PREDICTING INDUCTIVE BIASES OF PRE-TRAINED MODELS

Charles Lovering[†], **Rohan Jha**[†], **Tal Linzen**[§], **Ellie Pavlick**[†]

[†] Brown University
Department of Computer Science
{charles_lovering@, rohan_jha@alumni., ellie_pavlick@} brown.edu

[§] New York University
Department of Linguistics and Center for Data Science
linzen@nyu.edu

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 naturalistic 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. Still, our definition of a feature accommodates multiple spurious or target features. In fact, some of our spurious features already encompass multiple features: the lexical feature, for example, is a combination of several individual-word features because it holds if one of a set of words is in the sentence. This type of spurious feature is common in real datasets: E.g., the hypothesis-only baseline in NLI is a disjunction of lexical features (with semantically unrelated words like "no", "sleeping", etc.) (Poliak et al., 2018b; Gururangan et al., 2018).

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 (Elkahky et al., 2018; Zmigrod et al., 2019; Min et al., 2020).

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.

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 clarity, we sometimes drop the S_* notation (e.g., s-only in place of $S_{s\text{-only}}$).

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

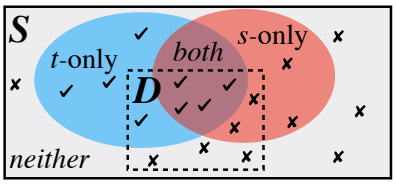


Figure 1: We partition datasets into four sections, defined by the features (spurious and/or target) that hold. We sample training datasets D, which provide varying amounts of *evidence* against the spurious feature, in the form of s-only examples. In the illustration above, the s-only rate is $\frac{2}{10} = 0.2$, 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 $= |D_{s\text{-only}}| / |D|$, 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)** Rissanen (1978)–first used by Voita & Titov (2020) for probing–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]

[1] Without loss of generality, we define t in our datasets s.t. $t(x) = y, \forall x, y \in S$. We do this to iron out the case where t outputs the opposite value of y.

[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

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 $\text{MDL}(s)/\text{MDL}(t)$, will yield models that achieve better performance despite less training evidence.

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 ($\text{MDL}(s)/\text{MDL}(t)$) and model performance, 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.

4 EXPERIMENTS WITH NATURALISTIC DATA

We investigate whether the same trend holds for language models fine-tuned with naturalistic data, e.g grammar-generated English sentences. 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 formal linguistic theory guides how we define the target features, as well as

longer blocks, using classifiers that are trained on previously transmitted data. The high MDL's are a result of overfitting by classifiers that are trained on limited data–and therefore, the classifiers have worse compression performance than the uniform baseline.

[4]Note, all models are ultimately able to learn to detect t (achieve high test accuracy) on the both partition, but not on the t-only partition.

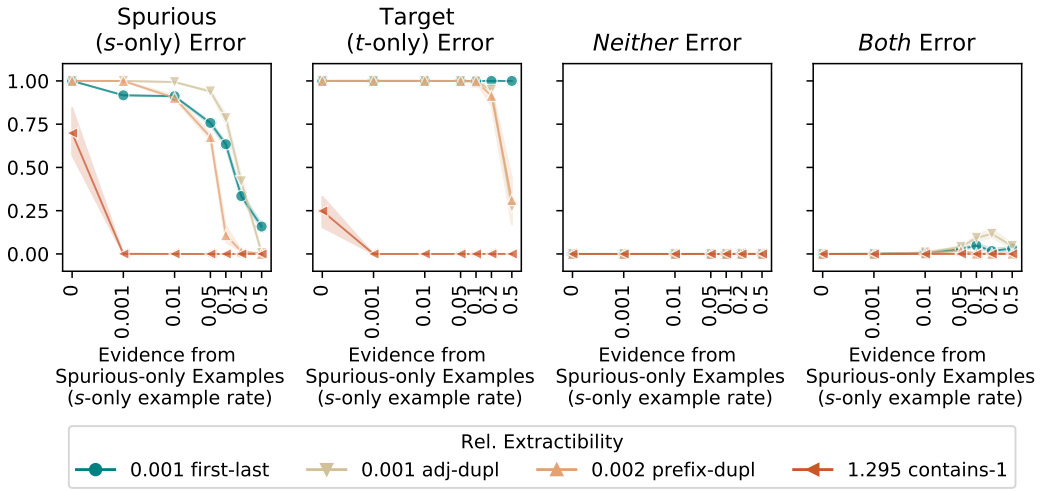


Figure 2: **Results on Synthetic Data**. Error on each partition of the test set, as a function of s-only rate. A model that has learned to use the target feature alone to predict the label will achieve zero error across all partitions. s-only and t-only error reach 0 quickly when t is as easy to extract as s (i.e., the relative 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.

recent work in computational linguistics showing that neural language models can be sensitive to spurious features in this task (Marvin & Linzen, 2018; Warstadt et al., 2020a).

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 ($\text{MDL}(s)/\text{MDL}(t)$). 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, grammatical sentences begin with specific lexical items (e.g., *"often"*); 2) length, grammatical sentences are longer; 3) recent-noun, verbs in grammatical sentences agree with the immediately preceding noun (in addition to their subject); and 4) plural,- 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 evaluate T5, BERT, RoBERTa, GPT-2 and an LSTM with GloVe embeddings (Raffel et al., 2019; Devlin et al., 2018; Liu et al., 2019b; Radford 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).

