

# Probing Across Time: What Does RoBERTa Know and When?

Leo Z. Liu<sup>♣\*</sup> Yizhong Wang<sup>♣\*</sup> Jungo Kasai<sup>♣</sup>  
Hannaneh Hajishirzi<sup>♣♡</sup> Noah A. Smith<sup>♣♡</sup>

<sup>♣</sup>Paul G. Allen School of Computer Science & Engineering,  
University of Washington, Seattle, WA, USA

<sup>♡</sup>Allen Institute for Artificial Intelligence, Seattle, WA, USA

{zeyuliu2,yizhongw,jkasai,hannaneh,nasmith}@cs.washington.edu

## Abstract

Models of language trained on very large corpora have been demonstrated useful for natural language processing. As fixed artifacts, they have become the object of intense study, with many researchers “probing” the extent to which they acquire and readily demonstrate linguistic abstractions, factual and commonsense knowledge, and reasoning abilities. Recent work applied several probes to intermediate training stages to observe the developmental process of a large-scale model (Chiang et al., 2020). Following this effort, we systematically answer a question: for various types of knowledge a language model learns, *when* during (pre)training are they acquired? Using RoBERTa as a case study, we find: linguistic knowledge is acquired fast, stably, and robustly across domains. Facts and commonsense are slower and more domain-sensitive. Reasoning abilities are, in general, not stably acquired. As new datasets, pretraining protocols, and probes emerge, we believe that probing-across-time analyses can help researchers understand the complex, intermingled learning that these models undergo and guide us toward more efficient approaches that accomplish necessary learning faster.

## 1 Introduction

Current NLP approaches lean heavily on language models trained on very large corpora (Peters et al., 2018; Devlin et al., 2019; Radford et al., 2018; Liu et al., 2019b; Brown et al., 2020). Many researchers have sought to interpret what kinds of knowledge are acquired during this “pretraining” phase (Clark et al., 2019; Hao et al., 2019; Kovalova et al., 2019; Belinkov et al., 2020). Extending Chiang et al. (2020), we systematically conduct probing across the pretraining iterations, to understand not just what is learned (as explored in numerous past analyses of fixed, already-trained

models), but also *when*. In this work, we aim to inform future work on more efficient pretraining (e.g., fewer iterations are needed to acquire some kinds of knowledge) and on understanding dependencies among different kinds of knowledge.

Specifically, we apply a *probing across time* framework to the widely used RoBERTa masked language model (Liu et al., 2019b). We reproduce the pretraining of RoBERTa and apply a suite of probes at many checkpoint iterations across pretraining (§3). Our rich probe suite covers a diverse range of desirable knowledge types: linguistic properties (Liu et al., 2019a), factual knowledge (Petroni et al., 2019), and commonsense (Zhou et al., 2020) and basic reasoning capabilities (Talmor et al., 2019). Our main finding is that linguistic information tends to be acquired fast, factual and commonsense knowledge slower, and reasoning abilities are largely unlearned.

We next apply probing across time to instances of RoBERTa trained on text from varying domains and with varying amounts of data (§4). Our experiments show that the learning order and learning patterns of different types of knowledge generally hold regardless of the data variation. However, different data choices do have an impact on the learning speed and the final performance. Our findings suggest that the inclusion of data in more diverse domains is more important than the quantity alone.

Finally, we compare probes across time with research benchmark task performance across time (§5). We find that most of these benchmark tasks (e.g., SST-2, Socher et al., 2013, and SQuAD, Rajpurkar et al., 2016) require a relatively small number of pretraining steps to achieve high performance, which is similar to the fast learning patterns shown by linguistic probes. Some other tasks that are designed to test more complex knowledge (e.g., ReCoRD, Zhang et al., 2018 and WSC, Levesque et al., 2012) benefit from longer pretraining time, aligning well with our findings for the correspond-

\*Equal contribution.

ing type of probes.

We expect that, as new pretrained models and new probes emerge, probing-across-time analyses can help synthesize evidence for models’ capabilities. We release our code, as well as all the pretraining checkpoints at <https://github.com/leo-liuzy/probe-across-time> to benefit future research.

## 2 Probing Across Time

The great success of pretrained language models has motivated researchers to characterize what kinds of knowledge they encode. Probing seeks to determine how much is known by the pretrained model, so its representations are used without fine-tuning. Few or zero additional parameters are estimated to target the probing task so that success on the probing task is attributable to pretraining alone. Past probing work has applied probing analysis after pretraining is complete and compares *different* models (e.g., BERT vs. GloVe). We extend probing to different parameter states (i.e., training checkpoints) over the *same* model’s pretraining trajectory. By observing how probe performance changes over time, we hope to understand not just what the pretrained language model knows, but also when.

To that end, we adopt a diverse set of probes aiming at different types of knowledge (§2.1) and apply those probes at different iterates of the estimated model across pretraining. This requires pretraining from scratch and checkpointing intermediate parameter estimates (§2.3). Considering our computational budget, we choose RoBERTa<sup>1</sup> as our case study because of its popularity in supporting downstream probes and tasks. We leave exploration of different objectives and architectures, e.g., GPT-2 (Radford et al., 2018), as future work. We also set up baselines as additional relative references to stand for the best and the worst expected results from pretraining (§2.2). See §A for implementation details and the computational cost.

### 2.1 Probe Suite Construction

Belinkov et al. (2020) categorize existing probes into two families. *Structural probes* train a lightweight classifier that predicts a label on top of the model’s internal representations. Such probes are mostly used to test linguistic knowledge like parts of speech. *Behavioral probes*, on the other

hand, do not rely on additional parameters or training, but use the model as it is to make predictions. For example, if a masked language model can predict “Honolulu” for the input “Barack Obama was born in [MASK],” we can conclude that the pretrained model learned Obama’s birth place.

We adopt a rich set of probing tasks from five existing and publicly available probing packages to systematically evaluate different types of encoded knowledge: LKT and BLIMP for linguistic knowledge, LAMA for factual and commonsense knowledge, CAT for commonsense knowledge, and OLMPICS for reasoning knowledge. See §A for links to those probes. For clarity, we focus here on four or five probing tasks from each package and present the rest (which showed similar patterns) in §B. See Table 1 for an overview of these probing tasks and examples. We briefly describe the goal and our setup of these five packages as follows.

**LKT** Liu et al. (2019a) introduce a *structural linguistic* probe suite for testing linguistic knowledge and transferability (LKT) of contextual representations. For all tasks in LKT, we train a linear classifier model to predict the linguistic annotation of each word in a sentence. Through the performance of the classifier, we measure how closely the encoded information in the word representations conforms to linguistic annotations from human experts. Following Liu et al. (2019a), we use learnable coefficients to weigh a sum of representations from all the transformer layers, and compute the input vector to the classifier. We measure the probe performance by accuracy or  $F_1$  on the test sets.

