Conditional probing: measuring usable information beyond a baseline

John Hewitt **Kawin Ethayarajh** **Percy Liang** **Christopher D. Manning**
Department of Computer Science
Stanford University
{johnhew,kawin,pliang,manning}@cs.stanford.edu

Abstract

Probing experiments investigate the extent to which neural representations make properties—like part-of-speech—predictable. One suggests that a representation encodes a property if probing that representation produces higher accuracy than probing a baseline representation like non-contextual word embeddings. Instead of using baselines as a point of comparison, we're interested in measuring information that is contained in the representation but not in the baseline. For example, current methods can detect when a representation is more useful than the word identity (a baseline) for predicting part-of-speech; however, they cannot detect when the representation is predictive of just the aspects of part-of-speech *not* explainable by the word identity. In this work, we extend a theory of *usable* information called $\mathcal{V}$-information and propose *conditional probing*, which explicitly conditions on the information in the baseline. In a case study, we find that after conditioning on non-contextual word embeddings, properties like part-of-speech are accessible at deeper layers of a network than previously thought.

1 Introduction

Neural language models have become the foundation for modern NLP systems (Devlin et al., 2019; Radford et al., 2018), but what they understand about language, and how they represent that knowledge, is still poorly understood (Belinkov and Glass, 2019; Rogers et al., 2020). The *probing* methodology grapples with these questions by relating neural representations to well-understood properties. Probing analyzes a representation by using it as input into a supervised classifier, which is trained to predict a property, such as part-of-speech (Shi et al., 2016; Ettinger et al., 2016; Alain and Bengio, 2016; Adi et al., 2017; Belinkov, 2021).

One suggests that a representation encodes a property of interest if probing that representation produces higher accuracy than probing a baseline representation like non-contextual word embeddings. However, consider a representation that encodes *only* the part-of-speech tags that aren't determined by the word identity. Probing would report that this representation encodes *less about part-of-speech* than the non-contextual word baseline, since ambiguity is relatively rare. Yet, this representation clearly encodes interesting aspects of part-of-speech. How can we capture this?

In this work, we present a simple probing method to explicitly condition on a baseline.[1] For a representation and a baseline, our method trains two probes: (1) on just the baseline, and (2) on the concatenation of the baseline and the representation. The performance of probe (1) is then subtracted from that of probe (2). We call this process *conditional probing*. Intuitively, the representation is not penalized for *lacking* aspects of the property accessible in the baseline.

We then theoretically ground our probing methodology in $\mathcal{V}$-information, a theory of *usable* information introduced by Xu et al. (2020) that we additionally extend to multiple predictive variables. We use $\mathcal{V}$-information instead of mutual information (Shannon, 1948; Pimentel et al., 2020b) because any injective deterministic transformation of the input has the same mutual information as the input. For example, a representation that maps each unique sentence to a unique integer must have the same mutual information with any property as does BERT's representation of that sentence, yet the latter is more useful. In contrast, $\mathcal{V}$-information is defined with respect to a family of functions $\mathcal{V}$ that map one random variable to (a probability distribution over) another. $\mathcal{V}$-information can be constructed by deterministic transformations that make a property more accessible to the functions in the family. We show that conditional probing

[1] Our code is available at https://github.com/john-hewitt/conditional-probing.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1626–1639
November 7–11, 2021. ©2021 Association for Computational Linguistics

provides an estimate of *conditional $\mathcal{V}$-information* $I_{\mathcal{V}}(\text{repr} \rightarrow \text{property} \mid \text{baseline})$.

In a case study, we answer an open question posed by Hewitt and Liang (2019): how are the aspects of linguistic properties that *aren't explainable by the input layer* accessible across the rest of the layers of the network? We find that the part-of-speech information not attributable to the input layer remains accessible much deeper into the layers of ELMo (Peters et al., 2018a) and RoBERTa (Liu et al., 2019) than the overall property, a fact previously obscured by the gradual loss across layers of the aspects attributable to the input layer. For the other properties, conditioning on the input layer does not change the trends across layers.

2 Conditional $\mathcal{V}$-information Probing

In this section, we describe probing methods and introduce conditional probing. We then review $\mathcal{V}$-information and use it to ground probing.

2.1 Probing setup

We start with some notation. Let $X \in \mathcal{X}$ be a random variable taking the value of a sequence of tokens. Let $\phi(X)$ be a representation resulting from a deterministic function of X; for example, the representation of a single token from the sequence in a layer of BERT (Devlin et al., 2019). Let $Y \in \mathcal{Y}$ be a property (e.g., part-of-speech of a particular token), and $\mathcal{V}$ a *probe family*, that is, a set of functions $\{f_\theta : \theta \in \mathbb{R}^p\}$, where $f_\theta : z \rightarrow \mathcal{P}(\mathcal{Y})$ maps inputs z to probability distributions over the space of the label.[2] The input $z \in \mathbb{R}^m$ may be in the space of $\phi(X)$, that is, $\mathbb{R}^d$, or another space, e.g., if the probe takes the concatenation of two representations. In each experiment, a training dataset $\mathcal{D}_{\text{tr}} = \{(x_i, y_i)\}_i$ is used to estimate θ, and the probe and representation are evaluated on a separate dataset $\mathcal{D}_{\text{te}} = \{(x_i, y_i)\}_i$. We refer to the result of this evaluation on some representation R as $\text{Perf}(R)$.

2.2 Baselined probing

Let $B \in \mathbb{R}^d$ be a random variable representing a baseline (e.g., non-contextual word embedding of a particular token.) A common strategy in probing is to take the difference between a probe performance on the representation and on the baseline (Zhang and Bowman, 2018); we call this **baselined probing performance**:

$$\text{Perf}(\phi(X)) - \text{Perf}(B). \quad (1)$$

This difference in performances estimates how much more *accessible* Y is in $\phi(X)$ than in the baseline B, under probe family $\mathcal{V}$.

But what if B and $\phi(X)$ capture distinct aspects of Y? For example, consider if $\phi(X)$ captures parts-of-speech that aren't the most common label for a given word identity, while B captures parts-of-speech that are the most common for the word identity. Baselined probing will indicate that $\phi(X)$ explains less about Y than the baseline, a "negative" probing result. But clearly $\phi(X)$ captures an interesting aspect of Y; we aim to design a method that measures just what $\phi(X)$ *contributes beyond* B in predicting Y, not what B has and $\phi(X)$ lacks.

2.3 Our proposal: conditional probing

In our proposed method, we again train two probes; each is the concatenation of two representations of size d, so we let $z \in \mathbb{R}^{2d}$. The first probe takes as input $[B; \phi(X)]$, that is, the concatenation of B to the representation $\phi(X)$ that we're studying. The second probe takes as input $[B; \mathbf{0}]$, that is, the concatenation of B to the $\mathbf{0}$ vector. Conditional probing method takes the difference of the two probe performances, which we call **conditional probing performance**:

$$\text{Perf}([B; \phi(X)]) - \text{Perf}([B; \mathbf{0}]). \quad (2)$$

Including B in the probe with $\phi(X)$ means that $\phi(X)$ only needs to contribute what is missing from B. In the second probe, the $\mathbf{0}$ is used as a placeholder, representing the lack of knowledge of $\phi(X)$; its performance is subtracted so that $\phi(X)$ isn't given credit for what's explainable by B.[3]

2.4 $\mathcal{V}$-information

$\mathcal{V}$-information is a theory of *usable* information—that is, how much knowledge of random variable Y can be extracted from r.v. R when using functions in $\mathcal{V}$, called a *predictive family* (Xu et al., 2020). Intuitively, by explicitly considering computational constraints, $\mathcal{V}$-information can be *constructed* by computation, in particular when said computation makes a variable easier to predict. If $\mathcal{V}$ is the set of all functions from the space of R to the set of

[2] We discuss mild constraints on the form that $\mathcal{V}$ can take in the Appendix. Common probe families including linear models and feed-forward networks meet the constraints.

