Squib

Probing Classifiers: Promises, Shortcomings, and Advances

Yonatan Belinkov*
Technion – Israel Institute of Technology
belinkov@technion.ac.il

Probing classifiers have emerged as one of the prominent methodologies for interpreting and analyzing deep neural network models of natural language processing. The basic idea is simple — a classifier is trained to predict some linguistic property from a model's representations — and has been used to examine a wide variety of models and properties. However, recent studies have demonstrated various methodological limitations of this approach. This article critically reviews the probing classifiers framework, highlighting their promises, shortcomings, and advances.

1 Introduction

The opaqueness of deep neural network models of natural language processing (NLP) has spurred a line of research into interpreting and analyzing them. Analysis methods may aim to answer questions about a model's structure or its decisions. For instance, one might ask which parts of a neural neural model are responsible for certain linguistic properties, or which parts of the input led the model to make a certain decision. A common methodology to answer questions about the structure of models is to associate internal representations with external properties, by training a classifier on said representations that predicts a given property. This framework, known as **probing classifiers**, has emerged as a prominent analysis strategy in many studies of NLP models.[1]

Despite its apparent success, the probing classifiers paradigm is not without limitations. Critiques have been made about comparative baselines, metrics, the choice of classifier, and the correlational nature of the method. In this short article, we first define the probing classifiers framework, taking care to consider the various involved components. Then we summarize the framework's shortcomings, as well as improvements and advances. This article provides a roadmap for NLP researchers who wish to examine probing classifiers more critically and highlights areas in need of additional research.

2 The Probing Classifiers Framework

On the surface, the probing classifiers idea seems straightforward. We take a model that was trained on some task, such as a language model. We generate representations using the model, and train another classifier that takes the representations and predicts some

* Supported by the Viterbi Fellowship in the Center for Computer Engineering at the Technion.

Submission received: 4 March 2021; revised version received: 31 July 2021; accepted for publication: 8 September 2021

1 For an overviews of analysis methods in NLP, see the survey by Belinkov and Glass (2019), as well as the tutorials by Belinkov, Gehrmann, and Pavlick (2020) and Wallace, Gardner, and Singh (2020). For an overview of explanation methods in particular, see the survey by Danilevsky et al. (2020).

property. If the classifier performs well, we say that the model has learned information relevant for the property. However, upon closer inspection, it turns out that much more is involved here. To see this, we now define this framework a bit more formally.

Let us denote by $f : x \mapsto \hat{y}$ a model that maps input x to output $\hat{y}$. We call this model the original model. It is trained on some annotated dataset $\mathcal{D}_O = \{x^{(i)}, y^{(i)}\}$, which we refer to as the original dataset. Its performance is evaluated by some measure, denoted $\text{PERF}(f, \mathcal{D}_O)$. The function f is typically a deep neural network that generates intermediate representations of x, for example $f_l(x)$ may denote the representation of x at layer l of f.[2] A probing classifier $g : f_l(x) \mapsto \hat{z}$ maps intermediate representations to some property $\hat{z}$, which is typically some linguistic feature of interest. As a concrete example, f might be a sentiment analysis model, mapping a text x to a sentiment label y, while g might be a classifier mapping intermediate representations $f_l(x)$ to part-of-speech tags z. The classifier g is trained and evaluated on some annotated dataset $\mathcal{D}_P = \{x^{(i)}, z^{(i)}\}$, and some performance measure $\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P)$ (e.g., accuracy) is reported. Note that the performance measure depends on the probing classifier g and the probing dataset $\mathcal{D}_P$, as well as on the original model f and the original dataset $\mathcal{D}_O$.

From an information theoretic perspective, training the probing classifier g can be seen as estimating the mutual information between the intermediate representations $f_l(x)$ and the property z (Belinkov 2018, p. 42; Pimentel et al. 2020b; Zhu and Rudzicz 2020), which we write $\text{I}(\mathbf{z}; \mathbf{h})$, where $\mathbf{z}$ is a random variable ranging over properties z and $\mathbf{h}$ is a random variable ranging over representations $f_l(x)$.

The above careful definition of the probing classifiers framework reveals that it is comprised of multiple concepts and components, depicted in Figure 1a. The choice of each such component, and the interactions between them, lead to non-trivial questions regarding the design and implementation of any probing classifier experiment. Before we turn to these considerations in Section 4, we briefly review some history and promises of probing classifiers in the next section.

