Learning Helpful Inductive Biases from Self-Supervised Pretraining

Anonymous EMNLP submission

Abstract

Large pretrained language models demonstrate strong, language-specific biases during fine-tuning that allow them to solve language tasks better than models without pretraining. We aim to characterize these biases, and identify the amount of pretraining that is necessary to acquire them. We introduce a new English language diagnostic set called MSGS (Mixed Signals Generalization Set) which contains two types of data: *mixed* data in which the labels are consistent with both a linguistic classification (e.g., Is the main verb in the progressive form?) and a superficial surface one (e.g., Does "the" precede "a"?); and *unmixed* data in which the labels align only with the linguistic feature. We fine-tune RoBERTa on mixed data (with and without small amounts of inoculating unmixed data) and test on unmixed data to see which feature it has bias in favor of. We pretrain RoBERTa from scratch on quantities of data ranging from 1M to 1B words and compare their performance on MSGS to the publicly available RoBERTa-Base. We find steady growth in linguistic bias with increased pretraining data. The models we test can usually represent the linguistic features, but they only learn to prefer to generalize based on these features with significant pretraining. In the absence of inoculating data, only RoBERTa-Base consistently demonstrates a linguistic bias with any regularity.

1 Introduction

How does self-supervised pretraining on large datasets shape the inductive biases of models like BERT and RoBERTa? There is significant evidence that these models can learn to implicitly encode linguistic features like syntactic dependencies and part-of-speech from self-supervised tasks like language modeling (Tenney et al., 2019; Hewitt and Manning, 2019). But the very same models are susceptible to making incorrect generalizations based on surface features in the data (McCoy et al., 2019). In this paper, we investigate how different amounts of pretraining data help RoBERTa make more robust generalizations by altering its inductive biases, making it more likely to arrive at generalizations on target tasks based on helpful linguistic features, and less likely to rely on misleading surface features.

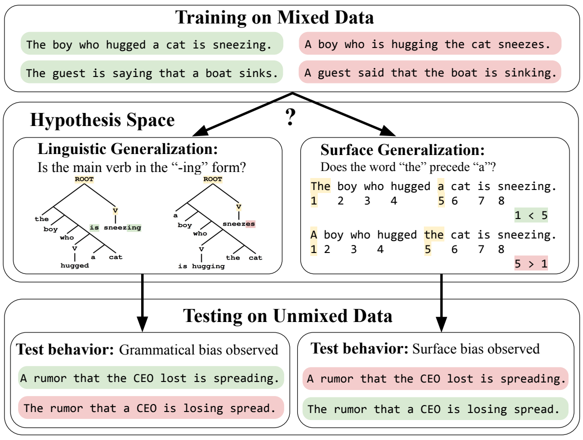
Training on Mixed Data
The boy who hugged a cat is sneezing.
A boy who is hugging the cat sneezes.
The guest is saying that a boat sinks.
A guest said that the boat is sinking.
Hypothesis Space
?
Linguistic Generalization:
Is the main verb in the "-ing" form?
ROOT
the
boy
who
V
hugged
a
cat
V
is sneezing
ROOT
a
boy
who
V
is hugging
the
cat
V
sneezes
Surface Generalization:
Does the word "the" precede "a"?
The boy who hugged a cat is sneezing.
1 2 3 4 5 6 7 8
1 < 5
A boy who hugged the cat is sneezing.
1 2 3 4 5 6 7 8
5 > 1
Testing on Unmixed Data
Test behavior: Grammatical bias observed
A rumor that the CEO lost is spreading.
The rumor that a CEO is losing spread.
Test behavior: Surface bias observed
A rumor that the CEO lost is spreading.
The rumor that a CEO is losing spread.

Figure 1: Example of our experimental design without inoculating data. A model is shown mixed data consistent with two independent generalizations during training, and tested on data that disambiguate between the two generalizations. Green and red shading represents data or features associated with the positive and negative classes in a binary classification task, respectively. Yellow shading represents properties of the input that the model must identify in order to extract these predictive features.