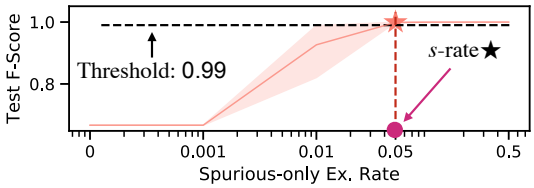
4.3 Results

For each (s, t) feature pair, 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**$\star$ as the lowest s-only example rate at which the fine-tuned model achieves 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 infer that 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 ($\rho = -0.82$) and T5 ($\rho = -0.67$). That is, the more extractable t is relative to s, the

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

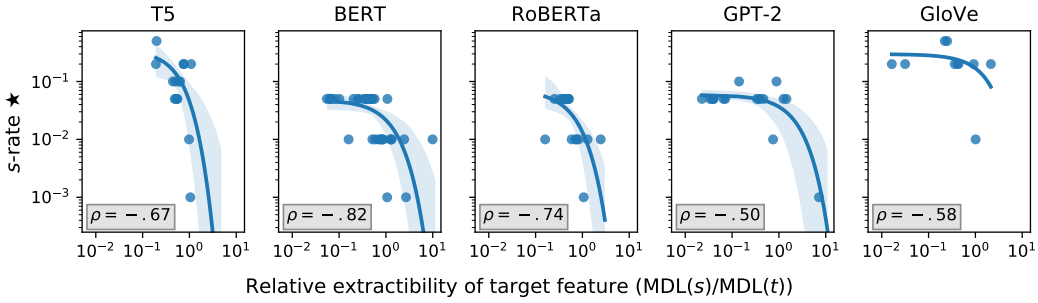
[6] See Appendix for *both* error and *neither* error; both are stable and low in general.



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

	Absolute		Relative (t to s)	
	Target	Spurious	Ratio	Difference
T5	**0.48**	-0.27	**-0.67**	**-0.74**
BERT	0.44	**-0.71**	**-0.82**	**-0.84**
RoBERTa	0.14	**-0.73**	**-0.74**	**-0.73**
GPT2	-0.08	**-0.45**	**-0.50**	**-0.52**
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 the relative extractability of s, t via (MDL(s)/MDL(t)).

Figure 3: **Relative Extractability Correlates with Adoption of Target Feature**. (a) An illustrative chart of how s-rate $\star$ is determined: Intuitively, it the amount of evidence required before a fine-tuned model adopts the target feature. (b) Spearman's ρ between s-rate$\star$ and various measures of extractability over the (s, t) pairs. Bold indicates a significant correlation. Relative extractability, whether ratio (MDL(s)/MDL(t)) or difference (MDL(s) − MDL(t)) explains learning behavior better than absolute extractability of either feature. (c) The logistic regression plots between s-rate$\star$ and extractability of t relative to s via the ratio (MDL(s)/MDL(t)). The Spearman's ρ of tbe correlation between the ratio and s-rate $\star$ is also detailed in the bottom-left corner of each plot.

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 55% (11/20) feature pairs. This partially explains why the GloVe model results are less clear.

Figure 3c shows that T5 (compared to BERT, GPT2, and RoBERTa) requires more data (a higher s-rate $\star$) to perform well. We believe that this may be because we fine-tuned T5 with a linear classification head, rather than only-text input and output it was pre-trained on. We made this decision (1) because we had trouble training T5 in the original manner, and (2) using a linear classification head was consistent with the other model architectures.

Figure 4 shows the performance curves for T5 (with BERT, GPT2, RoBERTA, and GloVe in Appendix A), 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). Note that each data point is the test performance on a dataset that varies by the amount of evidence from spurious-only examples along the x-axis (with

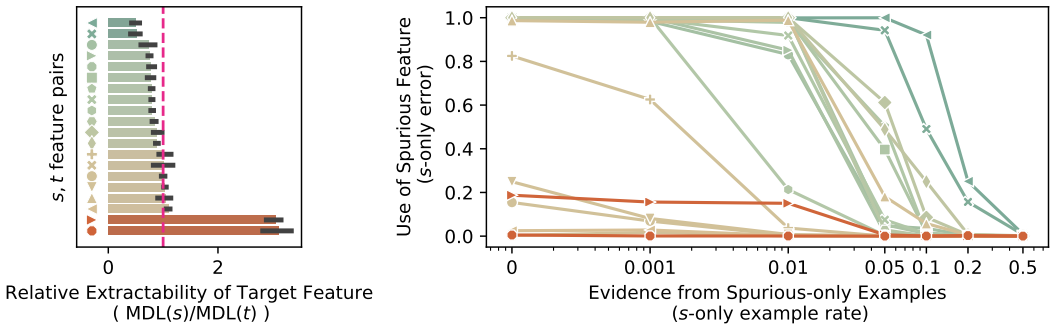


Figure 4: **Learning Curves for T5.** Curves show use of spurious feature (s-only accuracy) as a function of training evidence (s-only rate). Each line represents one (t, s) pair (described in §4.1). Pairs vary in the relative extractability of t vs. s (measured by the ratio $\text{MDL}(s)/\text{MDL}(t)$ 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 BERT, GPT2 and RoBERTa; see Appendix A. GloVe has difficulty learning the features, and the relationship is less clear.

each line corresponding to a different s, t feature pair.) 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 Discussion

Our experimental results provide support for our hypothesis: 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. 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). Our findings additionally connect to new probing techniques which have received significant attention in NLP but have yet to be connected to explanations of or predictions about state-of-the-art models' decisions in practice.

Fine-tuning may not uncover new features. The models are capable of learning both the s and t features in isolation, so our experiments show that if the relative extractibility is highly skewed, one feature may hide the other – a fine-tuned model may not use the harder-to-extract feature. Thus, if one classifier does not pick up on a feature readily enough, another classifier (or, rather, the same classifier trained with different data) may not be sensitive to that feature at all. This has ramifications for how we view fine-tuning, which is generally considered to be beneficial because it allows models to learn new, task-relevant features. Our findings suggest that if the needed feature is not already extractable-enough after pretraining, fine-tuning may not have the desired effect.

