INFORMATION-THEORETIC PROBING EXPLAINS RELIANCE ON SPURIOUS HEURISTICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Most current NLP systems are based on a pre-train-then-fine-tune paradigm, in which a large neural network is first trained in a self-supervised way designed to encourage the network to extract broadly-useful linguistic features, and then fine-tuned for a specific task of interest. Recent work attempts to understand why this recipe works and explain when it fails. Currently, such analyses have produced two sets of apparently-contradictory results. Work that analyzes the representations that result from pre-training (via "probing classifiers") finds evidence that rich features of linguistic structure can be decoded with high accuracy, but work that analyzes model behavior after fine-tuning (via "challenge sets") indicates that decisions are often not based on such structure but rather on spurious heuristics specific to the training set. In this work, we test the hypothesis that the extent to which a feature influences a model's decisions can be predicted using a combination of two factors: The feature's *extractability* after pre-training (measured using information-theoretic probing techniques), and the *evidence* available during fine-tuning (defined as the feature's co-occurrence rate with the label). In experiments with both synthetic and natural language data, we find strong evidence (statistically significant correlations) supporting this hypothesis.

1 INTRODUCTION

Large pre-trained language models (LMs) (Devlin et al., 2018; Raffel et al., 2019; Brown et al., 2020) have demonstrated impressive empirical success on a range of benchmark NLP tasks. However, analyses have shown that such models are easily fooled when tested on distributions that differ from those they were trained on, suggesting they are often "right for the wrong reasons" (McCoy et al., 2019). Recent research which attempts to understand why such models behave in this way has primarily made use of two analysis techniques: *probing classifiers* (Adi et al., 2017; Hupkes et al., 2018), which measure whether or not a given feature is encoded by a representation, and *challenge sets* (Cooper et al., 1996; Linzen et al., 2016b; Rudinger et al., 2018), which measure whether model behavior in practice is consistent with use of a given feature. The results obtained via these two techniques currently suggest different conclusions about how well pre-trained representations encode language. Work based on probing classifiers has consistently found evidence that models contain rich information about syntactic structure (Hewitt & Manning, 2019; Bau et al., 2019; Tenney et al., 2019a), while work using challenge sets has frequently revealed that models built on top of these representations do not behave as though they have access to such rich features, rather they fail in trivial ways (Dasgupta et al., 2018; Glockner et al., 2018a; Naik et al., 2018).

In this work, we attempt to link these two contrasting views of feature representations. We assume the standard recipe in NLP, in which linguistic representations are first derived from large-scale self-supervised *pre-training* intended to encode broadly-useful linguistic features, and then are adapted for a task of interest via transfer learning, or *fine-tuning*, on a task-specific dataset. We test the hypothesis that the extent to which a fine-tuned model uses a given feature can be explained as a function of two metrics: The *extractability* of the feature after pre-training (as measured by probing classifiers) and the *evidence* available during fine-tuning (defined as the rate of co-occurrence with the label). We first show results on a synthetic task, and second using state-of-the-art pre-trained LMs on language data. Our results suggest that probing classifiers can be viewed as a measure of the pre-trained representation's inductive biases: The more extractable a feature is after pre-training, the less statistical evidence is required in order for the model to adopt the feature during fine-tuning.

Contribution. This work establishes a relationship between two widely-used techniques for analyzing LMs. Currently, the question of how models' internal representations (measured by probing classifiers) influence model behavior (measured by challenge sets) remains open (Belinkov & Glass, 2019; Belinkov et al., 2020). Understanding the connection between these two measurement techniques can enable more principled evaluation of and control over neural NLP models.

2 Setup and Terminology

2.1 Formulation

Our motivation comes from McCoy et al. (2019), which demonstrated that, when fine-tuned on a natural language inference task (Williams et al., 2018, MNLI), a model based on a state-of-the-art pre-trained LM (Devlin et al., 2018, BERT) categorically fails on test examples which defy the expectation of a "lexical overlap heuristic". For example, the model assumes that the sentence *"the lawyer followed the judge"* entails *"the judge followed the lawyer"* purely because all the words in the latter appear in the former. While this heuristic is statistically favorable given the model's training data, it is not infallible. Specifically, McCoy et al. (2019) report that 90% of the training examples containing lexical overlap had the label "entailment", but the remaining 10% did not. Moreover, the results of recent studies based on probing classifiers suggest that more robust features are extractable with high reliability from BERT representations. For example, given the example *"the lawyer followed the judge"/"the judge followed the lawyer"*, if the model can represent that *"lawyer"* is the agent of *"follow"* in the first sentence, but is the patient in the second, then the model should conclude that the sentences have different meanings. Such semantic role information can be recovered at $> 90\%$ accuracy from BERT embeddings (Tenney et al., 2019b). Thus, the question is: Why would a model prefer a weak feature over a stronger one, if both features are extractable from the model's representations and justified by the model's training data?

