Learning Which Features Matter: RoBERTa Acquires a Preference for Linguistic Generalizations (Eventually)

Alex Warstadt,[1] **Yian Zhang,**[2] **Haau-Sing Li,**[3] **Haokun Liu,**[3] **Samuel R. Bowman**[1,2,3]
[1]Dept. of Linguistics, [2]Dept. of Computer Science, [3]Center for Data Science
New York University
Correspondence: warstadt@nyu.edu

Abstract

One reason pretraining on self-supervised linguistic tasks is effective is that it teaches models features that are helpful for language understanding. However, we want pretrained models to learn not only to represent linguistic features, but also to *use* those features preferentially during fine-turning. With this goal in mind, we introduce a new English-language diagnostic set called MSGS (the Mixed Signals Generalization Set), which consists of 20 ambiguous binary classification tasks that we use to test whether a pretrained model prefers linguistic or surface generalizations during fine-tuning. We pretrain RoBERTa models from scratch on quantities of data ranging from 1M to 1B words and compare their performance on MSGS to the publicly available RoBERTa$_{\text{BASE}}$. We find that models can learn to represent linguistic features with little pretraining data, but require far more data to learn to *prefer* linguistic generalizations over surface ones. Eventually, with about 30B words of pretraining data, RoBERTa$_{\text{BASE}}$ does demonstrate a linguistic bias with some regularity. We conclude that while self-supervised pretraining is an effective way to learn helpful inductive biases, there is likely room to improve the rate at which models learn which features matter.

1 Introduction

Self-supervised pretraining through language modeling on massive datasets has revolutionized NLP. One reason this method works is that pretraining shapes a model's hypothesis space, giving it inductive biases that help it learn linguistic tasks (Howard and Ruder, 2018). Numerous probing studies have provided support for this idea by showing that language models learn representations that encode linguistic features (Gulordava et al., 2019; Tenney et al., 2019; Hewitt and Manning, 2019).

However, feature learning is just the first step to acquiring helpful inductive biases. Models must also be able to learn which features matter. The NLU datasets these models are often fine-tuned on are ambiguous and contain artifacts, and often support multiple possible generalizations. Neural networks are not mind readers: Models that have been shown to represent linguistic features sometimes fail to use them during fine-tuning on NLU tasks, instead adopting shallow surface generalizations (Jia and Liang, 2017; McCoy et al., 2019). To this end, recent work in probing pretrained models advocates for shifting the focus of study away from whether they represent linguistic features and in favor of whether they learn *useful* representations of those features (Voita and Titov, 2020; Pimentel et al., 2020; Elazar et al., 2020).

We investigate how RoBERTa (Liu et al., 2019b) acquires language-specific inductive biases during self-supervised pretraining. We track separately

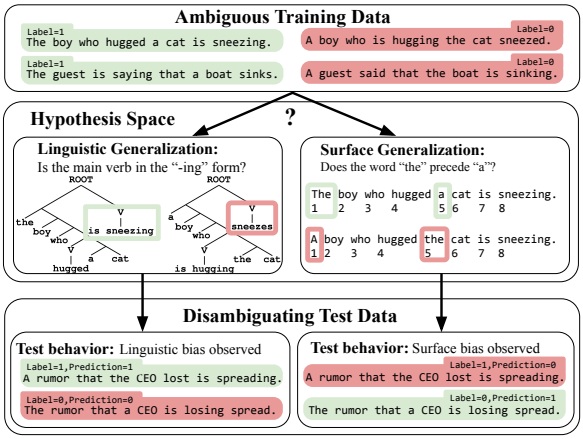

Figure 1: Example of an ambiguous experiment (without inoculation). A model is trained on ambiguous data whose labels are consistent with either a linguistic or a surface generalization, and tested on disambiguating data whose labels support only the linguistic generalization. Light green and dark red shading represents data or features associated with the positive and negative labels/predictions, respectively.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 217–235,
November 16–20, 2020.

	Feature type	Feature description	Positive example	Negative example
Surface	Absolute position	Is the first token of S "the"?	The cat chased a mouse.	A cat chased a mouse.
	Length	Is S longer than n (e.g., 3) words?	The cat chased a mouse.	The cat meowed.
	Lexical content	Does S contain "the"?	That cat chased the mouse.	That cat chased a mouse.
	Relative position	Does "the" precede "a"?	The cat chased a mouse.	A cat chased the mouse.
	Orthography	Does S appear in title case?	The Cat Chased a Mouse.	The cat chased a mouse.
Linguistic	Morphology	Does S have an irregular past verb?	The cats slept.	The cats meow.
	Syn. category	Does S have an adjective?	Lincoln was tall.	Lincoln was president.
	Syn. construction	Is S the control construction?	Sue is eager to sleep.	Sue is likely to sleep.
	Syn. position	Is the main verb in "ing" form?	Cats who eat mice are purring.	Cats who are eating mice purr.

Table 1: Schematic examples of the linguistic and surface features in our experiments.

how RoBERTa's representation of linguistic features and its preferences for linguistic generalizations over surface generalizations change as the amount of pretraining data increases. We pretrain RoBERTa from scratch on datasets ranging from 1M to 1B words and evaluate these models alongside $\text{RoBERTa}_{\text{BASE}}$ in a series of experiments to probe the inductive biases of a pretrained model at the time of fine-tuning on a downstream task.

We probe these models in three kinds of experiments: First, we conduct *control* experiments where we fine-tune models on unambiguous binary classification tasks to test whether they learn to represent simple linguistic and surface features. Second, we conduct *ambiguous* experiments following the *poverty of the stimulus* design (Wilson, 2006), as illustrated in Figure 1. In these experiments, we fine-tune a pretrained model on an ambiguous binary classification task in which the training set is consistent with both a linguistic generalization and a surface one. We then test the classifier on disambiguating data to reveal which generalization the model adopted, and by extension its preference among the two features. Third, we conduct *inoculation* experiments (following Liu et al., 2019a) to test how hard it is to sway a model with a surface bias to adopt a linguistic generalization. We do this by introducing small amounts of disambiguating data into an otherwise ambiguous training set. We automatically generate data for all these tasks, and call the resulting dataset MSGS (Mixed Signals Generalization Set), pronounced "messages".

The results show that RoBERTa acquires a stronger linguistic bias as pretraining increases. $\text{RoBERTa}_{\text{BASE}}$ has the strongest linguistic bias, and requires little to no inoculating data to reliably make the linguistic generalization. In general, models with more pretraining data can generally be induced to adopt linguistic generalizations with less inoculating data. We also find a large gap between the amount of pretraining data that RoBERTa needs to learn the linguistic features necessary to generalize out-of-domain and the amount it needs to learns that it should *prefer* those features when generalizing. The control experiments on unambiguous data reveal that models with little pretraining do actually represent the linguistic features, but nonetheless show a strong surface bias. In other words, the main contribution of pretraining to linguistic bias learning is devoted not to extracting features, but to learning which features matter.

We conclude that helpful inductive biases can be learned through pretraining, but current models require abundant data to do so. The implications of this conclusion point in two directions: First, we can probably continue to pretrain on increasingly massive training sets to improve on the generalization and few-shot learning abilities of models like T5 (Raffel et al., 2019) and GPT-3 (Brown et al., 2020). Second, since models learn useful features early, there is hope that future advances could accelerate by reducing the amount of data needed to learn which features matter. To aid in this effort, we release the MSGS dataset, our pretrained RoBERTas, and all our code: `https://github.com/nyu-mll/msgs`.

2 Inductive Bias

Background: Learning Inductive Bias Any finite set of training examples shown to a learning algorithm like a neural network is consistent with infinitely many generalizable decision functions. Inductive biases are a learner's preferences among these functions. An inductive bias can eliminate certain possible functions altogether, or result in a preference for some over others (Haussler, 1988). For instance, an RNN classifier is capable of representing *any* function, but prefers ones that focus

mostly on local relationships within the input sequence (Dhingra et al., 2018; Ravfogel et al., 2019).

Some recent work seeks to design neural architectures that build in desirable inductive biases (Dyer et al., 2016; Battaglia et al., 2018), or compares the immutable biases of different architectures (McCoy et al., 2020; Hu et al., 2020). However, inductive biases can also be *learned* by biological (Harlow, 1949) and artificial systems alike (Lake et al., 2017). In the language model finetuning paradigm proposed by Howard and Ruder (2018) and popularized by models such as BERT (Devlin et al., 2019), a pretrained neural network plays the role of the learner. Pretraining adjusts a model's weights so that it will navigate the hypothesis space during training on a downstream task more effectively than a randomly initialized model.

There is a difference between learning to extract a linguistic feature and acquiring a bias towards using it when generalizing. There is ample evidence that BERT encodes features such as syntactic category and constituency (Tenney et al., 2019; Clark et al., 2019; Hewitt and Manning, 2019). The acquisition of linguistic features is a *prerequisite* for a linguistic bias. However, these findings do not tell us if the model will make use of these features to form generalizations during target task training, or if it will fall back on surface features that account for most of the data.

