

Probing What Different NLP Tasks Teach Machines about Function Word Comprehension

Najoung Kim^{†,*}, Roma Patel^ϕ, Adam Poliak[†], Alex Wang[∂],
Patrick Xia[†], R. Thomas McCoy[†], Ian Tenney[△], Alexis Ross[◇],
Tal Linzen[†], Benjamin Van Durme[†], Samuel R. Bowman[∂], Ellie Pavlick^{ϕ,*}

[†]Johns Hopkins University ^ϕBrown University

[∂]New York University [△]Google AI Language [◇]Harvard University

Abstract

We introduce a set of nine challenge tasks that test for the understanding of function words. These tasks are created by structurally mutating sentences from existing datasets to target the comprehension of specific types of function words (e.g., prepositions, *wh*-words). Using these probing tasks, we explore the effects of various pretraining objectives for sentence encoders (e.g., language modeling, CCG supertagging and natural language inference (NLI)) on the learned representations. Our results show that pretraining on language modeling performs the best on average across our probing tasks, supporting its widespread use for pretraining state-of-the-art NLP models, and CCG supertagging and NLI pretraining perform comparably. Overall, no pretraining objective dominates across the board, and our function word probing tasks highlight several intuitive differences between pretraining objectives, e.g., that NLI helps the comprehension of negation.

1 Introduction

Many recent advances in NLP have been driven by new approaches to representation learning—i.e., the design of models whose primary aim is to yield representations of words or sentences that are useful for a range of downstream applications (Bowman et al., 2017). Approaches to representation learning typically differ in the architecture of the model used to learn the representations, the objective used to train that network, or both. Varying these factors can significantly impact performance on a broad range of NLP tasks (McCann et al., 2017; Peters et al., 2018; Devlin et al., 2019).

This paper investigates the role of pretraining objectives of sentence encoders, with respect to

their capacity to understand function words (e.g., prepositions, conjunctions). Although the importance of finding an effective pretraining objective for learning better (or more generalizable) representations is well acknowledged, relatively few studies offer a controlled comparison of diverse pretraining objectives, holding model architecture constant.

We ask whether the linguistic properties implicitly captured by pretraining objectives measurably affect the types of linguistic information encoded in the learned representations. To this end, we explore whether qualitatively different objectives lead to demonstrably different sentence representations. We focus our analysis on function words because they play a key role in compositional meaning—e.g., introducing and identifying discourse referents or representing relationships between entities or ideas—and are not yet considered to be well-modeled by distributional semantics (Bernardi et al., 2015). Our results suggest that different pretraining objectives give rise to differences in function word comprehension; for instance, we see that natural language inference helps understanding negation, and CCG supertagging helps recognizing meaningful sentence boundaries. However, overall, we find that the observed differences are not always straightforwardly interpretable, and further investigation is needed to determine which specific aspects of pretraining tasks yield good representations of function words.

The analyses we present contribute new results in an ongoing line of research aimed at providing a finer-grained understanding of what neural networks capture about linguistic structure (Conneau et al., 2018; Poliak et al., 2018b; Linzen et al., 2018; Tenney et al., 2019, *i.a.*). Our contributions are:

*Corresponding authors: Najoung Kim (n.kim@jhu.edu), Ellie Pavlick (ellie.pavlick@brown.edu)

Acceptability					
<i>wh</i> -word	why are you so chippy about posh people?				✓
	... a Mr. Nice Guy like Melcher, what is now 46				X
Def.	... the case is remarkable for the cooperation ...				✓
	... a case is remarkable for a cooperation ...				X
Coord.	I have also tried monthly data and the results are the same.				✓
	Rooms very clean but smelled very fresh.				X
EOS	the forehead is gathered in a frown // the mouth is slightly parted to reveal the teeth				✓
	the forehead is gathered in a frown the mouth // is slightly parted to reveal the teeth				X
NLI					
Prep.	With a single jerk the man’s head tore free.	→	The man’s head tore free from a single jerk.		✓
	With a single jerk the man’s head tore free.	→	The man’s head tore free without a single jerk.		X
Negation	This is a common problem.	→	This is not an uncommon issue we are facing.		✓
	This is not a common problem.	→	This is not an uncommon issue we are facing.		X
Spatial	To reach ... turn left up a small alleyway	→	do not turn right up the alleyway ...		✓
	To reach ... turn left up a small alleyway	→	Turn right up the alleyway ...		X
Quant.	all taken up yeah	→	There are not still some left		✓
	all taken up yeah	→	There are still some left		X
Comp.	Today there are more than 300,000.	→	Today there are not less than 300,000.		✓
	Today there are more than 300,000.	→	Today there are less than 300,000.		X

Table 1: Examples of sentences and sentence pairs corresponding to each of our probing datasets. The highlighted words are those that are relevant to the phenomena targeted by each set.

- We provide an in-depth exploration into how different pretraining objectives for sentence encoders affect the information encoded by the output representations. We isolate the effects of different pretraining objectives by holding the model architecture constant.
- We study function words, which have been under-studied in previous works on representation learning, but are critical to language understanding.
- We release nine new datasets,¹ quality-controlled by both linguists and non-linguist annotators, to facilitate ongoing work and follow-up analysis.

2 Function Word Probing Tasks

2.1 Approach

We introduce nine new probing tasks aimed at evaluating models’ understanding of function words. We focus on function words because although they are key building blocks of composi-

tional meaning and are highly frequent, they have received relatively little attention in the probing literature and in the distributional semantics literature. Each task targets the understanding of a specific type of function word; illustrative examples are given in Table 1. Our expectation is that different pretraining objectives (see Section 3.2) will yield sentence representations which measurably differ in their performance on these probing tasks.

We use two different formats for our probing tasks: acceptability judgment and natural language inference (NLI). The former uses a binary classification approach (acceptable/unacceptable) for probing a single sentence vector, in line with works such as Conneau et al. (2018) and Adi et al. (2017). The latter uses an entailment-based approach similar to White et al. (2017) and Poliak et al. (2018b), which is a ternary classification task (*entailment*, *contradiction*, *neutral*) over sentence pairs. The format is selected based on the suitability to the particular function word type in question.