Abstracting over details, we distill the basic NLP task setting described above into the following, to be formalized in the Section 2.2. We assume a binary sequence classification task where a *target* feature t perfectly predicts the label (e.g., the label is 1 *iff* t holds). Here, t represents features which actually determine the label by definition, e.g., whether one sentence semantically entails another. Additionally, there exists a *spurious* feature s that frequently co-occurs with t in training but is not guaranteed to generalize outside of the training set. Here, s (often called a "heuristic" or "bias" elsewhere in the literature) corresponds to features like lexical overlap, which are predictive of the label in some datasets but are not guaranteed to generalize.

Assumptions. In this work, we assume there is a single t and a single s; in practice there may be many s features. We assume that s and t frequently co-occur, but that only s occurs in isolation. This assumption reflects realistic NLP task settings since datasets always contain some heuristics, e.g., lexical cues, cultural biases, or artifacts from crowdsourcing (Gururangan et al., 2018). Thus, our experiments focus on manipulating the occurrence of s alone, but not t alone: This means giving the model evidence against relying on s. This is in line with prior applied work that attempts to influence model behavior by increasing the evidence against s during training (Min et al., 2020; Zmigrod et al., 2019; Elkahky et al., 2018).

2.2 Definitions

Let $\mathcal{X}$ be the set of all sentences and S be the space of all sentence-label pairs $(x, y) \in \mathcal{X} \times \{0, 1\}$. We use $\mathcal{D} \subset S$ to denote a particular training sample drawn from S. We define two types of binary features: *target* (t) and *spurious* (s). Each is a function from sentences $x \in \mathcal{X}$ to a binary label $\{0, 1\}$ that indicates whether the feature holds.

Target and spurious features. The *target* feature t is such that there exists some function $f : \{0, 1\} \rightarrow \{0, 1\}$ such that $\forall (x, y) \in S,\ f(t(x)) = y$. In other words, the label can always be perfectly predicted given the value of t.[1] A feature s is *spurious* if it is not a *target* feature.

[1] Without loss of generality, we define t in our datasets s.t. $t(x) = y, \forall x, y \in S$.

Partitions of S. To facilitate analysis, we partition S in four regions (Figure 1). We define $S_{s\text{-only}}$ to be the set of examples in which the spurious feature occurs alone (without the target). Similarly, $S_{t\text{-only}}$ is the set of examples in which the target occurs without the spurious feature. S_{both} and S_{neither} are analogous. For compactness, we sometimes drop the S_* notation (e.g., s-only in place of $S_{s\text{-only}}$).

$$
\begin{aligned}
S_{both} &= \{(x, y) \,|\, t(x) = 1 \wedge s(x) = 1\} \\
S_{neither} &= \{(x, y) \,|\, t(x) = 0 \wedge s(x) = 0\} \\
S_{t\text{-only}} &= \{(x, y) \,|\, t(x) = 1 \wedge s(x) = 0\} \\
S_{s\text{-only}} &= \{(x, y) \,|\, t(x) = 0 \wedge s(x) = 1\}
\end{aligned}
$$

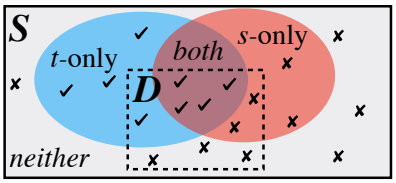


Figure 1: We partition datasets into four sections, defined by the features (spurious and/or target) that hold. We sample training datasets D, which provide varying amounts of *evidence* against the spurious feature, in the form of s-only examples. In the illustration above, the s-only rate is $\frac{2}{10} = 0.2$, i.e., 20% of examples in D provide evidence that s alone should not be used to predict y.

Evidence from Spurious-Only Examples. We are interested in spurious features which are highly correlated with the target during training. Given a training sample $\mathcal{D}$ and features s and t, we define the ***s*-only example rate** to be a measure of the model's evidence against the use of s as a predictor of y. Concretely, s-only rate $= \frac{|D_{s\text{-only}}|}{|D|}$, the proportion of training examples in which s occurs without t (and $y = 0$).

Use of Spurious Feature. If a model has falsely learned that the spurious feature s alone is predictive of the label, it will have a high error rate when classifying examples for which s holds but t does not. We define the ***s*-only error** to be the classifier's error computed only over examples drawn from $S_{s\text{-only}}$. When relevant, ***t*-only error**, **both error**, and **neither error** are defined analogously.

Extractability of a Feature. We want to compare features in terms of how *extractable* they are given a representation. For example, given a sentence embedding, it may be possible to predict multiple features with high accuracy, e.g., whether the word *"dog"* occurs, and also whether the word *"dog"* occurs as the subject of the verb *"run"*. However, detecting the former will no doubt be an easier task than detecting the latter. We use the prequential **minimum description length (MDL)**–introduced by Voita & Titov (2020)–in order to quantify this intuitive difference.[2] MDL is an information-theoretic metric that measures how accurately a feature can be decoded and the amount of effort required to decode it. Formally, MDL measures the number of bits required to communicate the labels given the representations. Conceptually, MDL can be understood as a measure of the area under the loss curve: If a feature is highly *extractable*, a model trained to detect that feature will converge quickly to high accuracy, resulting in a low MDL. Computing MDL requires repeatedly training a model over a dataset labeled by the feature in question. To compute MDL(s), we train a classifier (without freezing any parameters) to differentiate $S_{s\text{-only}}$ vs. S_{neither}, and similarly compute MDL(t). See Voita & Titov (2020) for additional details on MDL.[3]

2.3 Hypothesis

Stated using the above-defined terminology, our hypothesis is that a model's *use of the target feature* is modulated by two factors: The relative *extractability* of the target feature t (compared to the spurious feature s), and the *evidence* from s-only examples provided by the training data. In particular, we expect that higher extractability of t (relative to s), measured by MDL(s)/MDL(t), will yield models that achieve better performance despite less training evidence.