Methods: Measuring Inductive Bias We conduct three kinds of experiments to probe a model's preference for linguistic or surface generalizations: unambiguous control experiments, fully ambiguous experiments, and partially ambiguous inoculation experiments. Figure 1 gives an overview of the ambiguous experiment design.

First, it only makes sense to compare a model's preference between two features if it actually represents both features. This is the goal behind *control experiments*, in which we fine-tune RoBERTa to classify sentences based on a single linguistic or surface feature in a totally unambiguous setting.

Second, we conduct *ambiguous experiments* on models that pass the controls. We fine-tune a pretrained model on a binary sentence classification task using *ambiguous* data, which equally supports both a simple linguistic generalization and a simple surface one. For example, Figure 1 shows a linguistic task where sentences in the positive class are defined by having a main verb in the "ing" form. We make the training data ambiguous by introducing a surface feature that distinguishes the two classes: In all (and only) training examples with label 1, the word "the" precedes the word "a". Based on this training data, a model could reasonably adopt either a linguistic generalization or a surface one.

We then test the classifier on disambiguating data to observe which generalization it made. In this kind of data, the labels align with the linguistic generalization, and contradict the surface one: For example, in Figure 1, "a" now always precedes "the" in the positive test examples with label 1. We quantify a model's inductive bias using a metric we call the *linguistic bias score* (LBS). We define LBS as the Matthews correlation between the model predictions and the labels on the disambiguating test set (Matthews, 1975). If LBS is 1, the learner shows a systematic linguistic bias. If LBS is -1, it shows a systematic surface bias. If LBS is 0, it shows neither bias.

Finally, while the fully ambiguous experiments probe models' biases in an idealized setting, training data in more naturalistic contexts often does contain some evidence supporting a linguistic generalization over a simple surface one. To simulate this, we also conduct a series of *inoculation experiments* (following Liu et al., 2019a), in which we introduce small amounts of disambiguating data into an otherwise ambiguous training set. For each experiment, we replace 0.1%, 0.3%, or 1% of the training data with examples that support the linguistic generalization and contradict the surface one. These experiments allow us to compare the strength of linguistic bias in models that show an overall surface bias: If two models adopt the surface generalization in the fully ambiguous case, we can still say that one has a stronger linguistic bias than the other if it requires less inoculation data to be swayed towards the linguistic generalization.

3 Evaluation Data

We introduce MSGS (Mixed Signals Generalization Set), pronounced "messages", a dataset we design to be used in poverty of the stimulus and inoculation experiments. With the goal of contrasting inductive biases that are helpful and harmful in most NLP applications, the tasks in MSGS test a model's preferences for generalizations based on linguistic or surface features.

Features under Study Table 3 illustrates the 4 linguistic features and 5 surface features we con-

Dom.	Split	$\mathbf{L_L}$	$\mathbf{L_S}$	Sentence
In	Train (Ambiguous)	1	1	These men weren't hating that this person who sang tunes destroyed the vase.
		0	0	These men hated that this person who sang tunes was destroying some vase.
	Inoc. (Disamb.)	1	0	These men weren't hating that this person who sang tunes destroyed some vase.
		0	1	These men hated that this person who sang tunes was destroying the vase.
Out	Test (Disamb.)	1	0	These reports that all students built that school were impressing some children.
		0	1	These reports that all students were building the school had impressed some children.
	Aux. (Ambiguous)	1	1	These reports that all students built the school were impressing some children.
		0	0	These reports that all students were building that school had impressed some children.

Table 2: A full paradigm from the SYNTACTIC POSITION × LEXICAL CONTENT task. $\mathbf{L_L}$ and $\mathbf{L_S}$ mark the presence of the linguistic feature (*Is the main verb in the "ing" form?*) and surface feature (*Does S contain "the"?*), respectively. *Dom.* is short for *domain*.

sider.[1] Each feature is meant to be representative of a broad category of features (e.g. morphological features), though the precise implementation of each feature is necessarily much narrower (e.g. *Does the sentence have an irregular past verb?*). Forming generalizations based on surface features entails knowledge of the identity of certain words (in our case, only "the" and "a"), the positional indices of words in the string, the total number of words in a string, or whether certain characters are lowercase or uppercase.[2] Forming generalizations based on linguistic features requires more abstract knowledge of tense and inflectional morphemes, parts of speech, the control construction,[3] and hierarchical syntactic structures, none of which are encoded in the surface string.

Dataset Structure MSGS contains 20 ambiguous binary classification tasks each gotten by pairing one of 4 linguistic features with one of 5 surface features. We write $\text{FEAT}_1 \times \text{FEAT}_2$ to denote a task that combines features FEAT_1 and FEAT_2. Each ambiguous dataset contains 50k sentences split into training, evaluation, and inoculation sets. MSGS also includes 9 unambiguous *control tasks*—one for each feature. Each control dataset contains 30k sentences split into training and evaluation sets.

For ambiguous tasks, we generate data in paradigms of 8 sentences following a $2 \times 2 \times 2$ design, as shown in Table 2. We vary the following three features: a binary linguistic feature, a binary surface feature, and the domain from which the sentence is sampled. We generate in-domain and out-of-domain sentences from different templates (see §3:Data Generation for more detail).

As shown in Table 2, we split the data into four contrasting pairs with different purposes: (1) *Training data* is ambiguous in-domain data makes up 99% to 100% of the training set. (2) *Inoculating data* is disambiguating in-domain data which makes up 0.1% to 1% of the training set in experiments with inoculation. We show the classifier only the linguistic label ($\mathbf{L_L}$) to nudge it towards adopting a linguistic generalization. (3) *Test data* is disambiguating out-of-domain data used to test whether the model adopted the linguistic or surface generalization. (4) *Auxiliary data* is ambiguous out-of-domain data used to test how well the model adapts to the out-of-domain templates, regardless of which generalization it makes.

For control tasks, we generate data in paradigms of 4 sentences following a 2×2 design by varying the feature and domain. We use control tasks to test whether each pretrained model represents each feature well enough to fine-tune an effective classifier in an unambiguous setting.

Data Generation The data is generated from templates using a generation toolkit from Warstadt et al. (2020). This toolkit includes a vocabulary of over 3000 entries labeled with grammatical fea-

[1] We explored a slightly larger set of linguistic features and excluded several based on initial experiments showing our models did not encode them. For example, we constructed a task with the objective of identifying sentences that contain antonyms (e.g. *The little girl likes the big dog.*), but found that only $\text{RoBERTa}_{\text{BASE}}$ could solve the unambiguous control task.

[2] Although these are surface properties of the string, they are not all trivial for RoBERTa due to its subword tokenization.

[3] The *control construction* is a syntactic construction in which a semantic argument of a predicate fills or *controls* an argument slot of an embedded verb. The *raising construction* is superficially similar, but the filler of the embedded argument slot is not a *semantic* argument of the main predicate (Sag et al., 2003). For instance, *Sue is eager to sleep* is an example of control because the NP *Sue* is the semantic subject of both *eager* and *sleep*. By contrast, *Sue is likely to sleep* is an example of raising because *Sue* is the semantic subject of *sleep*, but not of *likely*. These two phenomena have different syntactic derivations in some theories (Chomsky, 1981).

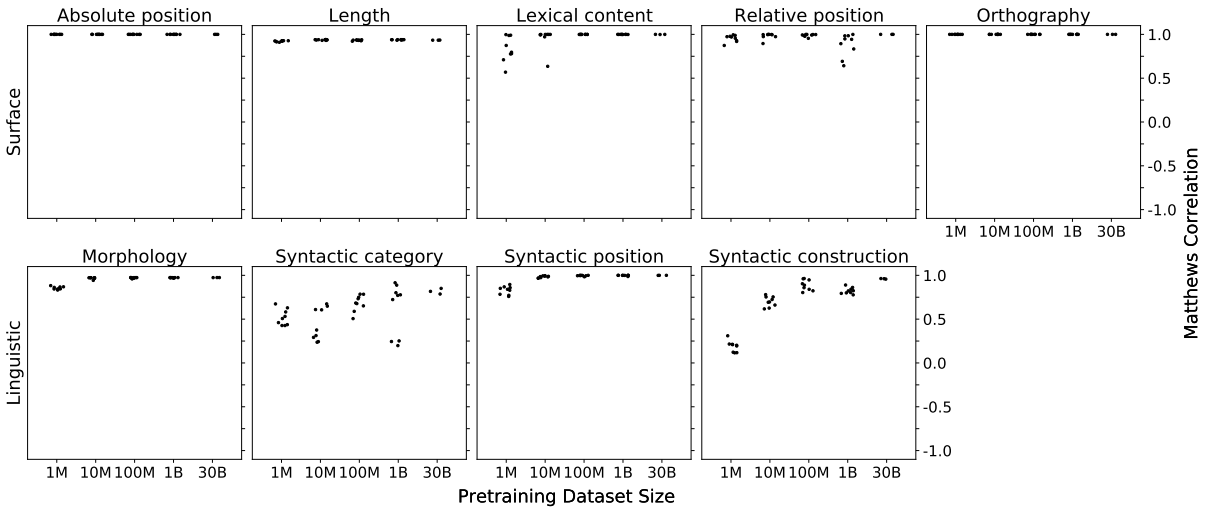

Figure 2: Results on the main experiments measured in Matthews correlation for the surface control tasks (top) and linguistic control tasks (bottom). Note: For surface tasks a positive score represents a surface generalization.

tures that allow for lexical variation in the data while maintaining grammatical well-formedness. Although generated sentences often describe unlikely or implausible scenarios (e.g., *The lawyer was sinking all canoes*), semantic plausibility is independent of all the features we examine, so this should not affect a model that genuinely encodes these features. To prevent out-of-vocabulary tokens affecting our results, we ensure that every word stem in the vocabulary appears in the pretraining datasets for our RoBERTa models (see §4.1).