Probing classifiers can be viewed as measures of a pre-trained representation's inductive biases. Analysis with probing classifiers has primarily focused on whether important linguistic features can be decoded from representations at better-than-baseline rates, 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 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 result from pre-trained representations has exciting implications for those interested in sample ef-

ficiency and cognitively-plausible language learning (Warstadt & Bowman, 2020; Linzen, 2020). We note this work has not established that the relationship between extractability and feature use is causal. This could be explored using intermediate task training (Pruksachatkun et al., 2020) in order to influence the extractability of features prior to fine-tuning for the target task; Merchant et al. (2020) suggests that fine-tuning on parsing might improve the extractability of syntactic features.

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) suggests 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). Similar techniques 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 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).

Concurrent work (Warstadt et al., 2020b) also investigates the inductive biases of large pre-trained models (RoBERTa), in particular, they ask when (at what amount of pre-training data) such models shift from a surface feature (what we call spurious features) to a linguistic feature (what we call a target feature). In our work, we focus on how to predict which of these two biases characterize the model (via relative MDL).

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.

ACKNOWLEDGEMENTS

We would like to thank Michael Littman for helpful suggestions on how to better present our findings and Ian Tenney for insightful comments on a previous draft of this work. This work is supported by DARPA under grant number HR00111990064. This research was conducted using computational resources and services at the Center for Computation and Visualization, Brown University.

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.

Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

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, 2020a.

Alex Warstadt, Yian Zhang, Haau-Sing Li, Haokun Liu, and Samuel R Bowman. Learning which features matter: Roberta acquires a preference for linguistic generalizations (eventually). *arXiv preprint arXiv:2010.05358*, 2020b.

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.

PREDICTING INDUCTIVE BIASES OF PRE-TRAINED MODELS

A ADDITIONAL RESULTS

Figure 5, 6, 7 show additional results for all models – (*both* accuracy, *neither* accuracy, and F-score). These charts appear at the end of the Appendix. We also find similar results with GPT2 and RoBERTA, shown in Figures 8, 9.

Details on the MDL statistics are available in Table 3.

	MDL(t)				
	BERT	GPT2	GloVe	RoBERTa	T5
gap_base_length	443	614	2921	21	537
gap_base_lexical	223	337	1815	17	323
gap_base_plural	299	346	1664	14	401
gap_base_tense	278	417	1962	16	373
gap_hard_length	421	479	2845	18	493
gap_hard_lexical	292	341	1846	17	321
gap_hard_none	116	77	946	8	182
gap_hard_plural	322	440	1862	15	387
gap_hard_tense	354	353	1685	15	332
npi_length	300	240	952	16	273
npi_lexical	225	345	1249	12	245
npi_plural	292	335	1267	15	282
npi_tense	220	258	999	13	286
sva_base_agreement	113	463	1581	12	149
sva_base_lexical	150	629	5872	15	154
sva_base_plural	93	428	5122	14	166
sva_hard_agreement	173	579	1582	14	176
sva_hard_length	183	708	1466	17	197
sva_hard_lexical	181	684	1467	16	189
sva_hard_plural	192	698	1449	18	201
	MDL(s)				
gap_base_length	28	23	92	3	103
gap_base_lexical	297	425	452	12	241
gap_base_plural	3421	3794	1906	45	1924
gap_base_tense	159	141	710	9	235
gap_hard_length	42	16	46	4	98
gap_hard_lexical	313	493	410	12	246
gap_hard_none	144	85	957	9	179
gap_hard_plural	3200	3161	3987	37	1795
gap_hard_tense	131	131	694	7	198
npi_length	78	214	499	8	297
npi_lexical	46	48	475	6	123
npi_plural	15	22	1165	6	125
npi_tense	14	18	435	5	150
sva_base_agreement	305	348	1879	16	198
sva_base_lexical	12	14	162	5	85
sva_base_plural	77	199	535	11	174
sva_hard_agreement	319	835	1314	15	230
sva_hard_length	94	28	465	8	103
sva_hard_lexical	12	27	891	6	92
sva_hard_plural	168	274	695	14	256

Table 3: Summary of extractability (MDL in bits) for t and s for each template and each model.

B Implementation Details & Reproduciblity

Our code is available at:
`https://github.com/cjlovering/predicting-inductive-biases`.

There are two major parts to this project in terms of reproducibility: (1) the data and (2) the model implementations. We describe the templates for the data below in Appendix E – the full details are in the project source. For the transformer models, we use Hugging Face for the implementations and access to the pre-trained embeddings (Wolf et al., 2019). We use PyTorch Lightning to organize the training code (Falcon, 2019). We fix all hyperparameters, which are reported in Table 4.

We want to call attention to BERT requiring much less data than T5 to capture our target features. At face value, it seems that BERT requires much less data than T5 to capture our target features. However, we are wary about making such strong claims. Something to consider here (noted in the Appendix B is that for T5 we used a linear model rather than formatting the task in text (which is how T5 is trained). We made this decision (1) because we had trouble training T5 in this purely textual manner, and (2) using a linear classification head over two classes is consistent with the other model architectures. Again, GPT2 and RoBERTa performed on par with BERT, so the difference between the performance of BERT and T5 may be due to how we trained T5.