**BLIMP** Salazar et al. (2020) introduce a *behavioral linguistic* probe suite on the benchmark of linguistic minimal pairs (BLIMP, Warstadt et al., 2020). This benchmark isolates specific phenomena in syntax, morphology, or semantics such as island effects and subject-verb agreement. As seen in Table 1, input sentence pairs differ only by a word or a short phrase, but contrast in grammatical acceptability. We test whether RoBERTa scores the grammatical sentence higher than the ungrammatical one. The score for a sentence is calculated by sequentially masking one word at a time and averaging the log probabilities of the masked words. Since no additional parameters or training are involved, BLIMP provides a complementary perspective to LKT— if probing *without* training shows the same pattern as *with* training, it strength-

<sup>1</sup>In our work, RoBERTa stands for RoBERTa-base.

Package	Knowledge	Task	Formulation	Examples
LKT	Linguistic	POS Tagging	Token Labeling	PRON AUX VERB ADV ADP DET NOUN PUNCT I 'm staying away from the stock
		Syntactic Chunking		B-NP B-VP B-PP B-NP I-NP I-NP O Shearson works at American Express Co
		Name Entity Recognition		O O I-ORG I-ORG I-ORG O O O O By stumps Kent County Club had reached 108 .
		Syntactic Arc Predication	Token Pair Labeling	
		Syntactic Arc Classification		
BLIMP	Linguistic	Irregular Forms	Comparing Sentence Scores Expected: $\mathbb{S}(\checkmark) > \mathbb{S}(\times)$	✓ Aaron <i>broke</i> the unicycle. ✗ Aaron <i>broken</i> the unicycle.
		Determiner-Noun Agree.		✓ Rachelle had bought that <i>chair</i> . ✗ Rachelle had bought that <i>chairs</i> .
		Subject-Verb Agreement		✓ These casseroles <i>disgust</i> Kayla. ✗ These casseroles <i>disgusts</i> Kayla.
		Island Effect		✓ Which <i>bikes</i> is John fixing? ✗ Which is John fixing <i>bikes</i> ?
		Filler Gap		✓ Brett knew <i>what</i> many waiters find. ✗ Brett knew <i>that</i> many waiters find.
LAMA	Factual	Google RE	Masked LM Expected: $\forall w \in V_{\text{RoBERTa}} \setminus \{\checkmark\},$ $\mathbb{P}(\checkmark   C) > \mathbb{P}(w   C)$	Albert Einstein was born in [MASK] ✓: [MASK] = 1879
		T-REX		Humphrey Cobb was a [MASK] and novelist ✓: [MASK] = screenwriter
	SQuAD	A Turing machine handles [MASK] on a strip of tape. ✓: [MASK] = symbols		
	ConceptNet	You can use [MASK] to bathe your dog. ✓: [MASK] = shampoo		
CAT	Commonsense	Conjunction Acceptability	Comparing Sentence Scores Expected: $\forall \times, \checkmark$ $\mathbb{S}(\checkmark) > \mathbb{S}(\times)$	✓ Jim yelled at Kevin <i>because</i> Jim was so upset. ✗ Jim yelled at Kevin <i>and</i> Jim was so upset.
		Winograd		✓ The fish ate the worm. The <i>fish</i> was hungry. ✗ The fish ate the worm. The <i>worm</i> was hungry.
		Sense Making		✓ Money can be used for buying <i>cars</i> . ✗ Money can be used for buying <i>stars</i> .
		SWAG		✓ Someone unlocks the door and they go in. <i>Someone leads the way in</i> .
				✗ Someone unlocks the door and they go in. <i>Someone opens the door and walks out</i> .
Argument Reasoning	✗ Someone unlocks the door and they go in. <i>Someone walks out of the driveway</i> .			
OLMPICS	Reasoning	Taxonomy Conjunction	Multiple Choice Masked LM Expected: $\forall \times, \checkmark$ $\mathbb{P}(\checkmark   C) > \mathbb{P}(\times   C)$	A ferry and a floatplane are both a type of [MASK]. ✓ vehicle ✗ airplane ✗ boat
		Antonym Negation		It was [MASK] hot, it was really cold. ✓ not ✗ really
		Object Comparison		The size of a airplane is usually much [MASK] than the size of a house. ✗ smaller ✓ larger
		Always Never		A chicken [MASK] has horns. ✓ never ✗ rarely ✗ sometimes ✗ often ✗ always
		Multi-Hop Composition		When comparing a 23, a 38 and a 31 year old, the [MASK] is oldest. ✓ second ✗ first ✗ third

Table 1: Representative tasks from selected probe packages.  $\mathbb{S}(\cdot)$  scores a sentence by sequentially masking each word in the sentence and averaging the log probabilities.  $C$  denotes the rest of the sentence, and  $\mathbb{P}(\cdot | C)$  is the conditional probability distribution over the vocabulary given  $C$ .  $V_{\text{RoBERTa}}$  is the vocabulary of RoBERTa.

ens our observation on linguistic knowledge.

**LAMA** Petroni et al. (2019) introduce a *behavioral* probing package that tests *factual* and *commonsense* knowledge. Each example in LAMA is a cloze-style question with its subject or object masked. By predicting the masked word with RoBERTa, we measure its ability to recover real-world facts. We only consider the examples whose masked words exist in the RoBERTa vocabulary and measure whether RoBERTa predicts the masked word with the highest probability.

**CAT** Zhou et al. (2020) introduce *behavioral commonsense* probes based on a series of existing commonsense datasets. These probes measure whether the pretraining model can give a higher score to positive examples (sentences that align with commonsense) than negative examples (sentences that don't). The score of each sentence is computed by sequentially masking one word at a time and averaging the log probabilities.

**OLMPICS** Talmor et al. (2019) introduce a *be-*

*havioral* probe package that tests the model's *reasoning* abilities including object comparison, taxonomy conjunction, and multi-hop composition. We adopt the multiple choice masked LM setup where the pretrained RoBERTa is required to fill in the mask by selecting words from 2–5 candidates. Different from other probing packages, Talmor et al. (2019) show that pretrained LMs do not get a large improvement over baselines on most of the probing tasks. This suggests that these reasoning tasks present challenges for current pretrained models, but we still include this probe package because it offers tests with a different aim and thus different insights into pretraining RoBERTa.

## 2.2 Baselines for Relative Performance

Probes are not a perfect, absolute measure of encoded knowledge. In particular, Hewitt and Liang (2019) find that probing classifiers can memorize labeling decisions independently of the linguistic knowledge of the representations. Pimentel et al. (2020) argue that a tighter estimate of the encoded

knowledge can be obtained by complex probing models. We ask whether targeted knowledge can be easily extracted with few or zero additional parameters (i.e., *ease of extraction*, as suggested by Pimentel et al., 2020). We treat the probing scores as relative performance; and we address their concerns by comparing RoBERTa with the following baselines:

- **Random Guess** randomly selects one class label or token from the candidate pool.
- **Random Vector + Linear Classifier** uses a random vector to represent each type, and trains a linear classifier on the top to predict the label with the token vector being frozen.
- **GloVe + Linear Classifier** uses GloVe vectors (Pennington et al., 2014), and trains a linear classifier on top to predict the label.
- **Original RoBERTa** probes the officially released checkpoint<sup>2</sup> of RoBERTa base to see if our checkpoints are pretrained properly and can achieve reasonable performance.