3 Promises

Perhaps the first studies that can be cast in the framework of probing classifiers are by Köhn (2015) and Gupta et al. (2015), who trained classifiers on static word embeddings to predict various morphological, syntactic, and semantic properties. Their goals were to provide more nuanced evaluations of word embeddings compared to prior work, which only integrated them in downstream tasks. Other early work classified hidden states of a recurrent neural network machine translation system into morpho-syntactic properties (Shi, Padhi, and Knight 2016). They were motivated by the end-to-end nature of the neural machine translation system, which, compared to a phrase/syntax-based system, did not explicitly integrate such properties (so they ask: "What kind of syntactic information is learned, and how much?"). The framework has taken up a more stable form by several groups who studied sentence embeddings (Ettinger, Elgohary, and Resnik 2016; Adi et al. 2017; Conneau et al. 2018) and recurrent/recursive neural networks (Belinkov et al. 2017a; Hupkes, Veldhoen, and Zuidema 2018).[3] The same idea had been concurrently proposed for investigating computer vision models (Alain and Bengio 2016).

2 We use $f_l(x)$ to refer more generally to any intermediate output of f when applied to x, so the framework includes analyses of other model components, such as attention weights (Clark et al. 2019).

3 For chronological completeness, workshop and preprint versions of Hupkes, Veldhoen, and Zuidema (2018) and Adi et al. (2017) appeared earlier (Veldhoen, Hupkes, and Zuidema 2016; Adi et al. 2016).

$x \mapsto y$	Original task
$\mathcal{D}_O = \{x^{(i)}, y^{(i)}\}$	Original dataset
$f : x \mapsto y$	Original model
$\text{PERF}(f, \mathcal{D}_O)$	Performance on the original task
$f_l(x)$	Representations of x from f
$f_l(x) \mapsto z$	Probing task
$\mathcal{D}_P = \{x^{(i)}, z^{(i)}\}$	Probing dataset
$g : f_l(x) \mapsto z$	Probing classifier
$\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P)$	Probing performance

(a) Basic Components.

$\bar{f} : x \mapsto y$	Skyline model or upper bound
$\underline{f} : x \mapsto y$	Baseline model
$x \mapsto y_{Rand}$	Control task (Hewitt and Liang 2019)
$c : f_l(x) \mapsto c(f_l(x))$	Control function (Pimentel et al. 2020b)
$\mathcal{D}_{P,Rand}$	Control task dataset (Hewitt and Liang 2019)
$\mathcal{D}_{O,z}$	Control dataset (Ravichander, Belinkov, and Hovy 2021)
$\text{SEL}(g, f, \mathcal{D}_O, \mathcal{D}_P, \mathcal{D}_{P,Rand})$	Probing selectivity (Hewitt and Liang 2019)
$\mathcal{G}(\mathbf{z}, \mathbf{h}, c)$	Information gain w.r.t control function (Pimentel et al. 2020b)
$\text{MDL}(g, f, \mathcal{D}_O, \mathcal{D}_P)$	Probe minimum description length (Voita and Titov 2020)
$\tilde{f}_l(x)$	Representations of x from f, after an intervention

(b) Additional Components.

Figure 1: Components comprising the probing classifiers framework.

A main motivation in this body of work is the *opacity* of the representations.[4] Compared to performance on downstream tasks, probing classifiers aim to provide more nuanced evaluations w.r.t *simple properties*.[5] Indeed, following the initial studies, a plethora of work has applied the framework to various models and properties, alleviating some of the opacity, at least in terms of properties encoded in the representations. See Belinkov and Glass (2019) for a comprehensive survey up to early 2019.[6]

However, what can be inferred from successful probing performance is less obvious. Good probing performance is often taken to indicate several potential situations: good *quality* of the representations w.r.t the probing property,[7] *readability* of information found in the representations,[8] or its *extractability*.[9] In contrast, low probing performance is taken to indicate that the probing property is not present in the representations or is

4 "little is known about the information that is captured by different sentence embedding learning mechanisms" (Adi et al. 2017); "a poor understanding of what they are capturing" (Conneau et al. 2018); "little is known about what and how much these models learn." (Belinkov et al. 2017a).