Hyperparameter	**Value**
Hyperparameters	
random seed	1, 2, 3
batch size	128
cumulative mdl block sizes (%)	0.1, 0.1, 0.2, 0.4, 0.8, 1.6, 3.05, 6.25, 12.5, 25
s-only rates (%)	0, 0.1, 1, 5, 10, 20, 50
T5 Hyperparameters	
model keycode	t5-base
model architecture	T5Model
warmup	linear (default values)
optimizer	AdamW
pooler	last hidden state
lr	2e-5
batch size	128
classifier head	Linear
BERT/RoBERTa/GPT2 Hyperparameters	
model keycode	bert-base-uncased
model architecure	BertForSequenceClassification
warmup	cosine (default values)
optimizer	AdamW
pooler	first hidden state
lr	2e-5
batch size	128
classifier head	MLP
LSTM (Toy and GloVe) Hyperparameters	
optimizer	Adam
pooler	last hidden state
lr	1e-3
batch size	128
classifier head	MLP with 1 hidden-layer tanh activation
hidden size	300

Table 4: **Hyper and System Parameters.** We use Hugging Face for the underlying model implementations.

	Absolute		Relative (t to s)	
	Target	Spurious	Ratio	Diff.
T5	**0.49**	**-0.53**	**-0.81**	**-0.88**
BERT	**0.47**	**-0.62**	**-0.64**	**-0.57**
GPT2	0.1	**-0.47**	**-0.56**	**-0.54**
RoBERTa	0.04	**-0.72**	**-0.72**	**-0.67**
GloVe	0.05	0.06	0.02	-0.07

Table 5: AUC. Each column presents the Spearman correlation between the given measure and the s-only rate at which the model first acquired the target feature. * indicates a significant correlation. A feature is considered acquired when its test-performance is above 0.99 F-score. All (s, t) pairs (20 of 20 each) were eventually learned by T5, BERT, GPT2, and RoBERTa; 11 of 20 (s, t) pairs were learned by the GloVe-based LSTM.

C Results using AUC instead of MDL

See Table 5. A metric similar to MDL for capturing the same intuition is the ***area under the validation loss curve (AUC)***. This metric is highly related to online MDL in computation. We find it sometimes produces different results in practice, so we consider both metrics in our analysis of extractability.

D Measuring Extractability Indirectly

We measure the MDL for t with *both* and s-only examples. In the simulated setting we can compare this approach with measuring the MDL directly (t-only vs *neither*). See Table 6 for MDL results. The ordering of the feature's difficulty holds across the two methods.

Feature	Probe	MDL in k-bits
adj-dupl	(s-only vs *both*)	242.19
	(t-only vs *neither*)	248.68
contains-1	(s-only vs *both*)	0.29
	(t-only vs *neither*)	0.34
first-last	(s-only vs *both*)	397.64
	(t-only vs *neither*)	607.46
prefix-dupl	(s-only vs *both*)	175.74
	(t-only vs *neither*)	193.92

Table 6: We measure the target MDL directly on the toy data, where we can access the target feature. Recall that in natural conditions we can not generate examples with the target feature (free of spurious features), so we used a dataset comprising *both* and s-only examples. In the simulated setting, our approach for measuring the target extractability indirectly by using *both* and s-only examples reports results similar to those when directly using *target* and *neither*. These values are in $k$bits.

E Templates for Naturalistic Data

Each template corresponds to a combination of target features, grammars, and spurious features (the target and spurious features are discussed in Section 4.1). See Table 7 for a complete list of templates. See Table 9 for further details about the templates that are used for each of the the target features. Complete details about implementation of these templates (and all data) will be released upon acceptance.

Target feature	Spurious features	grammar	#
SVA	closest-noun, length[1], lexical, plurality	base, nested	7
NPI	length, lexical, plurality, tense	base	4
GAP	length, lexical, plurality, tense	base, islands[2]	9

[1]Length is not applicable for the base template

[2]We also run the islands template without additional spurious features.

Table 7: List of templates that are used in Section 4. These are discussed in greater detail below.

Target	Spurious	Example
Subject agrees with verb	N before V is singular	[*neither*] * The piano teachers of the lawyers wounds the handyman.
NPI in down. -entailing context	Contains negation word	[*neither*] * The student who was wrong ever resigned.
Correct filler-gap dependency	Main verb is in past tense	[*neither*] * I know that he recognized __ yesterday.

Table 8: *neither* examples for Table 2.

Subcase	Template	Example
NPI ✓	No N_{p-neg} ever V.	No girl who was sad ever resigned.
NPI ✓	$Det\ N_p\ V$.	Some student who no girl ever followed ran.
NPI X	The N_p ever V.	*The man who some lawyer hated ever traveled.
NPI X	The $N_{p-ever}\ V$	*The boy who ever smiled shouted.
SVA ✓	The N_1^x of the (N_i of the)* N_2 V_1^x the N_3.	The piano teacher of the lawyers wounds the handyman.
SVA X	The N_1^x of the (N_i of the)* N_2 V_1^y the N_3.	*The piano teacher of the lawyers wound the handyman.
	Whether or not a noun and verb agree is given by whether their superscripts match.	
GAP ✓	(N_i V_i that)* N_1 V_1 who (N_i V_i that)* (N_i V_i).	I know who he believed.
GAP ✓	(N_i V_i that)$^{1+}$ N_2.	I know that he believed them.
GAP X	(N_i V_i that)* N_1 V_1 who (N_i V_i that)* N_1 V_1 N_2.	*I know who he believed them.
GAP X	(N_i V_i that)* N_1 V_1.	*I know that he believed.
ISL X	N_1 V_1 who N_2 V_2 N_4 that N_3 V_3.	*I know who he knew the lawyers that she believed.