We define *surface features* as shallow features, such as the length of a sequence or the linear position of a token, that can be extracted without any understanding of linguistic structure or meaning and *linguistic features* as features, such as the syntactic category of a phrase, that must be defined in terms of abstract grammatical structures. Real natural language understanding tasks generally cannot be solved using surface features alone, and a model that preferentially generalizes using surface features over linguistic features is bound to fail. Currently, the best examples of models with a known bias towards grammatically informed generalizations use tree structured architectures or pre-parsed inputs (Dyer et al., 2016; Wilcox et al., 2019; McCoy et al., 2020). However, such models are more computationally expensive and have not proven to give significant advantages on downstream tasks.

We explore the extent to which inductive biases, rather than being built into the architecture of a model, can be acquired by a general purpose sentence processing model like BERT or RoBERTa through pretraining. To test the extent to which this occurs, we conduct a battery of experiments inspired by the *poverty of the stimulus* experimental design (Wilson, 2006) to probe the models' inductive biases, as illustrated in Figure 1. First, we fine-tune a pretrained model using *mixed* data, in which the label of an input sentence is consistent with both a linguistic feature and a surface feature in the sentence. We then test the classifier with *unmixed* data, in which the label of a sentence is correlated with the linguistic feature and perfectly anti-correlated with the surface feature. This experimental design allows us to observe which feature (if either) the classifier bases its generalization on, and therefore what biases the model learns from pretraining. We select five surface features and pit each one against four linguistic features, giving a total of 20 binary classification tasks, for which we automatically generate data.

We call the resulting dataset MSGS (Mixed Signals Generalization Set), pronounced "messages". In addition, we repeat these experiments with small amounts of unmixed *inoculating* data introduced into the mixed training data, meant to sway the model toward the linguistic generalization (following Liu et al., 2019a).

To track the evolution of inductive biases as pretraining data increases, we pretrain RoBERTa from scratch on datasets ranging from 1M to 1B words and evaluate these models alongside RoBERTa-Base. We find a clear relationship between the amount of pretraining data and the model's tendency to adopt a linguistic generalization in the face of mixed signals: Models with more pretraining data can generally be induced to adopt linguistic generalizations with less inoculating data. RoBERTa-Base has the strongest linguistic bias, and requires little to no inoculating data to reliably make the linguistic generalization. For models with less pretraining data, we continue to observe the surface generalization even in the presence of unmixed data that contradicts it. Control experiments on only unmixed data reveal that these models are fully able to acquire the linguistic generalization, but nonetheless show a strong inductive bias against it. Thus, we observe a long gap in the amount of pretraining between that causes a model learns the features it would need to use to generalize out-of-domain, and when it learns that it should *prefer* those features when generalizing.

We plan to release all our data, pretrained models, and code upon acceptance.

2 Methods

Learning Inductive Bias Any finite set of training examples shown to a learning algorithm like a neural network is in principle consistent with infinitely many generalizable decision functions. Inductive biases are a model'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). An RNN classifier, for instance, is capable of representing *any* function, but prefers functions that focus mostly on local relationships within the input sequence (Dhingra et al., 2018; Ravfogel et al., 2019).

Crucially, inductive biases need not be immutable properties of learning algorithm or model architecture. In the language model fine-tuning 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. When such a model is fine-tuned on a downstream task, a pretrained model undoubtedly navigates the hypothesis differently than a model with a similar architecture without pretraining. In this sense a model like BERT *learns* inductive biases through pretraining.

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,

	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.

or if it will fall back on a combination of surface features that account for most of the data.

Measuring Inductive Bias To probe these biases in RoBERTa, we adopt the poverty of the stimulus design introduced by Wilson (2006), which we modify slightly with a data inoculation condition. Figure 1 gives an overview of the design:

First, we fine-tune the model on mixed data that is compatible with two reasonable but very different generalizations. For example, in the training data in Figure 1, every sentence in the positive class has two unique properties distinguishing it from the negative class sentences: the main verb appears in the progressive form (our linguistic feature), and the word "the" precedes the word "a" (our surface feature). The model's inductive biases determine which of these features it recognizes and generalizes based on.

