

# The Architectural Bottleneck Principle

Tiago Pimentel\*, Josef Valvoda\*, Niklas Stoehr Ryan Cotterell


 University of Cambridge  ETH Zürich

{tp472, jv406}@cam.ac.uk

{niklas.stoehr, ryan.cotterell}@inf.ethz.ch

## Abstract

In this paper, we seek to measure how much information a component in a neural network could extract from the representations fed into it. Our work stands in contrast to prior probing work, most of which investigates how much information a model’s representations *contain*. This shift in perspective leads us to propose a new principle for probing, the **architectural bottleneck principle**: In order to estimate how much information a given component could extract, a probe should look exactly like the component. Relying on this principle, we estimate how much syntactic information is available to transformers through our attentional probe, a probe that exactly resembles a transformer’s self-attention head. Experimentally, we find that, in three models (BERT, ALBERT, and RoBERTa), a sentence’s syntax tree is mostly extractable by our probe, suggesting these models have access to syntactic information while composing their contextual representations. Whether this information is actually used by these models, however, remains an open question.

 <https://github.com/rycolab/attentional-probe>

## 1 Introduction

The surprising performance of pretrained language models on diverse natural language processing tasks has sparked interest in their analysis. Probing is one of the most prevalent methods employed to engage in such an analysis. In a typical probing study (Alain and Bengio, 2016; Belinkov et al., 2017; Adi et al., 2017, *inter alia*), the weights of the model under consideration are first frozen. A probe is then trained on top of the model’s contextual representations in an attempt to predict one of the input sentence’s properties, e.g., its syntactic parse. Unfortunately, best practices on how to design such probes remain contested.

On one side of the debate, some argue for simplicity, suggesting that simple probes are to be

preferred so that we can distinguish probing from simply learning an NLP task (Hewitt and Liang, 2019). On the other side of the debate, some argue we need complex probes in order to extract all relevant information from the representations (Saphra and Lopez, 2019; Pimentel et al., 2020b). Bridging the gap, some have also called for a compromise, advocating that all probes on the complexity–accuracy Pareto curve should be considered (Pimentel et al., 2020a).

In this paper, we propose the **architectural bottleneck principle** (ABP) as a guideline for constructing useful probes. Under the ABP, a probe’s architecture should mirror a component of the model being probed. Previous work has mostly focused on how much information is contained in a set of representations. However, if we care about whether the information is in fact used by the model, we should instead ask how much information the model in question could use.<sup>1</sup> Under this perspective, the probed model’s architecture acts as a natural bottleneck to how much information the model could use—and should thus also act as a constraint when probing.<sup>2</sup>

As a concrete example, we posit that a transformer’s attention head serves as a bottleneck to its use of syntactic information, as these are the only components in a transformer with access to multiple tokens at once. Following the ABP, we thus propose the **attentional probe**, which looks exactly like an attention head. This probe allows us to answer one specific question: How much syntactic information could a transformer use while computing its attention weights?

Our results reveal that most—albeit not all—syntactic information is extractable with this simple attention head architecture: While we estimate English sentences to contain on average

<sup>1</sup>Explicitly, we use the bigram *could extract* to refer to the total amount of information a component is able to extract from the representations fed into it; this upper-bounds the information that component actually uses.

<sup>2</sup>For related work investigating whether a model uses some information, see Ravfogel et al. (2021) and Lasri et al. (2022).

\*Equal contribution.

31.2 bits of information about their syntactic tree structure, the attentional probe can extract up to 28.0 bits. Furthermore, while these results hold for three popular transformer-based language models (BERT, ALBERT and RoBERTa), they do not for a similar but untrained model. This suggests that training a model shapes its representations to encode syntactic information. We find this trend holds across four typologically diverse languages (Basque, English, Tamil, and Turkish). In contrast, when we keep BERT’s pretrained parameters frozen and analyze the weights of its pretrained attention heads, we observe that they do not seem to encode syntax under our operationalisation. Ergo, while we know these models could use syntactic information to compute attention weights, whether they actually do remains an open question.

## 2 A Taxonomy of Probing Principles

There are many competing approaches for how to design an effective probe (Belinkov and Glass, 2019). We taxonomise them into principles here.

**1. The Linearity Principle** (Alain and Bengio, 2016). *A neural network’s purpose is to make information linearly separable for its final layer. Thus, probes should be linear models.*

Focusing on how much information a model could use in its final layer,<sup>3</sup> Alain and Bengio (2016) propose what we term the linearity principle; many subsequent studies then adopted it in designing their probes (Shi et al., 2016; Ettinger et al., 2016; Bisazza and Tump, 2018; Liu et al., 2019a). Other researchers, however, argued that a model’s non-final layers do not necessarily encode information linearly (Conneau et al., 2018; Pimentel et al., 2020b). They then suggested that a probe should measure the *total* amount of information present in a model’s representations—*independent* of whether it is actually used by the model. This led to a second principle, which we outline below.

**2. The Maximum Information Principle** (Pimentel et al., 2020b). *A probe’s goal is to estimate how much information is encoded in a set of representations. Thus, probes should be as complex as necessary to extract all relevant information.*

Following this principle, some authors have found, unsurprisingly, that non-linear probes estimate larger amounts of information to be encoded in

a representation than linear ones (Qian et al., 2016; Belinkov et al., 2017; White et al., 2021). Pimentel et al. (2020b), however, argued that all contextual representations, e.g., the ones produced by BERT, encode as much information about a target attribute as the original sentence. It follows that probing only makes sense with some constraint on probe complexity. Taking complexity into account suggests another natural principle for probe design.

**3. The Easy-extraction Principle** (Hewitt and Liang, 2019). *The goal of probing is to reveal how easy it is to extract the information encoded in the representations. Thus, probes should be as simple as possible without sacrificing performance.*

The idea of preferring simple probes goes by many names in the literature. Some authors discuss the complexity of probing architectures (Hewitt and Liang, 2019; Voita and Titov, 2020; Pimentel et al., 2020a; Cao et al., 2021), while others discuss the amount of data required to train the probe (Pimentel and Cotterell, 2021). None of the work above, however, discusses how the model actually uses the information about the target attribute (Elazar et al., 2021; Lasri et al., 2022). If we are interested in whether information can be used by the model, we need a new principle. In this work, we argue that a model’s architecture should factor into the probe’s design, because the model’s architecture constrains the amount of information the model can use. This leads us to propose the following principle.

