BAP and DEPPROBE on IT-ISDT (Test)** grouped according to UD taxonomy.

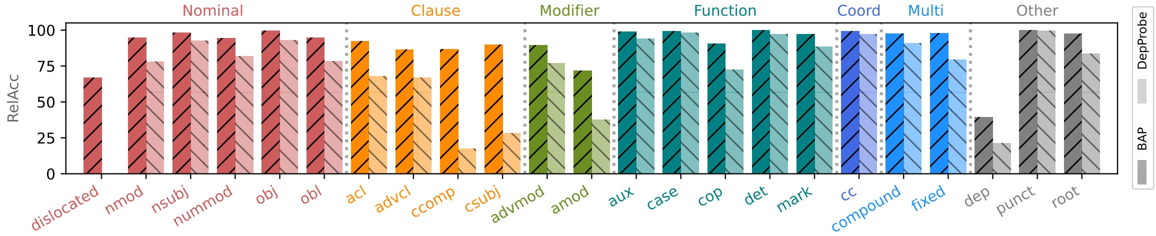


Figure 14: **RelAcc of BAP and DEPPROBE on JA-GSD (Test)** grouped according to UD taxonomy.

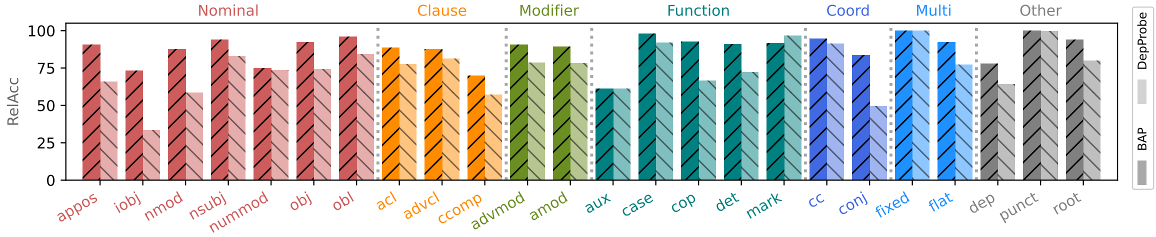


Figure 15: **RelAcc of BAP and DEPPROBE on KO-GSD (Test)** grouped according to UD taxonomy.

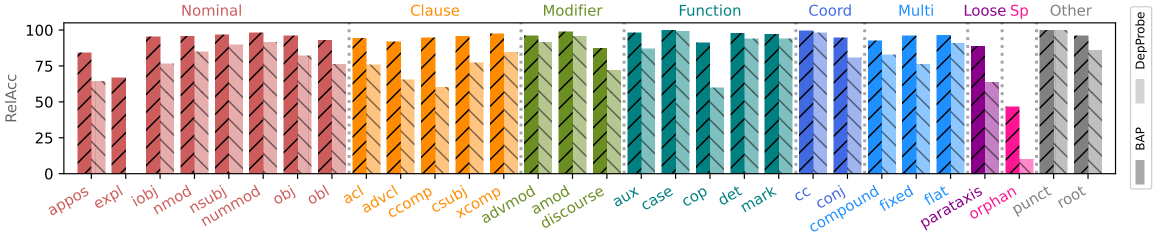


Figure 16: **RelAcc of BAP and DEPPROBE on RU-SynTagRus (Test)** grouped according to UD taxonomy.

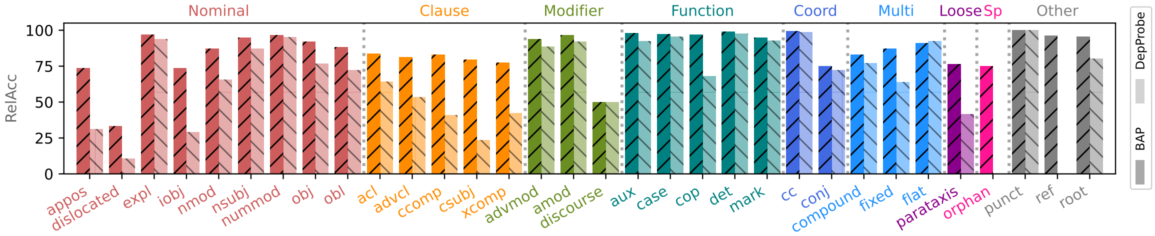


Figure 17: **RelAcc of BAP and DEPPROBE on SV-Talbanken (Test)** grouped according to UD taxonomy.

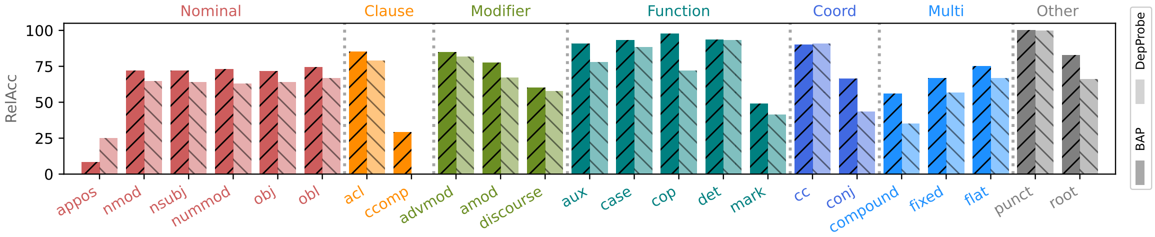


Figure 18: **RelAcc of BAP and DEPPROBE on TR-IMST (Test)** grouped according to UD taxonomy.

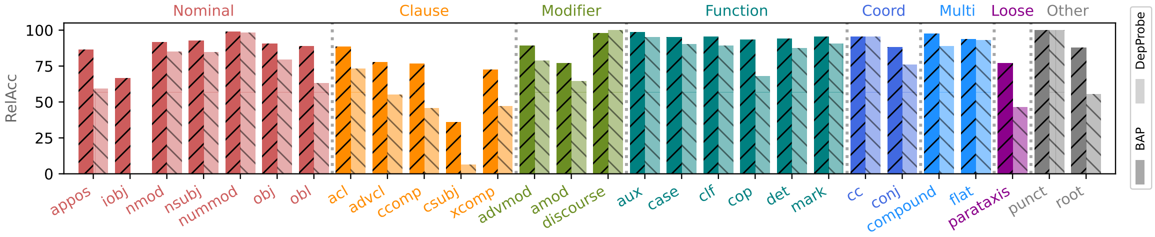


Figure 19: **RelAcc of BAP and DEPPROBE on ZH-GSD (Test)** grouped according to UD taxonomy.