[2] We observe similar overall trends when using an alternative metric based on validation loss (Appendix C).

[3] Note that our reported MDL is higher in some cases than that given by the uniform code (the number of sentences that are being encoded). The MDL is computed as a sum of the costs of transmitting successively longer blocks, using classifiers that are trained on previously transmitted data. The high MDL's are likely a result of overfitting by classifiers that are trained on limited data–and therefore, bounds computed with classifiers that have worse compression performance than the uniform baseline.

3 EXPERIMENTS WITH SYNTHETIC DATA

Since it is often difficult to fully decouple the target feature from competing spurious features in practice, we first use synthetic data in order to test our hypothesis in a clean setting. We use a simple classifier with an embedding layer, a 1-layer LSTM, and an MLP with 1 hidden layer with tanh activation. We use a synthetic sentence classification task with k-length sequences of numbers as input and binary labels as output. We use a symbolic vocabulary V with the integers $0 \ldots |V| - 1$. We fix $k = 10$ and $|V| = 50$K. We begin with an initial training set of 200K, evenly split between examples from S_{both} and S_{neither}. Then, varied across runs, we manipulate the evidence against the spurious feature (i.e., the s-only rate) by replacing a percentage p of the initial data with examples from $S_{s\text{-only}}$ for $p \in \{0\%, 0.1\%, 1\%, 5\%, 10\%, 20\%, 50\%\}$. Test and validation sets consist of 1,000 examples each from $S_{\text{both}}, S_{\text{neither}}, S_{t\text{-only}}, S_{s\text{-only}}$. In all experiments, we set the spurious feature s to be the presence of the symbol `2`. We consider several different target features t (Table 1), intended to vary in their extractability. Table 1 contains MDL metrics for each feature (computed on training sets of $200K$, averaged over 3 random seeds). We see a gradation of feature extractability, as desired.[4]

Target Feature	Description	MDL(s)	MDL(t)	Rel. MDL	Example
`contains-1`	`1` occurs in sequence	0.36	0.29	1.259	2 4 11 1 4
`prefix-dupl`	Sequence begins with duplicate	0.42	175.74	0.002	5 5 11 12 2
`adj-dupl`	Adjacent duplicate in seq.	0.37	242.20	0.001	11 12 3 3 2
`first-last`	First number equals last number	0.37	397.64	0.001	7 2 11 12 7

Table 1: Instantiations of the target feature t in our synthetic experiments. The spurious feature s is always the presence of the symbol `2`. Features are intended to differ in how hard they are for an LSTM to detect given sequential input (measured by MDL per §2.2, reported in k-bits).

Figure 2 shows model performance as a function of s-only rate for each of the four features described above. Here, performance is reported using error rate (lower is better) on each partition ($S_{s\text{-only}}$, $S_{t\text{-only}}$, S_{both}, S_{neither}) separately. We are primarily interested in whether the relative extractability of the target feature (compared to the spurious feature) predicts model performance. We indeed see a fairly clear relationship between the relative extractability ($\frac{\text{MDL}(s)}{\text{MDL}(t)}$) and the performance of the model, at every level of training evidence (s-only rate). For example, when t is no less extractable than s (i.e., `contains-1`), the model achieves zero error at an s-only rate of 0.001, meaning it learns that t alone predicts the label despite having only a handful of examples that support this inference. In contrast, when t is harder to extract than s (e.g., `first-last`), the model fails to make this inference, even when a large portion of training examples provide evidence supporting it.

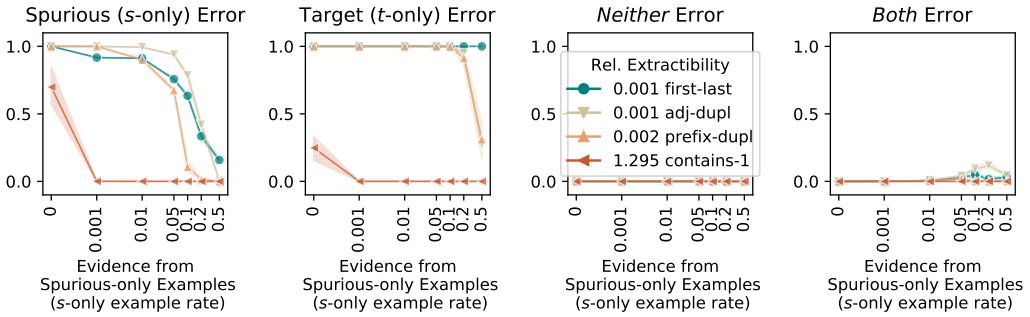


Figure 2: **Results on Synthetic Data**. Error on each partition of the test set, as a function of s-only rate. A model that has learned to use the target feature alone to predict the label will achieve zero error across all partitions. s-only and t-only error reach 0 quickly when t is as easy to extract as s (i.e., the realtive extractability is 1). However, when t is harder to extract than s (rel. extractability < 1), performance lags until evidence from s-only examples is quite strong.