Our experimental logic only makes sense if we are reasonably confident that models can only achieve high test performance by genuinely adopting a linguistic generalization. However, training models on generated data can easily lead to overfitting, and classifiers trained and tested on data from the same domain can achieve perfect performance even on arbitrary tasks with random labels (Hewitt and Liang, 2019). For this reason, our primary evaluations test models' ability to *generalize* out-of-domain. We manipulate domain in two ways:

First, we generate training data and test data for each dataset from separate in-domain and out-of-domain templates. Thus a model cannot succeed at test time simply by recognizing a template or a key part of a template. For example, in the SYNTACTIC POSITION × LEXICAL CONTENT paradigm shown in Table 2, the in-domain data contrasts the main verb with a verb in a relative clause embedded in the complement clause of a verb; while the out-of-domain data contrasts the main verb with a verb in the complement clause of a noun. In most tasks, each domain itself is generated from multiple templates as well to widen the domain and encourage better generalization during training.

Second, on tasks that test lexical knowledge (for instance, the knowledge that *slept* is an irregular past verb and *meow* is not), we divide the crucial lexical items into in-domain and out-of-domain sets. Thus, a model cannot succeed by memorizing the keywords associated with each class. See the Appendix for a more detailed description of the implementation details for each feature.

4 Models, Pretraining, & Fine-Tuning

We test 13 RoBERTa models in our main experiments: We pretrain 12 from scratch, and also test RoBERTa$_{\text{BASE}}$ pretrained by Liu et al. (2019b).

4.1 Pretraining

Pretraining Data We pretrain RoBERTa using scaled-down recreations of the dataset used by Devlin et al. (2019) to train BERT, i.e English Wikipedia (2.5 billion tokens) and BookCorpus (800 million tokens). Both are included in the RoBERTa pretraining data.[4] We download the latest Wikipedia dump as of Feb 1, 2020. The original BookCorpus (Zhu et al., 2015) is no longer available, so we collect similar data from Smashwords, the original source of BookCorpus.[5]

[4]RoBERTa uses English Wikipedia, BookCorpus, CC-News, OpenWebText, and STORIES in pretraining.

[5]We collect our data using the Wikipedia XML dump https://dumps.wikimedia.org/mirrors.html and data-processing code https://github.com/attardi/wikiextractor, and a Smashwords crawler https://github.com/soskek/bookcorpus.

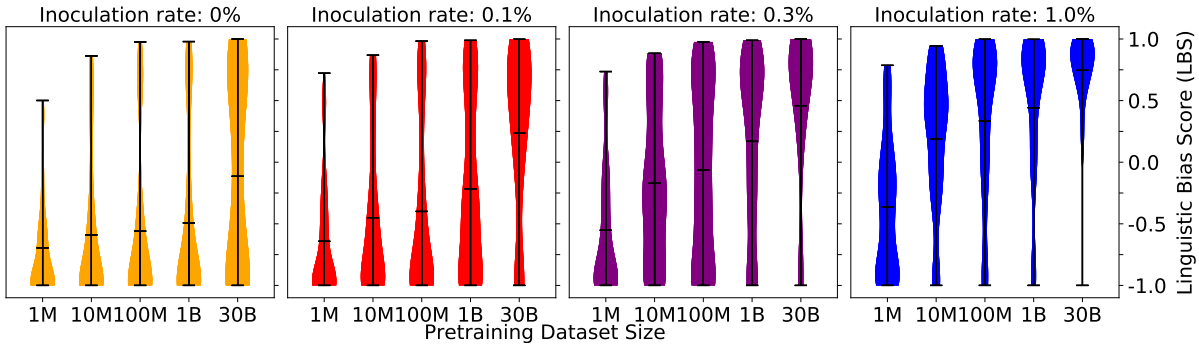

Figure 3: Results measured in LBS for each pretraining and inoculating data amount, aggregated over the 20 tasks in MSGS. We exclude models that fail the corresponding controls, as described in Section 5. High density near LBS of 1 means many models in that group have a linguistic bias; high density near -1 means many models have a surface bias. Models with stronger linguistic bias achieve higher LBS with less inoculation data.

We pretrain RoBERTa on four training sets containing different numbers of words: 1M, 10M, 100M, and 1B.[6] To make these datasets, we sample entire Wikipedia articles and Smashwords books independently, keeping the proportions of Wikipedia and Smashwords text approximately constant.

Model Sizes Model size is the only hyperparameter we systematically search over during pretraining. We consider smaller model sizes to prevent overfitting on small training sets. The detailed configurations of the model sizes are summarized in the Appendix. We use RoBERTa$_{\text{BASE}}$ from Liu et al. (2019b) as our largest model size. The other configurations represent a scale roughly based on settings used in Sanh et al. (2019), Vaswani et al. (2017), Jiao et al. (2019), and Tsai et al. (2019).

Search Range For dropout, attention dropout, learning rate decay, weight decay and the Adam parameters, we adopt the same parameter values used in Liu et al. (2019b). We fix warm up steps to be 6% of max steps, peak learning rate to be 5e-4, early stopping patience to be 100M tokens, and heuristically define the search range of model size, max steps and batch size for each training set.

Search Results We randomly sample hyperparameters from the search range and train 25 models for each of the 1M, 10M, 100M datasets. We train 10 models on the largest (1B) dataset due to resource limitations. For each training set size, we choose three of the resulting models to evaluate. In order to avoid confounds caused by different model sizes, for each training set we choose three models of the same size that have the lowest perplexity. The hyperparameters and validation perplexities of the selected models are listed in the Appendix.

[6]The publicly available RoBERTa$_{\text{BASE}}$ is trained on 160GB of data, which we estimate to be about 30B words.

4.2 Fine-Tuning

We loosely follow the hyperparameter settings that Liu et al. (2019b) used for fine-tuning on GLUE tasks (Wang et al., 2018), and use the following learning rates: {1E-5, 2E-5, 3E-5}. We depart from Liu et al. in using a batch size of 16 and training for 5 epochs without early-stopping in all runs. These changes are based on pilots that showed that larger batch sizes and longer fine-tuning were no more effective for our tasks.

We conduct 3,471 fine-tuning runs: We fine-tune 13 RoBERTa models: (3 random initializations) $\times$ (4 pretraining data amounts) + (1 RoBERTa$_{\text{base}}$). We fine-tune each model 267 times: (3 learning rates) $\times$ ((9 control tasks) + (20 ambiguous tasks) $\times$ (4 inoculation amounts)). We evaluate model performance using LBS (see §2:Methods: Measuring Inductive Bias).

5 Results & Discussion

We have several main findings: (1) models learn to *represent* both surface features and linguistic features with relatively little data; (2) RoBERTa begins to acquire a linguistic bias with over 1B words of pretraining data; (3) increasing pretraining data strengthens linguistic bias; (4) there is considerable variation in models' preferences between specific pairs of linguistic and surface features.

Control results Figure 2 shows the results for the controls. Performance is near ceiling for most models and features. Because we evaluate all the