Table 9: List of templates that are used in Section 4. These do not include the spurious features which are discussed in Section 4.1. For NPI, each N_* represents a noun phrase. N_{p-neg} is valid after a negation (might contain a polarity item). N_p is valid after a determiner (cannot contain an unlicensed polarity item). N_{p-ever} is not valid after a determiner (contains an unlicensed polarity item). These phrases have complex nesting behavior and can become arbitrary long. In addition, a sentence might consist of multiple independent clauses, each of which is given by one of these templates. For SVA, the base templates do not have the additional nouns in the starred parentheticals, while the nested templates have zero or more. For GAP, the harder set of templates include ISL (island) examples as an additional s-only example that force the model to not violate (one specific type) of island constraint. For complete details and lexicons see the source.

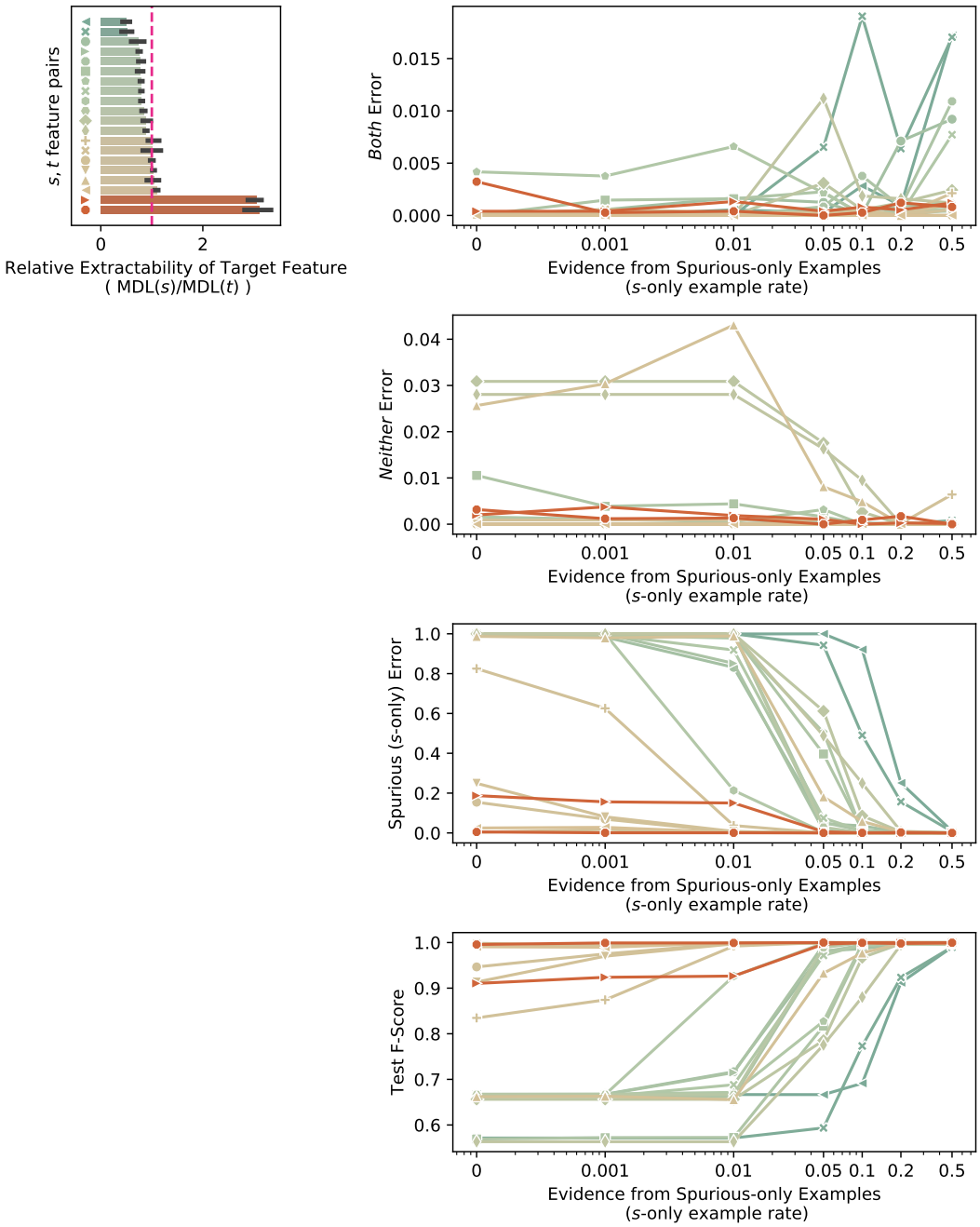


Figure 5: T5.

F MDL ISSUES: OVERFITTING IN THE SYNTHETIC EXPERIMENTS

We found that the MDL exceeds the uniform code length in some of the synthetic experiments. We found that this occurs because the model overfits on the small early-block sizes. See Figure 10.