5 "fine-grained measurement of some of the information encoded in sentence embeddings" (Adi et al. 2017); "simple linguistic properties of sentences" (Conneau et al. 2018); "assessing the specific semantic information that is being captured in sentence representations" (Ettinger, Elgohary, and Resnik 2016).

6 There have also been numerous other studies using the probing classifier framework as is. For a partial list, see https://github.com/boknilev/nlp-analysis-methods/issues/5. For recent analyses focusing on the BERT model (Devlin et al. 2019), see Rogers, Kovaleva, and Rumshisky (2020).

7 "evaluate the quality of the trained classifier on the given task as a proxy to the quality of the extracted representations" (Belinkov et al. 2017a).

8 "If the classifier succeeds, it means that the pre-trained encoder is storing readable tense information into the embeddings it creates" (Conneau et al. 2018).

9 "testing for extractability of semantic information by testing classification accuracy.." (Ettinger, Elgohary, and Resnik 2016); "if a sequential model is computing certain information, or merely keeping track of it,

not usable.[10] Sometimes, good performance is taken to indicate *how* the original model achieves its behavior on the original task (Hupkes, Veldhoen, and Zuidema 2018). A linear probing classifier is thought to reveal features that are used by the original model, while a more complex probe "bears the risk that the classifier infers features that are not actually used by the network" (Hupkes, Veldhoen, and Zuidema 2018). Often, different terms (*quality*, *readability*, *usability*, etc.) appear abstractedly without precise definitions.

As we shall see, some of the above assumptions and conclusions are better accounted for than others by the probing classifiers paradigm. Indeed, the community has recently taken a more critical look at the methodology, which we turn to now.

4 Shortcomings and Advances

In light of the promises discussed above, this section reviews several limitations of the probing classifiers framework, as well as existing proposals for addressing them. We discuss comparisons and controls, how to choose the probing classifier, which causal claims can be made, the difference between datasets and tasks, and the need to define the probed properties. We formalize new additional components (Figure 1b) in a unified framework, along with the basic components (Figure 1a).

4.1 Comparisons and controls

A first concern with the framework is how to interpret the results of a probing classifier experiment. Suppose we run such an experiment and obtain a performance of $\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P) = 87.8$. Is that a high/low number? What should we compare it to? We will denote a baseline model with $\underline{f}$ and an upper bound or skyline model with $\bar{f}$.

Some studies compare with majority baselines (Belinkov et al. 2017a; Conneau et al. 2018) or with classifiers trained on representations that are thought to be simpler than what the original model f produces, such as static word embeddings (Belinkov et al. 2017a; Tenney et al. 2019). Others advocate for random baselines, training the classifier g on a randomized version of f (Conneau et al. 2018; Zhang and Bowman 2018; Tenney et al. 2019; Chrupała, Higy, and Alishahi 2020). These studies show that even random features capture significant information that can be decoded by the probing classifier, so performance on learned features should be viewed in such a perspective.

On the other hand, some studies compare $\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P)$ to skylines or upper bounds $\bar{f}$, in an attempt to provide a point of comparison for how far probing performance is from the possible performance on the task of mapping $x \mapsto z$. Examples include estimating human performance (Conneau et al. 2018), reporting the state of the art from the literature (Liu et al. 2019), or training a dedicated model to predict z from x, without restricting to (frozen) representations from f (Belinkov et al. 2017b).

Others have proposed to design controls for possible confounders. Hewitt and Liang (2019) observe that the probing performance $\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P)$ may tell us more about the probe g than about the model f. The probe g may memorize information from $\mathcal{D}_P$, rather than evaluate information found in representations $f(x)$. They design control tasks, which a probe may only solve by memorizing. In particular, they randomize the labels in $\mathcal{D}_P$, creating a new dataset $\mathcal{D}_{P,Rand}$. Then, they define *selectivity* as the difference between the probing performance on the probing task and the control task:

it should be possible to extract this information from its internal state space" (Hupkes, Veldhoen, and Zuidema 2018).

10 "low accuracy suggests this information is not represented in the hidden state" (Hupkes, Veldhoen, and Zuidema 2018); "if we cannot train a classifier to predict some property of a sentence based on its vector representation, then this property is not encoded in the representation (or rather, not encoded in a useful way, considering how the representation is likely to be used)" (Adi et al. 2017).