To generate our probing datasets, we make structural modifications to sentences drawn from existing corpora, targeting a particular type of function word. We heuristically apply modifica-

¹The datasets are released as part of the Diverse Natural Language Inference Collection (DNC, Poliak et al., 2018b), available at <http://decomp.io>.

tions which we believe are likely to produce a specific label, and then recruit human annotators in order to produce the final labels used in our evaluations. The result is a publicly available suite of nine task datasets (four acceptability tasks and five NLI tasks) consisting of 3,544 annotated examples. Appendix C lists the sizes of each dataset.

2.2 Acceptability Judgment-Based Tasks

We cast acceptability as a binary classification task following the format of such judgments commonly used in linguistics, in a similar manner to Warstadt et al. (2018). All tasks follow a common protocol of first identifying sentences that contain the construction that we are interested in, and then mutating half of the identified sentences to generate infelicitous versions of the original sentences. Unless stated otherwise, the original sentences are drawn from the test set of the Billion Word Benchmark (BWB, Chelba et al., 2013).

Wh-Words Understanding *wh*-words (i.e., *who*, *what*, *where*, *when*, *why*, *how*) depends on understanding the context and correctly identifying the antecedent, which may not be overtly present in the sentence. For instance, recognizing the infelicity of *I talked about who I live* requires knowing that the (unstated) antecedent must be a place and not a person. Our dataset consists of sentences that contain one of the six *wh*-words listed above. Half of these sentences are mutated versions of the original which are generated by replacing the original *wh*-word with a different *wh*-word randomly selected from the remaining five options.

Definite-Indefinite Articles The definiteness task probes the understanding of definiteness that arises by the use of the definite article (*the*) versus indefinite articles (*a* and *an*). We find sentences containing multiple occurrences of *the* or multiple occurrences of *a*, and, for half of them, swap all such occurrences (i.e., replacing *the* with *a*² or vice-versa). This gives us four types of sentences: unchanged sentences with multiple definite articles, unchanged sentences with multiple indefinite articles, sentences with all definite articles replaced by the indefinite article, and sentences with all indefinite articles replaced by the definite article. Our intent is that the former two types will be judged felicitous while the latter two will be infelicitous despite the fact that the sentence would be

²When we replace *the* with *a*, we choose *alan* as necessary based on the word it precedes.

syntactically well-formed. We only focus on the cases with multiple occurrences of the same article, because replacing a single article most of the time did not significantly affect the acceptability (although it often did affect the actual meaning).

Coordinating Conjunctions Correct understanding of coordinating conjunctions (*and*, *but*, *or*) requires contextual comprehension of the two conjoined linguistic units, since different coordinating conjunctions express different logical relations, meaning their use is often restricted by the meanings of the conjoined items. We take sentences that contain coordinating conjunctions, and replace half of them with a version that contains a different conjunction. For example, the sentence *Room’s very clean but smelled very fresh* is infelicitous despite being syntactically well-formed; *but* is unnatural here because the conjoined clauses do not form a clear contrast. Judging this sentence to be infelicitous requires a proper understanding of the ideas expressed in the clauses and how they relate to each other.

End-of-Sentence The end-of-sentence (EOS) task tests a model’s ability to identify semantically coherent chunks (i.e., sentences) in running text. In written text this is often indicated by punctuation marks such as periods, but humans are able to easily identify sentences even without overt markers. Thus, we take pairs of sentences from the same paragraph of the WikiText-103 (Merity et al., 2017) test set and remove all punctuation marks and capitalization, and concatenate each sentence pair to create a line of running text.³ Half of the dataset consists of a pair of valid sentences, and the other half consists of a pair of potentially invalid sentences generated from an incorrect segmentation of the running text, where the incorrect segmentation index is obtained by sampling from a Gaussian distribution centered around the correct index ($\sigma = 2$) and rounding to the nearest integer.

2.3 NLI-Based Tasks

Our NLI-based probing tasks ask whether the choice of function word affects the inferences licensed by a sentence. These tasks consist of a pair of sentences—a premise *p* and a hypothesis *h*—and ask whether or not *p* entails *h*. We exploit the label changes induced by a targeted mutation of

³We use WikiText instead of BWB because adjacent sentences in BWB are not logically contiguous and therefore may not be from the same discourse context.

the sentence pairs taken from the Multi-genre Natural Language Inference dataset (MNLI, Williams et al., 2018). The rationale is that, if a change to a single function word in the premise changes the entailment label, that function word must play a significant role in the semantics of the sentence.

Prepositions We manually curate a list of prepositions (see Appendix D) that are likely to be swapped with each other without affecting the grammaticality of the sentence. We generate mutated NLI pairs by finding occurrences of the prepositions in our list and randomly replacing them with other prepositions in the list. Our list consists of a set of locatives⁴ and several other manually-selected prepositions that are not strictly locatives but are likely to be substitutable (*about, for, to, with, without*).

Comparatives Comparatives express qualitative or quantitative differences between entities. For instance, a sentence that states *A is more than B* and another that states *B is more than A* lead to different inferences. We select a list of common comparatives (e.g., *more/less, bigger/smaller*) and select pairs from MNLI that contain a comparative phrase in both the premise and the hypothesis. We mutate the sentences by negating the hypothesis.⁵

Quantification The quantification task tests the understanding of natural language expressions of quantities, including common quantifiers (*all, some*), number words (*two, twenty*), and proportion (*half, one-third, quarter*). We select NLI pairs that contain at least one quantifier in both the premise and the hypothesis, and again apply mutations by negating the hypothesis.

Spatial Expressions The spatial expressions task probes the understanding of words that denote spatial relations between entities. Changing the spatial configuration often leads to different inferences; for instance, *A is to the left of B* implies that *B is to the right of A*, but not that *A is to the right of B*. We select a set of words that describe spatial configurations which are not necessarily prepositions (e.g., *left, right, close, far*). Again, we find MNLI pairs containing these words and negate to generate mutated pairs.

⁴Locative prepositions are those that denote place or position: e.g., *in, on, near*, etc.

⁵We note that Dasgupta et al. (2018) also focus on comparatives, but they exclusively look at artificial sentences containing *more/less*.