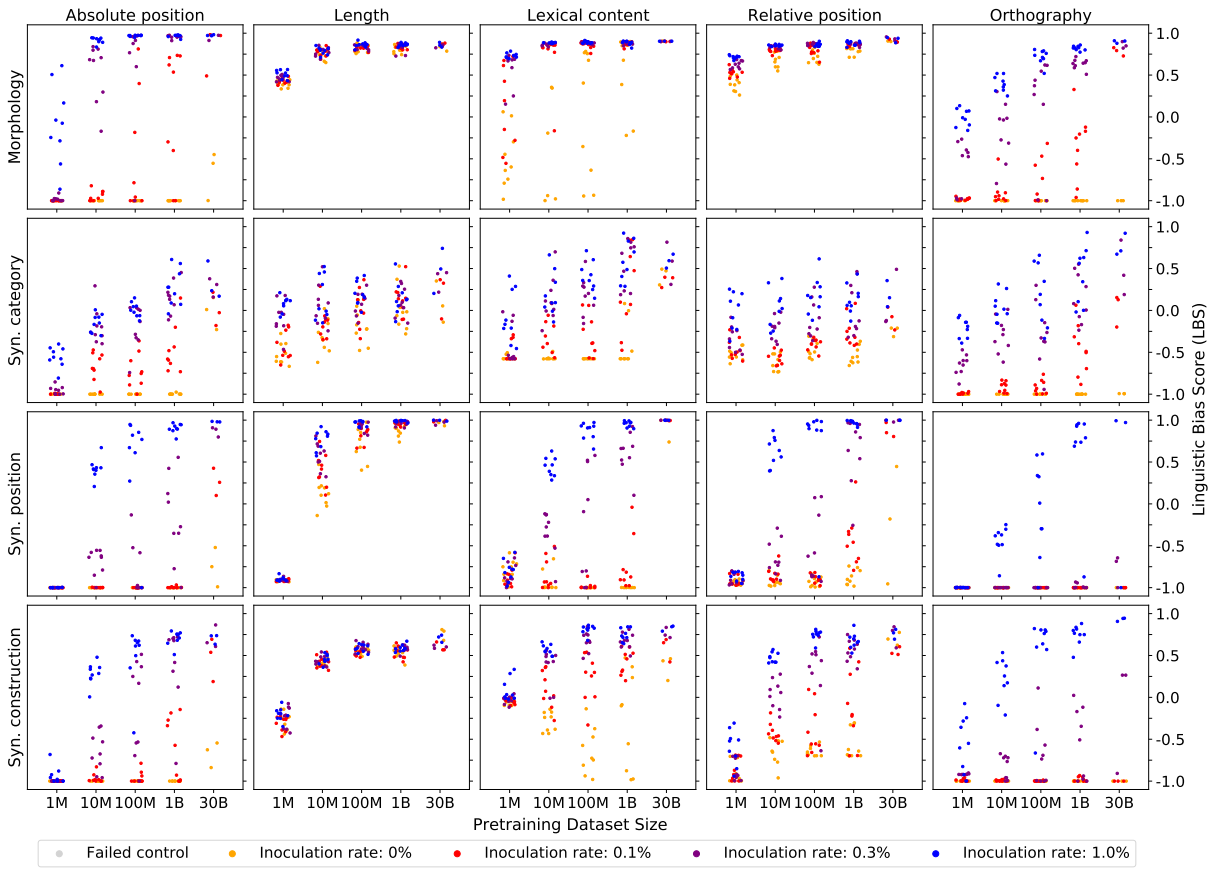

Figure 4: Results of the ambiguous binary classification tasks measured in LBS for every (linguistic feature, surface feature) pair. Each plot in the matrix shows the results on the disambiguating test items after training on an ambiguous task. All experiments on the same row investigate the same linguistic feature; all experiments on the same column investigate the same surface feature. Each data point represents one run. The x-axis of the point is the pretraining size of the model, and the y-axis is its LBS. Models with stronger linguistic bias achieve higher LBS with less inoculation data. Gray points show runs where the corresponding controls did not pass. A black-and-white version of this figure separating color channels into separate plots, can be found in the Appendix.

models out-of-domain, this result cannot be explained by the models simply memorizing the features from the task training data. Thus, we conclude that most pretrained models we test encode both linguistic and surface features.

The only exceptions are the syntactic category and syntactic construction features, for which models with less than 100M perform poorly. In subsequent plots, we filter out results where the controls are not passed. Specifically, if a particular combination of model checkpoint and learning rate achieves a Matthews correlation of less than 0.7 on the control task for feature F, we eliminate all results with this combination for any task involving F in Figure 3, or represent them as gray points in Figure 4.

Main Experiment Results Figure 3 summarizes the main experiment results. For a given amount of pretraining and inoculation data, we consider all classifiers trained on all 20 tasks in MSGS and plot the density of their linguistic bias scores (LBSs).

The results in the leftmost box (with 0% inoculation) show that only RoBERTa$_{\text{BASE}}$ demonstrates a consistent linguistic bias in the fully unambiguous setting. That said, it still adopts the surface bias much of the time. The other models show a clear surface bias overall. The results of experiments with inoculation data show that models with more pretraining data require less inoculation data to be swayed towards the linguistic generalization. We consistently observe, for each pretraining quantity, a phase transition where the linguistic generalization begins to overtake the surface generalization upon exposure to a certain amount of inoculating data. For example, the 1B model goes through this transition between 0.1% and 0.3% inoculating data. The 100M and 10M models go through this transition between 0.3% and 1% inoculating data. The phase transition comes earlier for models with

more pretraining, indicating they have a stronger linguistic bias. We also notice distinctive behavior for the models at the extreme ends of pretraining data quantity: The 1M model never completes the transition, suggesting it has a strong surface bias, and RoBERTa$_{\text{BASE}}$ appears to be in the middle of this transition with 0% inoculating data, suggesting that even more pretraining data could produce a model with a more consistent linguistic bias.

These findings are echoed in individual task results in Figure 4.[7] In each plot, models with the same amount of inoculation data (i.e. points with a given color) have higher LBS as the amount of pretraining data increases. Notably, on ambiguous tasks involving LEXICAL CONTENT, RoBERTa$_{\text{BASE}}$ usually favors generalizations based on linguistic features without any inoculating data, which no other pretrained model does. We find this result quite striking: Even if the labels are perfectly correlated with the presence or absence of the word "the", RoBERTa$_{\text{BASE}}$ overlooks that fact in favor of a deeper generalization based on an abstract feature like the inflectional form of a verb in a particular syntactic position. Furthermore, this preference is clearly *acquired* through additional pretraining. The results for MORPHOLOGY × ORTHOGRAPHY is a typical illustration of the differences between models. The 1M model never adopts the linguistic generalization based on the morphological feature, though it eventually rejects the surface generalization. The 100M and 1B models make robust linguistic generalizations only with 1.0% inoculating data. By contrast, RoBERTa$_{\text{BASE}}$ requires only 0.1% inoculating data (i.e. 10 out of 10k examples) to adopt the linguistic generalization.

Surface Biases of RoBERTa Our results also suggest some specific conclusions about which kinds of surface features RoBERTa pays attention to.[8] For instance, these models have little preference for sentence length. As shown in the second column of Figure 4, most of the models form linguistic generalizations rather than generalizations based on sentence length, even with no inoculating data. By contrast, the models strongly prefer generalizations based on orthography—and to a lesser extent lexical content and word order—over linguistic generalizations.

The Success of Pretrained Models Our findings provide insight into why pretraining on massive datasets is so successful. While linguistic feature learning is a major effect of pretraining, it is far from the end of the story: Pretraining also helps models learn which features are central to language. However, this second kind of learning seems to require far more exposure to data with current models and pretraining techniques. Therefore, massive datasets are needed to teach models which features are useful for generalizing.

The data scale at which we observe RoBERTa beginning to show a linguistic bias (between 1B and 30B words) is similar to the amount of pretraining data used by the first pretrained LMs to achieve major successes at NLU tasks, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). This suggests a crucial data threshold below which language model pretraining is unlikely to be significantly helpful for most applications with current model architectures, and may explain the many-year gap between the development of neural LMs and the first major applications of LM pretraining: The early LMs must have not have been trained sufficiently to cross that threshold, yielding consistently poor results.

6 Related work

There is increasing interest in studying the inductive biases of neural networks. Much of this work has grown out of numerous findings that these models often fail to generalize in ways that task designers intend. For example, Jia and Liang (2017) and McCoy et al. (2019) demonstrate that ambiguity in widely used NLU datasets like SQuAD (Rajpurkar et al., 2016) and MultiNLI (Williams et al., 2018) leads models like BERT to adopt some surface generalizations, despite the fact that they represent linguistic features. This continues to be a problem for models like RoBERTa$_{\text{BASE}}$ which show an overall linguistic bias in our experiments. However, for tasks like NLI, the underlying linguistic feature depends on a combination of significant syntactic knowledge, semantic knowledge, and world knowledge. It stands to reason that representations and preferences for such high level features require more data to learn than the features we probe.

Other work has used the poverty of stimulus design to study inductive biases associated with particular neural architectures during syntactic gen-

[7] Analogous results for the held out training-condition data, inoculation data, and auxiliary data are in the Appendix.

[8] MSGS does not come close to representing the full range of possible relevant lexical or syntactic features, preventing us from making strong conclusions about which specific linguistic features RoBERTa has biases in favor of.

eralization. Ravfogel et al. (2019) train RNNs on a morphological prediction task using artificial languages derived from naturally occurring English text, finding that RNNs show a recency bias in acquiring agreement rules. McCoy et al. (2018, 2020) train a seq2seq models on generated data ambiguous between a surface and a structural generalization to learn the subject-auxiliary inversion rule in English question formation. They find that, while tree-structured models show a structural bias, sequence models do not. Warstadt and Bowman (2020) conduct related experiments on subject-auxiliary inversion and other English structural rules, and find that BERT likely acquires a structural bias from pretraining.

More abstract inductive biases have also been studied. Using zero-shot learning in an artificial language, Lake and Baroni (2018) show that RNNs lack a bias in favor of learning compositional meanings for new symbols. Gandhi and Lake (2019) and Gulordava et al. (2020) explore conditions under which neural networks exhibit a bias towards learning mutually exclusive meanings for new symbols.

Data augmentation and inoculation have also been explored previously as a way to influence how models generalize. McCoy et al. (2019) and Min et al. (2020) show that small amounts of inoculating data during training on textual entailment help BERT overlook certain surface generalizations. Jha et al. (2020) study inoculation using a constructed language of numerical sequences. Like us, they generate ambiguous datasets, though they only compare features that resemble our surface features. They find that it is relatively easy to nudge models away from shallow generalizations, but harder to nudge them towards deeper ones.