$\text{SEL}(g, f, \mathcal{D}_O, \mathcal{D}_P, \mathcal{D}_{P,Rand}) = \text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P) - \text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_{P,Rand})$. They show that probes may have high accuracy, but low selectivity, and that linear probes tend to have high selectivity, while non-linear probes tend to have low selectivity. This indicates that high accuracy of non-linear probes may come from memorization of surface patterns by the probe g, rather than from information captured in the representations $f_l(x)$. The control tasks introduced by Hewitt and Liang are particularly suited for word-level properties z as they evaluate memorization of word types; it is less clear how to apply this idea more broadly, such as in sentence-level properties.

Taking an information-theoretic perspective on probing, Pimentel et al. (2020b) proposed to use control functions instead of control tasks in order to compare probes. Their control function is any function applied to the representation, $c : f_l(x) \mapsto c(f_l(x))$, and they compare the information gain, which is the difference in mutual information between the property z and the representation before and after applying the control function: $\mathcal{G}(\mathbf{z}, \mathbf{h}, c) = \text{I}(\mathbf{z}; \mathbf{h}) - \text{I}(\mathbf{z}; \mathbf{c}(\mathbf{h}))$. While Pimentel et al. (2020b) posit that their control function are a better criterion than the control tasks of Hewitt and Liang (2019), subsequent work showed that the two criteria are almost equivalent, both theoretically and empirically (Zhu and Rudzicz 2020).

Another kind of control is proposed by Ravichander, Belinkov, and Hovy (2021), who design control datasets, where the linguistic property z is not discriminative w.r.t the original task of mapping x to y. That is, they modify $\mathcal{D}_O$ and create a new dataset, $\mathcal{D}_{O,z}$, where all examples have the same value for property z. Intuitively, a model f trained on $\mathcal{D}_{O,z}$ should not pick up information about z, since it is not useful for the task of f. They show that a probe g may learn to predict property z incidentally, even when it is not discriminative w.r.t the original task of mapping $x \mapsto y$, casting doubts on causal claims concerning the effect that a property encoded in the representation may have on the original task. While they create control datasets for probing sentence-level information, the same idea can be applied to word-level properties.

4.2 Which classifier to use?

Another concern is the choice of the probing classifier g: What should be its structure? What role does its expressivity play in drawing conclusions about the original model f?

Some studies advocate for using simple probes, such as linear classifiers (Alain and Bengio 2016; Hupkes, Veldhoen, and Zuidema 2018; Liu et al. 2019; Hall Maudslay et al. 2020). Somewhat anecdotally, a few studies observed better performance with more complex probes, but reported similar relative trends (Conneau et al. 2018; Belinkov 2018). That is, a ranking $\text{PERF}(g, f_1, \mathcal{D}_O, \mathcal{D}_P) > \text{PERF}(g, f_2, \mathcal{D}_O, \mathcal{D}_P)$, of two representations $f_1(x)$ and $f_2(x)$, holds across different probes g. However, this pattern may be flipped under alternative measures, such as selectivity (Hewitt and Liang 2019).

Several studies considered the complexity of the probe g in more detail. Pimentel et al. (2020b) argue that, in order to give the best estimate about the information that model f has about property z, the most complex probe should be used. In a more practical view, Voita and Titov (2020) propose to measure both the performance of the probe g and its complexity, by estimating the minimum description length of the code required to transmit property z knowing the representations $f_l(x)$: $\text{MDL}(g, f, \mathcal{D}_O, \mathcal{D}_P)$. Note that this measure again depends on the probe g, the model f, and their respective datasets $\mathcal{D}_O$ and $\mathcal{D}_P$. They found that MDL provides more information about how a probe g works, for instance by revealing differences in complexity of probes when performing control tasks from $\mathcal{D}_{P,Rand}$, as in Hewitt and Liang (2019). Pimentel et al. (2020a) argue that probing work should report the possible trade-offs between accuracy and complexity, along a range of probes g, and call for using probes that are both

simple and accurate. While they study a number of linear and non-linear multi-layered perceptrons, one could extend this idea to other classes of probes. Indeed, Cao, Sanh, and Rush (2021) design a pruning-based probe, which learns a mask on weights of f and obtains a better accuracy–complexity trade-off than a non-linear probe.