After training, we use unmixed test data to observe which generalization the model converges on. In the unmixed data, the two features are perfectly *anti*-correlated, and the two hypotheses lead to opposite predictions. In the test data in Figure 1, it is *never* the case that "the" precedes "a" in a sentence with a main verb in the progressive form. If the model makes a linguistic generalization, its predictions will depend only on the form of the main verb and not on the relative position of "the" and "a", and vice-versa if it makes the surface generalization.

We also experiment with introducing small amounts of *inoculating* data. For each experiment, we introduce different amounts of unmixed data to the mixed training data, and rerun all the experiments. The sizes of inoculating unmixed data are 1%, 3%, and 10% of the size of the mixed data. These experiments are meant to test a model's sensitivity to *weak* signals in favor of linguistic generalizations.

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 all mix signals from a linguistic feature and a surface feature.

Features under Study Table 1 illustrates the 4 linguistic features and 5 surface features we consider.[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.

[1] In developing MSGS we explored a slightly larger set of linguistic features. We excluded several based on initial experiments showing our models did not robustly encode them.

[2] Although these are all surface properties of the original string, they are not all trivial for RoBERTa due to its use of subword tokenization. For instance, the case of individual characters and the presence of word boundaries must be inferred for individual token types.

[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. For instance, in *Sue is eager to sleep*, the NP *Sue* is the subject of *eager*, but *Sue* is also understood as the subject of *sleep*. This contrasts with the *raising* construction in *Sue is likely to sleep*, where *Sue* is

Dom.	Split	L_L	L_S	Sentence
In	Train (Mixed)	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. (Unmixed)	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 (Unmixed)	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. (Mixed)	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*.

Dataset Structure and Evaluation MSGS contains 20 *mixed* binary classification tasks where the signal from each of the 4 linguistic features has been mixed with the signal from each of the 5 surface features. MSGS includes an additional 9 unmixed *control tasks*—one for each feature. For mixed tasks, data is generated in paradigms of 8 sentences following a 2 × 2 × 2 design, as shown in Table 2. In addition to a binary linguistic feature and a binary surface feature, we also vary the domain from which the sentence is sampled (see §3). In some tasks, this means the in-domain and out-of-domain sentences in a given paradigm bear little resemblance, as in Table 2. The mixed in-domain sentences where both labels are the same fall into the training set, while the unmixed in-domain sentences with non-matching labels are reserved for the inoculation experiments. The unmixed out-of-domain test data is used to evaluate model bias in our main results, and mixed out-of-domain auxiliary data is used to measure how well the model adapts to the out-of-domain data, regardless of which generalization it makes. We write $\text{FEAT}_1 \times \text{FEAT}_2$ to denote a task that mixes features FEAT_1 and FEAT_2. The unmixed control data is generated in paradigms of 4 sentences following a 2 × 2 design by varying the feature and domain. The training and test sets both consist of examples from 5k paradigms, giving 10k training and test examples for each task.

To evaluate how a model generalizes, we look at its outputs on the unmixed test examples. We define the *linguistic bias score* (LBS) as the Matthews correlation coefficient between the outputs and the linguistic labels on a test set (Matthews, 1975). If LBS is 1, the learner shows a systematic linguistic bias. If LBS is -1, the learner shows a systematic surface bias. If LBS is 0, the learner fails to arrive at either generalization, and therefore shows no bias of either kind.

Data Generation The data is generated from templates using a generation toolkit from Warstadt et al. (2019). This toolkit includes a vocabulary of over 3000 items labeled with grammatical features that allow for lexical variation in the data while maintaining grammatical well-formedness. Despite this, generated sentences often describe unlikely or implausible scenarios (e.g., *Every lawyer was sinking a canoe*). However, 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).