Finally, several earlier studies explored how increasing training data impacts linguistic knowledge in LMs. Unlike the present study, these studies evaluate LMs using an unsupervised acceptability judgment task on minimal pairs (i.e. not during fine-tuning), and do not attempt to separate feature learning from feature preferences. van Schijndel et al. (2019) find the greatest increase in sensitivity to acceptability contrasts occurs between training on 2M and 10M words. Warstadt et al. (2020) find that while LMs learn agreement phenomena at a similarly early stage, other phenomena require more data to learn. Finally, Hu et al. (2020) find that adopting architectures that build in linguistic bias, such as RNNGs (Dyer et al., 2016), has a bigger effect on the acceptability task than increasing training data from 1M to 40M words.

7 Future Work & Conclusion

Our experiments shed light on the relationship between pretraining data and an inductive bias towards linguistic generalization. Our results indicate that, although some abstract linguistic features are learnable from relatively small amounts of pretraining data, models require significant pretraining after discovering these features to develop a bias towards *using* them when generalizing. This gives some insight into why extensive pretraining helps general-purpose neural networks adapt to downstream tasks with relative ease.

We also introduce MSGS, a new diagnostic dataset for probing the inductive biases of learning algorithms using the poverty of the stimulus design and inoculation, and also introduce a set of 12 RoBERTa models we pretrain on smaller data quantities. These models could prove to be a helpful resource for future studies looking to study learning curves of various kinds with respect to the quantity of pretraining data.

Finally, while our results naturally lead to the conclusion that we should continue to pursue models with ever more pretraining, such as GPT-3 (Brown et al., 2020), we do not wish to suggest that this will be the only or best way to build models with stronger inductive biases. Future work might use MSGS as a diagnostic tool to measure how effectively new model architectures and self-supervised pretraining tasks can more efficiently equip neural networks with better inductive biases.

Acknowledgments

This project has benefited from financial support to SB by Eric and Wendy Schmidt (made by recommendation of the Schmidt Futures program), by Samsung Research (under the project *Improving Deep Learning using Latent Structure*), by Intuit, Inc., and in-kind support by the NYU High-Performance Computing Center and by NVIDIA Corporation (with the donation of a Titan V GPU). This material is based upon work supported by the National Science Foundation under Grant No. 1850208 and 1922658. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstral, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li Li, and Razvan Pascanu. 2018. Relational inductive biases, deep learning, and graph networks.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. ArXiv preprint 2005.14165.

Noam Chomsky. 1981. *Lectures on government and binding*. Walter de Gruyter.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does BERT look at? An analysis of BERT's attention. ArXiv preprint 1906.04341.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 42–48.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A. Smith. 2016. Recurrent neural network grammars. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 199–209, San Diego, California. Association for Computational Linguistics.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. When BERT forgets how to POS: Amnesic probing of linguistic properties and MLM predictions. ArXiv preprint 2006.00995.

Kanishk Gandhi and Brenden M Lake. 2019. Mutual exclusivity as a challenge for neural networks. *arXiv preprint arXiv:1906.10197*.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2019. Colorless green recurrent networks dream hierarchically. *Proceedings of the Society for Computation in Linguistics*, 2(1):363–364.

Kristina Gulordava, Thomas Brochhagen, and Gemma Boleda. 2020. Which one is the dax? achieving mutual exclusivity with neural networks. *arXiv preprint arXiv:2004.03902*.

Harry F Harlow. 1949. The formation of learning sets. *Psychological review*, 56(1):51.

David Haussler. 1988. Quantifying inductive bias: Ai learning algorithms and valiant's learning framework. *Artificial intelligence*, 36(2):177–221.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

John Hewitt and Christopher D Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger P Levy. 2020. A systematic assessment of syntactic generalization in neural language models. *arXiv preprint arXiv:2005.03692*.

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

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Brenden Lake and Marco Baroni. 2018. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *International Conference on Machine Learning*, pages 2879–2888.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences*, 40.

Nelson Liu, Roy F Schwartz, and Noah A Smith. 2019a. Challenge. *manuscript*.

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

Brian W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, 405(2):442–451.

R Thomas McCoy, Robert Frank, and Tal Linzen. 2018. Revisiting the poverty of the stimulus: hierarchical generalization without a hierarchical bias in recurrent neural networks. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*.

R. Thomas McCoy, Robert Frank, and Tal Linzen. 2020. Does syntax need to grow on trees? sources of hierarchical inductive bias in sequence-to-sequence networks. *Transactions of the Association for Computational Linguistics*, 8:125–140.

R Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *Proceedings of the Association for Computational Linguistics*.

Junghyun Min, R Thomas McCoy, Dipanjan Das, Emily Pitler, and Tal Linzen. 2020. Syntactic data augmentation increases robustness to inference heuristics. *arXiv preprint arXiv:2004.11999*.

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

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of rnns with synthetic variations of natural languages. In *Proceedings of NAACL-HLT*, pages 3532–3542.

Ivan A. Sag, Thomas Wasow, and Emily M. Bender. 2003. *Syntactic Theory: A Formal Introduction*, 2 edition. CSLI Publications.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Marten van Schijndel, Aaron Mueller, and Tal Linzen. 2019. Quantity doesn't buy quality syntax with neural language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5831–5837, Hong Kong, China. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. In *Proceedings of ICLR*.

Henry Tsai, Jason Riesa, Melvin Johnson, Naveen Arivazhagan, Xin Li, and Amelia Archer. 2019. Small and practical bert models for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3623–3627.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. ArXiv preprint 2003.12298.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Alex Warstadt and Samuel R Bowman. 2020. Can neural networks acquire a structural bias from raw linguistic data? In *Proceedings of the 42nd Annual Conference of the Cognitive Science Society.*

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

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of NAACL 2018*, volume 1, pages 1112–1122.

Colin Wilson. 2006. Learning phonology with substantive bias: An experimental and computational study of velar palatalization. *Cognitive science*, 30(5):945–982.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Data Description

MSGS contains 5 surface features and 4 linguistic features, summarized in Table 3 (repeated from the main body of the paper, for convenience). Implementation details for the features are described below. The implementation of one feature sometimes depends on other feature it is paired with in an ambiguous dataset.

Absolute position This feature is 1 *iff* the sentence begins with the word "the". We generally ensure that sentences bearing a value for this feature contains two clauses and four determiners total. Some sentences in SYNTACTIC CATEGORY × ABSOLUTE POSITION contain fewer than four NPs. The in-domain and out-of-domain sentences differ in the order or position of the clauses.

Length This feature is 1 *iff* the sentence exceeds some number of tokens. The exact threshold varies depending on the linguistic feature in an ambiguous task, since different linguistic features lead to sentences of different length, on average. In mixed tasks, we vary the length of sentences by adjoining subordinate clauses (e.g. *If Sue wakes*) of varying length to the clause in which the linguistic feature is varied.

Lexical content This feature is 1 *iff* the sentence contains *the*. The sentences generally contain at least two clauses and four determiners. The position of *the* varies between in-domain and out-of-domain sentences.

Relative position This feature is 1 when *the* precedes *a*, and 0 when *a* precedes *the*. The sentences generally contain at least two clauses and four determiners. Thus, there are six different configurations in which *the* precedes *a*, and these are separated into in-domain and out-of-domain templates.

Orthography This feature is 1 *iff* the sentence appears in title case. In the control paradigm, the sentences generally contain two clauses, whose positions are varied between in-domain and out-of-domain sentences.

Lexical semantics This feature is 1 *iff* the sentence contains a pair of antonyms. In sentences with label 0, there is a pair of words in a hypernym/hyponym or synonym relation. There are 21 pairs of adjective antonyms and 21 pairs of verb antonyms (not accounting for different inflectional forms). To prevent the task being solvable using lexical content, these pairs are divided into in-domain and out-of-domain sets. There are different templates corresponding to whether the antonyms are adjectives, intransitive verbs, or transitive verbs. Each template appears in both in-domain and out-of-domain sentences.

Morphology This feature is 1 when the sentence contains an irregular past tense verb, and 0 when it contains a 3rd person present plural verb (identical to the bare form). Sentences either contain an irregular past tense verb or a regular 3rd person present plural verb (identical to the bare form). We do this because other verb forms can be identified by inflectional morphemes such as *-s* or auxiliaries such as *have*, and so discrimination between them could in some cases reduce to a lexical content task. The verbs are divided into in-domain and out-of-domain sets.

Syntactic category This feature is 1 *iff* the sentence contains an adjective. To diversify the templates, we consider all grammatical combinations of a noun, an adjective, a locative PP, and a proper name (e.g., *Sue is the tall actress in the park*, or *The actress is Sue*). In out-of-domain sentences we also include single-word nominal predicates like *president* (see the example in Table 3 to control for the fact that predicative adjectives are always a single, lowercase word. This gives a total of 19 templates divided into in-domain and out-of-domain sets, some with adjectives and some without. The set of adjectives is also split between domains.