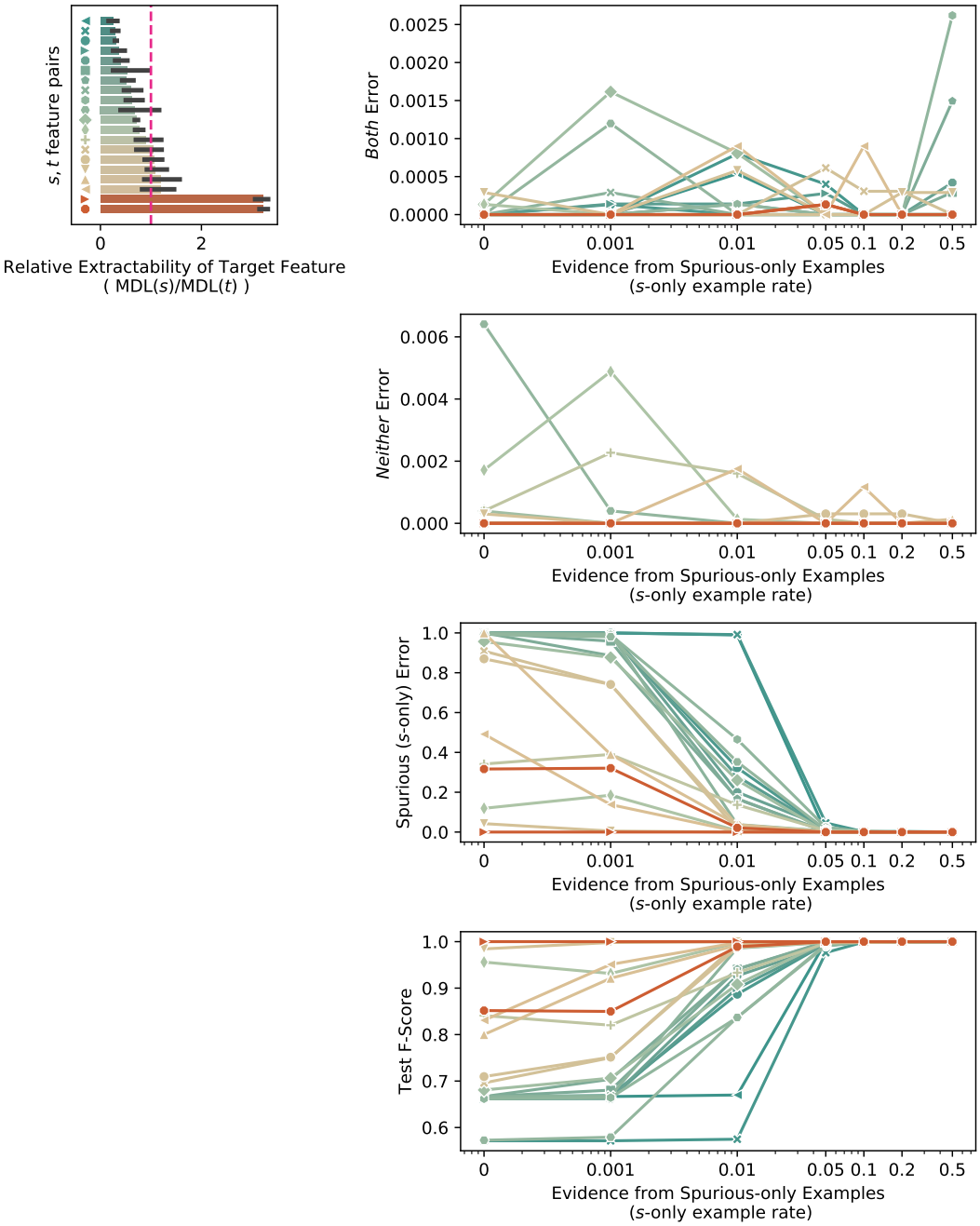


Figure 6: BERT.