[3] The value $\mathbf{0}$ is arbitrary; any constant can be used, or one can train the probe on just B.

probability distributions over the space of Y, then $\mathcal{V}$-information is mutual information (Xu et al., 2020). However, if the predictive family is the set of all functions, then no representation is more useful than another provided they are related by a bijection. By specifying a $\mathcal{V}$, one makes a hypothesis about the functional form of the relationship between the random variables R and Y. One could let $\mathcal{V}$ be, for example, the set of log-linear models.

Using this predictive family $\mathcal{V}$, one can define the uncertainty we have in Y after observing R as the $\mathcal{V}$-entropy:

$$H_{\mathcal{V}}(Y|R) = \inf_{f \in \mathcal{V}} \mathbb{E}_{r,y \sim R,Y}\big[-\log f[r](y)\big], \quad (3)$$

where $f[r]$ produces a probability distribution over the labels. Information terms like $I_{\mathcal{V}}(R \to Y)$ are defined analogous to Shannon information, that is, $I_{\mathcal{V}}(R \to Y) = H_{\mathcal{V}}(Y) - H_{\mathcal{V}}(Y|R)$. For brevity, we leave a full formal description, as well as our redefinition of $\mathcal{V}$-information to multiple predictive variables, to the appendix.

2.5 Probing estimates $\mathcal{V}$-information

With a particular performance metric, baselined probing estimates a difference of $\mathcal{V}$-information quantities. Intuitively, probing specifies a function family $\mathcal{V}$, training data is used to find $f \in V$ that best predicts Y from $\phi(X)$ (the infimum in Equation 7), and we then evaluate how well Y is predicted. If we use the negative cross-entropy loss as the Perf function, then **baselined probing** estimates

$$I_{\mathcal{V}}(\phi(X) \to Y) - I_{\mathcal{V}}(B \to Y),$$

the difference of two $\mathcal{V}$-information quantities. This theory provides methodological best practices as well: the form of the family $\mathcal{V}$ should be chosen for theory-external reasons,[4] and since the probe training process is approximating the infimum in Equation 3, we're not concerned with sample efficiency.

Baselined probing appears in existing information-theoretic probing work: Pimentel et al. (2020b) define conditional mutual information quantities wherein a lossy transformation $c(\cdot)$ is performed on the sentence (like choosing a single word), and an estimate of the gain from knowing the rest of the sentence is provided; $I(\phi(X); Y|c(\phi(X))) = I(X; Y|c(X))$.[5] Methodologically, despite being a conditional information, this is identical to baselined probing, training one probe on just $\phi(X)$ and another on just $c(\phi(X))$.[6]

2.6 Estimating conditional information

Inspired by the transparent connections between $\mathcal{V}$-information and probes, we ask what the $\mathcal{V}$-information analogue of conditioning on a variable in a mutual information, that is, $I(X, Y|B)$. To do this, we extend $\mathcal{V}$-information to multiple predictive variables, and design conditional probing (as presented) to estimate

$$\begin{aligned} &I_{\mathcal{V}}(\phi(X) \to Y|B) \\ &\qquad = H_{\mathcal{V}}(Y|B) - H_{\mathcal{V}}(Y|B, \phi(X)), \end{aligned}$$

thus having the interpretation of probing what $\phi(X)$ explains about Y apart from what's already explained by B (as can be accessed by functions in $\mathcal{V}$). Methodologically, the innovation is in providing B to the probe on $\phi(X)$, so that the information accessible in B need not be accessible in $\phi(X)$.

3 Related Work

Probing—mechanically simple, but philosophically hard to interpret (Belinkov, 2021)—has led to a number of information-theoretic interpretations.

Pimentel et al. (2020b) claimed that probing should be seen as estimating mutual information $I(\phi(X); Y)$ between representations and labels. This raises an issue, which Pimentel et al. (2020b) notes: due to the data processing inequality, the MI between the representation of a sentence (from e.g., BERT) and a label is upper-bounded by the MI between the sentence itself and the label. Both an encrypted document X and an unencrypted version $\phi(X)$ provide the same mutual information with the topic of the document Y. This is because MI allows unbounded work in using X to predict Y, including the enormous amount of work (likely) required to decrypt it without the secret key. Intuitively, we understand that $\phi(X)$ is more useful than X, and that this is because the function ϕ performs useful "work" for us. Likewise, BERT can perform useful work to make interesting properties

[4] There are also PAC bounds (Valiant, 1984) on the estimation error for $\mathcal{V}$-information (Xu et al., 2020); simpler families $\mathcal{V}$ with lower Rademacher complexity result in better bounds.

[5] Equality depends on the injectivity of ϕ; otherwise knowing the representation $\phi(X)$ may be strictly less informative than knowing X.

[6] This is because of the data processing inequality and the fact that $c(\phi(X))$ is a deterministic function of $\phi(X)$.

more accessible. While Pimentel et al. (2020b) conclude from the data processing inequality that probing is not meaningful, we conclude that estimating mutual information is not the goal of probing.

Voita and Titov (2020) propose a new probing-like methodology, *minimum description length (MDL) probing*, to measure the number of bits required to transmit both the specification of the probe and the specification of labels. Intuitively, a representation that allows for more efficient communication of labels (and probes used to help perform that communication) has done useful "work" for us. Voita and Titov (2020) found that by using their methods, probing practitioners could pay less attention to the exact functional form of the probe. $\mathcal{V}$-information and MDL probing complement each other; $\mathcal{V}$-information does not measure sample efficiency of learning a mapping from $\phi(X)$ to Y, instead focusing solely on how well any function from a specific family (like linear models) allows one to predict Y from $\phi(X)$. Further, in practice, one must choose a family to optimize over even in MDL probing; the complexity penalty of communicating the member of the family is analogous to choosing $\mathcal{V}$. Further, our contribution of conditional probing is orthogonal to the choice of probing methodology; it could be used with MDL probing as well.

$\mathcal{V}$-information places the functional form of the probe front-and-center as a *hypothesis* about how structure is encoded. This intuition is already popular in probing, For example, Hewitt and Manning (2019) proposed that syntax trees may emerge as squared Euclidean distance under a linear transformation. Further work refined this, showing that a better structural hypothesis may be hyperbolic (Chen et al., 2021) axis-aligned after scaling (Limisiewicz and Mareček, 2021), or an attention-inspired kernel space (White et al., 2021).

In this work, we intentionally avoid claims as to the "correct" functional family $\mathcal{V}$ to be used in conditional probing. Some work has argued for simple probe families (Hewitt and Liang, 2019; Alain and Bengio, 2016), others for complex families (Pimentel et al., 2020b; Hou and Sachan, 2021). Pimentel et al. (2020a) argues for choosing multiple points along an axis of expressivity, while Cao et al. (2021) define the family through the weights of the neural network. Other work performs structural analysis of representations without direct supervision (Saphra and Lopez, 2019; Wu et al., 2020).