Syntactic construction This feature has value 1 *iff* the sentence contains the control construction. In the control construction a semantic argument of a predicate fills or *controls* an argument slot of an embedded verb (Sag et al., 2003). For instance, in *Sue is eager to sleep*, the NP *Sue* surfaces as the syntactic subject of *eager*, but *Sue* is also understood as the semantic subject of *sleep*. This contrasts with the *raising* construction in *Sue is likely to sleep*, where *Sue* is again surfaces as the syntactic subject of *likely* in the main clause, and is the semantic subject of *sleep* in the embedded position, but is not a semantic argument of *likely*. Different predicates are compatible with control and raising: *eager* is a control predicate and *likely* is a raising predicate. We include control and raising predicates of three kinds: subject control/raising verbs, object control/raising verbs, and control/raising adjectives. Specific predicates are divided into in-domain and

	Feature type	Feature description	Positive example	Negative example
Surface	Absolute position	Is the first token of S "the"?	The cat chased a mouse.	A cat chased a mouse.
	Length	Is S longer than n (e.g., 3) words?	The cat chased a mouse.	The cat meowed.
	Lexical content	Does S contain "the"?	That cat chased the mouse.	That cat chased a mouse.
	Relative position	Does "the" precede "a"?	The cat chased a mouse.	A cat chased the mouse.
	Orthography	Does S appear in title case?	The Cat Chased a Mouse.	The cat chased a mouse.
Linguistic	Morphology	Does S have an irregular verb?	The cat slept.	The cat meows.
	Syn. category	Does S have an adjective?	Lincoln was tall.	Lincoln was president.
	Syn. construction	Is S the control construction?	Sue is eager to sleep.	Sue is likely to sleep.
	Syn. position	Is the main verb in "ing" form?	Cats who eat mice are purring.	Cats who are eating mice purr.

Table 3: Schematic examples of the linguistic and surface features.

out-of-domain sets, but all three kinds of predicates appear in both domains.

Syntactic position All sentences contain at one or two embedded clauses. We include six sentence types, divided into in-domain and out-of-domain. For example, some sentences contain a relative clause within a relative clause, or a verb phrase with a complement clause. Each sentence type is generated from multiple templates varying the position of the clauses. The set of *-ing* verbs is not split between domains.

B Pretraining Details

Name	L	AH	HS	FFN	P
Base	12	12	768	3072	125M
Med	6	12	768	3072	82M
Med-Small	6	8	512	2048	45M
Small	4	8	384	1200	26M
XSmall	3	4	256	1024	15M

Table 4: The RoBERTa model sizes we search over during pretraining. AH = number of attention heads; HS = hidden size; FFN = feed-forward network dimension; P = number of parameters.

Training Size	Model Size	Max Steps	Batch Size	Validation Perplexity
1B	BASE	31K	4096	3.84
1B	BASE	100K	512	3.93
1B	BASE	31K	1024	4.25
100M	BASE	31K	1024	4.61
100M	BASE	100K	512	4.99
100M	BASE	31K	512	5.02
10M	BASE	10K	512	10.78
10M	BASE	10K	1024	11.31
10M	BASE	31K	512	11.58
1M	MED-SMALL	10K	512	134.18
1M	MED-SMALL	31K	512	139.39
1M	MED-SMALL	100K	512	153.38

Table 5: The pretraining parameters of the 12 models we use in our experiments.

C Additional Results

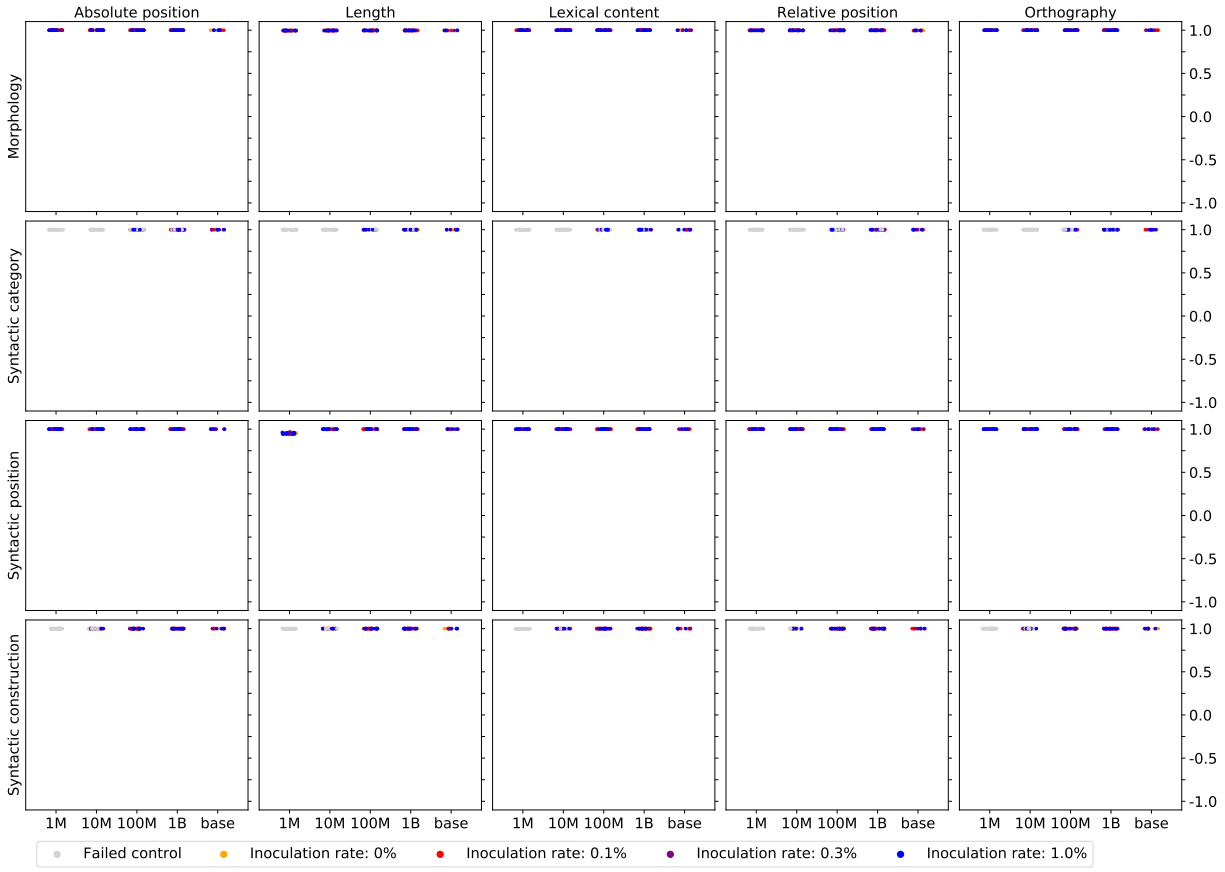

Figure 5: Results on the held-out training-condition items (in-domain/mixed) measured in LBS.

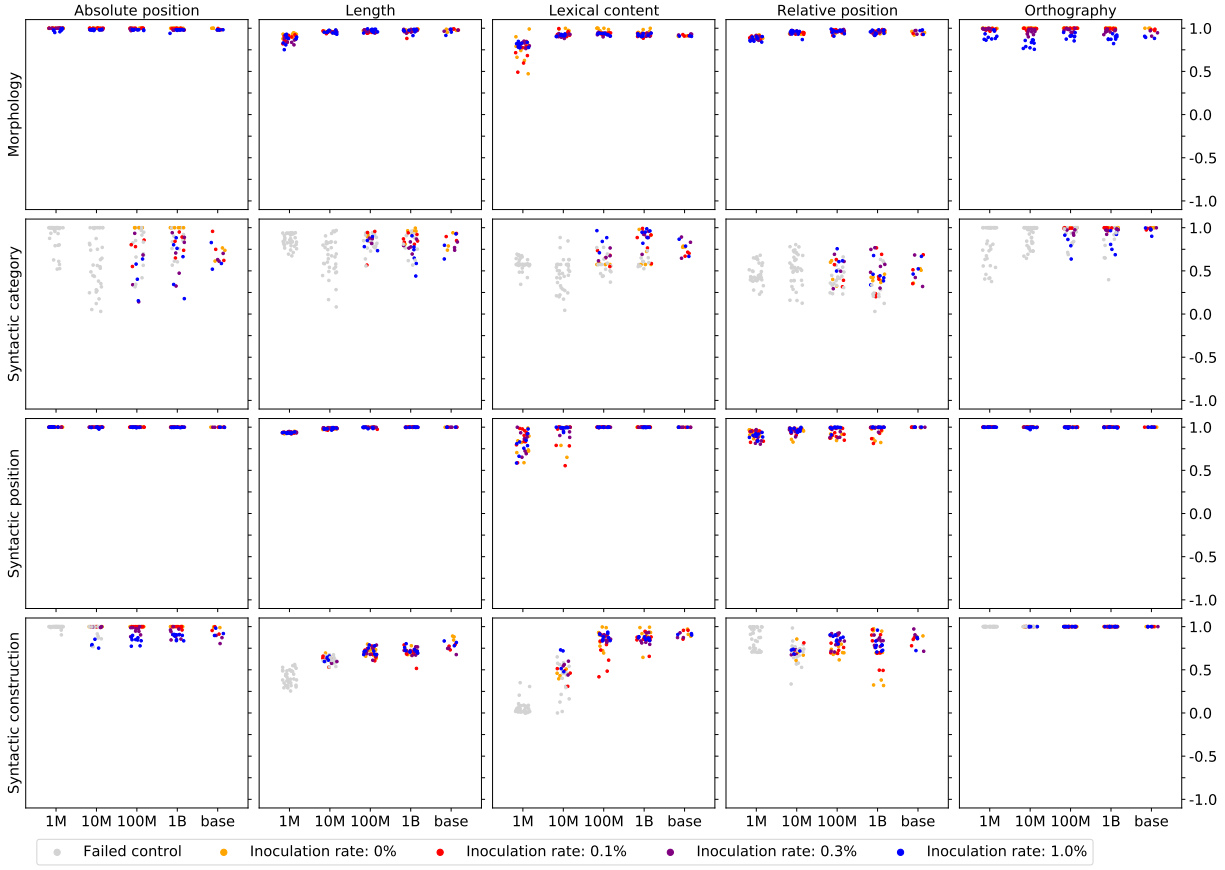

Figure 6: Results on the held-out auxiliary-condition items (out-of-domain/mixed) measured in LBS.