**4. The Architectural Bottleneck Principle (ABP).** *A probe should measure how much information a component of a model could use. Thus, a probe’s architecture should mirror that component.*

We believe the ABP naturally connects the first three principles. Importantly, the ABP generalises the linearity principle: If a model employs a linear projection coupled with a softmax in its final layer, and our probe mirrors that layer, as linear probes do, then the ABP will be equivalent to the linearity principle. Furthermore, the ABP also relates to the maximum information principle: If we probe a component that is expressive enough, it should be able to extract all relevant information from a set of representations. Finally, the ABP also implicitly controls for ease of extraction by restricting the capacity of probes.

## 3 Probing with Information Theory

In this paper, we take the position that the goal of probing is to determine how much information one

<sup>3</sup>We assume throughout this paper that a model’s final layer is a linear projection coupled with a softmax nonlinearity.

can extract from the representations being probed. Following Pimentel et al. (2020b), we now operationalise this value formally using information theory, which offers us a clean framework to quantify information. Specifically, the measure we are interested in is a  $\mathcal{V}$ -information (Xu et al., 2020).<sup>4</sup>

### 3.1 Mutual Information

Shannon (1948) famously quantified the amount of information that a random variable ( $\mathbf{R}$ ) contains about another ( $\mathbf{A}$ ) as their mutual information

$$I(\mathbf{R}; \mathbf{A}) \stackrel{\text{def}}{=} H(\mathbf{A}) - H(\mathbf{A} | \mathbf{R}) \quad (1)$$

where  $H(\mathbf{A})$  and  $H(\mathbf{A} | \mathbf{R})$  are, respectively, the entropy of  $\mathbf{A}$  and the conditional entropy of  $\mathbf{A}$  given  $\mathbf{R}$ . Given that  $\mathbf{R}$  is a continuous-valued representation with values  $\mathbf{r} \in \mathcal{R}$ , and  $\mathbf{A}$  is a discrete-valued attribute with values  $\mathbf{a} \in \mathcal{A}$ , these quantities are defined formally as

$$H(\mathbf{A}) \stackrel{\text{def}}{=} \sum_{\mathbf{a} \in \mathcal{A}} p(\mathbf{a}) \log \frac{1}{p(\mathbf{a})} \quad (2)$$

$$H(\mathbf{A} | \mathbf{R}) \stackrel{\text{def}}{=} \int_{\mathcal{R}} \sum_{\mathbf{a} \in \mathcal{A}} p(\mathbf{r}, \mathbf{a}) \log \frac{1}{p(\mathbf{a} | \mathbf{r})} d\mathbf{r} \quad (3)$$

The maximum information principle seeks to estimate Eq. (1). Notably, the relationship between  $\mathbf{R}$  and  $\mathbf{A}$ , represented by distribution  $p(\mathbf{a} | \mathbf{r})$ , may be arbitrarily complex, and this distributions' computational complexity has no direct effect on the conditional entropy's value  $H(\mathbf{A} | \mathbf{R})$ .

### 3.2 $\mathcal{V}$ -information

Under the architectural bottleneck principle, we are interested in how much information we can extract from  $\mathbf{R}$  about  $\mathbf{A}$ , when constrained to only using extraction functions in a set  $\mathcal{V}$ , the set of functions a model's component can represent. The  $\mathcal{V}$ -information (Xu et al., 2020), a generalisation of Shannon's (1948) mutual information, naturally operationalises this value as

$$I_{\mathcal{V}}(\mathbf{R} \rightarrow \mathbf{A}) \stackrel{\text{def}}{=} H_{\mathcal{V}}(\mathbf{A}) - H_{\mathcal{V}}(\mathbf{A} | \mathbf{R}) \quad (4)$$

where the conditional  $\mathcal{V}$ -entropy is defined as

$$H_{\mathcal{V}}(\mathbf{A} | \mathbf{R}) \stackrel{\text{def}}{=} \inf_{q \in \mathcal{V}} \int_{\mathcal{R}} \sum_{\mathbf{a} \in \mathcal{A}} p(\mathbf{r}, \mathbf{a}) \log \frac{1}{q(\mathbf{a} | \mathbf{r})} d\mathbf{r} \quad (5)$$

<sup>4</sup>We operationalise our measure of interest as a  $\mathcal{V}$ -information here, since *information* is usually the term used to describe probing questions. Our principle, however, can be easily extended to other measures. For instance, we can define  $\mathcal{V}$ -UAS as the supremum unlabelled attachment score (UAS) achievable by an architecture.

The unconditional  $\mathcal{V}$ -entropy is defined similarly.

In words, the  $\mathcal{V}$ -information computes the maximum information that can be extracted by a model with an architecture  $\mathcal{V}$ . Notably, if  $\mathcal{V}$  is sufficiently expressive, i.e., if  $p(\mathbf{a} | \mathbf{r}) \in \mathcal{V}$ , the  $\mathcal{V}$ -information will be equivalent to the traditional mutual information. Further,  $\mathcal{V}$ -information is bounded above by the mutual information, which leads to a new value termed here the  **$\mathcal{V}$ -coefficient**

$$C_{\mathcal{V}}(\mathbf{A} | \mathbf{R}) \stackrel{\text{def}}{=} \frac{I_{\mathcal{V}}(\mathbf{R} \rightarrow \mathbf{A})}{I(\mathbf{R}; \mathbf{A})} \quad (6)$$

In short, the  $\mathcal{V}$ -coefficient computes the percentage of information we can extract from a random variable when restricted to variational family  $\mathcal{V}$ .

## 4 An Attentional Probe

In our experiments, we will focus on a transformer's attention mechanism. Concretely, many researchers (e.g., Vig and Belinkov, 2019; Htut et al., 2019; Manning et al., 2020) have asserted that syntactic information is used by transformers when computing their attention weights (albeit not uncontroversially; for a review, see Rogers et al., 2021). Further, attention heads are the only components in a transformer which have access to multiple words at the same time. Thus, exploring the ABD in the context of attention heads is a natural starting point. Specifically, following the ABP, we will investigate how much information a transformer's attention head could extract from the representations fed into it.

Given an input sentence  $\mathbf{s}$ , a transformer (Vaswani et al., 2017) will generate a set of representations  $\mathbf{r} \in \mathcal{R} \stackrel{\text{def}}{=} \mathbb{R}^{|\mathbf{s}| \times d_1}$  at layer  $\ell$ . An attention head then uses these representations to compute the attention weights

$$\alpha_{ij} = (\mathbf{K}\mathbf{r}_i)^{\top} \mathbf{Q}\mathbf{r}_j, \quad w_{ij} = \frac{e^{\alpha_{ij}}}{\sum_{1 \leq j' \leq |\mathbf{s}|} e^{\alpha_{ij'}}} \quad (7)$$

where  $i$  and  $j$  index word positions in a sentence  $\mathbf{s}$ ,  $\mathbf{K}, \mathbf{Q} \in \mathbb{R}^{d_2 \times d_1}$  are the key and query matrices, and  $\alpha, \mathbf{w} \in \mathbb{R}^{|\mathbf{s}| \times |\mathbf{s}|}$  are, respectively, the self-attention logits and attention weights.