Hewitt and Liang (2019) suggested that differences in ease of identifying the word identity across layers could impede comparisons between the layers; our conditional probing provides a direct solution to this issue by conditioning on the word identity. Kuncoro et al. (2018) and Shapiro et al. (2021) use control tasks, and Rosa et al. (2020) measures word-level memorization in probes. Finally, under the possible goals of probing proposed by Ivanova et al. (2021), we see $\mathcal{V}$-information as most useful in *discovering emergent structure*, that is, parsimonious and surprisingly simple relationships between neural representations and complex properties.

4 Experiments

In our experiments, we aim for a case study in understanding how conditioning on the non-contextual embeddings changes trends in the accessibility of linguistic properties across the layers of deep networks.

4.1 Tasks, models, and data

Tasks. We train probes to predict five linguistic properties, roughly arranged in order from lower-level, more concrete properties to higher-level, more abstract properties. We predict five linguistic properties Y: (i) **upos**: coarse-grained (17-tag) part-of-speech tags (Nivre et al., 2020), (ii) **xpos**: fine-grained English-specific part-of-speech tags, (iii) **dep rel**: the label on the Universal Dependencies edge that governs the word, (iv) **ner**: named entities, and (v) **sst2**: sentiment.

Data. All of our datasets are composed of English text. For all tasks except sentiment, we use the Ontonotes v5 corpus (Weischedel et al., 2013), recreating the splits used in the CoNLL 2012 shared task, as verified against the split statistics provided by Strubell et al. (2017).[7][8] Since Ontonotes is annotated with constituency parses, not Universal Dependencies, we use the converter provided in CoreNLP (Schuster and Manning,

[7] In order to provide word vectors for each token in the corpus, we heuristically align the subword tokenizations of RoBERTa with the corpus-specified tokens through character-level alignments, following Tenney et al. (2019).

[8] Ontonotes uses the destructive Penn Treebank tokenization (like replacing brackets { with -LCB- (Marcus et al., 1993)). We perform a heuristic de-tokenization process before subword tokenization to recover some naturalness of the text.

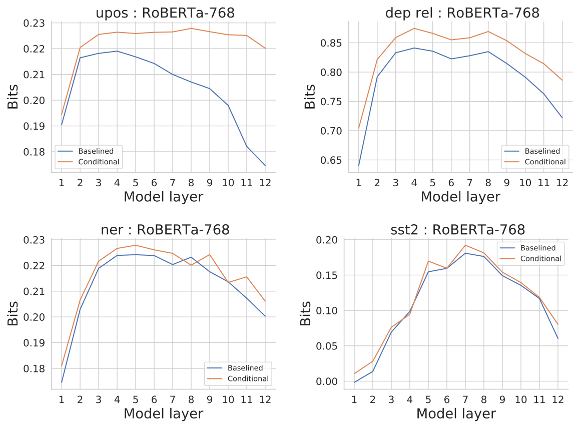

Figure 1: Probing results on RoBERTa. Results are reported in bits of $\mathcal{V}$-information; higher is better.

2016; Manning et al., 2014). For the sentiment annotation, we use the binary GLUE version (Wang et al., 2019) of the the Stanford Sentiment Treebank corpus (Socher et al., 2013). All results are reported on the development sets.

Models. We evaluate the popular RoBERTa model (Liu et al., 2019), as provided by the HuggingFace Transformers package (Wolf et al., 2020), as well as the ELMo model (Peters et al., 2018a), as provided by the AllenNLP package (Gardner et al., 2017). When multiple RoBERTa subwords are aligned to a single corpus token, we average the subword vector representations.

Probe families. For all of our experiments, we choose $\mathcal{V}$ to be the set of affine functions followed by softmax.[9] For word-level tasks, we have

$$f_\theta(\phi_i(X)_j) = \text{softmax}(W\phi_i(X)_j + b) \qquad (4)$$

where i indexes the layer in the network and j indexes the word in the sentence. For the sentence-level sentiment task, we average over the word-level representations, as

$$f_\theta(\phi_i(X)) = \text{softmax}(W \text{ avg}(\phi_i(X)) + b) \quad (5)$$

4.2 Results

Results on ELMo. ELMo has a non-contextual embedding layer ϕ_0, and two contextual layers ϕ_1 and ϕ_2, the output of each of two bidirectional LSTMs (Hochreiter and Schmidhuber, 1997). Previous work has found that ϕ_1 contains more syntactic information than ϕ_2 (Peters et al., 2018b; Zhang and Bowman, 2018). Baselined probing performance, in Table 1, replicates this finding. But Hewitt and Liang (2019) conjecture that this may be due to accessibility of information from ϕ_0. Conditional probing answers shows that when only measuring information not available in ϕ_0, there is still more syntactic information in ϕ_1 than ϕ_2, but the difference is much smaller.

	Baselined		Conditional	
	ϕ_1	ϕ_2	ϕ_1	ϕ_2
upos	0.20	0.16	0.22	0.20
xpos	0.20	0.16	0.21	0.20
dep rel	0.99	0.81	1.00	0.87
ner	0.24	0.23	0.25	0.24
sst2	0.18	0.13	0.17	0.13

Table 1: Results on ELMo, reported in bits of $\mathcal{V}$-information; higher is better. ϕ_i refers to layer i.

Results on RoBERTa. RoBERTa-base is a pretrained Transformer consisting of a word-level embedding layer ϕ_0 and twelve contextual layers ϕ_i, each the output of a Transformer encoder block (Vaswani et al., 2017). We compare baselined probing performance to conditional probing performance for each layer. In Figure 1, baselined probing indicates that part-of-speech information decays in later layers. However, conditional probing shows that information *not* available in ϕ_0 is maintained into deeper layers in RoBERTa, and only the information already available in ϕ_0 decays. In contrast for dependency labels, we find that the difference between layers is lessened after conditioning on ϕ_0, and for NER and sentiment, conditioning on ϕ_0 does not change the results.

5 Conclusion

In this work, we proposed *conditional probing*, a simple method for conditioning on baselines in probing studies, and grounded the method theoretically in $\mathcal{V}$-information. In a case study, we found that after conditioning on the input layer, usable part-of-speech information remains much deeper into the layers of ELMo and RoBERTa than previously thought, answering an open question from Hewitt and Liang (2019). Conditional probing is a tool that practitioners can easily use to gain additional insight into representations.[10]

[9] We used the Adam optimizer (Kingma and Ba, 2014) with starting learning rate 0.001, and multiply the learning rate by 0.5 after each epoch wherein a new lowest validation loss is not achieved.

[10] An executable version of the experiments in this paper is on CodaLab, at this link: https://worksheets.codalab.org/worksheets/0x46190ef741004a43a2676a3b46ea0c76.

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*.

Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations*.

Yonatan Belinkov. 2021. Probing classifiers: Promises, shortcomings, and alternatives. *CoRR*, abs/2102.12452.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

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.

Boli Chen, Yao Fu, Guangwei Xu, Pengjun Xie, Chuanqi Tan, Mosha Chen, and Liping Jing. 2021. Probing {bert} in hyperbolic spaces. In *International Conference on Learning Representations*.

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

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.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*.

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)*, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yifan Hou and Mrinmaya Sachan. 2021. Bird's eye: Probing for linguistic graph structures with a simple information-theoretic approach. 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 1844–1859, Online. Association for Computational Linguistics.