Moreover, our probing results on different checkpoints can illustrate the relative performance change during pretraining.

### 2.3 Pretraining Setups

We choose base-size RoBERTa as a case study. In order to conduct probing over time, we replicate the RoBERTa pretraining procedure and periodically save checkpoints for later probing. To ensure that probe-task relevant text is uniformly distributed over batches, the entire data is shuffled before every epoch. Our training setting follows closely the one prescribed in Liu et al. (2019b), except for the following differences: **1**) we use the hyperparameter setting with 1M update steps and a reduced batch size of 256, instead of 125K steps and a batch size of 2,048;<sup>3</sup> **2**) we use static masking (Devlin et al., 2019) during data processing instead of dynamic masking to run the code on TPUs. These differences can result in slightly worse performance on downstream tasks (Liu et al., 2019b). However, due to the large data size, we believe it won't significantly change the learning patterns we found in later sections.

We save a checkpoint every 20K training steps and more frequently during the first 12,800 steps, resulting in about 62 checkpoints for each pretrain-

ing setting.<sup>4</sup> Then we probe all these checkpoints to estimate the knowledge encoded by the model at different training steps.

**Pretraining data** The original RoBERTa was pretrained on BOOKCORPUS (4 GB, Zhu et al., 2015), English WIKIPEDIA (12 GB), CC-News (76 GB, Nagel, 2016), OPENWEBTEXT (38 GB, Gokaslan and Cohen, 2019), and STORIES (31 GB, Trinh and Le, 2018). Since we do not have access to their version of filtered CC-NEWS, we use REALNEWS (120 GB, Zellers et al., 2019) instead, which is similar, according to Liu et al. (2019b). This difference in training data might partly explain the performance degradation from the original RoBERTa. All the other corpora remain the same. This leads to a total of 205 GB<sup>5</sup> unprocessed text, and each training epoch makes 360,851 update steps (3 epochs in total for the 1M update steps). In our later controlled experiments, we sample these corpora to compare domains and data sizes in §4.

## 3 Learning Patterns

In this section, we use our reproduced RoBERTa to ask and answer, **at which stage does the model acquire each kind of knowledge?** In addition to plotting probe performance across time (i.e., parameter updates during learning; see Fig. 1), a useful measurement is the number of updates required to reach  $x\%$  of the maximum performance achieved by our model across all iterations. We denote this measurement by “Learning Progress- $x\%$ ” for  $x \in \{90, 95, 97\}$ ; it is indicated by the bottom horizontal bars in each probe's plot in Fig. 1. We consider each type of knowledge in turn. All of the following discussion is supported by Fig. 1.

### 3.1 Linguistic Learning

The structural linguistic probes test how closely the information in RoBERTa representations conforms to annotations developed from linguistic theories. The behavioral probes test how sensitively the language model can respond to some detailed syntactic error. In most cases, RoBERTa shows great success in learning linguistic knowledge with high speed and stability, and this pattern is consistent both in the classifier-based LKT and behavioral

<sup>4</sup>In a pilot study, we observed that the training and validation loss start to plateau at 50K training steps.

<sup>5</sup>We follow Liu et al. (2019b) to report data size by gigabytes of the uncompressed text in this paper. Our entire pretraining data contain 46 billion tokens after tokenization.

<sup>2</sup><https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.md>.

<sup>3</sup>We chose a smaller batch size for more fine-grained observations.