We now consider an attentional probe parameterised using the head in Eq. (7), but with randomly initialised  $\mathbf{K}$  and  $\mathbf{Q}$  matrices. Our goal is to train this probe, as we explain towards the end of this section. We use the attention weights, defined in Eq. (7), to compute the probability of

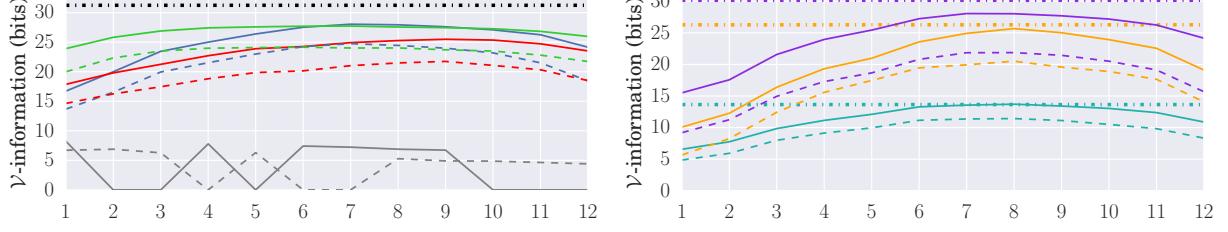


Figure 1:  $\mathcal{V}$ -information per layer of our probes evaluated on: (left) English using **BERT**, **RoBERTa**, **ALBERT** and **untrained** representations; (right) **Basque**, **Turkish**, and **Tamil** using BERT representations. Line patterns represent: ---- Mutual information; — Attentional  $\mathcal{V}$ -information; .... Structural  $\mathcal{V}$ -information.

a specific directed spanning tree  $\mathbf{a}$ , which encodes the syntactic dependencies

$$q_{\theta}(\mathbf{a} \mid \mathbf{r}) = \frac{\prod_{(i,j) \in \mathbf{a}} w_{ij}}{\sum_{\mathbf{a}' \in \mathcal{A}_{|\mathbf{s}|}} \prod_{(i,j) \in \mathbf{a}'} w_{ij}} \quad (8)$$

where tree  $\mathbf{a}$  is represented as a set of pairs  $(i, j)$  which index an edge in it,  $\mathcal{A}_{|\mathbf{s}|}$  represents the set of all directed spanning trees with a specific number of nodes  $|\mathbf{s}|$ , and  $\theta = [\mathbf{K}; \mathbf{Q}] \in \mathbb{R}^{d_2 \times (2d_1)}$  represents the probe’s parameters. We can easily compute the numerator in this equation for a specific tree. The normalising factor in the denominator is more complex, as it requires looping through a prohibitively large sum. Luckily, we can efficiently compute it with Koo et al.’s (2007) variation of the matrix–tree theorem (MTT) for root-constrained directed spanning trees (Tutte, 1984; Zmigrod et al., 2020).<sup>5</sup>

**The Variational Family.** The attentional probe architecture is defined by Eqs. (7) and (8). We now define the equivalent variational family

$$\mathcal{V} = \left\{ q_{\theta}(\mathbf{a} \mid \mathbf{r}) \mid \mathbf{K}, \mathbf{Q} \in \mathbb{R}^{d_2 \times d_1} \right\} \quad (9)$$

This variational family includes the set of all distributions computable by an attention head architecture. In practice, however, we cannot compute the infimum over the set  $\mathcal{V}$  as required in Eq. (5). As an approximation, we train our attentional probe to minimise a cross-entropy loss, which gives us an estimate of the  $\mathcal{V}$ -entropy. We expand on this point in App. A.1.<sup>6</sup>

<sup>5</sup>Importantly, Koo et al.’s (2007) method requires a set of weights between each word and a sentence’s root. To handle this, we feed an extra root representation  $\mathbf{r}_0$ , initialised as a vector with all zeros, to our attentional probe, making our attention weights actually have shape  $\mathbf{w} \in \mathbb{R}^{|\mathbf{s}|+1 \times |\mathbf{s}|+1}$ . Explicitly, adding a root to an undirected dependency tree is equivalent to making it directed. We then use Zmigrod et al.’s (2021) implementation of the matrix–tree theorem.

<sup>6</sup>We note that Hewitt et al. (2021) first noted the equivalence between estimating a  $\mathcal{V}$ -information and probing.

## 5 Experiments

**Data.** We use the universal dependencies’ (UD) treebanks (Zeman et al., 2020). Specifically, we analyse results in four typologically diverse languages: Basque, English, Tamil, and Turkish. Furthermore, we focus our analysis on unlabelled dependency trees. We note that UD uses a particular syntax formalism, which could impact our results (Kuznetsov and Gurevych, 2020).

**Models.** Empirically, we study multilingual BERT in all four languages under consideration (Devlin et al., 2019) as well as RoBERTa and ALBERT (Liu et al., 2019b; Lan et al., 2020), which are only available in English. In line with the ABD, we keep our probe’s hidden size the same as in the probed architectures. Finally, we also probe an *untrained* transformer model with the same architecture as BERT as a baseline.

**Training.** We train our probes with AdamW (Loshchilov and Hutter, 2019) using its default hyper-parameters in PyTorch (Paszke et al., 2019).<sup>7</sup>

**Baselines and Skylines.**<sup>8</sup> We contrast our attentional probe’s  $\mathcal{V}$ -information against two other values. First, as a baseline, we investigate a special case of our model where  $\mathbf{K} = \mathbf{Q}$ , inspired by recent work on structural probing (Hewitt and Liang, 2019; Maudslay et al., 2020; White et al., 2021). Notably, this equality leads to symmetric attention weight matrices; by modelling the root explicitly, however, we still get a distribution over directed trees. This baseline evaluates whether previous work, by over-constraining their probes, has underestimated the amount of information available to a transformer’s attention mechanism. We report this value as **structural  $\mathcal{V}$ -information**. Second, as a skyline, we compare our attentional probe to an estimate of the true **mutual information**  $I(\mathbf{R}; \mathbf{A})$ ,

<sup>7</sup>The estimation of  $H_{\mathcal{V}}(A)$  is described in App. D.