Anna A. Ivanova, John Hewitt, and Noga Zaslavsky. 2021. Probing artificial neural networks: insights from neuroscience. In *Proceedings of the Brain2AI Workshop at the Ninth International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Adhiguna Kuncoro, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom. 2018. LSTMs can learn syntax-sensitive dependencies well, but modeling structure makes them better. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1426–1436.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

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

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018a. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 2227–2237.

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

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 Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *Open AI Technical Report*.

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.

Rudolf Rosa, Tomáš Musil, and David Mareček. 2020. Measuring memorization effect in word-level neural networks probing. In *Text, Speech, and Dialogue*, pages 180–188, Cham. Springer International Publishing.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with svcca. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*. Association for Computational Linguistics.

Sebastian Schuster and Christopher D. Manning. 2016. Enhanced English Universal Dependencies: An improved representation for natural language understanding tasks. In *Language Resources and Evaluation (LREC)*.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Naomi Tachikawa Shapiro, Amandalynne Paullada, and Shane Steinert-Threlkeld. 2021. A multilabel approach to morphosyntactic probing. *arXiv preprint arXiv:2104.08464*.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*.

Leslie G Valiant. 1984. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142.

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*, pages 5998–6008.

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.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. *OntoNotes release 5*. LDC2013T19.

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.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, 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*.

Kelly W Zhang and Samuel R Bowman. 2018. Language modeling teaches you more syntax than translation does: Lessons learned through auxiliary task analysis. *arXiv preprint arXiv:1809.10040*.

A Multivariable $\mathcal{V}$-information

In this section we introduce Multivariable $\mathcal{V}$-information. $\mathcal{V}$-information as introduced by Xu et al. (2020) was defined in terms of a single predictive variable X, and is unwieldy to extend to multiple variables due to its use of a "null" input outside the sample space of X (Section D.3).[11] Our multivariable V-information removes the use of null variables and naturally captures the multivariable case. Consider an agent attempting to predict $Y \in \mathcal{Y}$ from some information sources $X_1, \ldots, X_n$, where $X_i \in \mathcal{X}_i$. Let $\mathcal{P}(\mathcal{Y})$ be the set of all probability distributions over Y.

At a given time, the agent may only have access to a subset of the information sources. Let the known set $C \in \mathcal{C}$ and unknown set $\bar{C} \in \bar{\mathcal{C}}$ be a binary partition of $X_1, \ldots, X_n$. Though the agent isn't given the true value of $\bar{C}$ when predicting Y, it is instead provided with a constant value $\bar{a} \in \bar{\mathcal{C}}$, which does not vary with Y.[12]

We first specify constraints on the set of functions that the agent has at its disposal for predicting Y from X:

Definition 1 (Multivariable Predictive Family)**.** *Let $\Omega = \{f : \mathcal{X}_1 \times \cdots \times \mathcal{X}_n \to \mathcal{P}(\mathcal{Y})\}$. We say that $\mathcal{V} \subseteq \Omega$ is a predictive family if, for any partition of $\mathcal{X}_1, \ldots, \mathcal{X}_n$ into $\mathcal{C}, \bar{\mathcal{C}}$, we have*

$$\begin{aligned} &\forall f, x_1, \ldots, x_n, \in \mathcal{V} \times \mathcal{X}_1 \times \cdots \times \mathcal{X}_n, \\ &\exists f' \in \mathcal{V} : \ \forall \bar{c}' \in \bar{\mathcal{C}},\ f(c, \bar{c}) = f'(c, \bar{c}'), \end{aligned} \tag{6}$$

where we overload $f(c, \bar{c})$ to equal $f(x_1, \ldots, x_n)$ for the values of $x_1, \ldots, x_n$ specified by $c, \bar{c}$.

Intuitively, the constraint on $\mathcal{V}$ states that for any binary partition of the $X_1, \ldots, X_n$ into known and unknown sets, if a function is expressible given some constant assignment to the unknown variables, the same function is expressible if the unknown variables are allowed to vary arbitrarily. Intuitively, this means one can assign zero weight to those variables, so their values don't matter. This constraint, which we refer to as *multivariable optional ignorance* in reference to Xu et al. (2020), will be used to ensure non-negativity of information; when some X_ℓ is moved from $\bar{C}$ to C as a new predictive variable for the agent to use, optional ignorance ensures the agent can still act as if that variable were held constant.

Example 1. *Let $X_1, \ldots, X_n \in \mathbb{R}^{d_1}, \ldots, \mathbb{R}^{d_n}$ and $Y \in \mathcal{Y}$ be random variables. Let Ω be defined as in Definition 1. Then $V = \{f : f(x_1, \ldots, x_n) = \mathrm{softmax}(W_2\, \sigma(W_1[x_1; \cdots ; x_n] + b) + b)\}$, the set of 1-layer multi-layer perceptrons, is a predictive family. Ignorance of some x_i can be achieved by setting the corresponding rows of W_1 to zero.*

[11] In particular, the null input encodes *not* knowing the value of X; technical conditions in the definition of $\mathcal{V}$-information as to the behavior of this null value increase in number exponentially with the number of predictive variables.

[12] The exact value of $\bar{a}$ will not matter, as a result of Definition 1.

Given the predictive family of functions the agent has access to, we define the multivariable $\mathcal{V}$-information analogue of entropy:

Definition 2 (Multivariable Predictive $\mathcal{V}$-entropy)**.** *Let $X_1, \ldots, X_n \in \mathcal{X}_1, \ldots, \mathcal{X}_n$. Let $C \in \mathcal{C}$ and $\bar{C} \in \bar{\mathcal{C}}$ form a binary partition of $X_1, \ldots, X_n$. Let $\bar{a} \in \bar{\mathcal{C}}$. Then the $\mathcal{V}$-entropy of Y conditioned on C is defined as*

$$H_{\mathcal{V}}(Y|C) = \inf_{f \in V} \mathbb{E}_{c,y}\big[-\log f(c, \bar{a})[y]\big]. \quad (7)$$

Note that $\bar{a}$ does not vary with y; thus it is 'informationless'. The notation $f(c, \bar{a})$ takes the known value of $C \subseteq \{X_1, \ldots, X_n\}$, and the constant value $\bar{a}$, and produces a distribution over $\mathcal{Y}$, and $f(c, \bar{a})[y]$ evaluates the density at y.

If we let $V = \Omega$, the set of all functions from the $\mathcal{X}_i$ to distributions over $\mathcal{Y}$, then $\mathcal{V}$-entropy becomes exactly Shannon entropy (Xu et al., 2020). And just like for Shannon information, the multivariable $\mathcal{V}$-information from some variable X_ℓ to Y is defined as the reduction in entropy when its value becomes known. In our notation, this means some X_ℓ is moving from $\bar{C}$ (the unknown variables) to C (the known variables), so this definition encompasses the notion of *conditional* mutual information if C is non-empty to start.