[4]Note, all models are ultimately able to learn to detect t (achieve high test accuracy).

4 EXPERIMENTS ON NATURAL LANGUAGE

We now investigate whether the same trend holds for widely-used language models fine-tuned on natural language data. To do this, we fine-tune models for the linguistic acceptability task, a simple sequence classification task as defined in Warstadt & Bowman (2019), in which the goal is to differentiate grammatical sentences from ungrammatical ones. We focus on acceptability judgments since there exists substantial formal linguistic theory that can inform how we define the target features, as well as recent work in computational linguistics showing that neural language models can be sensitive to spurious features in this task (Warstadt et al., 2020; Marvin & Linzen, 2018).

4.1 DATA

We design a series of simple natural language grammars that generate a variety of feature pairs (s, t), which we expect will exhibit different levels of relative extractability ($\frac{\text{MDL}(s)}{\text{MDL}(t)}$). We focus on three syntactic phenomena (described below). In each case, we consider the target feature t to be whether a given instance of the phenomenon obeys the expected syntactic rules. We then introduce several spurious features s which we deliberately correlate with the positive label during fine-tuning.

The **Subject-Verb Agreement (SVA)** construction requires detecting whether the verb agrees in number with its subject, e.g., *"the girls are playing"* is acceptable while *"the girls is playing"* is not. In general, recognizing agreement requires some representation of hierarchical syntax, since subjects may be separated from their verbs by arbitrarily long clauses. We introduce four spurious features: 1) lexical, in which grammatical sentences begin with specific lexical items (e.g., *"often"*); 2) length, in which grammatical sentences are longer; 3) recent-noun, in which verbs in grammatical sentences agree with the immediately preceding noun (in addition to their subject); and 4) plural, in which verbs in grammatical sentences are preceded by singular nouns as opposed to plural ones.

The **Negative Polarity Items (NPI)** construction requires detecting whether a negative polarity item (e.g., *"any"*, *"ever"*) is grammatical in a given context, e.g., *"no girl ever played"* is acceptable while *"a girl ever played"* is not. In general, NPIs are only licensed in contexts that fall within the scope of a downward entailing operator (such as negation). We again consider four types of spurious features: 1) lexical, in which grammatical sentences always include one of a set of lexical items (*"no"* and *"not"*); 2) length (as above); 3) plural, in which each noun in a grammatical sentence is singular, as opposed to plural; and 4) tense, in which grammatical sentences are in present tense.

Some verbs (e.g. *"recognize"*) require a direct object. However, in the right syntactic contexts (i.e., when in the correct syntactic relation with a wh-word), the object position can be empty, creating what is known as a "gap". E.g., *"I know what you recognized __"* is acceptable while *"I know that you recognized __"* is not. The **Filler-Gap Dependencies (GAP)** construction requires detecting whether a sentence containing a gap is grammatical. For our GAP tasks, we again consider four spurious features (lexical, length, plural, and tense), defined similarly to above.

Target	Spurious	Example
Subject agrees with verb	N before V is singular	[*both*] The piano teachers of the lawyer wound the handyman.
		[s-only] *The piano teachers of the lawyer wounds the handyman.
NPI in down. -entailing context	Contains negation word	[*both*] No student who was wrong ever resigned.
		[s-only] *The student who was not wrong ever resigned.
Correct filler-gap dependency	Main verb is in past tense	[*both*] I knew what he recognized __ yesterday.
		[s-only] *I knew what he recognized someone yesterday.

Table 2: Examples of features used to generate fine-tuning sets with target/spurious features of varying extractability scores. Top examples show a case in which t and s both occur and the sentence is acceptable, and bottom examples show a case in which s occurs without t and the sentence is unacceptable. Only s is highlighted since t is often defined over the structure of the sentence (see text) and thus difficult to localize to a few tokens. Table 8 in the Appendix has *neither* examples.

The templates above (and slight variants) result in 20 distinct fine-tuning datasets, over which we perform our analyses (see Appendix for details). Table 2 shows several examples. For the purposes of this paper, we are interested only in the relative extractability of t vs. s given the pre-trained representation; we don't intend to make general claims about the linguistic phenomena *per se*. Thus, we do not focus on the details of the features themselves, but rather consider each template as generating one data point, i.e., an (s, t) pair representing a particular level of relative extractability.