<sup>8</sup>We describe both approaches in more detail in App. C.



Model	Layer	$I_V$	I	$C_V$
BERT	7	28.0	31.2	90%
RoBERTa	9	25.5	31.2	82%
ALBERT	7	27.7	31.2	89%

Table 1: Maximum  $V$ -informations ( $I_V$ ) and  $V$ -coefficients ( $C_V$ ) estimated in English in each probed model, together with the layer in which they occur. We also display the estimated mutual information (I).

for which we follow Pimentel et al. (2020b) in using a deep neural network (DNN) to estimate.<sup>9</sup>

## 6 Results

We present our main results in Fig. 1. First, our probes estimate that most syntactic information is extractable in the middle layers, as previously reported by Tenney et al. (2019). Second, Fig. 1 shows that a large amount of syntactic information is encoded in the representations fed to the attention heads. Further, while we estimate close to 31 bits of information to be encoded in English, Tamil, and Basque sentences, we only estimate around 15 bits to be encoded in Turkish sentences; we suspect this is due to Turkish having the shortest sentences in the corpus (see App. I for these lengths).

Third, we find that, out of the total syntactic information present in the sentences, nearly all is available to the transformer-based models under consideration. In English, for instance, we find the  $V$ -coefficient of the most informative layer to be 90%, 82%, and 89% in BERT, RoBERTa and ALBERT, respectively; see Tab. 1. This means they have access to roughly 85% of all syntactic information in a sentence. These trends are consistent across the four languages we have considered. Notably, this is not the case for the untrained BERT representations, which suggests this structure is a byproduct of the language models’ pretraining procedures.

Additionally, we find that our structural baseline considerably underestimates the models’ potential ability to reconstruct a syntax tree; the best English structural baseline recovers only 23 bits of information (versus 28 bits by the attentional probe). One can see this effect in Fig. 1, where all of the structural baseline results fall beneath their corresponding attentional probe counterparts.<sup>10</sup>

<sup>9</sup>We note that, as demonstrated by Pimentel et al. (2020b), the mutual information  $I(\mathbf{R}; \mathbf{A})$  is constant across contextual representations and equivalent to  $I(\mathbf{S}; \mathbf{A})$ . We thus use our single best approximation of it in each language as our estimate.

<sup>10</sup>We provide unlabelled attachment scores in App. F.

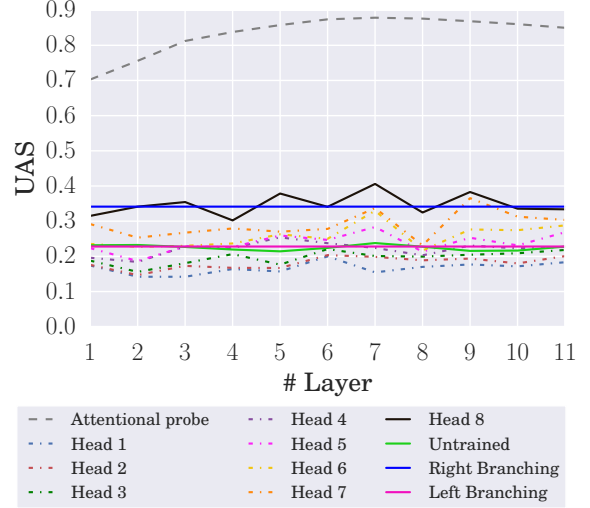


Figure 2: UAS of the attentional weights computed by BERT (with its pretrained weights frozen) in English. We also display the UAS achieved by the attentional probe, the best head per layer of an untrained BERT, and a right and left-branching baseline.

In a final experiment, we plug BERT’s attention weights, as computed with its pretrained attention heads, directly into Eq. (8) and analyse its resulting unlabelled attachment scores. These results are presented in Fig. 2 for English (as well as in App. H for the other analysed languages). In short, they reveal that, while attention heads *could* use a large amount of syntactic information, none of the actual heads compute weights that strongly resemble syntax trees; see Htut et al. 2019 for similar results. As BERT has 8 attention heads, however, it might be the case that the syntactic information is used in a distributed manner, with each head relying on a subset of this information (see Tab. 3 in Clark et al. 2019 for results partly supporting this hypothesis).

## 7 Conclusions

In this paper, we have approached probing from a new perspective. Rather than asking how much information is encoded by the model, we ask how much information its components could extract. We then quantify this amount using  $V$ -information. Evaluating the attention mechanism of popular transformer language models, we find that the majority of the information about the syntax tree of a sentence is in fact extractable by the model. This, however, is not true for randomly initialised transformer models. Our results, thus, lead us to conclude that a transformer’s training leads its attention heads to have the potential to decode syntax trees.

## Acknowledgements

We thank the reviewers and action editor for their helpful comments. We also thank Kevin Du, Clara Meister, Lucas Torroba Hennigen, and Afra Amini for their feedback on this manuscript.

## Limitations

In this paper, we propose a new principled way to choose a probe’s architecture, operationalising the question “how much information *could* a model extract from a set of representations?” in terms of a  $\mathcal{V}$ -information. We note, however, that this probe design principle is only applicable to answer the specific question above. Explicitly, we do *not* answer what we believe to be the more interesting question: “how much information *does* a model actually extract from a set of representations?” In practice, while our proposed probing method does not answer this second question, it does offer an upperbound for it; the amount of information a model could extract from a set of representations is strictly larger than the amount actually extracted. Quantifying how tight (or loose) this upperbound is remains future work.

## Ethical Concerns

We foresee no ethical concerns with this work.

## References

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. [Fine-grained analysis of sentence embeddings using auxiliary prediction tasks](#). In *International Conference on Learning Representations*.
- Itziar Aduriz, Maria Jesus Aranzabe, Jose Maria Arriola, Aitziber Atutxa, Arantza Diaz de Ilaraza, Aitzpea Garmendia, and Maite Oronoz. 2003. [Construction of a Basque dependency treebank](#). In *Treebanks and Linguistic Theories*.
- Guillaume Alain and Yoshua Bengio. 2016. [Understanding intermediate layers using linear classifier probes](#). *arXiv preprint arXiv:1610.01644*.
- Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. [What do neural machine translation models learn about morphology?](#) In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.
- Yonatan Belinkov and James Glass. 2019. [Analysis Methods in Neural Language Processing: A Survey](#). *Transactions of the Association for Computational Linguistics*, 7:49–72.
- Arianna Bisazza and Clara Tump. 2018. [The lazy encoder: A fine-grained analysis of the role of morphology in neural machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2871–2876, Brussels, Belgium. Association for Computational Linguistics.
- Steven Cao, Victor Sanh, and Alexander Rush. 2021. [Low-complexity probing via finding subnetworks](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 960–966, Online. Association for Computational Linguistics.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. [What does BERT look at? An analysis of BERT’s attention](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. Association for Computational Linguistics.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single  \$\\$ \&! \# \*\$  vector: Probing sentence embeddings for linguistic properties](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
- 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 *International Conference on Learning Representations*.
- Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. [Amnesic probing: Behavioral explanation with amnesic counterfactuals](#). *Transactions of the Association for Computational Linguistics*, 9:160–175.
- Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. [Probing for semantic evidence of composition by means of simple classification tasks](#). In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139, Berlin, Germany. Association for Computational Linguistics.