We want to be confident that systematic generalization on one of these datasets requires knowledge of the actual feature. The experimental logic would fail if, for example, a model could achieve a linguistic bias score of 1 on the SYNTACTIC POSITION dataset by making use of some surface heuristic. We take two precautions to guard against this:

First, we generate training data and test data for each dataset from separate in-domain and out-of-domain templates, so that 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 within a relative clause within a complement clause of a verb; while the out-of-domain data contrasts the main verb with a verb in a complement clause of a noun. In most

the subject of *sleep*, but is not a semantic argument of *likely*.

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 in total, 12 of which we pretrain from scratch. The remaining one is the RoBERTa-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. Since BookCorpus (Zhu et al., 2015) is no longer available, we collect similar data from Smashwords, the original source of BookCorpus.[5]

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

Model Sizes To prevent overfitting on small training sets, we include five model sizes in our search space. The detailed configurations of the model sizes are summarized in Table 3. We use RoBERTa-Base from Liu et al. (2019b) as our largest model size. Our other size 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).

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

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 only 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 our main experiments. 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 of the selected models are listed in the Appendix.

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 limiting training epochs to 5 without early-stopping for all experiments. These changes are based on pilot experiments in which we found that larger batch sizes were no more effective and that out-of-domain generalization on our tasks was stable after 5 epochs.

For each pretrained model selected, we fine-tune on every combination of learning rates and inoculation quantities, giving 12 runs per model per task. We evaluate model performance using LBS as described in §3.

5 Results & Discussion

Figure 2 plots the relationship between linguistic bias score, pretraining data, and inoculation data.

[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.

[6] We count tokens as whitespace-separated strings.

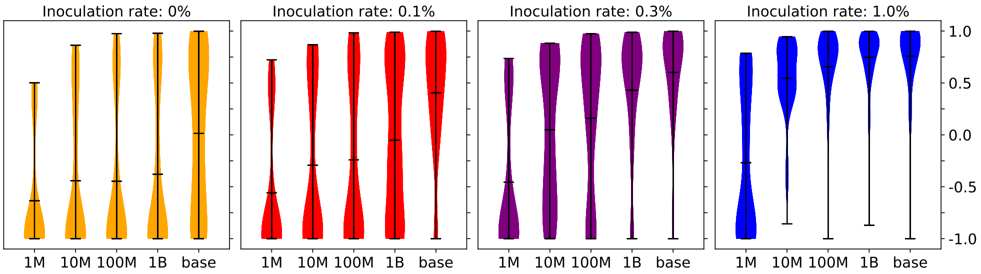

Figure 2: Results measured in LBS for each pretraining and inoculation data amount, aggregated over the 20 tasks in MSGS. Results for which the corresponding controls fail are excluded, as described in Section 5.

In this and 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 2, or represent them as gray points in Figure 3. Performance for the controls is near ceiling for all features except syntactic category and syntactic construction. This means all the models are able to perfectly extract these features in given texts. Results for the control tasks, training-condition data, inoculation-condition data, and auxiliary-condition data are given in the Appendix.

Pretraining strengthens linguistic bias Our main finding is that more pretraining data leads to a stronger linguistic bias. In Figure 2 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 phase 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. As is shown in the figure, the phase transition happens earlier for models with more pretraining, indicating they have a stronger linguistic bias. We notice distinctive behavior for the models at the extreme ends of pretraining data quantity: (1) The 1M model never completes the transition, suggesting that minimal linguistic bias can be acquired 1M words or fewer (2) RoBERTa-Base appears to already be in the middle of this transition with 0% inoculating data, suggesting that even more pretraining data would produce a model with a consistent linguistic bias.

These findings are echoed in individual task results in Figure 3[7]. In each plot, points with the same color (i.e. same amount of inoculating data) generally increase with pretraining size, suggesting that more pretraining data leads to a stronger linguistic bias. Notably, on tasks involving a linguistic feature mixed with LEXICAL CONTENT, RoBERTa-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 of the word “the”, RoBERTa-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 a generalization based on orthography with 1.0% inoculating data. The 100M and 1B models make robust linguistic generalizations only with 1.0% inoculating data. In contrast, RoBERTa-Base requires only 0.1% inoculating data (i.e. 10 out of 10k examples) to form a strong linguistic generalization.