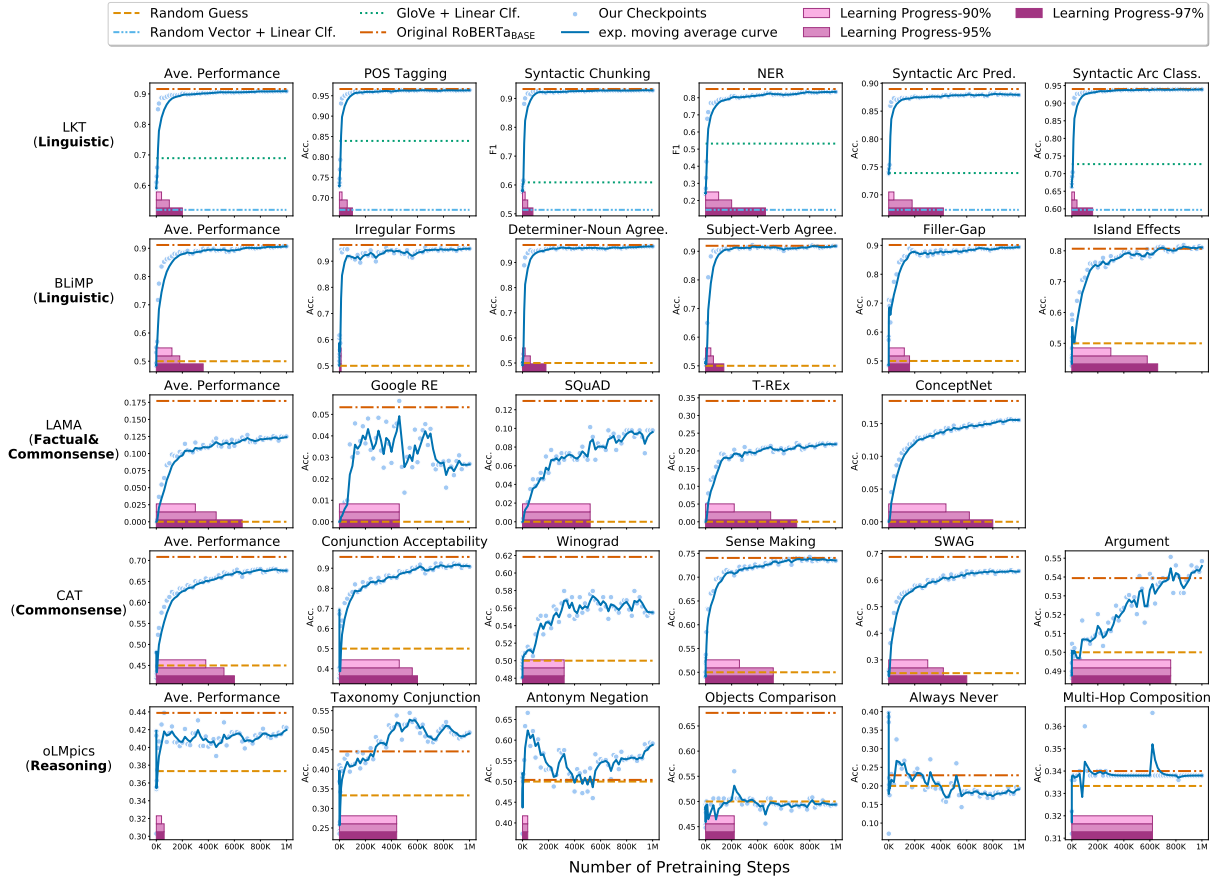


Figure 1: Probes across time performance from our reproduced RoBERTa. Linguistic information tends to be acquired fast, factual and commonsense knowledge slower, and reasoning abilities are largely unlearned. For a better visualization, we use exponential moving average curve with a coefficient of 0.5 to plot a smoothed curve. For each probe package, the first column is the average performance over the 4-5 selected tasks. See complete plots in §B.1. “Learning Progress- $x\%$ ” values are calculated with raw data, not smoothed data. The maximum performance of Always Never (OLMPICS) occurs when the model is initialized.

BLiMP probes (see the first two rows in Fig. 1). In the LKT measurements from all 62 checkpoints, we observe that 97% of the improvement in overall performance occurs within 20% of the total training updates. The variation of this fast pattern among tasks in LKT is small and all the performance converges closely to the originally reported results. In a majority of BLiMP tests, compared to LKT, RoBERTa shows similar or even faster learning speed to achieve the 97% threshold (Irregular Form, Determiner-Noun Agreement, and Subject-Verb Agreement), whereas some other tasks are more slowly learned. See §B.1 for more results.

### 3.2 Factual and Commonsense Learning

Overall, Fig. 1 shows *slower* learning speed and more *instability* in both LAMA and CAT than in the linguistic probes. Most of our measurements require more than half of pretraining steps to achieve 97% of the best performance. Compared with the

high consistency in linguistic probes, there is more variation among tests. For example, the SQuAD and ConceptNet tests reveal the LM is steadily, although slowly, learning some factual and commonsense knowledge. However, other tests like argument reasoning and Winograd show fluctuation or even a decrease in performance, and some factual knowledge such as Google RE is not easily learned. We also note that there are noticeable gaps in the final performance of some tasks between our reproduced RoBERTa and original RoBERTa. We suspect that this is because of the several differences in our replication of RoBERTa (e.g., batch size and REALNEWS vs. CC-NEWS, §2.3).

### 3.3 Reasoning

OLMPICS shows some drastically different patterns from the other knowledge types (Fig. 1). Many of the reasoning abilities are not learned during RoBERTa pretraining. For example, in Ob-

ject Comparison, our RoBERTa model is on par with random guessing, and the performance on Always Never keeps decreasing overall after the initialization. Some other tests included in §B.1 also show severe fluctuation or similar patterns. However, there are still signs of learning for some tasks. Taxonomy Conjunction shows the most promising sign of learning, and although to a small extent, RoBERTa also learns to do multi-hop reasoning. Another noteworthy observation is that RoBERTa’s performance on Antonym Negation first increases rapidly, and then decreases in the rest of the first half of training; however, it starts to increase again toward the end. This indicates that this knowledge is not stably stored in the model.

#### 4 Varying the Pretraining Corpus

Our experiments on the pretraining trajectory in the previous section demonstrate that, when acquired, different types of knowledge are learned at different stages. On the other hand, previous work (Raffel et al., 2020; Gururangan et al., 2020) demonstrates that language domain and data size are important factors for a pretrained model’s performance in downstream tasks. Thus, the question arises: do our observations on the pretraining trajectory hold regardless of the training corpus? In this section, we ask and answer **how do domains and corpus sizes affect the learning trajectory?**

**Domains** With the same setting as §3, we pre-train RoBERTa with three controlled domains: English WIKIPEDIA, REALNEWS, and STORIES. We downsample REALNEWS and STORIES to be roughly the same size as WIKIPEDIA (12 GB).

The first row of Fig. 2 compares probing performance across time over varying domains. On one hand, the general learning pattern from the previous section persists regardless of the domain: linguistic knowledge is acquired faster than the other types. On the other hand, we find that the change in the pretraining domain affects the final performance of all knowledge types but to different extents. LKT is generally less affected than BLIMP, probably due to the additional training of LKT’s classifier on the probing data. Although the model slowly acquires both factual and commonsense knowledge regardless of corpus domains, factual knowledge is much more affected by domain (in fact, most affected among all). LAMA especially shows very slow learning on the STORIES domain, implying that factual knowledge might be very

sparse in STORIES. The fact that the included factual tests—Google-RE, T-REx and SQuAD—are sourced from WIKIPEDIA might explain why the factual tests show RoBERTa trained on WIKIPEDIA learns faster than the one trained on REALNEWS. Though many existing works (Mostafazadeh et al., 2016; Bhagavatula et al., 2020; Qin et al., 2020) used story data to study commonsense, pretraining the model on STORIES still performs worst on our commonsense probes (ConceptNet in LAMA and most probes in CAT). On the OLMPICS probes, we keep observing large fluctuation on different domains, but do see more signs of learning on the WIKIPEDIA and REALNEWS domains than on STORIES. See §B.3 for more detailed plots.

**Corpus Sizes** To investigate the impact of corpus size, we experimented on downsampled English WIKIPEDIA (4 GB), the original English WIKIPEDIA (12 GB), and a combination of WIKIPEDIA with downsampled REALNEWS and downsampled STORIES (36 GB in total).

The second row of Fig. 2 compares probing performance over varying corpus sizes. In general, the learning order and learning patterns of different types of knowledge that we discuss in §3 still hold for all data sizes. Comparing different data sizes, we find that the biggest corpus (with more diverse inclusion of domains) generally learns faster and results in an ultimately better RoBERTa in all tested knowledge. On the other hand, interestingly, comparing the downsampled English WIKIPEDIA (4 GB) and original English WIKIPEDIA (12 GB), simply increasing the corpus size without changing domains does not improve the final probing performance substantially in all categories. This is even true for LAMA, which tests factual knowledge relevant to WIKIPEDIA. This suggests that diversity of data might be more important for pretraining than quantity. See §B.4 for more detailed plots.