- John Hewitt, Kawin Ethayarajh, Percy Liang, and Christopher Manning. 2021. [Conditional probing: measuring usable information beyond a baseline](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1626–1639, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- John Hewitt and Percy Liang. 2019. [Designing and interpreting probes with control tasks](#). 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.
- 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.
- Phu Mon Htut, Jason Phang, Shikha Bordia, and Samuel R. Bowman. 2019. [Do attention heads in BERT track syntactic dependencies?](#) *CoRR*, abs/1911.12246.
- Terry Koo, Amir Globerson, Xavier Carreras, and Michael Collins. 2007. [Structured prediction models via the matrix-tree theorem](#). In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, Prague, Czech Republic. Association for Computational Linguistics.
- Iliia Kuznetsov and Iryna Gurevych. 2020. [A matter of framing: The impact of linguistic formalism on probing results](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 171–182, Online. Association for Computational Linguistics.
- Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. [ALBERT: A lite BERT for self-supervised learning of language representations](#). In *The 8th International Conference on Learning Representations*.
- Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. [Probing for the usage of grammatical number](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. [Linguistic knowledge and transferability of contextual 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 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. [RoBERTa: A robustly optimized BERT pretraining approach](#). *CoRR*, abs/1907.11692.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *International Conference on Learning Representations*.
- Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. [Emergent linguistic structure in artificial neural networks trained by self-supervision](#). *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.
- 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.
- 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- Tiago Pimentel and Ryan Cotterell. 2021. [A Bayesian framework for information-theoretic probing](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2869–2887, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Tiago Pimentel, Naomi Saphra, Adina Williams, and Ryan Cotterell. 2020a. [Pareto probing: Trading off accuracy for complexity](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3138–3153, Online. Association for Computational Linguistics.
- Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020b. [Information-theoretic probing for linguistic structure](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.



- Peng Qian, Xipeng Qiu, and Xuanjing Huang. 2016. [Investigating language universal and specific properties in word embeddings](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1478–1488, Berlin, Germany. Association for Computational Linguistics.
- Loganathan Ramasamy and Zdeněk Žabokrtský. 2012. [Prague dependency style treebank for Tamil](#). In *Proceedings of Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pages 1888–1894, Istanbul, Turkey.
- Shauli Ravfogel, Grusha Prasad, Tal Linzen, and Yoav Goldberg. 2021. [Counterfactual interventions reveal the causal effect of relative clause representations on agreement prediction](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 194–209, Online. Association for Computational Linguistics.
- Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2021. [A primer in BERTology: What we know about how BERT works](#). *Transactions of the Association for Computational Linguistics*, 8:842–866.
- Naomi Saphra and Adam Lopez. 2019. [Understanding learning dynamics of language models with SVCCA](#). 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.
- Claude E. Shannon. 1948. [A mathematical theory of communication](#). *The Bell System Technical Journal*, 27(3):379–423.
- Xing Shi, Inkit Padhi, and Kevin Knight. 2016. [Does string-based neural MT learn source syntax?](#) In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1526–1534, Austin, Texas. Association for Computational Linguistics.
- Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Christopher D. Manning. 2014. [A gold standard dependency corpus for English](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*.
- 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 rediscovered the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- W. T. Tutte. 1984. *Graph Theory*. Addison-Wesley Publishing Company.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Jesse Vig and Yonatan Belinkov. 2019. [Analyzing the structure of attention in a transformer language model](#). In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.
- Elena Voita and Ivan Titov. 2020. [Information-theoretic probing with minimum description length](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.
- Jennifer C. White, Tiago Pimentel, Naomi Saphra, and Ryan Cotterell. 2021. [A non-linear structural probe](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 132–138, Online. Association for Computational Linguistics.
- 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.
- Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. [A theory of usable information under computational constraints](#). In *International Conference on Learning Representations*.
- Daniel Zeman, Joakim Nivre, Mitchell Abrams, Elia Ackermann, Noëmi Aepli, Željko Agić, Lars Ahrenberg, Chika Kennedy Ajede, Gabrielė Aleksandravičiūtė, Lene Antonsen, Katya Aplonova, Angelina Aquino, Maria Jesus Aranzabe, Gashaw Arutie, Masayuki Asahara, Luma Ateyah, Furkan Atmaca, Mohammed Attia, Aitziber Atutxa, Liesbeth Augustinus, Elena Badmaeva, Miguel Ballesteros, Esha Banerjee, Sebastian Bank, Verginica Barbu Mititelu, Victoria Basmov, Colin Batchelor, John Bauer, Kepa Bengoetxea, Yevgeni Berzak,





*Processing (EMNLP)*, pages 4809–4819, Online. Association for Computational Linguistics.

Ran Zmigrod, Tim Vieira, and Ryan Cotterell. 2021. [Efficient computation of expectations under spanning tree distributions](#). *Transactions of the Association for Computational Linguistics*, 9:675–690.