4.2 SETUP

We use three models: T5, BERT, and an LSTM with GloVe embeddings (Devlin et al., 2018; Raffel et al., 2019; Pennington et al., 2014).[5] Both T5 and BERT learn to perform well over the whole test set, whereas the GloVe model struggles with many of the tasks. We expect that this is because the contextualized pre-training encodes certain syntactic features which let the models better leverage small training sets (Warstadt & Bowman, 2020). Again, we begin with an initial training set of 2000 examples, evenly split between *both* and *neither*, and then introduce s-only examples at rates of 0%, 0.1%, 1%, 5%, 10%, 20%, and 50%, using three random seeds each. Test and validation sets consist of 1000 examples each from $S_{\text{both}}, S_{\text{neither}}, S_{s\text{-only}}$. In the natural language setting, it is often difficult to generate t-only examples, and thus we cannot compute extractability of the target feature t by training a classifier to distinguish $S_{t\text{-only}}$ from a random subset of S_{neither}, as we did in Section 3. Therefore, we estimate MDL by training a classifier to distinguish between examples from $S_{s\text{-only}}$ and examples from S_{both}. Using the simulated data from Section 3, we confirm that both methods ($S_{s\text{-only}}$ vs. S_{both} and $S_{t\text{-only}}$ vs. S_{neither}) produce similar estimates of MDL(t) (see Appendix).

4.3 RESULTS

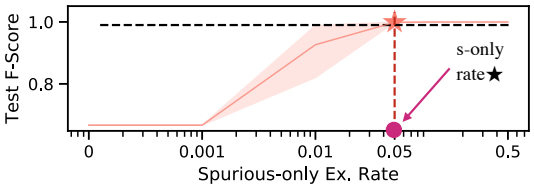
For each of our (s, t) feature pairs, we plot the use of the spurious feature (s-only error) as a function of the evidence against the spurious feature seen in training (s-only example rate).[6] We expect to see the same trend we observed in our synthetic data, i.e., the more extractable the target feature t is relative to the spurious feature s, the less evidence the model will require before preferring t over s. To quantify this trend, we compute correlations between 1) the relative extractability of t compared to s and 2) the amount of training evidence required for the model to adopt the target feature. For (2), we define ***s*-rate**$\star$ to be the lowest s-only example rate at which the fine-tuned model is able to achieve essentially perfect performance (F-score > 0.99) (see Fig. 3a). Intuitively, s-rate$\star$ is the (observed) minimum amount of evidence from which the model is able to infer that the t alone is predictive of the label.

Figure 3 shows these correlations and associated scatter plots. We can see that relative extractability is strongly correlated with s-rate$\star$ (Fig. 3b), showing highly significant negative correlation for both BERT ($\rho = -0.63$) and T5 ($\rho = -0.78$). That is, the more extractable t is relative to s, the less evidence the model requires before preferring t. This relationship holds regardless of whether relative extractability is computed using a ratio of MDL scores or an absolute difference. We also see that, in most cases, the relative extractability explains the model's behavior better than does the extractability of s or t alone. This trend is also apparent, albeit weak, for the GloVe model. BERT, T5, get > 0.99 accuracy on all feature pairs when testing the target feature in isolation, but the GloVe model solved the target task in isolation for only 60% (12/20) feature pairs. This partially explains why the GloVe model results are less clear.

Figure 4 shows the learning curves for T5 (BERT and GloVe in Appendix), i.e., use of the spurious feature (s-only error) as a function of the evidence from s-only examples seen in training (s-only example rate). For pairs with high MDL ratios (i.e., when t is actually easier to extract than s), the model learns to solve the task "the right way" even when the training data provides no incentive to do so. That is, in such cases, the models' decisions do not appear to depend on the spurious feature s even in cases when s and the target feature t perfectly co-occur in the fine-tuning data.

[5] In pilot studies, we found that standard BOW and CNN-based models were unable to learn the tasks at all.

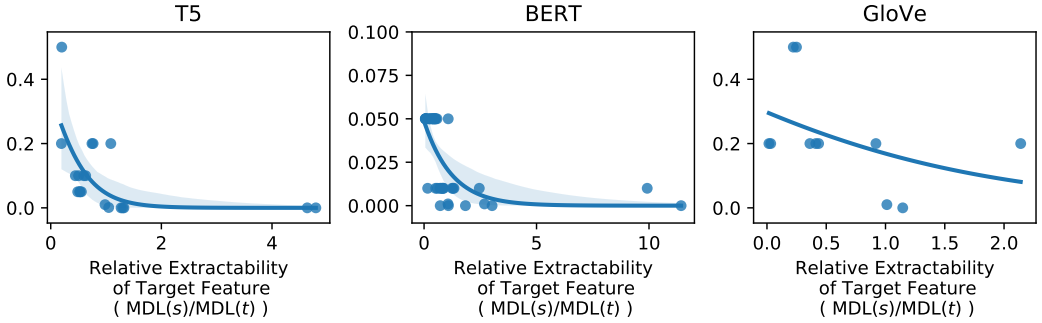
[6] See Appendix for *both* error and *neither* error; both are stable and low in general.



(a) s-rate$\star$ is determined at 0.99 test F-score.

	Absolute		Relative (t to s)	
	Tgt.	Spurious	Ratio	Diff.
T5	**0.48**	-0.27	**-0.67**	**-0.74**
BERT	0.44	**-0.71**	**-0.82**	**-0.84**
GloVe	0.36	-0.49	-0.58	-0.58

(b) Spearman's ρ: MDL vs. s-rate$\star$



(c) Logistic regression plots between s-rate$\star$ and extractability of t relative to s.