Negation This task probes whether models are able to understand negations, in particular explicit negation using the word *not*, lexical negation using antonyms, and the interaction between them. We first identify premise-hypothesis pairs from the MNLI dataset that contain antonym pairs (e.g., *dirty* appears in *p* and *clean* in *h*) and generate all possible patterns of negation with the two mutation strategies: swapping antonyms and adding explicit negation. That is, we use each of lexical negation, explicit negation, and their combination to mutate the premise and/or the hypothesis. We generate all 16 possible patterns of negation for a given premise-hypothesis (*p, h*) pair. For each of *p* and *h* we can either apply or not apply each of four possible mutations: lexical negation, explicit negation, both, and none.

2.4 Annotation

We recruit human annotators on Amazon Mechanical Turk to produce the final labels for the heuristically-generated datasets described above. We collect three labels per sentence (or per pair of sentences for EOS and NLI probing sets). We use the majority label in our final dataset, and discard examples on which there is no majority consensus. For more details about our annotation protocol, including compensation, refer to Appendix C.

Acceptability Tasks Human annotators are presented with a single (mutated or unmutated) sentence and are given the options $\{\textit{natural, unnatural, neither}\}$. We discard sentences in which the majority label does not agree with our expected label. That is, we only include mutated sentences with a majority label of *unnatural* and unmutated sentences with a majority label of *natural*. We collect around 500 annotated examples with balanced label ratio for each probing set. We release our sentences in small batches until we have approximately 250 *unnatural* examples per task. To create the final dataset, we pool all answers from all batches and take a subset of the *natural* sentences so that the label ratio is balanced, prioritizing examples with perfect inter-annotator agreement.

Natural Language Inference Tasks For the NLI tasks, we collect common-sense entailment judgments from annotators on a 5-point Likert scale on which 1 denotes ‘definitely contradiction’ and 5 denotes ‘definitely entailment’, following Zhang et al. (2017). This finer-grained scale is intended to avoid confounds arising from borderline

cases. Except for the use of scaled judgments, our instructions follow the MNLI guidelines. Specifically, our instructions said to assume that the sentences co-refer and that the first sentence (p) states a true fact, describes a scenario, or expresses an opinion, and to then indicate how likely it is that the second sentence (h) is also true, describes the same scenario, or expresses the same opinion.

Annotators could also select an option indicating that one or both of the sentences did not make sense; we discarded (p, h) pairs for which at least one annotator chose this option. We map judgments of 5 and 4 to *entailment*, 3 to *neutral*, and 2 and 1 to *contradiction*, and treat the majority label as the correct label after this mapping.

Agreement and Quality Control In constructing our final evaluation sets, we removed examples on which there was no majority consensus. For the binary acceptability tasks, we manually prefiltered sentences that were felicitous even after the heuristic modification. For the NLI tasks, we removed pairs that contained ungrammatical sentences that were not flagged by annotators via manual postfiltering. See Appendix C for more details.

3 Experimental Design

3.1 Pretraining Architecture

Since our focus is on comparing differences in pretraining objectives, we fix the architecture for all sentence encoders. We use the pre-trained character-level convolutional neural network (CNN) from ELMo (Peters et al., 2018) that replaces word embeddings (see Bowman et al. (2018) or Tenney et al. (2019) for similar usages of the CNN layer). This acts as a base input layer that uses no information beyond the word, and allows us to avoid potentially difficult issues surrounding unknown word handling in transfer learning.

We feed the word representations to a 2-layer 1024d bidirectional LSTM (Hochreiter and Schmidhuber, 1997). A downstream task-specific model sees both the top-layer hidden states of this model and, through a skip connection, the original representation of each word. We train a version of this model on each task in Section 3.2. Additional experimental details are in given in Appendix A. Our codebase is open-source⁶ and built

⁶<https://github.com/jsalt18-sentence-repl/jiant>

using AllenNLP (Gardner et al., 2017) and PyTorch (Paszke et al., 2017).

Classification Tasks For classification pretraining tasks (NLI, DisSent), we use an attention mechanism inspired by BiDAF (Seo et al., 2017). Given the sequence of output states of the core BiLSTM for both sentences in an example, we compute dot-product based attention between all pairs of words between the sentences to form a sequence of attention-contextualized word representations. We use an additional BiLSTM followed by max-pooling to obtain an attention-contextualized vector representation of each sentence h_1 and h_2 . We use the *heuristic matching* feature vector $[h_1; h_2; h_1 \cdot h_2; |h_1 - h_2|]$ (Mou et al., 2016) as input to an MLP.

Sequence-to-Sequence Tasks For sequence-to-sequence pretraining tasks (machine translation and skip-thought), we use a single-layer 1024d LSTM as the decoder, initialized with the max-pooled output of the encoder. We use a linear projection bottleneck layer to reduce the dimension of the output of the decoder by half before the output softmax layer.

3.2 Pretraining Tasks

Our main experiments compare seven pretraining tasks which we believe capture different aspects of linguistic meaning and which yield reasonable performance when used on a benchmark task such as MNLI.⁷ For our purposes, a *task* is a dataset-training objective pair. We attempt to select a set of tasks diverse enough to highlight performance differences due to pretraining objectives. We additionally report results using BERT (Devlin et al., 2019) (base, uncased) to demonstrate that our probing sets prove challenging even for state-of-the-art models.

Language Modeling We train a left-to-right word-level language model on BWB, which was successfully used by Peters et al. (2018) for pretraining sentence encoders. Because language modeling is trivial for a bidirectional LSTM, we follow Peters et al. (2018) by training separate forward and backward two-layer 1024d language models and concatenate their hidden states as token representations.

⁷Around 70% development set accuracy; see Appendix B.

Skip-Thought Drawing from [Kiros et al. \(2015\)](#) and [Tang et al. \(2017\)](#), we train a sequence-to-sequence model on skip-thought, which is a task of generating the next sentence in the discourse given the previous sentence. We use the learned encoder as our sentence encoder. Since this objective requires running text, we use sentences from WikiText-103 as training data.

CCG Supertagging We train a model to predict the Combinatory Categorical Grammar (CCG) supertag for each word, with sentences from CCG-Bank ([Hockenmaier and Steedman, 2007](#)). Supertags are similar to part-of-speech tags but capture more syntactic context (“almost-parsing”; [Bangalore and Joshi, 1999](#)).