#### 5 Experiments on Research Benchmarks

In §3-4, we used probes to understand the knowledge learning process, but **what do those observations mean to more practical scenarios where people use pretrained RoBERTa with finetuning?** In this section, we provide insights into this question by finetuning our RoBERTa checkpoints on eight representative research benchmarks (e.g., SQuAD). We conjecture that these benchmarks require more intermingled knowledge, whereas probing tasks usually target one specific phenomenon or

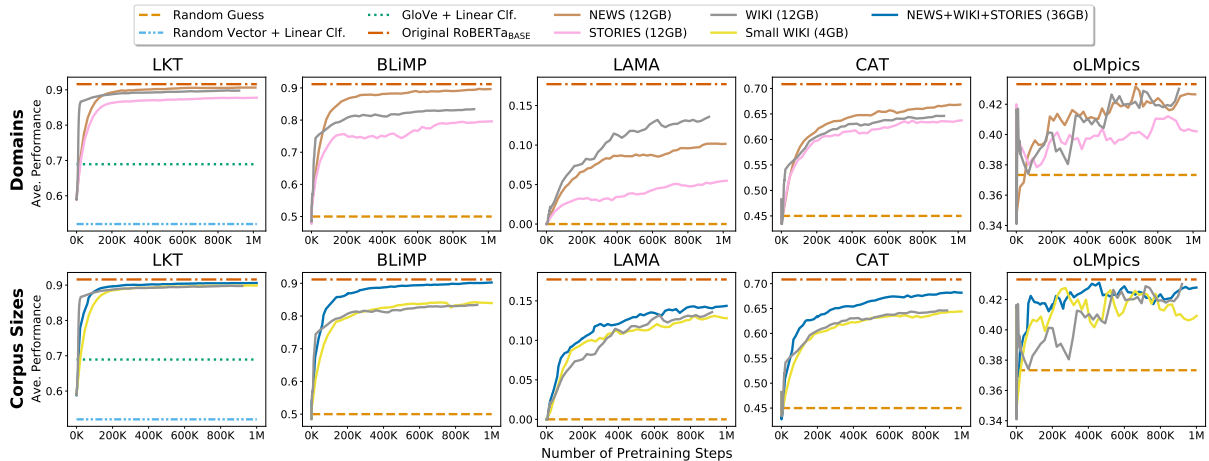


Figure 2: Performance of probes across time, comparing pretraining domains and corpus sizes. This plot reveals that the learning order and patterns of different knowledge types generally hold regardless of the data variation, with higher impact from corpus domains. Performance for each probing package is averaged over its 4–5 selected tasks and smoothed with a coefficient of 0.5 for better visualization (same as the first column in Fig. 1).

type of knowledge. By comparing them to the more controlled probes of §2, we aim to understand how acquisition of probed capabilities aligns with and perhaps accounts for performance on benchmarks.

**Experimental setup** We select 14 checkpoints from our pretraining to run our finetuning tasks: CoLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), MNLI (Williams et al., 2018), WNLI (WSC reconstructed as an inference task), MRPC (Dolan and Brockett, 2005), SQuAD (Rajpurkar et al., 2016), ReCoRD (Zhang et al., 2018), and Winograd Schema Challenge (WSC, Levesque et al. 2012). See Appendix A.2 for hyperparameters.<sup>6</sup>

These tasks are chosen to reflect the diverse downstream use cases of pretrained language models, including single-sentence (CoLA, SST-2) or sentence-pair (MNLI, WNLI, MRPC) classification, question answering (SQuAD, ReCoRD), and multiple-choice classification (WSC). Note that MNLI/WNLI and SQuAD/ReCoRD comprise two interesting contrastive pairs. Each pair shares a task format, with the latter tasks (i.e., WNLI and ReCoRD) designed to rely more on commonsense than the former ones. We expect more pretraining iterations for WNLI than MNLI and ReCoRD than SQuAD from the probing experiments (§3).

**Results** We plot our results in Fig. 3. Different patterns are observed for different tasks. CoLA and SST-2 require noticeably fewer pretraining steps,

<sup>6</sup>To avoid impractical hyperparameter search for the large number of checkpoints we have, for each finetuning task, we use the same hyperparameters for all experiments.

achieving 97% of the best performance within 16% and 32% of the pretraining time, respectively, which is even faster than LKT (20%) or BLiMP (36%). MNLI, WNLI, MRPC, and SQuAD are in a “middle” range (after linguistic knowledge, but before factual or commonsense knowledge), suggesting they are acquiring something the linguistic probes don’t test but that is learned faster than what RoBERTa can learn for LAMA / CAT. At last, ReCoRD and WSC are learned. In particular, WSC appears to be learned slower than all the tested probes, suggesting that pretraining is keeping learning knowledge that is beneficial to WSC but not tested in our probes. Note again that “learn the task” is relative to the best performance we observe by our RoBERTa on the task.

As expected, WNLI requires longer pretraining than MNLI and ReCoRD longer than SQuAD, which aligns well with our finding in §3.2 that the model learns commonsense slowly as pretraining progresses. In addition, we observe MNLI performance even drops towards the end of pretraining, implying that longer pretraining does not necessarily lead to better finetuning performance. For interested readers, we also include a plot of correlation among all experimental results in §B.2

## 6 Related Work and Further Discussion

**Learning dynamics** Early work (McClelland and Rumelhart, 1986) observed the dynamics in a feedforward neural network to assess the cognitive plausibility of a connectionist model. They found staged learning in past tense acquisition, similar to

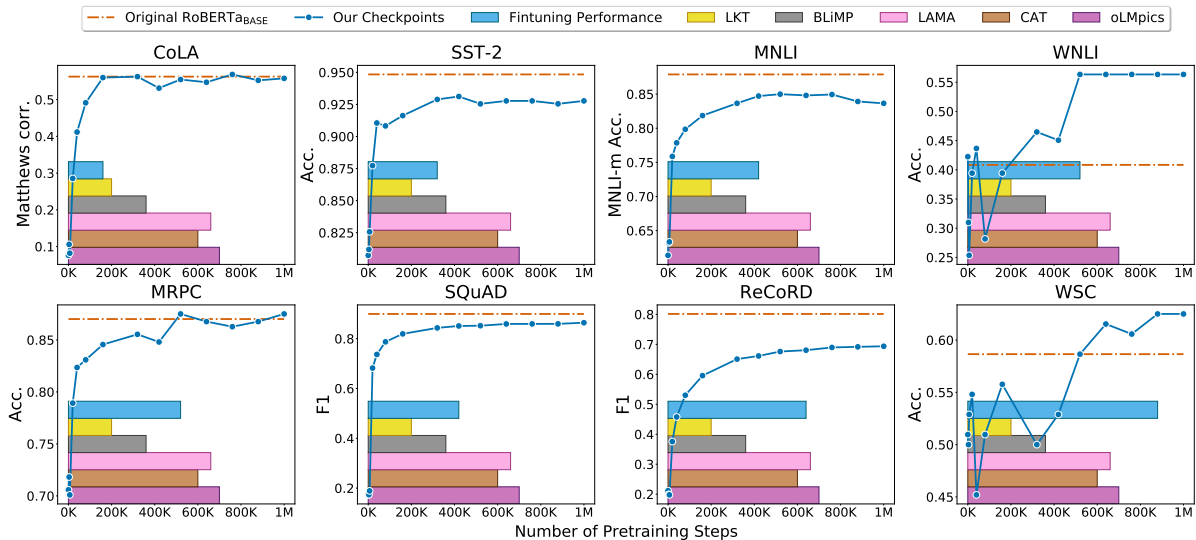


Figure 3: Finetuning performance on research benchmarks using the checkpoints from our reproduced RoBERTa. Most of these benchmark tasks require less than half of pretraining steps to achieve high performance, while tasks that are designed to test more complex knowledge (e.g., ReCoRD and WSC) benefit from longer pretraining time. We plot Learning Progress—97% of our RoBERTa in each probe package and finetuning (i.e., colored horizontal bars). The bottom five bars for the probes are shown for comparison. For Multi-genre NLI, we only show accuracy on the matched genres (same genres seen during finetuning), as we see similar curves on the unmatched genres.