Figure 3: **Relative Extractability Correlates with Adoption of Target Feature**. Table shows Spearman's ρ between s-rate$\star$ (the amount of evidence required before a fine-tuned model adopts the target feature) and various measures of extractability over the (s, t) pairs. Bold indicates a significant correlation. Relative extractability, whether ratio ($\frac{\text{MDL}(s)}{\text{MDL}(t)}$) or difference ($\text{MDL}(s) - \text{MDL}(t)$) explains learning behavior better than absolute extractability of either feature.

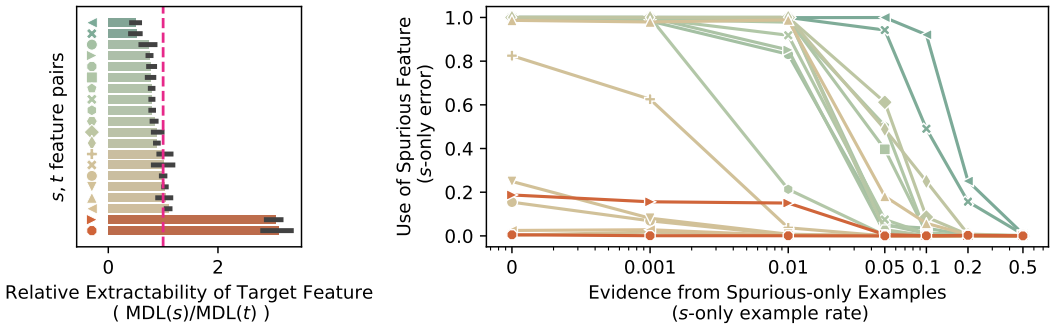


Figure 4: **Learning Curves for T5.** Curves show use of spurious feature (s-only accuracy) as a function of training evidence (s-only rate). Each line represents one (t, s) pair (described in §4.1). Pairs vary in the relative extractability of t vs. s (measured by the ratio $\text{MDL}(s)/\text{MDL}(t)$ and summarized in the bar chart). When t is much harder to extract relative to s (lower ratios), the classifier requires much more statistical evidence during training (higher s-only rate) in order to achieve low error. We find similar patterns for GPT2 Radford et al. (2019) and RoBERTa Liu et al. (2019b); see Appendix A.

5 Discussion

We hypothesized that the relative *extractability* of features given an input representation (as measured by information-theoretic probing techniques) is predictive of the decisions a trained model will make in practice. Our experimental results, both in simulation and in a realistic NLP setting, provide support for this hypothesis. In particular, we see evidence that models will tend to use imperfect features that are more readily-extractable over perfectly-predictive features that are harder to extract. This insight is highly related to prior work which has shown, e.g., that neural networks learn "easy" examples before they learn "hard" examples (Mangalam & Prabhu, 2019). What is novel about our

findings is the direct connection to new probing techniques which have received significant attention in NLP but have yet to be connected to explanations of or predictions about SOTA models' decisions in practice. One interpretation of our results is that probing classifiers can be viewed as measures of a pre-trained representation's inductive biases. Thus far, analysis using probing classifiers has primarily focused on whether important linguistic features can be decoded from representations at better-than-baseline levels, but there has been little insight about what it would mean for a representations' encoding of a feature to be "sufficient". Based on these experiments, we argue that a feature is "sufficiently" encoded if it is as available to the model as are surface features of the text. For example, if a fine-tuned model can access features about a word's semantic role as easily as it can access features about that word's lexical identity, the model may need little (or no explicit) training signal in order to prefer a decision rule based on the former structural feature. The desire for models with such behavior motivates the development of architectures with explicit inductive biases (e.g., TreeRNNs). Evidence that similar generalization behavior can result from pre-trained representations has exciting implications for those interested in sample efficiency and cognitively-plausible language learning (Warstadt & Bowman, 2020; Linzen, 2020). We note that this work has not established that the relationship between extractability and feature use is causal. This could be explored, for example, using intermediate task training (Pruksachatkun et al., 2020) in order to influence the extractability of features prior to fine-tuning for the target task. Recent work suggests, e.g., that fine-tuning on parsing might improve the extractability of syntactic features (Merchant et al., 2020).

6 Related Work

Significant prior work analyzes the representations and behavior of pre-trained LMs. Work using probing classifiers (Veldhoen et al., 2016; Adi et al., 2017; Conneau et al., 2018; Hupkes et al., 2018) has suggested that such models capture a wide range of relevant linguistic phenomena (Hewitt & Manning, 2019; Bau et al., 2019; Dalvi et al., 2019; Tenney et al., 2019a;b). Other techniques in this vein include attention maps/visualizations (Voita et al., 2019; Serrano & Smith, 2019), and relational similarity analyses (Chrupała & Alishahi, 2019). A parallel line of work uses challenge sets to understand model behavior in practice. Some works construct evaluation sets to analyze weaknesses in the decision procedures of neural NLP models (Jia & Liang, 2017b; Glockner et al., 2018b; Dasgupta et al., 2018; Gururangan et al., 2018; Poliak et al., 2018b; Elkahky et al., 2018; Ettinger et al., 2016; Linzen et al., 2016b; Isabelle et al., 2017; Naik et al., 2018; Jia & Liang, 2017a; Linzen et al., 2016a; Goldberg, 2019, and others). Others use such datasets to improve models' handling of linguistic features (Min et al., 2020; Poliak et al., 2018a; Liu et al., 2019a), or to mitigate biases (Zmigrod et al., 2019; Zhao et al., 2018; 2019; Hall Maudslay et al., 2019; Lu et al., 2018). Nie et al. (2020) and Kaushik et al. (2020) explore augmenting training sets with a human-in-the-loop methods.