Discourse (DisSent) We train a model on DisSent ([Jernite et al., 2017](#); [Nie et al., 2017](#)), which is an unsupervised task of predicting the discourse marker (e.g., *and*, *because*, or *so*) that connects two clauses. We train our model on a dataset created from WikiText-103 following [Nie et al. \(2017\)](#)’s protocol, which involves extracting pairs of clauses with a specific dependency relation.

Natural Language Inference Inspired by [Conneau et al. \(2017\)](#), we use the MNLI dataset for NLI pretraining. The task is to predict the entailment label for premise-hypothesis pairs; the possible labels are *entailment*, *contradiction*, *neutral*.

Machine Translation We train a sequence-to-sequence machine translation model with attention on WMT14 English-German ([Bojar et al., 2014](#)) and take the encoder as our sentence encoder. [McCann et al. \(2017\)](#) previously showed that pretraining an encoder on translation led to good performance on downstream NLP tasks.

Image-Caption Matching We train a model on the task of grounding sentences to the images they describe. We use image-caption pairs from the MSCOCO dataset ([Lin et al., 2014](#)) with an objective that minimizes the cosine distance between sentence representations and corresponding image features, as described in [Kiela et al. \(2018\)](#).

3.3 Classifiers for Probing Tasks

To probe the sentence encoders pretrained on the different objectives, we freeze the weights of the encoder after pretraining and train an additional model using the outputs of the fixed encoder as

inputs. We describe the implementation details for the NLI and acceptability probing sets below.

NLI Tasks For NLI-type probing, we train an NLI model on top of the representations produced by the pretrained sentence encoder that uses an attention mechanism inspired by [Seo et al. \(2017\)](#) that computes attention between all pairs of words in the two sentences (described in more detail in Section 3.1). We train this component on MNLI and evaluate directly on our NLI probing datasets with no further dataset-specific training.

Acceptability Classification Tasks For all acceptability tasks except the EOS task, we take the sequence of hidden state outputs from the pretrained encoder as the sentence representation. We aggregate this sequence into a single vector via max-pooling and train a 512d MLP on top of the resulting vector. For the EOS task, we also use max-pooling on each sentence in the pair. We then concatenate the resulting vectors and train an MLP on top of the joint representation.⁸ Each task has around 400 training examples (see Appendix C). Due to their small size, we use 10-fold cross validation where each fold is used as the test set exactly once, and report the average test set accuracy.

BERT For NLI-type probing tasks, we use the fine-tuned MNLI classifier from ([Devlin et al., 2019](#))⁹. For the acceptability classification tasks, we fine-tune the model by adding a sequence-level classifier on top of the pretrained BERT model. The sequence-level classifier is a linear layer that takes in as input the final hidden vector corresponding to the first input token as aggregate representation in the input sequence, and then classifies to the required number of classes for the task, where the label probabilities are computed with a standard softmax. The BERT fine-tuning setup allows a classification output to be indicated with a CLS token. Pairs of sequences are indicated with a SEP token between the pairs. All parameters are fine-tuned jointly to maximize the log-probability of the correct label while the hyperparameters are the same as in pretraining.

⁸We tried training a general acceptability model using CoLA and evaluating directly on our acceptability tasks, as an analogous evaluation setup to the NLI tasks, but all models performed around chance under this setup. This is likely due to the intrinsic difficulty of CoLA for our base model, as suggested by low performance from similar models (“GLUE Baselines”) on <https://gluebenchmark.com>.

⁹<https://github.com/google-research/bert>

whwords	0.50	0.51	0.63	0.53	0.55	0.52	0.54	0.55	0.67	0.50	0.86
spatial	0.46	0.54	0.45	0.55	0.62	0.40	0.40	0.55	0.54	0.76	0.86
quant	0.48	0.62	0.58	0.61	0.65	0.60	0.54	0.61	0.60	0.78	0.87
prep	0.37	0.46	0.45	0.47	0.50	0.41	0.42	0.47	0.45	0.52	0.77
neg	0.40	0.55	0.50	0.56	0.59	0.49	0.50	0.55	0.53	0.65	0.80
eos	0.50	0.51	0.82	0.59	0.62	0.70	0.61	0.68	0.71	0.90	0.82
def	0.50	0.59	0.66	0.60	0.61	0.62	0.59	0.61	0.72	0.52	0.86
conj	0.50	0.53	0.61	0.63	0.68	0.55	0.57	0.59	0.63	0.97	0.88
comp	0.49	0.48	0.48	0.55	0.53	0.54	0.45	0.56	0.55	0.74	0.85
all	0.47	0.54	0.60	0.57	0.60	0.55	0.54	0.58	0.62	0.69	0.83
	majority	rand	ccg 38K	dis 151K	nli 393K	img 592K	skip 4M	mt 3.4M	lm 30.3M	bert	human

Figure 1: Accuracy for each pretraining task on each probing set. The leftmost column shows the majority-class baseline, and the rightmost column shows individual annotator accuracy on the final probing set. Blue denotes performance improvement over randomly initialized encoder baseline and orange denotes performance decrease.

3.4 Variation from Random Restarts

In order to calibrate the degree of variation that can be expected due to random restarts, we run each of our probing tasks on five different random initializations of the sentence encoder weights. These sentence encoders were not pretrained, and we trained MLPs for each probing task on top of the randomly initialized sentence encoders. The expectation is that if pretraining has measurable effects on the probing results, the variance across different pretrained models would be greater than the variance across random restart models. Across five random restarts, the average standard deviation across our probing set was around 1 percentage point. The mean and standard deviation for each probing task are reported in Appendix E.

4 Results

4.1 Overall Performance

Figure 1 shows the performances of models trained on each pretraining task on our probing datasets. We also provide comparison with a randomly initialized encoder with no pretraining, which is known to be a strong baseline (Bowman et al., 2018). We observe that different pretraining tasks have different strengths and weaknesses; there is no single pretraining task that achieves the best (or worst) performance across the board. This implies that even the best encoders, such as BERT, are unable to capture function word semantics fully, and suggests further research into combining advantages of different tasks. Furthermore, most models are far from human performance, with only a few exceptions (e.g., BERT on conjunctions). This demonstrates that our probing

datasets serve as useful challenge sets, in addition to permitting fine-grained analysis.