Another line of work proposes methods to extract linguistic information from a trained model without learning additional parameters. In particular, much work has used some sort of pairwise importance score between words in a sentence as a signal for inferring linguistic properties, either full syntactic parsing or more fine-grained properties such as coreference resolution. These scores may come from attention weights (Raganato and Tiedemann 2018; Clark et al. 2019; Mareček and Rosa 2019; Htut et al. 2019) or from distances between word representations, perhaps including perturbations of the input sentence (Wu et al. 2020). The pairwise scores can feed into some general parsing algorithm, such as the Chu-Liu Edmonds algorithm (1965; 1967). Alternatively, some work has used representational similarity analysis (Kriegeskorte, Mur, and Bandettini 2008) to measure similarity between word or sentence representations and syntactic properties, both local properties like determining a verb's subject (Lepori and McCoy 2020) and more structured properties like inferring the full syntactic tree (Chrupała and Alishahi 2019). Also related is work on clustering representations w.r.t linguistic property and classifying by cluster assignment (Zhou and Srikumar 2021). This line of work can be seen as a parameter-less probing classifier g: a linguistic property is inferred from internal model components (representations, attention weights), without needing to learn new parameters. Thus, such work avoids some of the issues about what the probe learns. Additionally, from the perspective of an accuracy–complexity trade-off, such work should perhaps be placed on the low end of the complexity axis, although the complexity of the parsing algorithm could also be taken into account.

4.3 Correlation vs. causation

A main limitation of the probing classifier paradigm is the disconnect between the probing classifier g and the original model f. They are trained in two different steps, where f is trained once and only used to generate feature representations $f_l(x)$, which are fed into g. Once we have $f_l(x)$, we get a probing performance from g, which tells us something about the information in $f_l(x)$. However, in the process, we have forgotten about the original task assigned to f, which was to predict y. This raises an important question, which early work has largely taken for granted (Section 3): Does model f use the information discovered by probe g? In other words, the probing framework may indicate correlations between representations $f_l(x)$ and linguistic property z, but it does not tell us whether this property is involved in predictions of f. Indeed, several studies pointed out this limitation (Belinkov and Glass 2019), including reports on a mismatch between performance of the probe, $\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P)$, and performance of the original model, $\text{PERF}(f, \mathcal{D}_O)$ (Vanmassenhove, Du, and Way 2017). In contrast, Lovering et al. (2021) find that extractability of a property according to $\text{MDL}(g, f, \mathcal{D}_O, \mathcal{D}_P)$ is correlated with f making predictions consistent with that property. Relatedly, Tamkin et al. (2020) find a discrepancy between features $f_l(x)$ obtaining high probing performance, $\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P)$, and features identified as important when fine-tuning f while performing the probing task $f_l(x) \mapsto z$. They reveal this by randomizing the weights of specific layers when fine-tuning f, which can be seen as a kind of intervention.

Indeed, a number of studies have proposed improvements to the probing classifier paradigm, which aim to discover causal effects by *intervening* in representations of the model f. Giulianelli et al. (2018) use gradients from g to modify the representations in f and evaluate how this change affects both the probing performance and the original

model performance. In their case, f is a language model and g predicts subject–verb number agreement. They find that their intervention increases probing performance, as may be expected. Interestingly, while in the general language modeling case the intervention has a small effect on the original model performance, $\text{PERF}(f, \mathcal{D}_O)$, they find an increase in this performance on examples designed to assess number agreement. They conclude that probing classifiers can identify features that are actually used by the model. Tucker, Qian, and Levy (2021) also use probe gradients to update the representations $f_l(x)$ w.r.t z, resulting in what they call counterfactual representations, and measure the effect on other properties. Similarly, Elazar et al. (2021) remove certain properties z (such as parts of speech or syntactic dependencies) from representations in f by repeatedly training (linear) probing classifiers g and projecting them out of the representation. This results in a modified representation $\tilde{f}_l(x)$, which has less information about z. They compare the probing performance to the performance on the original task (in their case, language modeling) after the removal of said features. They find that high probing performance $\text{PERF}(g, f, \mathcal{D}_O, \mathcal{D}_P)$ does not necessarily entail a large drop in original task performance after their removal, that is, $\text{PERF}(\tilde{f}, \mathcal{D}_O)$. Thus, contrary to Giulianelli et al. (2018), they conclude that probing classifiers do not always identify features that are actually used by the model. In a similar vein, Feder et al. (2021) remove properties z from representations in f by training g adversarially. At the same time, another probing classifier g_C is trained positively, aiming to control for properties z_C that should not be removed from f. A major difference from standard probing classifiers work is the continued updating of f. They find that they can accurately estimate the effect of properties z on downstream tasks performed by f when it is fine-tuned.[11]