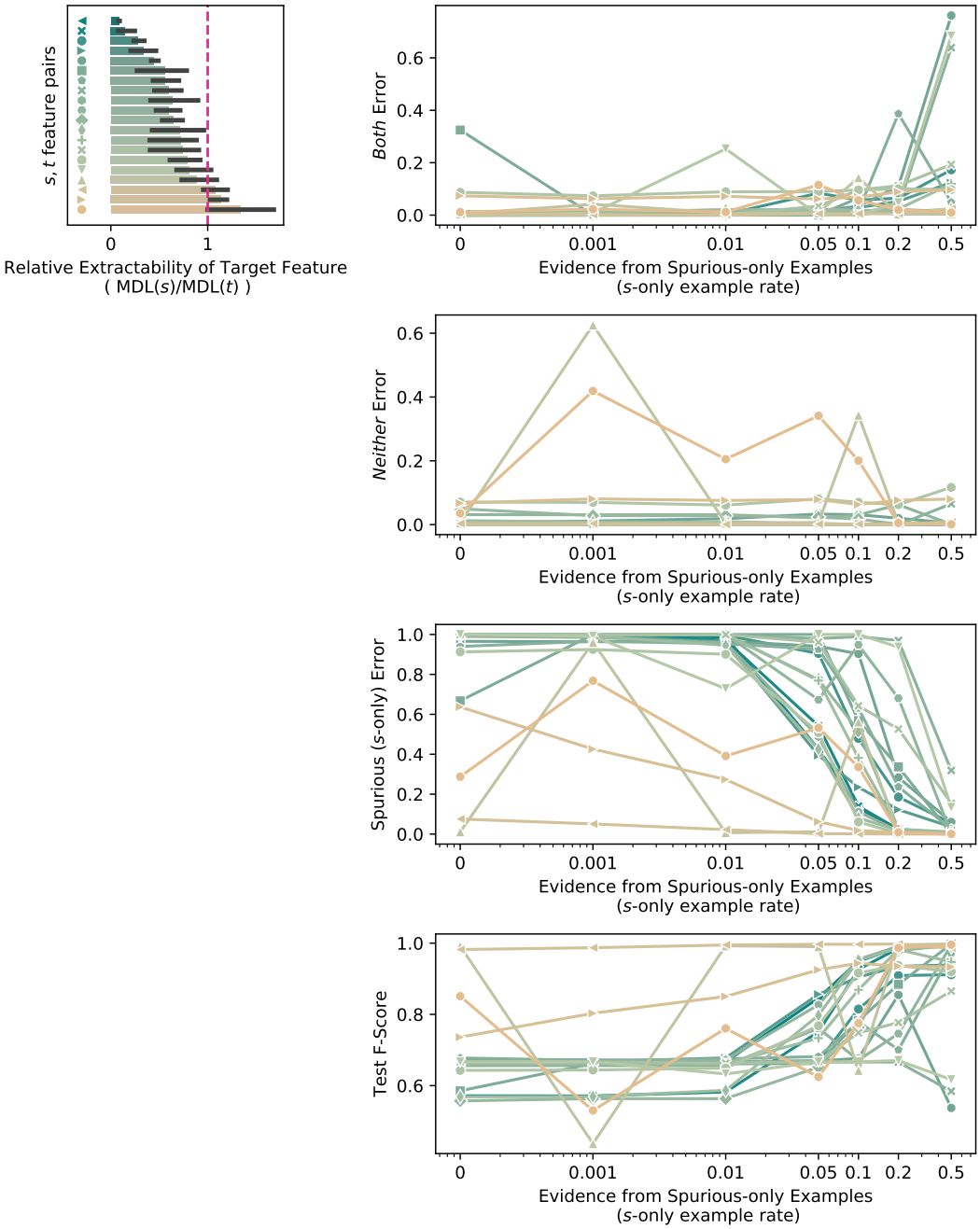


Figure 7: GloVe.

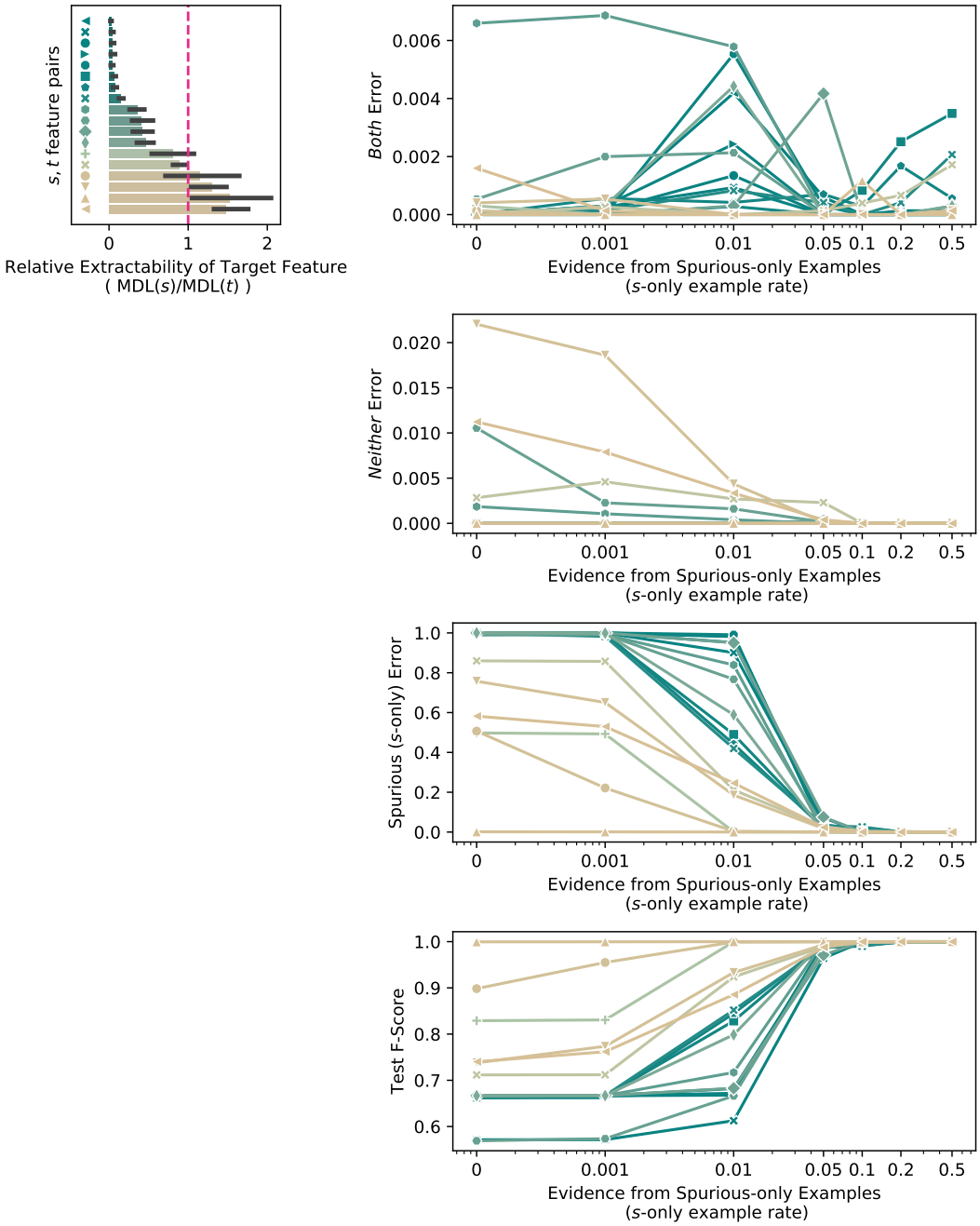


Figure 8: GPT2.