Our work is related to work on generalization of neural NLP models. Geiger et al. (2019) discusses ways in which evaluation tasks should be sensitive to models' inductive biases and Warstadt & Bowman (2020) discusses the ability of language model pre-training to encode such inductive biases. Work on data augmentation (Elkahky et al., 2018; Min et al., 2020; Zmigrod et al., 2019) is relevant, as the approach relies on the assumption that altering the training data distribution (analogous to what we call s-only rate in our work) will improve model behavior in practice. Kodner & Gupta (2020); Jha et al. (2020) discuss concerns about ways in which such approaches can be counterproductive, by introducing new artifacts. Work on adversarial robustness (Ribeiro et al., 2018; Iyyer et al., 2018; Hsieh et al., 2019; Jia et al., 2019; Alzantot et al., 2018; Hsieh et al., 2019; Ilyas et al., 2019; Madry et al., 2017; Athalye et al., 2018) is also relevant, as it relates to the influence of dataset artifacts on models' decisions. A still larger body of work studies feature representation and generalization in neural networks outside of NLP. Mangalam & Prabhu (2019) show that neural networks learn "easy" examples (as defined by shallow machine learning model performance) before they learn "hard" examples. Zhang et al. (2016) and Arpit et al. (2017) show that neural networks which are capable of memorizing noise nonetheless acheive good generalization performance, suggesting that such models might have an inherent preference to learn more general features. Finally, ongoing theoretical work characterizes the ability of over-parameterized networks to generalize in terms of complexity (Neyshabur et al., 2019) and implicit regularization (Blanc et al., 2019).

7 CONCLUSION

This work bears on an open question in NLP, namely, the question of how models' internal representations (as measured by probing classifiers) influence model behavior (as measured by challenge sets). We find that the feature extractability can be viewed as an inductive bias: the more extractable a feature is after pre-training, the less statistical evidence is required in order for the model to adopt the feature during fine-tuning. Understanding the connection between these two measurement techniques can enable more principled evaluation of and control over neural NLP models.

REFERENCES

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. Fine-grained Analysis of Sentence Embeddings Using Auxiliary Prediction Tasks. In *International Conference on Learning Representations (ICLR)*, 2017.

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. *arXiv preprint arXiv:1804.07998*, 2018.

Devansh Arpit, Stanislaw Jastrzkebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 233–242. JMLR.org, 2017. URL http://dl.acm.org/citation.cfm?id=3305381.3305406.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1z-PsR5KX.

Yonatan Belinkov and James Glass. Analysis Methods in Neural Language Processing: A Survey. *Transactions of the Association for Computational Linguistics*, 7:49–72, March 2019. doi: 10.1162/tacl_a_00254. URL https://www.aclweb.org/anthology/Q19-1004.

Yonatan Belinkov, Sebastian Gehrmann, and Ellie Pavlick. Interpretability and analysis in neural NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pp. 1–5, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-tutorials.1. URL https://www.aclweb.org/anthology/2020.acl-tutorials.1.

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. *arXiv preprint arXiv:1904.09080*, 2019.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Grzegorz Chrupała and Afra Alishahi. Correlating neural and symbolic representations of language. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2952–2962, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1283. URL https://www.aclweb.org/anthology/P19-1283.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 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)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://www.aclweb.org/anthology/P18-1198.

Robin Cooper, Dick Crouch, Jan van Eijck, Chris Fox, Josef van Genabith, Jan Jaspars, Hans Kamp, David Milward, Manfred Pinkal, Massimo Poesio, Steve Pulman, Ted Briscoe, Holger Maier, and Karsten Konrad. Using the framework. Technical report, The FraCaS Consortium, 1996.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, D. Anthony Bau, and James Glass. What Is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, January 2019.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ali Elkahky, Kellie Webster, Daniel Andor, and Emily Pitler. A challenge set and methods for noun-verb ambiguity. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2562–2572, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1277. URL https://www.aclweb.org/anthology/D18-1277.

Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 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*, pp. 134–139. Association for Computational Linguistics, 2016. doi: 10.18653/v1/W16-2524. URL http://www.aclweb.org/anthology/W16-2524.

WA Falcon. Pytorch lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning*, 3, 2019.

Atticus Geiger, Ignacio Cases, Lauri Karttunen, and Christopher Potts. Posing fair generalization tasks for natural language inference. 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)*, pp. 4484–4494, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1456. URL https://www.aclweb.org/anthology/D19-1456.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 650–655. Association for Computational Linguistics, 2018a. URL http://aclweb.org/anthology/P18-2103.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking NLI Systems with Sentences that Require Simple Lexical Inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 650–655. Association for Computational Linguistics, 2018b. URL http://aclweb.org/anthology/P18-2103.