Surface Biases of RoBERTa Our results also suggest some specific conclusions about which kinds of surface features RoBERTa does and does not bias. Specifically, as shown in the second column of figure 3, most of our models filtered by the control tasks form generalizations based on the linguistic features rather than the feature LENGTH with no inoculating data needed, suggesting a weak bias towards this feature. Similarly, the models show a relatively stronger bias towards ORTHOG-

[7] A black-and-white version of figure 3 can be found in the Appendix.

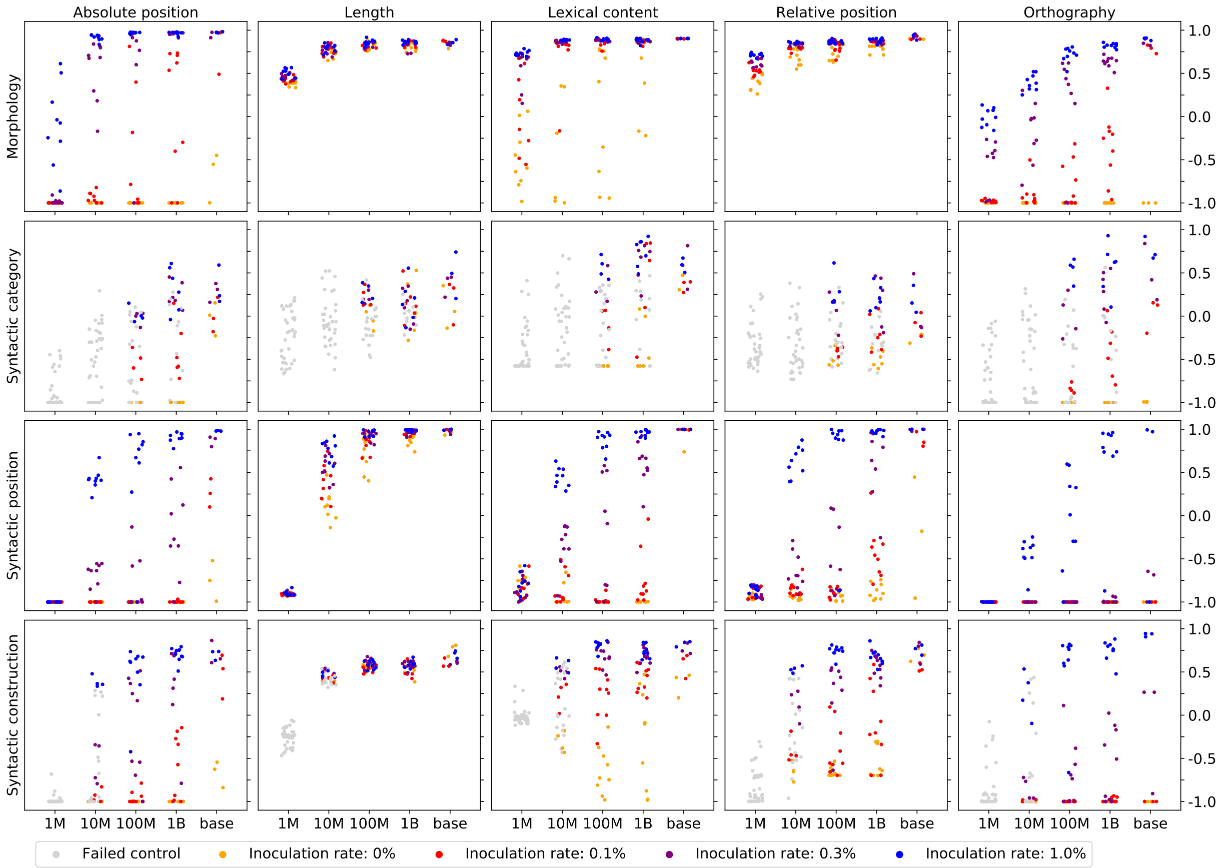

Figure 3: Results of the mixed binary classification tasks measured in LBS for every linguistic-surface feature pair. Each plot in the matrix shows the results on the unmixed test items after training a mixed task. All experiments on the same row investigate the same linguistic feature; all experiments 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. Gray points show runs where the corresponding controls did not pass.