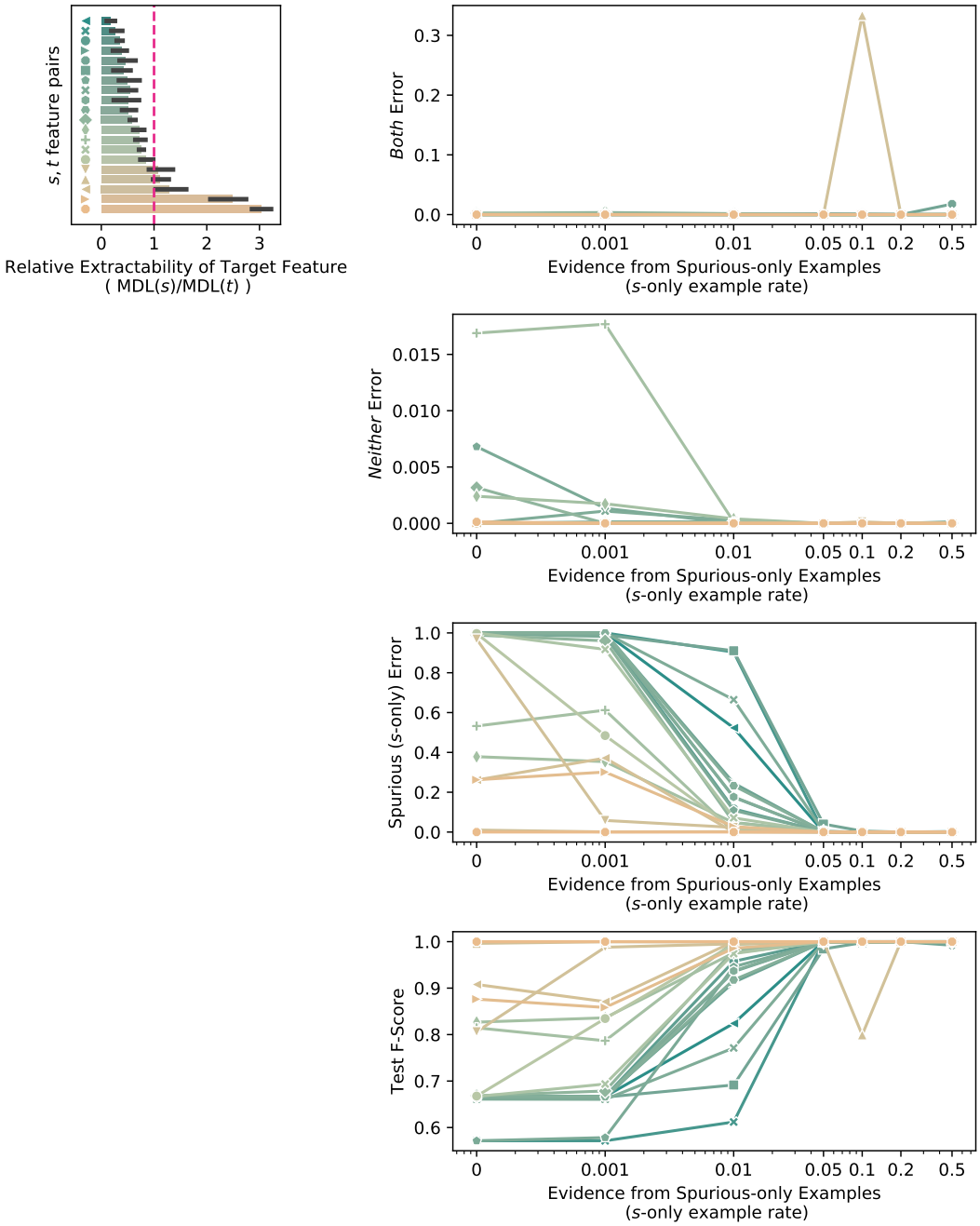


Figure 9: RoBERTa.

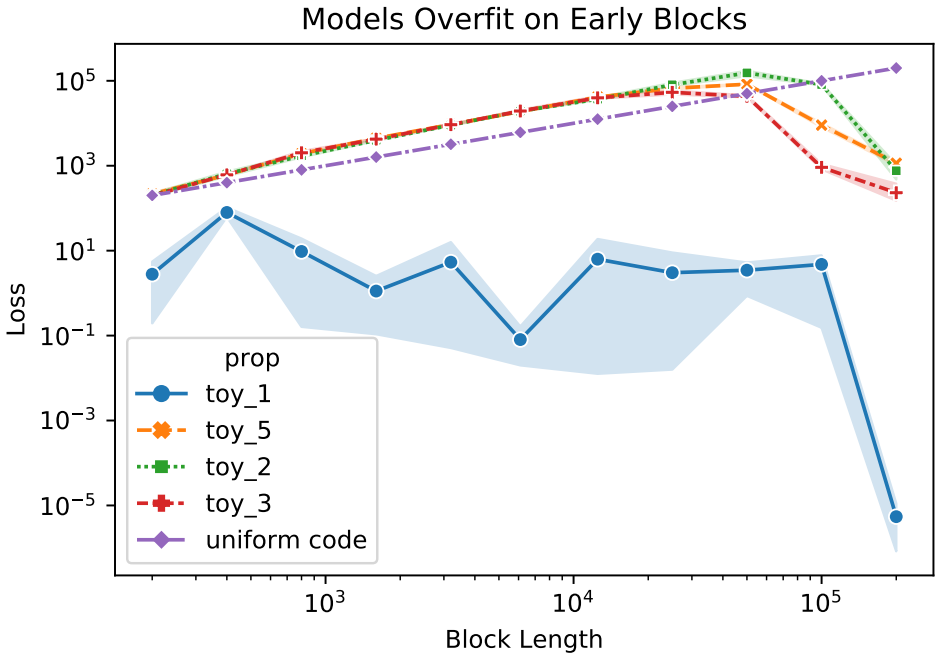


Figure 10: Overfitting on the Synthetic Tasks.