## A More on the $\mathcal{V}$ -information

### A.1 Probing as approximating $\mathcal{V}$ -information

In this section, we make a similar argument to [Hewitt et al.’s \(2021\)](#), who first pointed out the equivalence between the goals of probing and estimating a  $\mathcal{V}$ -information. When probing for some information, we typically train a probabilistic classifier  $q_\theta(\mathbf{a}|\mathbf{r})$  (with parameters  $\theta$ ) to approximate a target probability distribution  $p(\mathbf{a}|\mathbf{r})$ . We do this by using an empirical cross-entropy loss function

$$\mathcal{L}(\mathcal{D}_{\text{train}}; \theta) \stackrel{\text{def}}{=} \sum_{(\mathbf{r}, \mathbf{a}) \in \mathcal{D}_{\text{train}}} \log \frac{1}{q_\theta(\mathbf{a}|\mathbf{r})} \quad (10)$$

where  $\mathcal{D}_{\text{train}}$  is a training set composed of  $(\mathbf{r}, \mathbf{a})$  pairs, which are assumed to be sampled from the true distribution  $p(\mathbf{r}, \mathbf{a})$ . Further, we usually have access to a development set  $\mathcal{D}_{\text{dev}}$ , on which we estimate this same loss  $\mathcal{L}(\mathcal{D}_{\text{dev}}; \theta)$  and which we use to avoid overfitting. Together, these steps aim at making  $q_\theta(\mathbf{a}|\mathbf{r})$  approximate the distribution which minimises the true cross-entropy

$$H_\theta(\mathbf{A} | \mathbf{R}) \stackrel{\text{def}}{=} \int_{\mathbf{R}} \sum_{\mathbf{a} \in \mathcal{A}} p(\mathbf{r}, \mathbf{a}) \log \frac{1}{q_\theta(\mathbf{a}|\mathbf{r})} d\mathbf{r} \quad (11)$$

Minimising this cross-entropy is equivalent to finding the  $q(\mathbf{a}|\mathbf{r}) \in \mathcal{V}$  which minimises the conditional  $\mathcal{V}$ -entropy in Eq. (5). Furthermore, since  $H_\mathcal{V}(\mathbf{A})$  is constant with respect to the representations  $\mathbf{R}$ , this is also equivalent (up to an additive constant) to estimating the  $\mathcal{V}$ -information in Eq. (4)—where  $\mathcal{V}$  is defined by our choice of architecture for the probing classifier.

### A.2 On $\mathcal{V}$ , Expressivity and Learnability

Ideally, a trained probe  $q_\theta(\mathbf{r}|\mathbf{a})$  would converge to the infimum  $q \in \mathcal{V}$  from Eq. (5). In practice, however, limitations on the dataset size and optimisation algorithms may lead to poor approximations. Moreover, even with a trained probe, we still cannot compute Eq. (11), but must instead empirically approximate it with a test set and the loss function in Eq. (10). We can thus decompose our actually measured value into four terms

$$\mathcal{L}(\mathcal{D}_{\text{test}}; \theta) = H(\mathbf{A} | \mathbf{R}) + \epsilon_1 + \epsilon_2 + \epsilon_3 \quad (12)$$

where

$$\epsilon_1 \stackrel{\text{def}}{=} \underbrace{H_\mathcal{V}(\mathbf{A} | \mathbf{R}) - H(\mathbf{A} | \mathbf{R})}_{\text{Expressivity Constraint}} \quad (13)$$

$$\epsilon_2 \stackrel{\text{def}}{=} \underbrace{H_\theta(\mathbf{A} | \mathbf{R}) - H_\mathcal{V}(\mathbf{A} | \mathbf{R})}_{\text{Training Constraint}} \quad (14)$$

$$\epsilon_3 \stackrel{\text{def}}{=} \underbrace{\mathcal{L}(\mathcal{D}_{\text{test}}; \theta) - H_\theta(\mathbf{A} | \mathbf{R})}_{\text{Measurement Error}} \quad (15)$$

Given a large enough testset,  $\epsilon_3$  should be roughly zero, as the empirical loss in Eq. (10) is an unbiased estimator of the cross-entropy in Eq. (11). This leaves  $\epsilon_1$  and  $\epsilon_2$ . While  $\epsilon_1$  is intentionally imposed by the choice of  $\mathcal{V}$ , which defines the expressivity constraints on the structure of the learned information extractors,  $\epsilon_2$  is a byproduct of multiple factors:  $\mathcal{V}$  itself, the optimisation algorithm and both the train and devset sizes.

Analysing the  $\mathcal{V}$ -information of a very expressive variational family may thus be vacuous, as we may expect  $\epsilon_1$  to be relatively small compared to  $\epsilon_2$ ; this would likely be the case for a  $\mathcal{V}$  resembling the entire BERT architecture.<sup>11</sup> For smaller variational families, however, such as the ones we explore here, we can expect our learning procedures to be well behaved and for  $\epsilon_2$  to be relatively small.

## B Inverse Ablation Perspective

One could view our work as a reversed ablation study. In a typical ablation experiment a component of a model is removed to observe its effect on the functioning of the entire model. The idea is that the observed difference in performance of the model will indicate the relative importance of the component. However, with ablation it is impossible to tell what role the component plays in solving the target task. In comparison, we freeze the entire model up to a particular component we are interested in. Instead of asking how important the component is to the overall goal of the model, we ask how good it is at a task we believe is important towards achieving such goal.

## C Baseline and Skyline

### C.1 Structural Baseline

[Hewitt and Manning \(2019\)](#) propose the structural probe to investigate the encoding of syntactic structure in contextual representations. Intuitively, they

<sup>11</sup>In these scenarios, an information-theoretic measure that accounts for training set sizes might be more meaningful, such as the Bayesian information ([Pimentel and Cotterell, 2021](#)).

probe to which extent they can reconstruct a sentence’s syntactic tree purely from the distance between contextual representations  $\mathbf{r}$ . Instead of learning separate query  $\mathbf{Q}$  and key  $\mathbf{K}$  matrices as we do, however, they limit themselves to a single projection matrix  $\mathbf{B} \in \mathbb{R}^{d_2 \times d_1}$ . Their probe can thus be written as

$$\alpha_{ij} = (\mathbf{B}\mathbf{r}_i)^\top \mathbf{B}\mathbf{r}_j \quad (16)$$

Since we want to train the structural probe with the same cross-entropy parsing loss as our attentional probe, we softmax its distances

$$w_{ij} = \frac{e^{\alpha_{ij}}}{\sum_{1 \leq j' \leq |s|} e^{\alpha_{ij'}}} \quad (17)$$

making it similar to [White et al.’s \(2021\)](#) non-linear structural probe. This is necessary because the MTT we use to compute the denominator in Eq. (8) assumes non-negative inputs. We then train it with the same loss function as our proposed attentional probe, also making it similar to [Maudslay et al.’s \(2020\)](#) structural parser. In practice, thus, our structural baseline’s implementation can be seen as a non-linear structural parser.