RAPHY. This contrasts with features involving lexical content and word order, which appear to be relatively salient to our models.[8]

The success of pretrained models Our findings provide a new angle from which to view the widespread success of pretrained models like RoBERTa. Pretraining helps these models learn how to adapt to new target tasks by teaching them what kinds of features are central to language, and what kinds are not. Even though most models passed the controls, and therefore reliably represent both types of features, those with less pretraining nevertheless seem to prefer the surface features. In this way, language modeling pretraining can be seen as a kind of metalearning. The fact that RoBERTa-Base generally shows a linguistic bias aligns with its near-state-of-the-art performance on most language understanding tasks (Wang et al., 2018). The failure of the 1M model on the controls suggests that it has not learned sufficient basic linguistic features to be able to detect the correct generalization. This suggests a crucial data threshold below which language model pretraining is unlikely to be significantly helpful for most applications, 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 been too small or too slow 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, McCoy et al. (2019) find that supervised training on large crowdsourced textual entailment datasets like MultiNLI (Williams

[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.

et al., 2018) leads models like BERT to adopt some surface generalizations. As in our experiments, the models are given mixed signals by MultiNLI, though it is unintentional in this case: certain surface features such as lexical overlap are often correlated with the entailment label. Given a choice between a semantic generalization that accounts for the data generally, and shallow heuristics that work in subset of cases, the models opt for the latter.

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

Inductive biases can also be studied in a more abstract way. Using zero-shot learning in an artificial language, Lake and Baroni (2018) show that RNNs lack a bias towards ascribing a stable, compositional semantic content to new symbols. Gandhi and Lake (2019) and Gulordava et al. (2020) explore conditions under which neural networks do and do not exhibit a bias towards ascribing mutually exclusive semantic content to new symbols.

The concept of data augmentation using inoculating data has been explored previously as a way to change 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 mixed datasets including a shallow feature and a deep feature, though all their features including deep ones resemble our surface features. They find several differences between introducing inoculating data in favor of the deep generalization and data against the shallow generalization.

Finally, there have been prior related attempts to explore how increased self-supervised training data impacts linguistic generalizations in self-supervised models. Warstadt et al. (2019) and Hu et al. (2020) use an acceptability judgment task on minimal pairs (or sets) of sentences to evaluate language models trained on quantities of data ranging from <1M words to nearly 100M words. While Warstadt et al. (2019) find that increasing pretraining data in this range leads to steady increases in knowledge of acceptability (and by extension of linguistic features), Hu et al. (2020) find little to no effect. While our findings seem to align more closely with Warstadt et al.'s, a more comprehensive study of this learning curve would be valuable.

7 Future Work & Conclusion

Our experiments illuminate 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 preferentially 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. Another contribution is the 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 equip neural networks with better inductive biases.

References

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

Kevin Clark, Urvashi Khandelwal, Omer Levy, and

Christopher D. Manning. 2019. What does BERT look at? An analysis of BERT's attention. In *BlackBoxNLP@ACL*.

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 NAACL-HLT*, pages 199–209.

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

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

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

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*.

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.

Nelson F. Liu, Roy Schwartz, and Noah A. Smith. 2019a. Inoculation by fine-tuning: A method for analyzing challenge datasets. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

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.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. 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*.

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.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*.

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.

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, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R Bowman. 2019. BLiMP: A benchmark of linguistic minimal pairs for English. *arXiv preprint arXiv:1912.00582*.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019. Structural supervision improves learning of non-local grammatical dependencies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3302–3312.

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.