4.4 Datasets vs. tasks

The probing paradigm aims to study models performing some task ($f : x \mapsto \hat{y}$) via a classifier performing another task ($g : f_l(x) \mapsto \hat{z}$). However, in practice these *tasks* are operationalized via finite *datsaets*. Ravichander, Belinkov, and Hovy (2021) point out that datasets are imperfect proxies for tasks. Indeed, the effect of the choice of datasets—both the original dataset $\mathcal{D}_O$ and the probing dataset $\mathcal{D}_P$—has not been widely studied. Furthermore, we ideally want to disentangle the role of each dataset from the role of the original model f and probing classifier g. Unfortunately, models f tend to be trained on different datasets $\mathcal{D}_O$, making statements about models confounded with issues of datasets. Some prior work acknowledged that conclusions can only be made about the existing *trained models*, not about general *architectures* (Liu et al. 2019). However, in an ideal world, we would compare different architectures $\{f^i\}$ trained on the same dataset $\mathcal{D}_O$ or the same f trained on different datasets $\{\mathcal{D}_O^i\}$. Concerning the latter, Zhang et al. (2021) found that models require less data to encode syntactic and semantic properties compared to commonsense knowledge. More such experiments are currently lacking.

The effect of the probing dataset $\mathcal{D}_P$—its size, composition, etc.—is similarly not well studied. While some work reported results on multiple datasets when predicting the same property z (e.g., Belinkov et al. 2017a), more careful investigations are needed.

4.5 Properties must be pre-defined

Finally, inherent to the probing classifier framework is determining a property z to probe for. This limits the investigation in multiple ways: It constrains the work to

11 Other studies that perform interventions to interpret NLP models without involving probing classifiers (e.g., Bau et al. 2019; Lakretz et al. 2019; Vig et al. 2020) are left out of the present scope.

existing annotated datasets, which are often limited to English and certain properties. It also requires focusing on properties z that are thought to be relevant to the task of mapping $x \mapsto y$ a-priori, potentially leading to biased conclusions. In an isolated effort to alleviate this limitation, Michael, Botha, and Tenney (2020) propose to learn latent clusters useful for predicting a property z. They discover clusters corresponding to known properties (such as personhood) as well as new categories, which are not usually annotated in common datasets. Still, probing classifiers are so far mainly useful when one has prior expectations about which properties z might be relevant w.r.t a given task.

5 Summary

Given the various limitations discussed in this article, one might ask: What are probing classifiers good for? In line with the original motivation to alleviate the *opacity* of learned representations, work using probing classifiers has characterized them along a range of fine-grained properties. However, we have discussed several reservations regarding which insights can be drawn from a probing classifier experiment. Absolute claims about representation *quality* seem difficult to make. Yet recent improvements to the framework, such as better controls and metrics, allow us to make relative claims and answer questions like how *extractable* a property is from a representation. And causal approaches (Section 4.3) may reveal which properties are *used* by the original model.

One might hope that probing classifier experiments would suggest ways to improve the quality of the probed model or to direct it to be better tuned to some use or task. Presently, there are few such successful examples. For instance, earlier results showing that lower layers in language models focus on local phenomena while higher layers focus on global ones (using probing classifiers and other methods) motivated Cao et al. (2020) to decouple a question-answering model, such that lower layers process the question and the passage independently and higher layers process them jointly. An analysis of redundancy in language models (again using probing classifiers and other methods) motivated an efficient transfer-learning procedure (Dalvi et al. 2020). An analysis of phonetic information in layers of a speech recognition systems (Belinkov and Glass 2017) partly motivated Krishna, Toshniwal, and Livescu (2019) to propose multi-task learning with phonetic supervision on intermediate layers. Belinkov et al. (2020) discuss how their probing experiments can guide the selection of which machine translation models to use when translating specific languages. Finally, when considering using the representations for some downstream task, probing experiments can indicate which information is encoded, or can easily be extracted, from these representations.