humans. More recently, [Saphra and Lopez \(2019\)](#) studied linguistic and topic learning dynamics in hidden states of an LSTM language model. They found that syntactic information is encoded at an early training stage, which is consistent with our finding despite the difference in training objective and network architecture. Our work, instead, uses a rich set of probes to examine more diverse aspects of language and analyzes training iterations.

Concurrent work ([Chiang et al., 2020](#)) is the closest work to ours and uses probes to investigate the learning dynamics as well. However, they find that linguistic knowledge and factual knowledge do not generally improve as pretraining proceeds, we find that factual and commonsense knowledge *do* (§3); we attribute such difference to our more systematic choice of probes – not only adding two more categories (i.e. commonsense and reasoning) but also more tasks in the linguistic and factual knowledge categories. For example, we found that the factual knowledge probed using SQuAD and ConceptNet data still increases as the pretraining progresses. However, [Chiang et al. \(2020\)](#) only used a subset of T-REx, which plateaus quickly according to our experiments. Sharing their concern of how data affects pretraining, we empirically investigate how the domain of pretraining corpus affects the dynamics of different types of knowledge.

[Zhang et al. \(2020\)](#) investigate masked language

models trained on corpora of varying sizes in a *domain*. They experiment with linguistic probes and show that 90% of the improvement in syntactic probing performance can be achieved with a pretraining corpus of only about 10M words. In contrast, the probing performance in commonsense knowledge suffers from small training data. Different from the final models obtained in their work, we consider the *entire pretraining trajectories*. Nevertheless, our findings by varying corpus size (§4) are consistent with their conclusion and additionally we find that adding more data in diverse domains can improve both types of knowledge. Their results on the relation between the corpus size and probing performance also support our finding that linguistic knowledge is generally easy to learn, while other types of knowledge require more efforts in term of both the data size and the training iterations. Since their experiments consist of different hyperparameter settings, and the domains for some of our datasets (i.e., WIKIPEDIA and STORIES) are the same as theirs, we consider our observation complementary to theirs.

**Two views on probing** [Pimentel et al. \(2020\)](#) develop an *information-theoretic* perspective that differs from our *ease-of-extraction* view: contextual representations can not have more information than the original sentence because the embedding func-



tion is deterministic. However, this view does not consider how they are *structured* in the embedding space. These representations can only be taken advantage of if they are structured in an extractable way. All the “knowledge” mentioned in our paper refers to such structured and easy-to-extract information. In our pilot study, we experimented with MLP classifiers in structural probes. They achieved much higher (almost highest in some cases) scores than the linear classifier on even the randomly-initialized RoBERTa model. Though agreeing with Pimentel et al. (2020)’s conclusion, MLP classifiers cannot reflect the learning process of the targeted knowledge and fail to explain why RoBERTa improves in downstream tasks over time (Fig. 3).

## 7 Ethical Concerns

We estimate our carbon costs in Appendix §A. We recognize the possibility that changes to hyperparameters might lead to different conclusions and leave it to future work to balance the costs of such exploration with the value of a more detailed understanding of hyperparameter impact on learning over time. We note that the robustness of learning patterns is partly supported by results from Zhang et al. (2020), since they probed models trained with different hyperparameters and observed similar linguistic patterns as we did.

## 8 Conclusion

We have shown how *probing across time* reveals when, during pretraining iterations, a masked language model acquires various kinds of knowledge. RoBERTa, our case study model, is shown to learn linguistic knowledge faster than factual and commonsense knowledge, but struggle to learn reasoning abilities. We explored variation due to corpus domain and size, and related our findings to research benchmark tasks. As models evolve and new probes emerge, we believe our *probing across time* framework can serve as a general framework to inform progress on both fronts.

## Acknowledgments

In random order, we appreciate valuable discussion and feedback from Noah’s ARK group at the University of Washington, Nelson Liu at Stanford University, Victoria Lin at Salesforce research, Roy Schwartz at the Hebrew University of Jerusalem, and the anonymous reviewers. This research is supported in part by the Office of Naval Research

under grants N00014-18-1-2826 and N00014-18-1-2670.

## References

- Yonatan Belinkov, Sebastian Gehrmann, and Ellie Pavlick. 2020. [Interpretability and analysis in neural NLP](#). In *Proc. of ACL: Tutorial Abstracts*.
- Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen-tau Yih, and Yejin Choi. 2020. [Abductive commonsense reasoning](#). In *Proc. of ICLR*.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#).
- Cheng-Han Chiang, Sung-Feng Huang, and Hung-yi Lee. 2020. [Pretrained language model embryology: The birth of ALBERT](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6813–6828, Online. Association for Computational Linguistics.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. [What does BERT look at? an analysis of BERT’s attention](#). In *Proc. of BlackboxNLP*.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding](#). In *Proc. of NAACL*.
- William B. Dolan and Chris Brockett. 2005. [Automatically constructing a corpus of sentential paraphrases](#). In *Proc. of IWP*. Asian Federation of Natural Language Processing.
- Aaron Gokaslan and Vanya Cohen. 2019. OpenWebText corpus. <http://Skylion007.github.io/OpenWebTextCorpus>.
- Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. [Don’t stop pretraining: Adapt language models to domains and tasks](#). In *Proc. of ACL*.
- Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2019. [Visualizing and understanding the effectiveness of BERT](#). In *Proc. of EMNLP*.