Looking into each probing set in more detail, we see several intuitive patterns on how pretraining might affect probing performance. Among the pretrained models (not including BERT), the NLI model outperformed all other models on the negation¹⁰ and conjunction tasks, both of which involve words that play central roles in inferential reasoning. The CCG model yields the best result for EOS, which could be attributed to the task’s emphasis on structure; it is the only task where the target labels directly represent compositional structure.

Surprisingly, we find that pretraining can sometimes hurt performance. For instance, pretraining on the skip-thought objective or the image captioning objective appears to hurt performance on spatial relations and prepositions (compared to the performance of a randomly initialized encoder). In fact, for many probing sets, the choice of pretraining task affects whether it helps or hurts performance; for instance, pretraining on NLI helps with negation, whereas pretraining on image-caption matching and CCG lowers performance. This suggests that pretraining can be helpful, but *only* helpful if we pretrain on a task that provides useful information in solving the target task. For instance, in Section 4.3 we discuss how the image-caption matching objective may bias models to discard information about certain preposition senses. Overall, we observe that language modeling is the

¹⁰We additionally find that this improvement is specifically due to the NLI model’s capacity to understand explicit negation using *not*, rather than lexical negation with antonymy. See Appendix F for differences between negation subtypes.

most useful pretraining task, although CCG and NLI perform comparably (within random variation range). The strong performance of language modeling aligns with its effectiveness for pretraining models that achieve state-of-the-art NLP results. However, we note that CCG, our most syntactic task, achieves competitive results with the smallest number of training examples (an order of magnitude less data than NLI and two orders less than language modeling), suggesting that incorporating syntactic knowledge might yield more efficient learning of function word semantics.

We furthermore see that many of our probing sets are challenging even for BERT—although BERT substantially improves performance on many probing sets, and obtains superhuman performance on conjunctions and EOS,¹¹ it also shows clear weaknesses in several probing sets (e.g., *wh*-words and definite/indefinite articles) where it is outperformed even by a randomly initialized baseline with no pretraining.

4.2 Correlations between Pretraining Tasks

To further investigate whether our probing sets differentiate between pretraining objectives, we look into correlations between the model predictions; given two pretraining tasks i and j , how often does a model trained on i make the exact same prediction as a model trained on j ? Figure 2 shows the correlations across all probing sets in aggregate, and for the *wh*-words and prepositions sets specifically (see Appendix G for all sets).

We observe that models pretrained on different tasks do make different predictions overall, with image-caption matching and skip-thought being the tasks that make predictions that deviate the most from others (left). NLI and image-caption matching are the least correlated pair of tasks among all. The difference between image-captioning and other tasks is the most prominent in the preposition probing set; it makes predictions that are only weakly correlated with others (middle). We hypothesize that this is due to the duality of preposition semantics; most prepositions have both concrete and abstract senses, and the image model is biased to focus on the former.

To illustrate, consider the preposition *below*, which can denote a spatial configuration (e.g., *the boots end below the knee*) or an abstract rela-

tion (numeric or qualitative comparison; e.g., *her score is below sixty*). In the preposition dataset, *below* occurs 17 times, 11 of which are spatial and 6 abstract. For the spatial usage, both MNLI and image-caption models have 64% accuracy (7/11). The NLI model shows 50% accuracy for pairs containing abstract uses (3/6), but the image-captioning model answers none of them correctly (0/6). Here is an example of a numeric usage of *below* that the NLI model answered correctly but the image model answered incorrectly:

P: *Only those whose incomes do not exceed 125 percent of the federal poverty level qualify . . .*

H: *Those whose incomes are below 125 percent qualify . . .* (P→H)

The image model’s bias towards the spatial usage is intuitive, since the numeric usage of *below* (i.e., as a counterpart to *exceed*) is difficult to learn from visual clues only. This concrete-abstract duality, which is not specific to *below* but common to most other prepositions (Schneider et al., 2018), may partially explain why the image-caption model behaves so differently from all other models, which are not trained on a multimodal objective.

4.3 Data Size and Genre Effects

As can be seen from the varying sizes of the pretraining dataset reported in Figure 1, seeing more data during pretraining does not imply better performance on probing tasks. Also, as noted before, the fact that pretraining can hurt performance suggests that if the task is not the “right” task, adding more datapoints during pretraining can lead models to learn counterproductive representations.

Another potential confound is vocabulary overlap between pretraining and probing task datasets. Since all pretraining task datasets have different sets of vocabulary, the variance in the results could be attributed to the number of words in the probing set that were already seen in pretraining. To investigate this possibility, we compute the ratio of overlapping words between the pretraining and probing datasets. A regression analysis shows that vocabulary overlap overall is not a significant predictor of performance on the probing set ($p = 0.39$). No single probing set performance was significantly affected by vocabulary overlap either (all $p > .05$ after Bonferroni correction for multiple comparisons).

¹¹We speculate that this might be an effect of the next-sentence classification task that BERT is pretrained on.

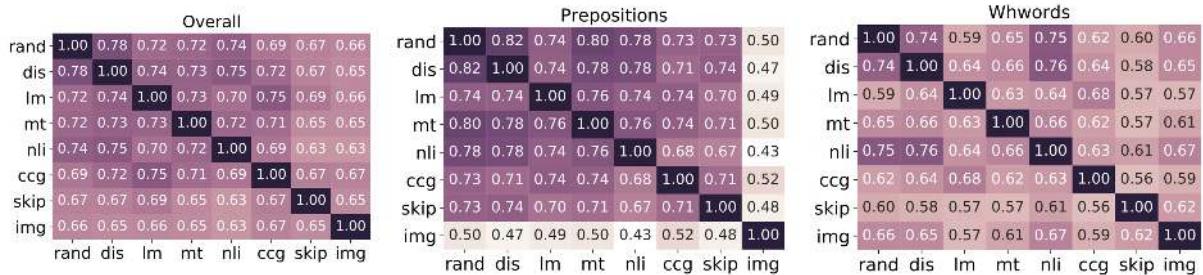


Figure 2: Prediction overlap on the probing tasks for models trained on different pretraining tasks (i.e., how often models make identical predictions on a particular probing set).