Definition 3 (Multivariable $\mathcal{V}$-information)**.** *Let $X_1, \ldots, X_n \in \mathcal{X}_1, \ldots, \mathcal{X}_n$ and $Y \in \mathcal{Y}$ be random variables. Let $\mathcal{V}$ be a multivariable predictive family. Then the conditional multivariable $\mathcal{V}$-information from X_ℓ to Y, where $\ell \in \{1, \ldots, n\}$, conditioned on prior knowledge of $C \subset \{X_1, \ldots, X_n\}$, is defined as*

$$I_{\mathcal{V}}(X_\ell \rightarrow Y|C) = H_{\mathcal{V}}(Y|C) - H_{\mathcal{V}}(Y|C \cup \{X_\ell\}) \quad (8)$$

A.1 Properties of multivariable $\mathcal{V}$-information

The crucial property of multivariable $\mathcal{V}$-information as a descriptor of probing is that it can be *constructed* through computation. In the example of the agent attempting to predict the sentiment (Y) of an encrypted message (X), if the agent has $\mathcal{V}$ equal to the set of linear functions, then $I_{\mathcal{V}}(X \rightarrow Y)$ is small[13]. A function ϕ that decrypts the message constructs $\mathcal{V}$-information about Y, since $I_{\mathcal{V}}(\phi(X) \rightarrow Y)$ is larger. In probing, ϕ is interpreted to be the contextual representation learner, which is interpreted as *constructing* $\mathcal{V}$-information about linguistic properties.

$\mathcal{V}$-information also has some desirable elementary properties, including preserving some of the properties of mutual information, like non-negativity. (Knowing some X_ℓ should not reduce the agent's ability to predict Y).

Proposition 1. *Let $X_1, \ldots, X_n \in \mathcal{X}_1, \ldots, \mathcal{X}_n$ and $Y \in \mathcal{Y}$ be random variables, and $\mathcal{V}$ and $\mathcal{U}$ be predictive families. Let $C, \bar{C}$ be a binary partition of $X_1, \ldots, X_n$.*

1. ***Independence** If Y, C are jointly independent of X_ℓ, then $I_{\mathcal{V}}(X_\ell \rightarrow Y|C) = 0$.*
2. ***Monotonicity** If $\mathcal{U} \subseteq \mathcal{V}$, then $H_{\mathcal{V}}(Y|C) \leq H_{\mathcal{U}}(Y|C)$.*
3. ***Non-negativity** $I_{\mathcal{V}}(X_\ell \rightarrow Y|C) \geq 0$.*

B Probing as Multivariable $\mathcal{V}$-information Estimation

We've described the $\mathcal{V}$-information framework, and discussed how it captures the intuition that usable information about linguistic properties is constructed through contextualization. In this section, we demonstrate how a small step from existing probing methodology leads to probing estimating $\mathcal{V}$-information quantities.

B.1 Estimating $\mathcal{V}$-entropy

In probing, gradient descent is used to pick the function in $\mathcal{V}$ that minimizes the cross-entropy loss,

$$\frac{1}{\mathcal{D}_{\text{tr}}} \sum_{x,y \in \mathcal{D}_{\text{tr}}} -\log p(y|\phi_i(x); \theta), \quad (9)$$

where θ are the trainable parameters of functions in $\mathcal{V}$. Recalling the definition of $\mathcal{V}$-entropy, this minimization performed through gradient descent is approximating the inf over $\mathcal{V}$, since $-\log p(y|x;\theta)$ is equal to $-\log f_\theta(x)[y]$. To summarize, this states that the supervision used in probe training can be interpreted as approximating the inf in the definition of $\mathcal{V}$-entropy. In traditional probing, the performance of the probe is measured on the test set $\mathcal{D}_{\text{te}}$ using the traditional metric of the task, like accuracy of F_1 score. In $\mathcal{V}$-information probing, we use $\mathcal{D}_{\text{te}}$ to approximate the expectation in the definition of $\mathcal{V}$-entropy. Thus, the performance of a single probe on representation R, where the performance metric is cross-entropy loss, is an estimate

[13] Where $I_{\mathcal{V}}(X \rightarrow Y)$ is defined to be $I_{\mathcal{V}}(X \rightarrow Y|\{\})$

of $H_{\mathcal{V}}(Y|R)$. This brings us to our framing of a probing experiment as estimating a $\mathcal{V}$-information quantity.

B.2 Baselined probing

Let baselined probing be defined as in the main paper. Then if the performance metric is defined as the negative cross-entropy loss, we have that $\mathrm{Perf}(B)$ estimates $-H_{\mathcal{V}}(Y|B)$, $\mathrm{Perf}(\phi(X))$ estimates $-H_{\mathcal{V}}(Y|\phi(X))$, and so baselined probing performance is an estimate of

$$\begin{aligned} &H_{\mathcal{V}}(Y|\{B\}) - H_{\mathcal{V}}(Y|\{\phi_i(X)\}) \\ &= I_{\mathcal{V}}(\phi_i(X) \to Y) - I_{\mathcal{V}}(B \to Y) \end{aligned} \tag{10}$$

B.3 Conditional probing

Let conditional probing be defined as in the main paper. Then if the performance metric is defined as the negative cross-entropy loss, we have that $\mathrm{Perf}([B;\mathbf{0}])$ estimates $-H_{\mathcal{V}}(Y|B)$, $\mathrm{Perf}([B;\phi(X)])$ estimates $-H_{\mathcal{V}}(Y|B,\phi(X))$, and so conditional probing performance is an estimate of

$$\begin{aligned} &H_{\mathcal{V}}(Y|\{B\}) - H_{\mathcal{V}}(Y|\{B,\phi_i(X)\}) \\ &= I_{\mathcal{V}}(\phi_i(X) \to Y|B) \end{aligned} \tag{11}$$

The first is estimated with a probe just on B—under the definition of predictive family, this means providing the agent with the real values of the baseline, and some constant value like the zero vector instead of $\phi_i(X)$. That is, holding $\bar{a} \in \phi_i(\mathcal{X})_i$ constant and sampling $b, y \sim B, Y$, the probability assigned to y is $f(b,\bar{a})[y]$ for $f \in \mathcal{V}$. The second term is estimate with a probe on both B and $\phi_i(X)$. So, sampling $b, x, y \sim B, X, Y$, the probability assigned to y is $f(b,\phi_i(x))[y]$ for $f \in \mathcal{V}$. Intuitively, conditional probing measures the *new* information in $\phi_i(X)$ because in both probes, the agent has access to B, so no benefit is gained from $\phi_i(X)$ supplying the same information.

C Proof of Proposition 1

Monotonicity *If $\mathcal{U} \subseteq \mathcal{V}$, then $H_{\mathcal{V}}(Y|C) \leq H_{\mathcal{U}}(Y|C)$.* Proof:

$$\begin{aligned} H_{\mathcal{U}}(Y|C) &= \inf_{f\in\mathcal{U}} \mathbb{E}_{c,y}\left[-\log f[c,\bar{a}](y)\right] \\ &\geq \inf_{f\in\mathcal{V}} \mathbb{E}_{c,y}\left[-\log f[c,\bar{a}](y)\right] \\ &= H_{\mathcal{V}}(Y|C) \end{aligned} \tag{12}$$

This holds because we are taking the infimum over $\mathcal{V}$ such that if $f \in \mathcal{U}$ then $f \in \mathcal{V}$.