- John Hewitt and Percy Liang. 2019. [Designing and interpreting probes with control tasks](#). In *Proc. of EMNLP*. Association for Computational Linguistics.
- Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. [Revealing the dark secrets of bert](#). In *Proc. of EMNLP*.
- Loïc Lanelongue, Jason Grealey, and Michael Inouye. 2020. [Green algorithms: Quantifying the carbon emissions of computation](#). *CoRR*, abs/2007.07610.
- Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. [The Winograd schema challenge](#). In *Proc. of KR&R*.
- Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. [Linguistic knowledge and transferability of contextual representations](#). In *Proc. of NAACL*.
- 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](#).
- James L. McClelland and David E. Rumelhart. 1986. *Parallel Distributed Processing*, volume 2.
- Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James F. Allen. 2016. [A corpus and cloze evaluation for deeper understanding of commonsense stories](#). In *Proc. of NAACL*.
- Sebastian Nagel. 2016. [CC-News](#). <http://web.archive.org/save/http://commoncrawl.org/2016/10/news-dataset-availabl>.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proc. of EMNLP*.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proc. of NAACL*.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick S. H. Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander H. Miller. 2019. [Language models as knowledge bases?](#) In *Proc. of EMNLP*.
- Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020. [Information-theoretic probing for linguistic structure](#). In *Proc. of ACL*.
- Lianhui Qin, Vered Shwartz, Peter West, Chandra Bhagavatula, Jena D. Hwang, Ronan Le Bras, Antoine Bosselut, and Yejin Choi. 2020. [Back to the future: Unsupervised backprop-based decoding for counterfactual and abductive commonsense reasoning](#). In *Proc. of EMNLP*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. [Language models are unsupervised multitask learners](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *JMLR*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. [Squad: 100, 000+ questions for machine comprehension of text](#). In *Proc. of EMNLP*.
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. [Masked language model scoring](#). In *Proc. of ACL*.
- Naomi Saphra and Adam Lopez. 2019. [Understanding learning dynamics of language models with SVCCA](#). In *Proc. of NAACL*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *Proc. of EMNLP*.
- Alon Talmor, Yanai Elazar, Yoav Goldberg, and Jonathan Berant. 2019. [oLMpics - on what language model pre-training captures](#). *TACL*.
- Trieu H. Trinh and Quoc V. Le. 2018. [A simple method for commonsense reasoning](#).
- Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. [BLiMP: The benchmark of linguistic minimal pairs for english](#). *TACL*.
- Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. [Neural network acceptability judgments](#). *TACL*.
- Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *Proc. of NAACL*.
- Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. [Defending against neural fake news](#). In *Proc. of NeurIPS*.
- Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. [ReCoRD: Bridging the gap between human and machine commonsense reading comprehension](#).
- Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R. Bowman. 2020. [When do you need billions of words of pretraining data?](#)
- Xuhui Zhou, Yue Zhang, Leyang Cui, and Dandan Huang. 2020. [Evaluating commonsense in pre-trained language models](#). In *Proc. of AAAI*.

Yukun Zhu, Ryan Kiros, Richard S. 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 *Proc. of ICCV*.

## A Implementation and Hyperparameters

### A.1 Pretraining

We chose the base-size RoBERTa model (125M parameters) as a case study for pretraining in this paper due to our computational budget.

We used the TPU implementation<sup>7</sup> of RoBERTa model in the official fairseq library.<sup>8</sup> However, because it needs to support experiments on TPUs, our implementation still has some differences compared to the original RoBERTa, which leads to the small performance gap between our pretrained models and the original RoBERTa baseline. However, we believe that these differences won't change the learning patterns very much. Specifically, for the static masking, since our data size is large, the whole pretraining (for reproducing RoBERTa) took less than 3 passes over the data. Therefore, using dynamic masking likely would not make much difference, especially considering that most of the probing performance becomes stable even before the first epoch ends.

The detailed pretraining hyperparameters are listed in Table 2. Each pretraining was run using 8 TPU-v3 cores, and 1M steps took around 15 days. To increase awareness about the potential environmental impact of our large-scale pretraining, we use a tool from Lannelongue et al. (2020)<sup>9</sup> to estimate the energy and carbon cost of our experiments. As a rough estimate, each pretraining (on one corpus) consumes about 684.02 kWh energy and has 173.19 kg CO<sub>2</sub>e carbon footprint. Therefore, in total, our pretraining experiments consume 4104.12 kWh energy and have 1039.14 kg CO<sub>2</sub>e carbon footprint.

### A.2 Probing and Finetuning

**Probing** We ran our probing on checkpoints reported in Table 3. All the probing packages used in our paper are publicly available. We use LKT (Liu et al., 2019a)<sup>10</sup> and BLIMP (Warstadt et al., 2020)<sup>11</sup> for probing linguistic knowledge; LAMA (Petroni et al., 2019)<sup>12</sup> for factual and common-

<sup>7</sup><https://cloud.google.com/tpu/docs/tutorials/roberta-pytorch>

<sup>8</sup><https://github.com/pytorch/fairseq>

<sup>9</sup><http://www.green-algorithms.org/>

<sup>10</sup><https://github.com/nelson-liu/contextual-repr-analysis>

<sup>11</sup><https://github.com/awsmlabs/mlm-scoring>

<sup>12</sup><https://cloud.google.com/tpu/docs/tutorials/roberta-pytorch>

sense knowledge; CAT (Zhou et al., 2020)<sup>13</sup> for commonsense knowledge; and OLMPICS (Talmor et al., 2019)<sup>14</sup> for reasoning ability. As noted before, LKT is the only probe package that requires additional training. All experiments in LKT are seeded and strictly follow the hyperparameters and training setup used in Liu et al. (2019a).

We estimate the total cost of probing: 165.54 kg CO<sub>2</sub>e carbon footprint and 345.26 kWh energy (Lanelongue et al., 2020).

**Finetuning** For convenience and reproducibility, we also use public package to conduct finetuning. We use `jiant`<sup>15</sup> for ReCoRD and WSC; and Huggingface for the rest. Due to the large amount of training involved, we choose 14 intermediate checkpoints. In addition, it is infeasible to find optimal hyperparameters for each individual checkpoint; to be as fair as we can, we use the same set of hyperparameters (see Table 4) for running each finetuning task on all our checkpoints and original RoBERTa. This might explain the sub-optimal performance from the original RoBERTa in Fig. 3.

We estimated that finetuning emits 54.98 kg CO<sub>2</sub>e carbon footprint and consumes 121.64 kWh energy (Lanelongue et al., 2020).

**Hardware** We run our probing and finetuning experiments on Intel Core i9-9820X CPU @ 3.30GHz and GTX 2080 Ti.

## B Additional Probing Results

### B.1 Complete Results for Learning Patterns

Here we show plots from all probing tasks we tested. Overall, the selected graphs in Fig. 1 are representative. See complete results of BLIMP in Fig. 5, showing fast learning speed; CAT in Fig. 6 showing slower learning speed; and OLMPICS in Fig. 7 shows a sign of “not learning” in many tests and “learning” only in a few tests.

### B.2 Correlation Plot

In Fig. 4, we show a plot of correlation among all experimental results run on our replicated RoBERTa (§3 and §5). Although successful probes usually correlate with each other, we note that low correlation with successful probes (e.g., linguistic probes) doesn’t necessarily imply failure. One such example is Ellipsis in BLIMP, which shows great

success in Fig. 5, yet has low correlation with other successful probes.

### B.3 Complete Results for Varying Pretrain Corpus Domain

In this section, we complement the domain results from Fig. 2. Note that the average performance shown here is calculated with all tested tasks in each probe package, whereas Fig. 2 only includes the same tasks as Fig. 1 for illustration purposes.