5 Related Work

An active line of work focuses on “probing” neural representations of language. [Ettinger et al. \(2016, 2017\)](#); [Zhu et al. \(2018\)](#), *i.a.*, use a task-based approach similar to ours, where tasks that require a specific subset of linguistic knowledge are used to perform qualitative evaluation. [Gulordava et al. \(2018\)](#), [Giulianelli et al. \(2018\)](#), [Rønning et al. \(2018\)](#), and [Jumelet and Hupkes \(2018\)](#) make a focused contribution towards a particular linguistic phenomenon (agreement, ellipsis, negative polarity). Using recast NLI, [Poliak et al. \(2018a\)](#) probe for semantic phenomena in neural machine translation encoders. [Staliūnaite and Bonfil \(2017\)](#); [Mahler et al. \(2017\)](#); [Ribeiro et al. \(2018\)](#) use similar strategies to our structural mutation method, although their primary goal was to break existing systems by adversarial modifications rather than to compare different models. [Ribeiro et al. \(2018\)](#) and our work both test for proper comprehension of the modified expressions, but our modifications are designed to induce semantic changes whereas their modifications are intended to preserve the original meaning. Our strategy is close to that of [Naik et al. \(2018\)](#), but our modifications are more constrained and lexically targeted.

The design of our NLI-style probing tasks follows the recent line of work which advocates for NLI as a general-purpose format for diagnostic tasks ([White et al., 2017](#); [Poliak et al., 2018b](#)). This idea is similar in spirit to [McCann et al. \(2018\)](#), which advocates for question answering as a general-purpose format, to edge probing ([Tenney et al., 2019](#)) which probes for syntactic and semantic structures via a common labeling format, and to GLUE ([Wang et al., 2018](#)) which aggregates a variety of tasks that share a common sentence-classification format. The primary difference in our work is that we focus specifically on the un-

derstanding of function words in context. We also present a suite of several tasks, but each one focuses on a particular structure, whereas tasks proposed in the works above generally aggregate multiple phenomena. Each of our tasks isolates each function word type and employ a targeted modification strategy that gives us a more narrowly-focused, informative scope of analysis.

6 Conclusion

We propose a new challenge set of nine tasks that focus on probing function word comprehension. Although we use our challenge set to compare the effects of pretraining, the probing sets themselves are architecture- and evaluation setup-agnostic. The results show that models pretrained with different objectives do generate different predictions (e.g., image models have a bias towards concrete preposition senses), and that no single objective leads to models that perform best or worst across all tasks. This suggests that there are ‘gaps’ in the linguistic knowledge learned from a single pretraining objective that could be complemented by others, and calls for further research into how different pretraining objectives could be productively combined. In addition to contributing to the discussion of finding effective pretraining tasks, we hope that our exploratory study initiates further discussions about modeling function words and their contribution to compositional meaning.

Acknowledgments

The work was conducted at the 2018 Frederick Jelinek Memorial Summer Workshop on Speech and Language Technologies, and supported by Johns Hopkins University with unrestricted gifts from Amazon, Facebook, Google, Microsoft and Mitsubishi Electric Research Laboratories.

References

- Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations*.
- Srinivas Bangalore and Aravind K Joshi. 1999. Supertagging: An approach to almost parsing. *Computational Linguistics*, 25(2):237–265.
- Raffaella Bernardi, Gemma Boleda, Raquel Fernandez, and Denis Paperno. 2015. **Distributional semantics in use**. In *Proceedings of the First Workshop on Linking Computational Models of Lexical, Sentential and Discourse-level Semantics*, pages 95–101, Lisbon, Portugal. Association for Computational Linguistics.
- Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. **Findings of the 2014 workshop on statistical machine translation**. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.
- Samuel Bowman, Yoav Goldberg, Felix Hill, Angeliki Lazaridou, Omer Levy, Roi Reichart, and Anders Sgaard, editors. 2017. *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*. Association for Computational Linguistics, Copenhagen, Denmark.
- Samuel R. Bowman, Ellie Pavlick, Edouard Grave, Benjamin Van Durme, Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, and Berlin Chen. 2018. **Looking for ELMo’s friends: Sentence-level pretraining beyond language modeling**. *CoRR*, abs/1812.10860.
- Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv:1312.3005*.
- Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.
- Alexis Conneau, Germán Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. **What you can cram into a single \$&!#* vector: Probing sentence embeddings for linguistic properties**. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136. Association for Computational Linguistics.
- Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv:1802.04302*.
- 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 Papers)*.
- Allyson Ettinger, Ahmed Elgohary, and Philip Resnik. 2016. **Probing for semantic evidence of composition by means of simple classification tasks**. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 134–139. Association for Computational Linguistics.
- Allyson Ettinger, Sudha Rao, Hal Daumé III, and Emily M. Bender. 2017. **Towards linguistically generalizable NLP systems: A workshop and shared task**. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 1–10, Copenhagen, Denmark. Association for Computational Linguistics.
- Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A deep semantic natural language processing platform.
- Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. 2018. **Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information**. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium. Association for Computational Linguistics.
- Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. **Colorless green recurrent networks dream hierarchically**. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1195–1205, New Orleans, Louisiana. Association for Computational Linguistics.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 770–778.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

- Julia Hockenmaier and Mark Steedman. 2007. [Ccg-bank: A corpus of ccg derivations and dependency structures extracted from the penn treebank](#). *Computational Linguistics*, 33(3).
- Yacine Jernite, Samuel R. Bowman, and David Sontag. 2017. Discourse-based objectives for fast unsupervised sentence representation learning. *arXiv:1705.00557*.
- Jaap Jumelet and Dieuwke Hupkes. 2018. [Do language models understand anything? On the ability of lstms to understand negative polarity items](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 222–231, Brussels, Belgium. Association for Computational Linguistics.
- Douwe Kiela, Alexis Conneau, Allan Jabri, and Maximilian Nickel. 2018. [Learning visually grounded sentence 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)*, pages 408–418, New Orleans, Louisiana. Association for Computational Linguistics.
- Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302.
- Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer.
- Tal Linzen, Grzegorz Chrupala, and Afra Alishahi, editors. 2018. *Proceedings of the First Workshop on Analyzing and Interpreting Neural Networks for NLP (BlackboxNLP)*. Association for Computational Linguistics, Brussels, Belgium.
- Taylor Mahler, Willy Cheung, Micha Elsner, David King, Marie-Catherine de Marneffe, Cory Shain, Symon Stevens-Guille, and Michael White. 2017. [Breaking NLP: Using morphosyntax, semantics, pragmatics and world knowledge to fool sentiment analysis systems](#). In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 33–39. Association for Computational Linguistics.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.
- Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv:1806.08730*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models. In *International Conference on Learning Representations*.
- Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. [Natural language inference by tree-based convolution and heuristic matching](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany. Association for Computational Linguistics.
- Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353.
- Allen Nie, Erin D. Bennett, and Noah D. Goodman. 2017. [DisSent: Sentence representation learning from explicit discourse relations](#). *arXiv:1710.04334*.
- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. *NIPS 2017 Autodiff Workshop*.
- 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)*, pages 2227–2237. Association for Computational Linguistics.
- Adam Poliak, Yonatan Belinkov, James Glass, and Benjamin Van Durme. 2018a. [On the evaluation of semantic phenomena in neural machine translation using natural language inference](#). 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 513–523, New Orleans, Louisiana. Association for Computational Linguistics.
- Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018b. [Collecting diverse natural language inference problems for sentence representation evaluation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 67–81. Association for Computational Linguistics.
- Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. 2018. On the convergence of Adam and beyond. In *International Conference on Learning Representations*.

- Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. [Semantically equivalent adversarial rules for debugging nlp models](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865. Association for Computational Linguistics.
- Ola Rønning, Daniel Hardt, and Anders Søgaard. 2018. [Linguistic representations in multi-task neural networks for ellipsis resolution](#). In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 66–73. Association for Computational Linguistics.
- Nathan Schneider, Jena D. Hwang, Vivek Srikumar, Jakob Prange, Austin Blodgett, Sarah R. Moeller, Aviram Stern, Adi Bitan, and Omri Abend. 2018. [Comprehensive supersense disambiguation of english prepositions and possessives](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 185–196. Association for Computational Linguistics.
- Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *International Conference on Learning Representations*.
- Ieva Staliūnaite and Ben Bonfil. 2017. [Breaking sentiment analysis of movie reviews](#). In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 61–64. Association for Computational Linguistics.
- Shuai Tang, Hailin Jin, Chen Fang, Zhaowen Wang, and Virginia de Sa. 2017. [Rethinking skip-thought: A neighborhood based approach](#). In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 211–218. Association for Computational Linguistics.
- Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. 2019. [What do you learn from context? probing for sentence structure in contextualized word representations](#). In *International Conference on Learning Representations*.
- Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv:1804.07461*.
- Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv:1805.12471*.
- Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. [Inference is everything: Recasting semantic resources into a unified evaluation framework](#). In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 996–1005. Asian Federation of Natural Language Processing.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.
- Sheng Zhang, Rachel Rudinger, Kevin Duh, and Benjamin Van Durme. 2017. Ordinal common-sense inference. *Transactions of the Association of Computational Linguistics*, 5(1):379–395.
- Xunjie Zhu, Tingfeng Li, and Gerard de Melo. 2018. [Exploring semantic properties of sentence embeddings](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 632–637, Melbourne, Australia. Association for Computational Linguistics.

Appendix

A Experimental Details

Image-Caption Matching We train on image-caption pairs from the MSCOCO dataset (Lin et al., 2014) to minimize the cosine distance between representations of the sentence and corresponding image. Specifically, we encode the sentence with the BiLSTM encoder. We use a Resnet-101 (He et al., 2016), a CNN pretrained on ImageNet to obtain a 1024-dimensional feature representation for the image. We linearly transform the encoder output of the sentence to the size of the image representation and use a cosine embedding loss against the two vectors, i.e., minimize the cosine distance between two representations to allow mapping sentences to their corresponding images.

Regularization We regularize with dropout with $p = 0.2$. Dropout is placed after the input layer, each LSTM layer, and each MLP layer in the task-specific classifier.

Preprocessing We use Moses tokenizer with a maximum sequence length of 40 tokens. Because we used a character-based word encoder, we have no word-level vocabulary, except for sequence-to-sequence tasks, where we use an output vocabulary of 20,000 tokens. For translation, we use BPE tokenization; for skip-thought we use the Moses tokenizer.

Training Details We optimize using AMSGrad (Reddi et al., 2018) with a learning rate of $1e-3$ for text generation tasks and $1e-4$ otherwise. We evaluate on the validation set every 1,000 iterations and stop training if we fail to get a best result after 20 evaluations. We multiply the learning rate by 0.5 whenever validation performance fails to improve for more than 4 validation checks. We also stop training if the learning rate falls below $1e-6$. At the end of training, we load the best checkpoint.

Acceptability task evaluation For the acceptability tasks, we use a 10-fold cross-validation evaluation setup because we are training task-specific classifiers for each probing task and the datasets are small. We split each dataset into 10 folds with balanced label ratio within each fold, and test on each fold using the other 9 as train and development sets (8 folds train, 1 fold dev). The accuracy reported in the paper for the acceptability tasks is test accuracy averaged across all folds.

B MNL Development Set Accuracy for Pretrained Models

	MNLI (dev)
Random	73.8
CCG	69.6
DisSent	73.6
MNLI	75.6
IMG	62.6
Skip	67.4
MT	72.0
LM	72.6

Table 2: MNLI development set accuracy for each pretrained model.

C Annotation Protocol

We recruited three annotators per sentence or sentence pair on Amazon Mechanical Turk to control the quality of the labels for our heuristically generated datasets. For the acceptability judgment task sentences, individual sentence or sentence pair example was presented to the annotators and they were asked to choose between the options *natural*, *unnatural*, *neither*, after reading the given example. The examples were presented in sets of five sentences (individual sentence tasks) or three sentence pairs (sentence pair tasks) in random order,

with the option to stop at any point during the process. The annotators were compensated with \$.1 per five sentences (or three sentence pairs). For NLI task sentences, the annotators were presented with six sentence pairs, for which they were asked to provide judgment on a five-point scale about the inferrability of the second sentence from the first. The annotators were compensated with \$.1 per six sentence pairs. See Table 3 for inter-annotator agreement and the final size of the dataset.