## C.2 DNN Parser

To approximate the true mutual information  $I(\mathbf{R}; \mathbf{A})$ , we follow [Pimentel et al. \(2020b\)](#) in using more powerful feed forward neural network probes. Specifically, we rely on a variant of [Dozat and Manning’s \(2017\)](#) parser. We first use two multi-layer perceptrons (MLP), one for the dependent and one for the head token in a dependency arc

$$\mathbf{r}'_i = \text{MLP}(\mathbf{r}_i), \quad \mathbf{r}'_j = \text{MLP}(\mathbf{r}_j) \quad (18)$$

These MLP’s are composed of a number of linear transformations, interweaved with ReLU non-linearities and dropout layers. We then feed both these transformed representations into a biaffine transformation to get the dependency logits

$$\alpha_{ij} = \mathbf{r}'_i{}^\top \mathbf{W} \mathbf{r}'_j \quad (19)$$

Finally, we again make these values non-negative by softmaxing them

$$w_{ij} = \frac{e^{\alpha_{ij}}}{\sum_{1 \leq j' \leq |s|} e^{\alpha_{ij'}}} \quad (20)$$

We train this model with the same cross-entropy loss function as our proposed attentional probe.

To choose the hyper-parameters of this model’s MLP we use random search, training 50 independent models. We random search for the number of layers in  $\{0, 1, 2\}$ , dropout in  $[0.0, 0.5]$ , and the hidden size in  $[32; 512]$ . Furthermore, we note that, as demonstrated by [Pimentel et al. \(2020b\)](#), the mutual information  $I(\mathbf{R}; \mathbf{A})$  is constant across contextual representations and equivalent to  $I(\mathbf{S}; \mathbf{A})$ , where  $\mathbf{S}$  is a random variable representing the original input sentence. We thus use our single best approximation of it in each language as our estimate.

## D Unconditional Entropy Parser

We still need to estimate the unconditional entropies  $H_V(\mathbf{A})$ . As these unconditional entropies are not conditioned on anything, however, we cannot estimate them using the previous parsers directly. Specifically, the representations  $\mathbf{r}_i$  and  $\mathbf{r}_j$  in Eqs. (7), (16) and (18) cannot be used. We sidestep this issue by dropping our contextual representations from these equations and using position embeddings in their place. Importantly, these position embeddings do not depend on the input sentences. In short, we compute these equations as

$$\alpha_{ij} = (\mathbf{K}\mathbf{p}_i)^\top \mathbf{Q}\mathbf{p}_j \quad (\text{attentional}) \quad (21)$$

$$\alpha_{ij} = (\mathbf{B}\mathbf{p}_i)^\top \mathbf{B}\mathbf{p}_j \quad (\text{structural}) \quad (22)$$

$$\mathbf{r}'_i = \text{MLP}(\mathbf{p}_i), \quad \mathbf{r}'_j = \text{MLP}(\mathbf{p}_j) \quad (\text{DNN}) \quad (23)$$

where  $\mathbf{p}_i \in \mathbb{R}^{d_1}$  is a randomly initialised position embedding and is trained with the rest of the probe.

## E Extra Information about Training

We use the base version of all our analysed pre-trained models (taken from the transformers library [Wolf et al., 2020](#)). We train the model with a batch size of 2048, evaluate the model every 100 batches, and stop training when the model does not improve over 10 consecutive evaluations. Both the attentional and structural probes are trained with a dropout of 0.2 (applied both on the raw input representations and on the key and query representations before being multiplied together) and with a hidden size (i.e.  $d_2$ ) of 64—this is the size of the query, and key representations in both BERT, RoBERTa and ALBERT. As our data, we used the treebanks: English EWT ([Silveira et al., 2014](#)); Basque BDT ([Aduriz et al., 2003](#)); Turkish IMST ([Sulubacak et al., 2016](#)); Tamil TTB ([Ramasamy and Žabokrtský, 2012](#)).



## F UAS Results

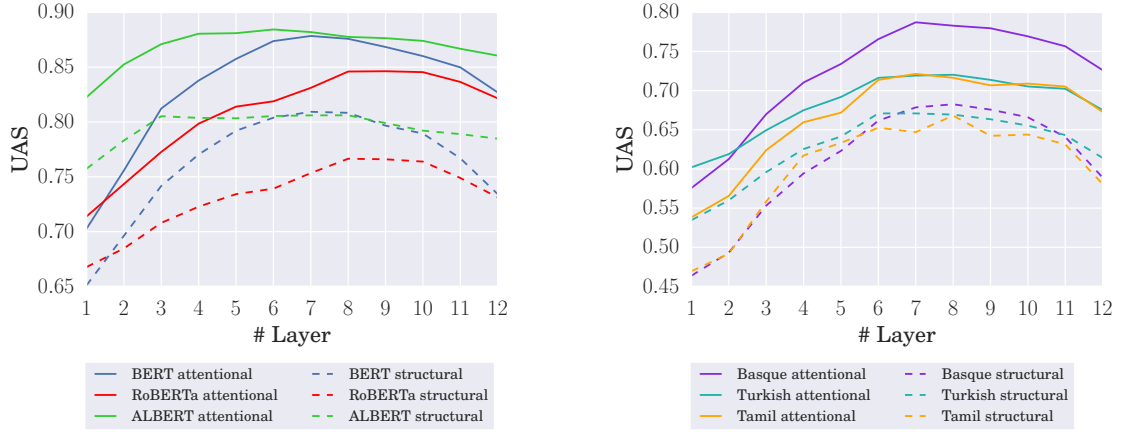


Figure 3: UAS of the probes evaluated on English using BERT, RoBERTa and ALBERT representations (left); Basque, Turkish and Tamil using BERT representations (right).