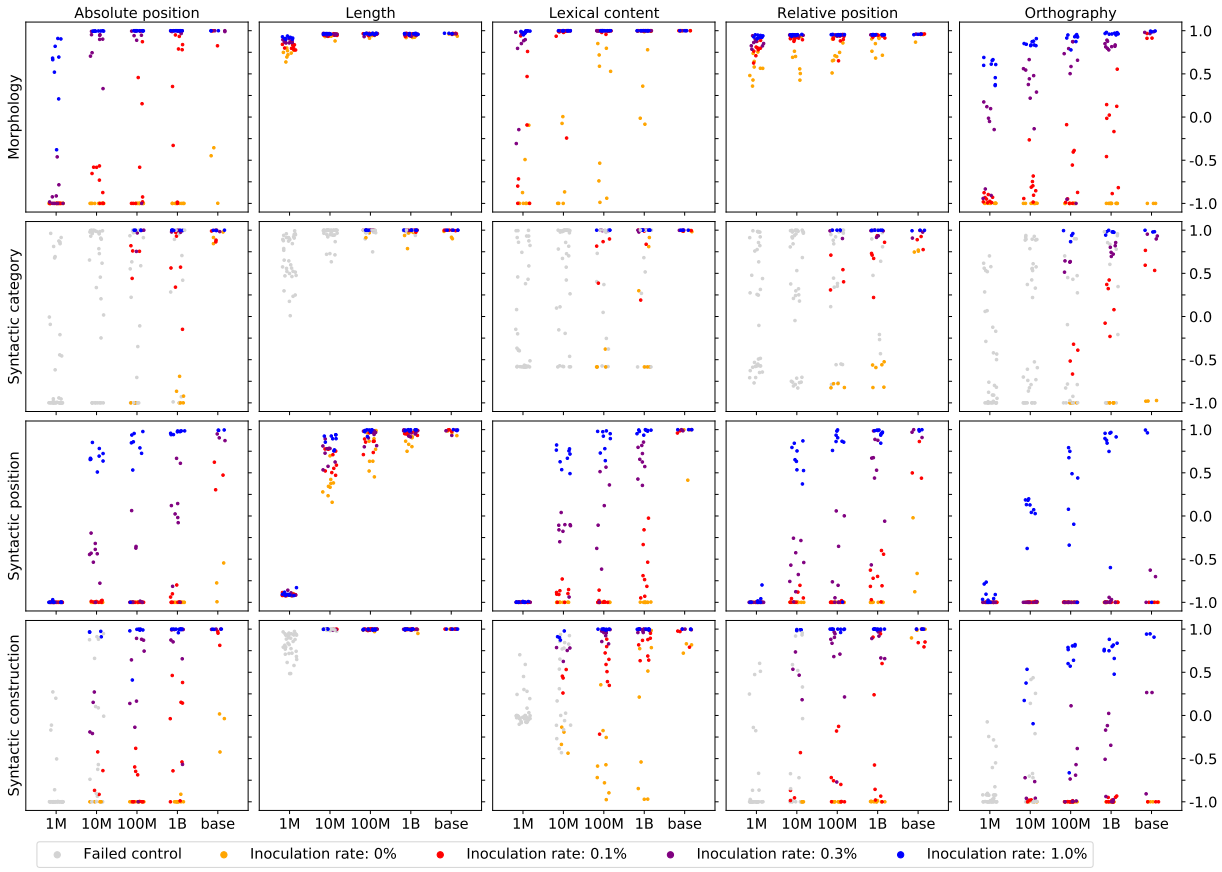

Figure 7: Results on the held-out inoculation-condition items (in-domain/unmixed) measured in LBS.

D Black and white versions of Fig. 4

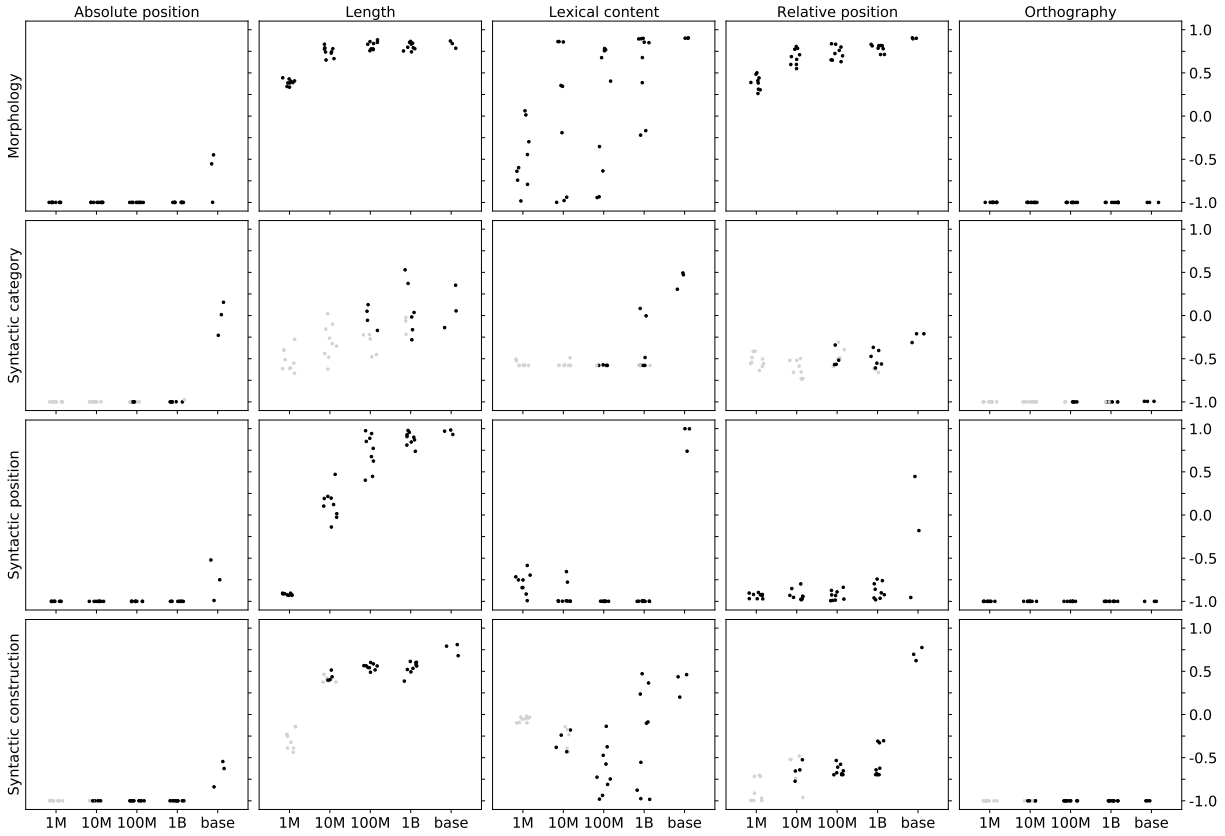

Figure 8: Results of the mixed binary classification tasks with no inoculation data.