To conclude, our critical review of the probing classifiers framework reveals that it is more complicated than may seem. When designing a probing classifier experiment, we advise researchers to take the various controls and alternative measures into account. Naturally, one should clearly define the original task/dataset/model and the probing task/dataset/classifier. It is important to set upper and lower bounds, and to consider proper controls, via either control tasks (for word-level properties) or datasets (for sentence-level properties). Depending on goals, one may want to measure the probe's complexity (if ease of extractability is in question), report the accuracy–complexity trade-off (when designing new probes), or perform an intervention (to measure usage of information by the original model). When possible, using parameter-free probes may circumvent some of the challenges with parameterized probes. We do not argue that every study must perform all the various controls and report all the alternative measures summarized here. However, future work seeking to use probing classifiers would do well to take into account the complexity of the framework, its apparent shortcomings, and available advances.

Acknowledgments

This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 448/20) and by an Azrieli Foundation Early Career Faculty Fellowship.

References

Adi, Yossi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *CoRR*, abs/1608.04207.

Adi, Yossi, 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 (ICLR)*.

Alain, Guillaume and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644v3*.

Bau, Anthony, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*.

Belinkov, Yonatan. 2018. *On Internal Language Representations in Deep Learning: An Analysis of Machine Translation and Speech Recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

Belinkov, Yonatan, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017a. 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, Association for Computational Linguistics, Vancouver, Canada.

Belinkov, Yonatan, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2020. On the linguistic representational power of neural machine translation models. *Computational Linguistics*, 46(1):1–52.

Belinkov, Yonatan, Sebastian Gehrmann, and Ellie Pavlick. 2020. Interpretability and analysis in neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 1–5, Association for Computational Linguistics, Online.

Belinkov, Yonatan and James Glass. 2017. Analyzing hidden representations in end-to-end automatic speech recognition systems. In *Advances in Neural Information Processing Systems*, volume 30, pages 2441–2451, Curran Associates, Inc.

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

Belinkov, Yonatan, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2017b. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Asian Federation of Natural Language Processing, Taipei, Taiwan.

Cao, Qingqing, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. DeFormer: Decomposing pre-trained transformers for faster question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4497, Association for Computational Linguistics, Online.

Cao, Steven, 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, Association for Computational Linguistics, Online.

Chrupała, Grzegorz and Afra Alishahi. 2019. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2952–2962, Association for Computational Linguistics, Florence, Italy.

Chrupała, Grzegorz, Bertrand Higy, and Afra Alishahi. 2020. Analyzing analytical methods: The case of phonology in neural models of spoken language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4146–4156, Association for Computational Linguistics, Online.

CHU, Y. 1965. On the shortest arborescence of a directed graph. *Science Sinica*, 14:1396–1400.

Clark, Kevin, 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, Association for Computational Linguistics, Florence, Italy.

Conneau, Alexis, 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, Association for Computational Linguistics, Melbourne, Australia.

Dalvi, Fahim, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926, Association for Computational Linguistics, Online.

Danilevsky, Marina, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable AI for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Association for Computational Linguistics, Suzhou, China.

Devlin, Jacob, 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, Association for Computational Linguistics, Minneapolis, Minnesota.

Edmonds, Jack. 1967. Optimum branchings. *Journal of Research of the national Bureau of Standards B*, 71(4):233–240.

Elazar, Yanai, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2021. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9(0):160–175.

Ettinger, Allyson, 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, Association for Computational Linguistics, Berlin, Germany.

Feder, Amir, Nadav Oved, Uri Shalit, and Roi Reichart. 2021. CausaLM: Causal Model Explanation Through Counterfactual Language Models. *Computational Linguistics*, 47(2):333–386.

Giulianelli, Mario, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Association for Computational Linguistics, Brussels, Belgium.

Gupta, Abhijeet, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Association for Computational Linguistics, Lisbon, Portugal.

Hall Maudslay, Rowan, 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, Association for Computational Linguistics, Online.

Hewitt, John 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, Association for Computational Linguistics, Hong Kong, China.