Non-Negativity $I_{\mathcal{V}}(X_\ell \to Y|C) \geq 0$. *Where $\mathcal{V}_{\bar{C}} \subset \mathcal{V}$ is the subset of functions that satisfies $f[c,\bar{c}] = f[c,\bar{c}'] \;\forall\; \bar{c}' \in \bar{C}$, and $\bar{a}, \bar{a}_{/\ell}$ denote the constant values of the unknown set with and without X_ℓ, the proof is as follows:*

$$\begin{aligned} H_{\mathcal{V}}(Y|C) &= \inf_{f\in\mathcal{V}} \mathbb{E}_{c,x_\ell,y}\left[-\log f[c,\bar{a}](y)\right] \\ &= \inf_{f\in\mathcal{V}_{\bar{C}}} \mathbb{E}_{c,x_\ell,y}\left[-\log f[c,x_\ell,\bar{a}_{/\ell}](y)\right] \\ &\geq \inf_{f\in\mathcal{V}} \mathbb{E}_{c,x_\ell,y}\left[-\log f[c,x_\ell,\bar{a}_{/\ell}](y)\right] \\ &= H_{\mathcal{V}}(Y|C \cup \{X_\ell\}) \end{aligned} \tag{13}$$

By definition, $I_{\mathcal{V}}(X_\ell \to Y|C) = H_{\mathcal{V}}(Y|C) - H_{\mathcal{V}}(Y|C \cup \{X_\ell\}) \geq 0$.

Independence *If Y, C are jointly independent of X_ℓ, then $I_{\mathcal{V}}(X \to Y|C) = 0$. Proof:*

$$\begin{aligned} &H_{\mathcal{V}}(Y|C \cup \{X_\ell\}) \\ &\quad= \inf_{f\in\mathcal{V}} \mathbb{E}_{c,x_\ell,y}\left[-\log f[c,x_\ell,\bar{a}_{/\ell}](y)\right] \\ &\quad= \inf_{f\in\mathcal{V}} \mathbb{E}_{x_\ell}\mathbb{E}_{c,y}\left[-\log f[c,x_\ell,\bar{a}_{/\ell}](y)\right] \\ &\quad\geq \mathbb{E}_{x_\ell}\left[\inf_{f\in\mathcal{V}} \mathbb{E}_{c,y}\left[-\log f[c,x_\ell,\bar{a}_{/\ell}](y)\right]\right] \\ &\quad= \mathbb{E}_{x_\ell}\left[\inf_{f\in\mathcal{V}_{\bar{C}}} \mathbb{E}_{c,y}\left[-\log f[c,x_\ell,\bar{a}_{/\ell}](y)\right]\right] \\ &\quad= \inf_{f\in\mathcal{V}_{\bar{C}}} \mathbb{E}_{c,y}\left[-\log f[c,\bar{a}](y)\right] \\ &\quad\geq \inf_{f\in\mathcal{V}} \mathbb{E}_{c,y}\left[-\log f[c,\bar{a}](y)\right] \\ &\quad= H_{\mathcal{V}}(Y|C) \end{aligned} \tag{14}$$

In the second line, we break down the expectation based on conditional independence. Then we apply Jensen's inequality and optional ignorance to remove the expectation w.r.t. x. Since $\mathcal{V}_{\bar{C}} \subset \mathcal{V}$, the former's infimum is at least as large as the latter's. Then

$$I_{\mathcal{V}}(X_\ell \to Y|C) = H_{\mathcal{V}}(C) - H_{\mathcal{V}}(Y|C \cup \{X_\ell\}) \leq 0$$

Combined with non-negativity (i.e., $I_{\mathcal{V}}(X_\ell \to Y|C) > 0$) we have inequality in both directions, so $I_{\mathcal{V}}(X_\ell \to Y|C) = 0$.

D Equivalence of Xu et al. (2020) and our $\mathcal{V}$-information

In order to define conditional probing, we needed a theory of $\mathcal{V}$-information that considered arbitrarily many predictive variables $\mathcal{X}_1, \ldots, \mathcal{X}_n$. $\mathcal{V}$-information as presented by Xu et al. (2020) considers only a single predictive variable $\mathcal{X}$. It becomes extremely cumbersome, due to the use of null variables in their presentation, to expand this to more, let alone arbitrarily many variables. So, we redefined and extended $\mathcal{V}$-information to more naturally capture the case with an arbitrary (finite) number of variables. In this section, we show that, in the single predictive variable case considered by Xu et al. (2020), our $\mathcal{V}$-information definition is equivalent to theirs. For the sake of this section, we'll call the $\mathcal{V}$-information of Xu et al. (2020) Xu-$\mathcal{V}$-information, and ours $\mathcal{V}$-information.

In particular, we show that there is a transformation from any predictive family of Xu-$\mathcal{V}$-information to predictive family for $\mathcal{V}$-information under which predictive $\mathcal{V}$-entropies are the same (and the same in the opposite direction.)

D.1 From Xu et al. (2020) to ours

We recreate the definition of predictive family from Xu et al. (2020) here:

Definition 4 (Xu predictive family). *Let $\Upsilon = \{f : \mathcal{X} \cup \{\varnothing\} \to \mathcal{P}(\mathcal{Y})\}$. We say that $\mathcal{U} \subseteq \Upsilon$ is a Xu predictive family if it satisfies*

$$\forall f \in \mathcal{U}, \forall P \in \textit{range}(f), \exists f' \in \mathcal{U}, s.t. \quad (15)$$

$$\forall x \in \mathcal{X}, f[x] = P, f'[\varnothing] = P \quad (16)$$

Now, we construct one of our predictive families from the Xu predictive family. Let $\mathcal{U} \subseteq \Upsilon$ be a Xu predictive family. We now construct a predictive family under our framework, $\mathcal{V} \subseteq \Omega$. For each $f \in \mathcal{U}$, $f : \mathcal{X} \cup \{\varnothing\} \to \mathcal{P}(\mathcal{Y})$, construct the following two functions: first, g, which recreates the behavior of f on the domain of $\mathcal{X}$:

$$g : \mathcal{X} \to \mathcal{P}(\mathcal{Y}) \quad (17)$$

$$g : x \mapsto f(x) \quad (18)$$

and second, g', which recreates the behavior of f on $\varnothing$, given any input from $\mathcal{X}$:

$$g' : \mathcal{X} \to \mathcal{P}(\mathcal{Y}) \quad (19)$$

$$g' : x \mapsto f(\varnothing) \quad (20)$$

Then we define our predictive family as the union of g, g' for all $f \in \mathcal{V}$:

$$\mathcal{V} = \bigcup_{f \in \mathcal{U}} \{g, g'\}, \quad (21)$$

where $\mathcal{V} \subseteq \Omega$ and $\Omega = \{f : \mathcal{X} \to \mathcal{P}(\mathcal{Y})\}$. Note from this construction that we've eliminated the presence of the null variable from the definition of predictive family.

We now show that $\mathcal{V}$, as defined in the construction above, is in fact a predictive family under our definition. Under our definition, there are two cases: either $\mathcal{X} \in C$ or $\mathcal{X} \in \bar{C}$. If $\mathcal{X} \in C$, then for all $f, x \in \mathcal{V} \times \mathcal{X}$, if we take any $f' \in \mathcal{V}$ (which is non-empty), then there is no $\bar{c}' \in \bar{C}$, so vacuously the condition holds. If $\mathcal{X} \in \bar{C}$, then for all $f, x \in \mathcal{V}$, we have that f was either g or g' for some function $h \in \mathcal{U}$ in the construction of $\mathcal{V}$ (because all functions in $\mathcal{V}$ were part of some pair g, g'.) Then we take $f' = g'$, and have that for all $\bar{c}' \in \bar{C}$, that is $x \in \mathcal{X}$, $f'(c, \bar{c}) = f'(c, \bar{c}') = g'(x) = h(\varnothing)$, satisfying the constraint.