Yoav Goldberg. Assessing bert's syntactic abilities. *arXiv preprint arXiv:1901.05287*, 2019.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*, 2018.

Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. It's all in the name: Mitigating gender bias with name-based counterfactual data substitution. 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)*, pp. 5266–5274, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1530. URL https://www.aclweb.org/anthology/D19-1530.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://www.aclweb.org/anthology/N19-1419.

Yu-Lun Hsieh, Minhao Cheng, Da-Cheng Juan, Wei Wei, Wen-Lian Hsu, and Cho-Jui Hsieh. On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1520–1529, 2019.

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

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pp. 125–136, 2019.

Pierre Isabelle, Colin Cherry, and George Foster. A Challenge Set Approach to Evaluating Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2486–2496. Association for Computational Linguistics, 2017. URL http://aclweb.org/anthology/D17-1263.

Mohit Iyyer, John Wieting, Kevin Gimpel, and Luke Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. *arXiv preprint arXiv:1804.06059*, 2018.

Rohan Jha, Charles Lovering, and Ellie Pavlick. When does data augmentation help generalization in nlp? *arXiv preprint arXiv:2004.15012*, 2020.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031. Association for Computational Linguistics, 2017a. URL http://aclweb.org/anthology/D17-1215.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031. Association for Computational Linguistics, 2017b.

Robin Jia, Aditi Raghunathan, Kerem Göksel, and Percy Liang. Certified robustness to adversarial word substitutions. *arXiv preprint arXiv:1909.00986*, 2019.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklgs0NFvr.

Jordan Kodner and Nitish Gupta. Overestimation of syntactic representation in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1757–1762, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.160. URL https://www.aclweb.org/anthology/2020.acl-main.160.

Tal Linzen. How can we accelerate progress towards human-like linguistic generalization? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5210–5217, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.465. URL https://www.aclweb.org/anthology/2020.acl-main.465.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016a. doi: 10.1162/tacl_a_00115. URL https://www.aclweb.org/anthology/Q16-1037.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016b. URL http://aclweb.org/anthology/Q16-1037.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. Inoculation by fine-tuning: A method for analyzing challenge datasets. 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)*, pp. 2171–2179, Minneapolis, Minnesota, June 2019a. Association for Computational Linguistics. doi: 10.18653/v1/N19-1225. URL https://www.aclweb.org/anthology/N19-1225.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019b.

Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. Gender bias in neural natural language processing. *arXiv preprint arXiv:1807.11714*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Karttikeya Mangalam and Vinay Uday Prabhu. Do deep neural networks learn shallow learnable examples first? 2019.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1192–1202, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1151. URL https://www.aclweb.org/anthology/D18-1151.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. What happens to bert embeddings during fine-tuning? 2020.

Junghyun Min, R. Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. Syntactic data augmentation increases robustness to inference heuristics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Seattle, Washington, July 2020. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. Stress Test Evaluation for Natural Language Inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2340–2353. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/C18-1198.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BygfghAcYX.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.441. URL https://www.aclweb.org/anthology/2020.acl-main.441.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 67–81, Brussels, Belgium, October-November 2018a. Association for Computational Linguistics. doi: 10.18653/v1/D18-1007. URL https://www.aclweb.org/anthology/D18-1007.

Adam Poliak, Jason Naradowsky, Aparajita Haldar, Rachel Rudinger, and Benjamin Van Durme. Hypothesis only baselines in natural language inference. *arXiv preprint arXiv:1805.01042*, 2018b.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R. Bowman. Intermediate-task transfer learning with pretrained language models: When and why does it work? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5231–5247, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.467. URL https://www.aclweb.org/anthology/2020.acl-main.467.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 856–865, 2018.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 8–14, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2002. URL https://www.aclweb.org/anthology/N18-2002.

Sofia Serrano and Noah A. Smith. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282. URL https://www.aclweb.org/anthology/P19-1282.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

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. What do you learn from context? Probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=SJzSgnRcKX.

Sara Veldhoen, Dieuwke Hupkes, and Willem Zuidema. Diagnostic Classifiers: Revealing how Neural Networks Process Hierarchical Structure. In *CEUR Workshop Proceedings*, 2016.

Elena Voita and Ivan Titov. Information-theoretic probing with minimum description length. *arXiv preprint arXiv:2003.12298*, 2020.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1580. URL https://www.aclweb.org/anthology/P19-1580.

Alex Warstadt and Samuel R Bowman. Grammatical analysis of pretrained sentence encoders with acceptability judgments. *arXiv preprint arXiv:1901.03438*, 2019.

Alex Warstadt and Samuel R. Bowman. Can neural networks acquire a structural bias from raw linguistic data?, 2020.

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392, 2020.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. URL http://arxiv.org/abs/1611.03530.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 15–20, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2003. URL https://www.aclweb.org/anthology/N18-2003.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. Gender bias in contextualized word embeddings. *arXiv preprint arXiv:1904.03310*, 2019.

Ran Zmigrod, Sebastian J. Mielke, Hanna Wallach, and Ryan Cotterell. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1651–1661, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1161. URL https://www.aclweb.org/anthology/P19-1161.