Htut, Phu Mon, Jason Phang, Shikha Bordia, and Samuel R Bowman. 2019. Do attention heads in bert track syntactic dependencies? *arXiv preprint arXiv:1911.12246*.

Hupkes, Dieuwke, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Köhn, Arne. 2015. What's in an embedding? analyzing word embeddings through multilingual evaluation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages

2067–2073, Association for Computational Linguistics, Lisbon, Portugal.

Kriegeskorte, Nikolaus, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4.

Krishna, Kalpesh, Shubham Toshniwal, and Karen Livescu. 2019. Hierarchical multitask learning for CTC-based speech recognition. *arXiv preprint arXiv:1807.06234*.

Lakretz, Yair, German Kruszewski, Theo Desbordes, Dieuwke Hupkes, Stanislas Dehaene, and Marco Baroni. 2019. The emergence of number and syntax units in LSTM language models. 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 11–20, Association for Computational Linguistics, Minneapolis, Minnesota.

Lepori, Michael and R. Thomas McCoy. 2020. Picking BERT's brain: Probing for linguistic dependencies in contextualized embeddings using representational similarity analysis. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3637–3651, International Committee on Computational Linguistics, Barcelona, Spain (Online).

Liu, Nelson F., Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. 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, Association for Computational Linguistics, Minneapolis, Minnesota.

Lovering, Charles, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pre-trained models. In *International Conference on Learning Representations*.

Mareček, David and Rudolf Rosa. 2019. From balustrades to pierre vinken: Looking for syntax in transformer self-attentions. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 263–275, Association for Computational Linguistics, Florence, Italy.

Michael, Julian, Jan A. Botha, and Ian Tenney. 2020. Asking without telling: Exploring latent ontologies in contextual representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6792–6812, Association for Computational Linguistics, Online.

Pimentel, Tiago, 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, Association for Computational Linguistics, Online.

Pimentel, Tiago, 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, Association for Computational Linguistics, Online.

Raganato, Alessandro and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 287–297, Association for Computational Linguistics, Brussels, Belgium.

Ravichander, Abhilasha, Yonatan Belinkov, and Eduard Hovy. 2021. Probing the probing paradigm: Does probing accuracy entail task relevance? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3363–3377, Association for Computational Linguistics, Online.

Rogers, Anna, 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.

Shi, Xing, 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, Association for Computational Linguistics, Austin, Texas.

Tamkin, Alex, Trisha Singh, Davide Giovanardi, and Noah Goodman. 2020. Investigating transferability in pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1393–1401,

Association for Computational Linguistics, Online.
Tenney, Ian, 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*.
Tucker, Mycal, Peng Qian, and Roger Levy. 2021. What if this modified that? syntactic interventions with counterfactual embeddings. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 862–875, Association for Computational Linguistics, Online.
Vanmassenhove, Eva, Jinhua Du, and Andy Way. 2017. Investigating 'aspect' in NMT and SMT: Translating the english simple past and present perfect. *Computational Linguistics in the Netherlands Journal*, 7:109–128.
Veldhoen, Sara, Dieuwke Hupkes, and Willem H Zuidema. 2016. Diagnostic classifiers revealing how neural networks process hierarchical structure. In *CoCo@ NIPS*.
Vig, Jesse, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401, Curran Associates, Inc.
Voita, Elena 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, Association for Computational Linguistics, Online.
Wallace, Eric, Matt Gardner, and Sameer Singh. 2020. Interpreting predictions of NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 20–23, Association for Computational Linguistics, Online.
Wu, Zhiyong, 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, Association for Computational Linguistics, Online.
Zhang, Kelly and Samuel Bowman. 2018. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 359–361, Association for Computational Linguistics, Brussels, Belgium.
Zhang, Yian, Alex Warstadt, Xiaocheng Li, and Samuel R. Bowman. 2021. When do you need billions of words of pretraining data? 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 1112–1125, Association for Computational Linguistics, Online.
Zhou, Yichu and Vivek Srikumar. 2021. DirectProbe: Studying representations without classifiers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5070–5083, Association for Computational Linguistics, Online.
Zhu, Zining and Frank Rudzicz. 2020. An information theoretic view on selecting linguistic probes. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9251–9262, Association for Computational Linguistics, Online.