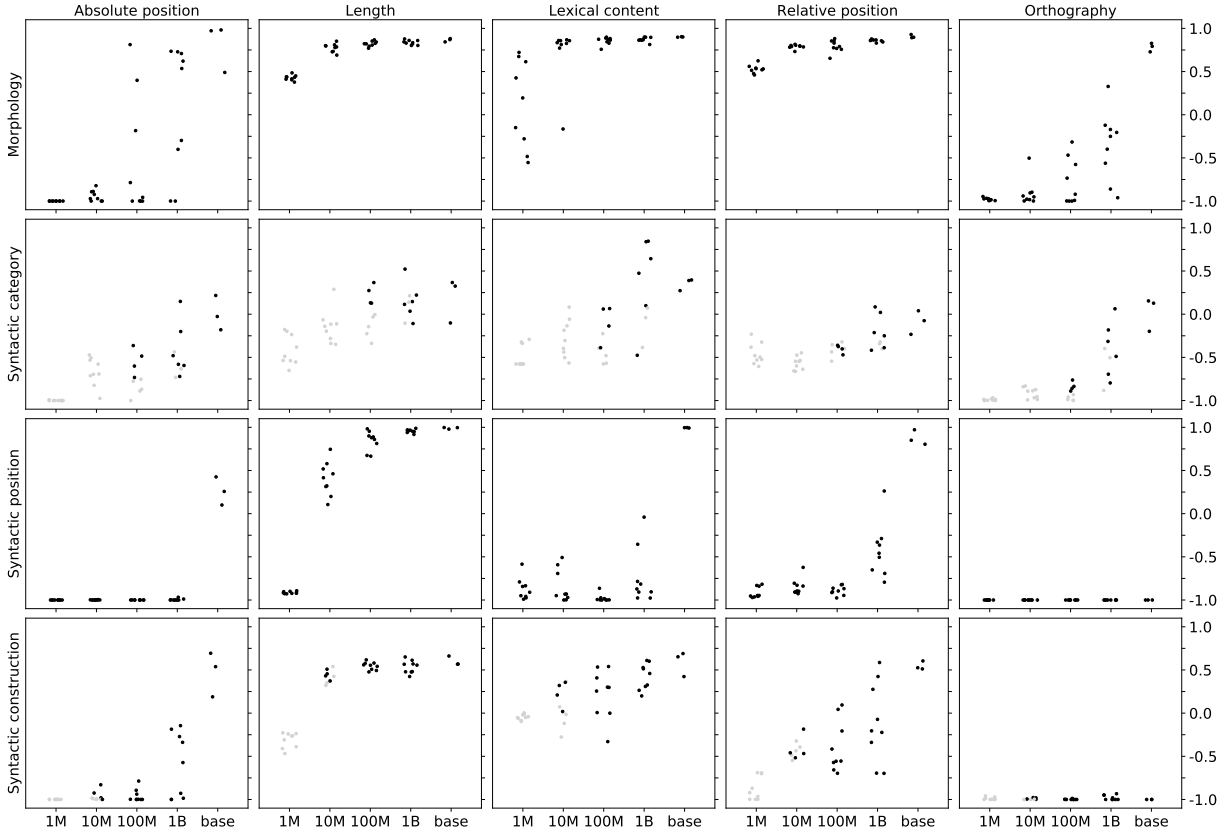

Figure 9: Results of the mixed binary classification tasks with 0.1% inoculation data.

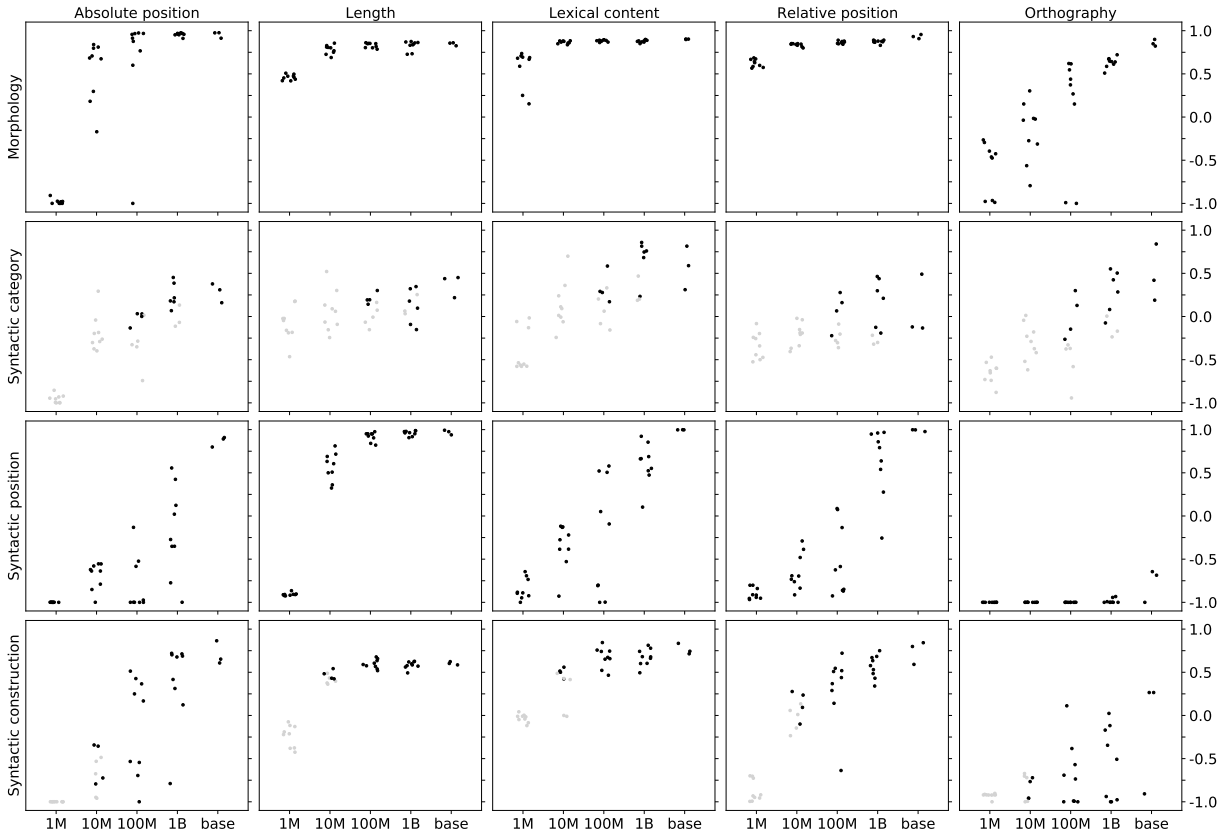

Figure 10: Results of the mixed binary classification tasks with 0.3% inoculation data.

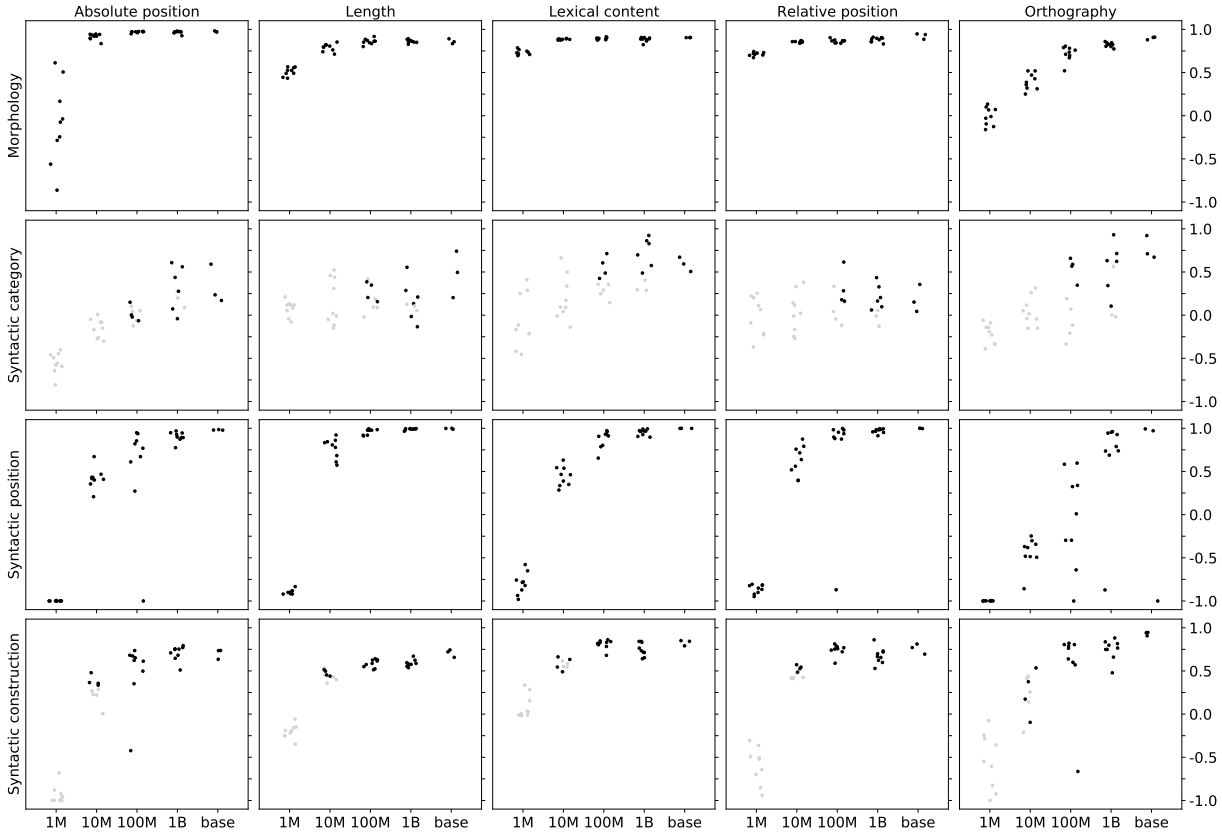

Figure 11: Results of the mixed binary classification tasks with 1% inoculation data.