	Agree	Unan.	Accuracy	Size
Negation	60.2	40.3	80.1	591
Spatial	71.1	56.7	85.6	157
Quant.	73.8	60.7	86.9	323
Comp.	70.0	55.1	85.0	89
Prep	54.7	32.1	77.4	358
<i>wh</i> -words	72.5	58.7	86.2	584
Def.	72.0	58.1	86.0	508
Coord.	75.3	62.9	87.6	456
EOS	64.9	47.3	82.4	478

Table 3: Pairwise inter-annotator agreement ($n = 3$), % of examples with unanimous agreement, and individual annotator accuracy according to the expected label for each task in the final probing dataset.

D List of Prepositions Used for the NLI Probing Set

about, above, across, after, against, ahead of, all over, along, among, around, at, before, behind, below, beneath, beside, by, for, from, in, in front of, inside, inside of, into, near, nearby, next to, on, on top of, out of, outside, outside of, over, past, through, to, under, up, within, with, without

E Random Initialization Variance

	$\mu (\pm\sigma)$
Prep	46.14 (± 0.89)
Negation	53.78 (± 0.84)
Spatial	56.18 (± 1.68)
Quant.	61.1 (± 1.55)
Comp.	51.22 (± 2.31)
<i>wh</i> -words	51.37 (± 0.36)
def/indef articles	57.77 (± 0.97)
coord.	54.39 (± 0.96)
EOS	52.74 (± 1.30)

Table 4: Mean and standard deviation of probing task performance across five different random initializations.

lexneg	0.43	0.65	0.62	0.68	0.65	0.55	0.57	0.66	0.65	0.69	0.81
expneg	0.41	0.43	0.37	0.42	0.53	0.39	0.39	0.42	0.40	0.58	0.82
all	0.40	0.55	0.50	0.56	0.59	0.49	0.50	0.54	0.53	0.64	0.80
	majority	rand	ccg 38K	dis 151K	nli 393K	img 592K	skip 4M	mt 3.4M	lm 30.3M	bert	human

Figure 3: Accuracy of each pretrained model on subsets of the negation probing set. *neg* is the accuracy for the whole negation probing set. *lexneg* shows accuracy for the subset of sentence pairs negated using antonyms and *expneg* sentences explicitly negated using *not*. The leftmost column shows the majority-class baseline, and the rightmost column shows individual annotator accuracy on the final evaluation set. Blue denotes performance improvement over randomly initialized encoder baseline and orange denotes performance decrease.

F Subset Accuracy for the Negation Probing Set

In Figure 3, we see that the NLI model’s improvement on the negation probing set mostly derives from its improvement on explicit negation rather than lexical negation.

G Prediction Overlap between Models

We show prediction overlap heatmaps for all probing tasks in Figure 4.

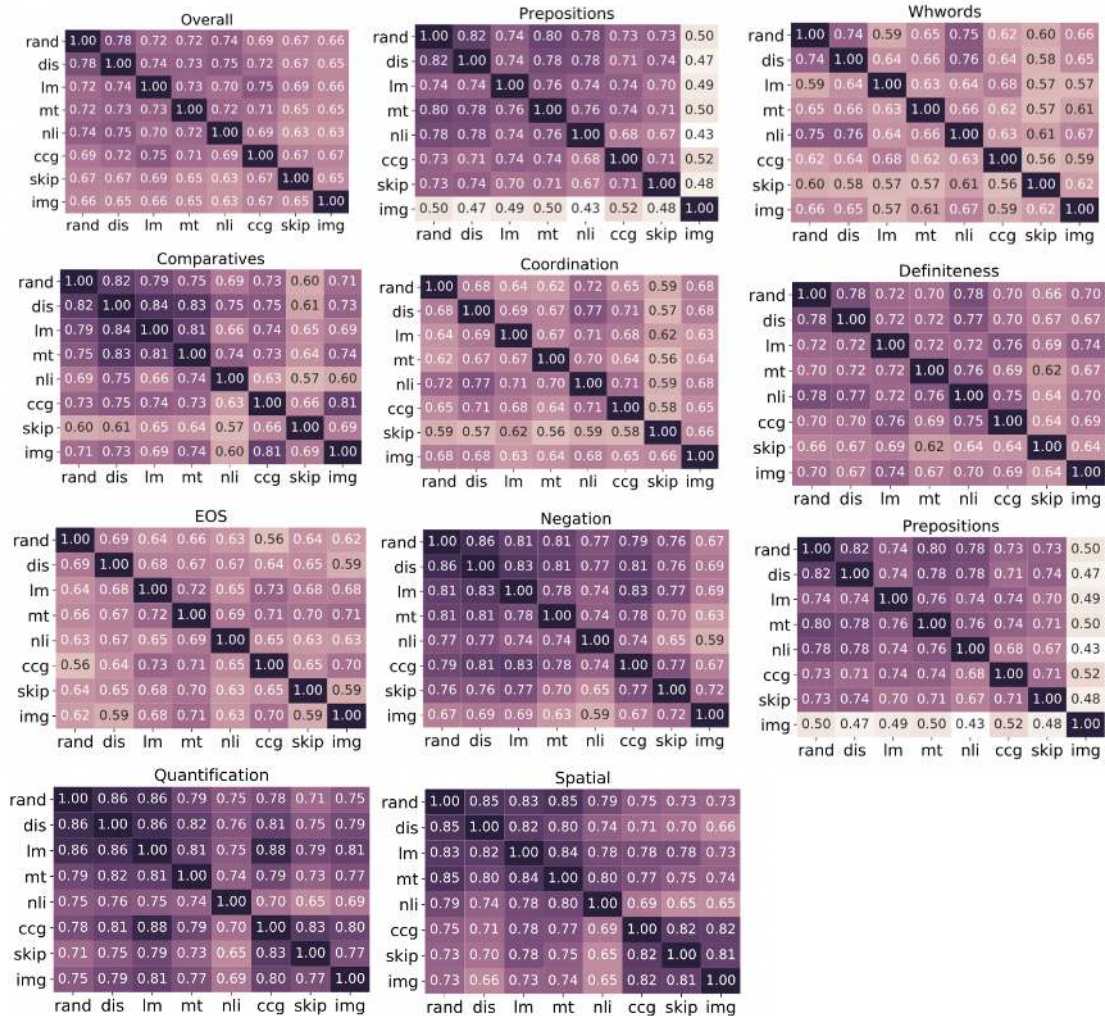


Figure 4: Heatmaps of prediction overlap for all probing tasks, between models pretrained with different objectives.