Finally, we show that the predictive $\mathcal{V}$-entropies of $\mathcal{V}$ (under our definition) and $\mathcal{U}$ (under that of Xu et al. (2020)) are the same. Consider Xu-predictive entropies:

$$H_{\mathcal{U}}(Y|X) = \inf_{f \in \mathcal{U}} \mathbb{E}_{x,y}[-\log f[x](y)] \quad (22)$$

$$H_{\mathcal{U}}(Y|\varnothing) = \inf_{f \in \mathcal{U}} \mathbb{E}_{y}[-\log f[\varnothing](y)] \quad (23)$$

First we want to show $H_{\mathcal{U}}(Y|X) = H_{\mathcal{V}}(Y|X)$. Consider the inf in Equation 22; the $f \in \mathcal{U}$ that achieves the inf corresponds to some $g \in \mathcal{V}$ by construction, and since $f(x) = g(x)$, we have that the value of the inf for $\mathcal{V}$ is at least as low as for $\mathcal{U}$. The same is true in the other direction; in our definition $H_{\mathcal{V}}(Y|\mathcal{X})$, the g that achieves the inf corresponds to some $f \in \mathcal{U}$ that produces the same probability distributions. So, $H_{\mathcal{U}}(Y|X) = H_{\mathcal{V}}(Y|X)$.

Now we want to show $H_{\mathcal{U}}(Y|\varnothing) = H_{\mathcal{V}}(Y)$. Now, consider the inf in Equation 23. The $f \in \mathcal{U}$ that achieves the inf corresponds to some g, g' in the construction of $\mathcal{V}$; that g' takes any $x \in \mathcal{X}$ and produces $f[\varnothing]$; hence the value of the inf for $\mathcal{V}$ is at least as low as for $\mathcal{U}$. The same is true in the other direction. We have that $H_{\mathcal{V}}(Y) = \inf_{f \in \mathcal{V}} \mathbb{E}_y[-\log f(\bar{a})[y]]$ for any $\bar{a} \in \mathcal{X}$. Either a g or a g' from the construction of $\mathcal{V}$ achieves this inf; if a g achieves it, then its corresponding g' emits the same probability distributions, so WLOG we'll assume it's a g'. We know that $g'(\bar{a}) = f(\varnothing)$ for all $\bar{a} \in \mathcal{X}$, so $H_{\mathcal{U}}(Y|\varnothing)$ is at most $H_{\mathcal{V}}(Y)$. So, $H_{\mathcal{U}}(Y|\varnothing) = H_{\mathcal{V}}(Y)$.

Since the V-entropies of the predictive family from Xu et al. (2020) and ours are the same, all the information quantities are the same. This shows that the predictive family we constructed in our theory is equivalent to the predictive family from Xu et al. (2020) that we started with.

D.2 From our $\mathcal{V}$-information to that of Xu et al. (2020)

Now we construct a predictive family $\mathcal{U}$ under the framework of Xu et al. (2020) from an arbitrary predictive family $\mathcal{V}$ under our framework. For each function $f \in \mathcal{V}$, we have from the definition that there exists $f' \in \mathcal{V}$ such that $\forall x \in \mathcal{X}$, $f'(x) = P$ for some $P \in \mathcal{P}(\mathcal{Y})$. We then define the function:

$$g : \mathcal{X} \cup \{\varnothing\} \to \mathcal{P}(\mathcal{Y}) \tag{24}$$

$$g(x) = \begin{cases} f(x) & x \in \mathcal{X} \\ f'(\bar{a}) & x = \varnothing \end{cases} \tag{25}$$

where $\bar{a} \in \mathcal{X}$ is an arbitrary element of $\mathcal{X}$, and the set of constant-valued functions

$$G = \{g' : g'(z) = P \mid P \in \text{range}(f)\}, \tag{26}$$

where $z \in \mathcal{X} \cup \{\varnothing\}$, and let

$$\mathcal{U} = \bigcup_{f \in \mathcal{V}} \{g\} \cup G \tag{27}$$

The set $\mathcal{U}$ is a predictive family under Xu-$\mathcal{V}$-information because for any $f \in \mathcal{U}$, f is either a g or a g' in our construction, and so optional ignorance is maintained by the set G that was either constructed for g or that g' was a part of. That is, from the construction, G contains a function for each element in the range of g (or g') that maps all $x \in \mathcal{X}$ as well as $\varnothing$ to that element, and $\mathcal{U}$ contains all elements in G.

Now we show that the predictive $\mathcal{V}$-entropies of $\mathcal{U}$ (from this construction) under Xu et al. (2020) are the same as for $\mathcal{V}$ under our framework.

First we want to show $H_{\mathcal{U}}(Y|X) = H_{\mathcal{V}}(Y|X)$. For the g that achieves the inf over $\mathcal{U}$ in Equation 22, we have there exists $f \in \mathcal{V}$ such that $g(x) = f(x)$ given that $x \in \mathcal{X}$, so $H_{\mathcal{V}}(y|x) \leq H_{\mathcal{U}}(y|x)$ The same is true in the other direction; the $f \in \mathcal{V}$ that achieves the inf in $\mathcal{V}$-entropy similarly corresponds to $g \in \mathcal{U}$, implying $H_{\mathcal{U}}(Y|X) \leq H_{\mathcal{V}}(Y|X)$, and thus their equality.

Now we want to show $H_{\mathcal{U}}(Y|\varnothing) = H_{\mathcal{V}}(Y)$. For the $g \in \mathcal{U}$ that achieves its inf, we have by construction that there is an $f' \in \mathcal{V}$ such that for any $\bar{a} \in \mathcal{X}$, it holds that $g(\varnothing) = f'(\bar{a})$. So, $H_{\mathcal{V}}(Y|X) \leq H_{\mathcal{U}}(Y|X)$. In the other direction, for the $f \in \mathcal{V}$ that achieves its inf given an arbitrary $\bar{a} \in \mathcal{X}$, there is the $f' \in \mathcal{V}$ from our construction of $\mathcal{U}$ such that $f(\bar{a}) = f'(x) = g(\varnothing)$ for all $x \in \mathcal{X}$. This implies $H_{\mathcal{U}}(Y|X) \leq H_{\mathcal{V}}(Y|X)$, and thus their equality.

D.3 Remarks on the relationship between our $\mathcal{V}$-information and that of Xu et al. (2020)

The difference between our $\mathcal{V}$-information and that of Xu et al. (2020) is in how the requirement of *optional ignorance* is encoded into the formalism. This is an important yet technical requirement that if a predictive agent has access to the value of a random variable X, it's allowed to *disregard* that value if doing so would lead to a lower entropy. An example of a subset of Ω for which this *doesn't* hold in the multivariable case is for multi-layer perceptrons with a frozen (and say, randomly sampled) first linear transformation. The information of, say, X_1 and X_2, are mixed by this frozen linear transformation, and so X_1 cannot be ignored in favor of just looking at X_2. However, if the first linear transformation is trainable, then it can simply assign 0 weights to the rows corresponding to X_1 and thus ignore it.