## G $\mathcal{V}$ -information by Language

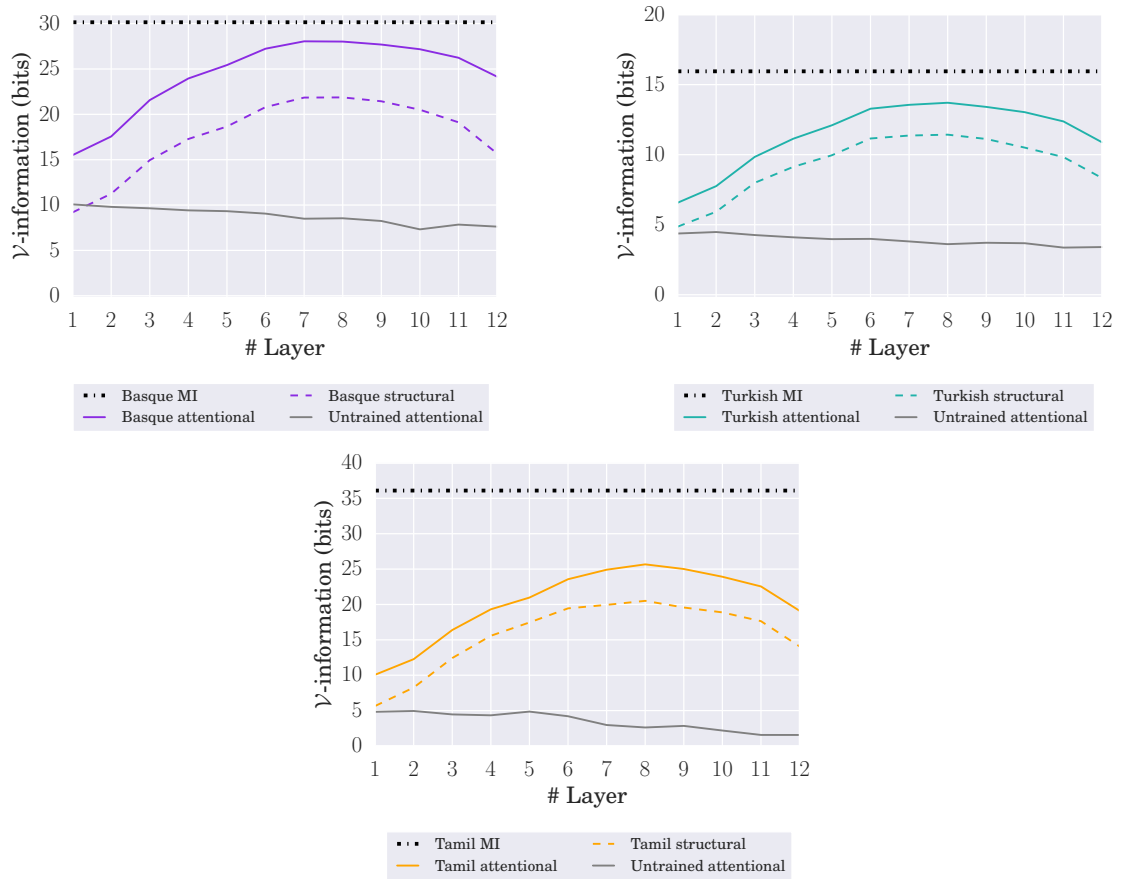


Figure 4: The estimated attentional  $\mathcal{V}$ -information, structural  $\mathcal{V}$ -information, and mutual information on Basque (top-left); Turkish (top-right); and Tamil (bottom).

## H Attention Head Weights Results

We additionally compute the parsing accuracy of the attention heads with their actual weights as taken from BERT (with its parameters as pretrained).<sup>12</sup> In Fig. 5 (as well as Fig. 2 in the main text), we label the heads in order of their performance (1 is always the least accurate per layer, 8 the most). These results show that an attention head’s potential to extract syntax trees is far above what each individual head actually extracts.

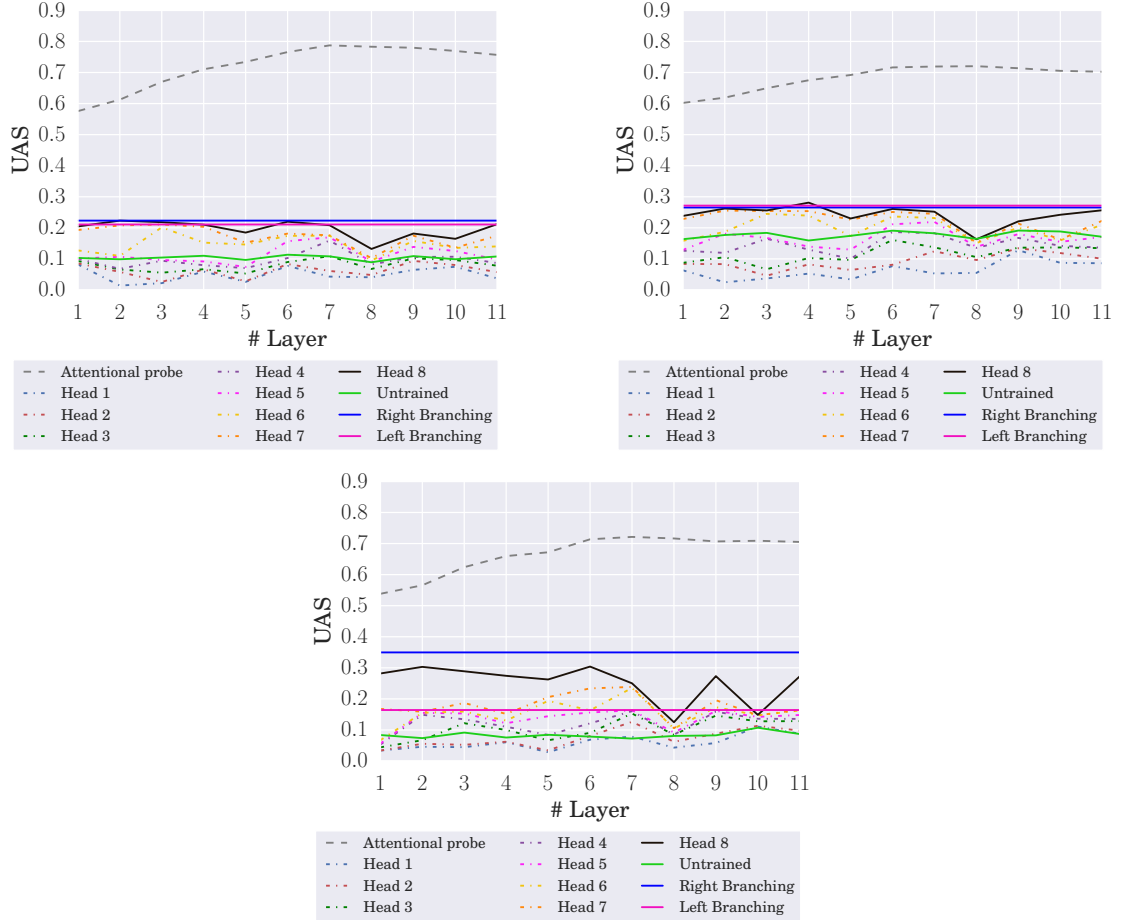


Figure 5: UAS of all attention heads’ weights computed by BERT (with its pretrained parameters frozen) in Basque (top-left); Turkish (top-right); and Tamil (bottom).

## I Average Sentence Lengths

Language	Average Sentence Length
Basque	13
English	15
Tamil	17
Turkish	10

Table 2: The average sentence length per language under consideration.

<sup>12</sup>We assign the weight between each word and the root node as zero, since it is not part of the attention weights.