The $\mathcal{V}$-information of Xu et al. (2020) ensures this option by introducing a null variable $\varnothing$ which is used to represent the lack of knowledge about their variable X – and for any probability distribution in the range of some $f \in \mathcal{U}$ under the theory, there must be some function f that produces the same probability distribution when given any value of X or $\varnothing$. This is somewhat unsatisfying because f should really be a function from $\mathcal{X} \to \mathcal{P}(\mathcal{Y})$, but this implementation of optional ignorance changes the domain to $\mathcal{X} \cup \{\varnothing\}$. When attempting to extend this to the multivariable case, the definition of optional ignorance becomes very cumbersome. With two variables, the domain of functions in a predictive family must be $(\mathcal{X}_1 \cup \{\varnothing\}) \times (\mathcal{X}_2 \cup \{\varnothing\})$. Because the definition of $\mathcal{V}$-entropy under Xu et al. (2020) treats using $\mathcal{X}$ separately from using $\varnothing$, one must define optional ignorance constraints separately for each subset of variables to be ignored, the number of which grows exponentially with the number of variables.

Our re-definition of $\mathcal{V}$-information gets around this issue by defining the optional ignorance constraint in a novel way, eschewing the $\varnothing$ and instead encoding it as the intuitive implementation that we

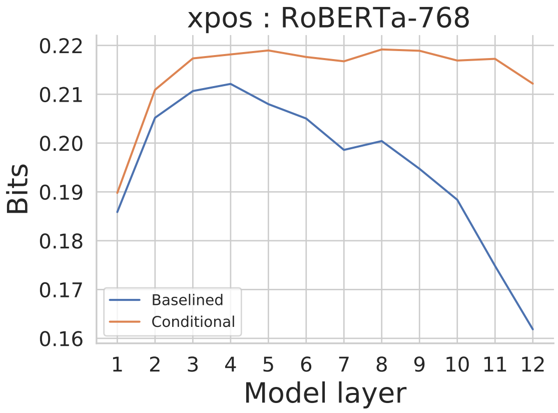

Figure 2: Probing results on RoBERTa for xpos. Results are reported in bits of $\mathcal{V}$-information; higher is better

described in the MLP – that for any function in the family and *fixed* value for some subset of the inputs (which will be the unknown subset), there's a function that behaves identically even if that subset of values is allowed to take *any* value. (Intuitively, by, e.g., having it be possible that the weights for those inputs are 0 at the first layer.)

E Full Results

In this section, we report all individual probing experiments: single-layer probes' $\mathcal{V}$-entropies in Table 2, single-layer probes' task-specific metrics in Table 3, two-layer probes' $\mathcal{V}$-entropies in Table 4, and two-layer probes' task-specific metrics in Table 5. In Figure 2, we report the xpos figure for RoBERTa corresponding to the other four figures in the main paper. We see that it shows roughly the same trend as the upos figure from the main paper.

Acknowledgements

The authors would like to thank the anonymous reviewers and area chair for their helpful feedback which led to a stronger final draft, as well as Nelson Liu, Steven Cao, Abhilasha Ravichander, Rishi Bommasani, Ben Newman, and Alex Tamkin, and much of the Stanford NLP Group for helpful discussions and comments on earlier drafts. JH was supported by an NSF Graduate Research Fellowship under grant number DGE-1656518, and by Two Sigma under their 2020 PhD Fellowship Program. KE was supported by an NSERC PGS-D scholarship.

RoBERTa Single-Layer $\mathcal{V}$-Entropy

Layer	upos	xpos	dep	ner	sst2
0	0.336	0.344	1.468	0.391	0.643
1	0.145	0.158	0.827	0.216	0.645
2	0.119	0.139	0.676	0.188	0.630
3	0.118	0.133	0.635	0.172	0.574
4	0.117	0.132	0.627	0.167	0.545
5	0.119	0.136	0.632	0.167	0.489
6	0.121	0.139	0.645	0.167	0.484
7	0.126	0.145	0.640	0.170	0.462
8	0.129	0.144	0.633	0.168	0.467
9	0.131	0.149	0.653	0.173	0.494
10	0.138	0.156	0.677	0.177	0.508
11	0.154	0.169	0.705	0.184	0.527
12	0.161	0.182	0.746	0.191	0.583

Table 2: $\mathcal{V}$-entropy results (in bits) on probes taking in one layer, for each layer of the network. Lower is better.

RoBERTa Single-Layer Metrics

Layer	upos	xpos	dep	ner	sst2
0	0.908	0.908	0.669	0.535	0.808
1	0.968	0.964	0.821	0.710	0.815
2	0.975	0.969	0.854	0.735	0.813
3	0.975	0.970	0.865	0.763	0.845
4	0.976	0.971	0.867	0.763	0.850
5	0.975	0.970	0.866	0.763	0.869
6	0.975	0.970	0.863	0.764	0.870
7	0.974	0.969	0.864	0.754	0.877
8	0.974	0.970	0.865	0.762	0.868
9	0.974	0.969	0.863	0.756	0.860
10	0.973	0.968	0.859	0.756	0.857
11	0.971	0.967	0.854	0.744	0.850
12	0.969	0.965	0.847	0.735	0.843

Table 3: Task-specific metric results on probes taking in one layer, for each layer of the network. For upos, xpos, dep, and sst2, the metric is accuracy. For NER, it's span-level F_1 as computed by the Stanza library (Qi et al., 2020). For all metrics, higher is better.

RoBERTa Two-Layer $\mathcal{V}$-entropy

Layer	upos	xpos	dep	ner	sst2
0-0	0.335	0.345	1.466	0.391	0.639
0-1	0.141	0.154	0.763	0.210	0.633
0-2	0.115	0.133	0.646	0.184	0.615
0-3	0.110	0.127	0.609	0.169	0.567
0-4	0.109	0.126	0.593	0.164	0.549
0-5	0.110	0.125	0.602	0.163	0.474
0-6	0.109	0.126	0.613	0.165	0.484
0-7	0.109	0.127	0.609	0.166	0.451
0-8	0.108	0.125	0.598	0.171	0.462
0-9	0.109	0.125	0.614	0.167	0.489
0-10	0.110	0.127	0.636	0.177	0.504
0-11	0.111	0.127	0.654	0.175	0.525
0-12	0.116	0.132	0.682	0.185	0.563

Table 4: $\mathcal{V}$-entropy results on probes taking in two layers: layer 0 and each other layer of the network. Lower is better.

RoBERTa Two-Layer Metrics

Layer	upos	xpos	dep	ner	sst2
0-0	0.908	0.907	0.670	0.543	0.808
0-1	0.969	0.965	0.834	0.722	0.825
0-2	0.975	0.970	0.861	0.744	0.822
0-3	0.976	0.971	0.870	0.765	0.850
0-4	0.977	0.972	0.874	0.767	0.849
0-5	0.977	0.972	0.872	0.772	0.875
0-6	0.977	0.972	0.869	0.766	0.875
0-7	0.977	0.972	0.869	0.760	0.869
0-8	0.977	0.972	0.872	0.762	0.862
0-9	0.977	0.972	0.869	0.766	0.864
0-10	0.977	0.972	0.864	0.755	0.857
0-11	0.977	0.972	0.861	0.753	0.859
0-12	0.975	0.971	0.856	0.743	0.847

Table 5: Task-specific metric results on probes taking in two layers: layer 0 and each other layer of the network. For upos, xpos, dep, and sst2, the metric is accuracy. For NER, it's span-level F_1 as computed by the Stanza library. For all metrics, higher is better.