See complete results of LKT in Fig. 8 and BLIMP in Fig. 9 showing relatively small domain impact; LAMA in Fig. 10 shows arguably the largest impact; CAT in Figure 11 showing noticeable impact on some tasks; we see large variation of the impact from the domain OLMPICS in Fig. 12. Note again that STORIES usually gives the slowest learning speed and worst final performance from tested probes.

### B.4 Complete Results for Varying Pretrain Corpus Size

In this section, we complement the corpus size results from Fig. 2. Note that the average performance shown here is calculated with all included tasks in each probe package, whereas Fig. 2’s use the same tasks as Fig. 1 for illustration purpose. See complete results of LKT in Fig. 13; BLIMP in Fig. 14; LAMA in Fig. 15; CAT in Fig. 16; OLMPICS in Fig. 17. To reiterate our conclusion from §4, the biggest corpus (with more diverse inclusion of domains) generally learns faster and results in an ultimately better RoBERTa in all tested knowledge; in contrast, we don’t observe this when we simply change from downsampled English WIKIPEDIA (4 GB) to original English WIKIPEDIA (12 GB).

<sup>13</sup><https://github.com/XuhuiZhou/CATS>

<sup>14</sup><https://github.com/alontalmor/olmpics>

<sup>15</sup><https://github.com/nyu-ml/jiant>

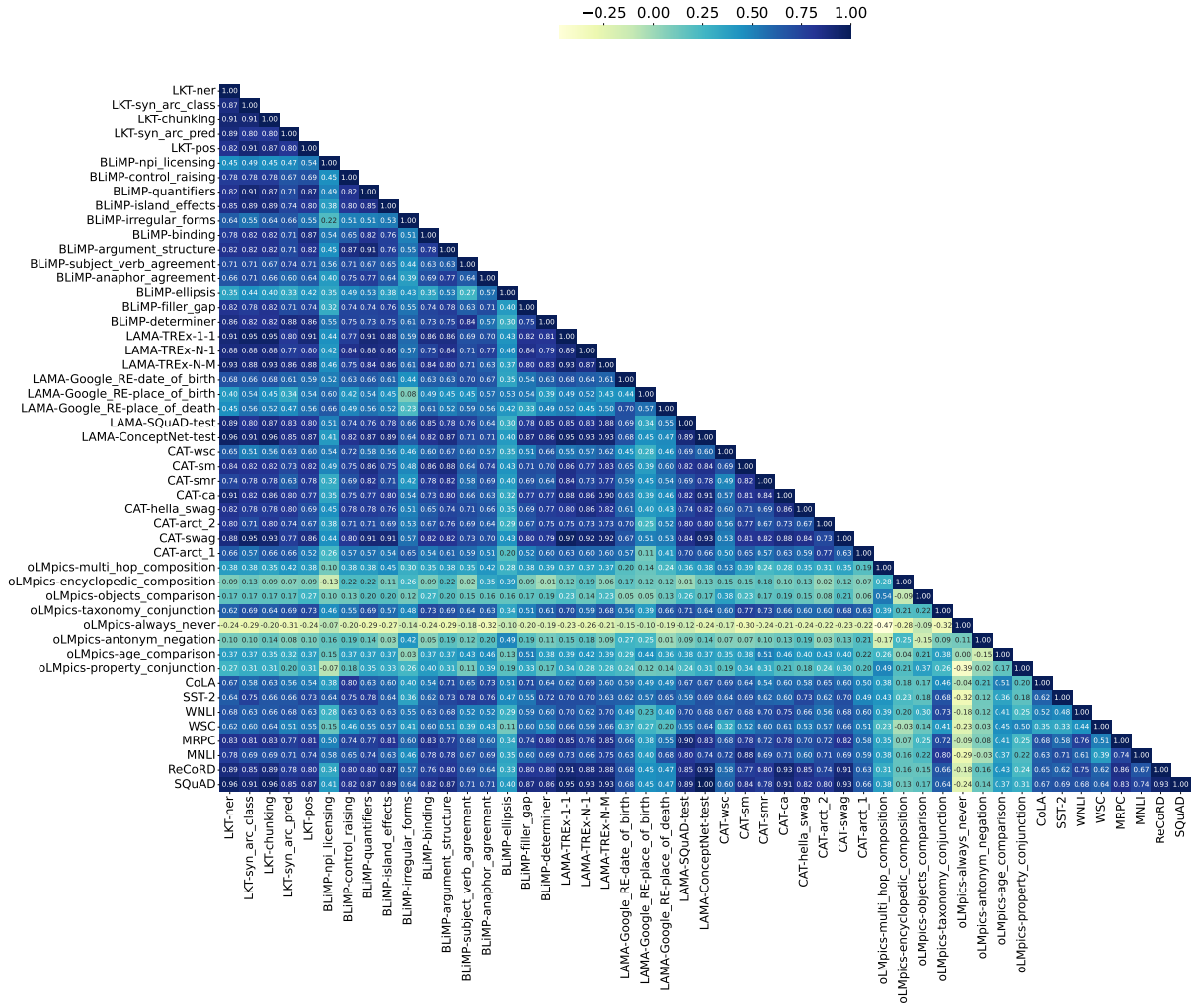


Figure 4: Plot of **Kendall Tau correlation** among all experiments included. We calculate based on results from the 14 checkpoints used in Fig. 3

Name	Values
Architecture	RoBERTa-base
Masking	Static
Update steps	1M
Batch size	256
Max length	512
Warmup steps	10K
Peak Learning rate	0.0005
Learning rate scheduler	Polynomial Decay
Dropout rate	0.1
Attention dropout rate	0.1
Weight Decay rate	0.01
Optimizer	Adam( $\beta_s = (0.9, 0.98)$ , $\epsilon = 1e - 6$ )

Table 2: Configurations for pretraining RoBERTa.

Domain	No. of Checkpoints
NEWS+WIKI+STORIES (36GB)	65
NEWS (12GB)	56
STORIES (12GB)	52
WIKI (12GB)	71
Small WIKI (4GB)	52
Reproduced	62

Table 3: Numbers of saved checkpoints during pretraining RoBERTa on different domains. “Reproduced” denotes our replication of model in (Liu et al., 2019b), detailed in §2.3. The rest corresponds to corpora experimented on basis of domains and corpus sizes, detailed in §4.

	CoLA	SST-2	MRPC	WNLI	MNLI	SQuAD	WSC	ReCoRD
<b>Batch size</b>	32	32	32	32	32	12	32	32
<b>Epoch</b>	3	3	3	3	3	2	10	2
<b>Learning rate</b>	2e-5	2e-5	2e-5	2e-5	2e-5	3e-5	1e-5	1e-5

Table 4: Important hyperparameters for finetuning experiments in §5. Experiments are seeded for reproducibility.

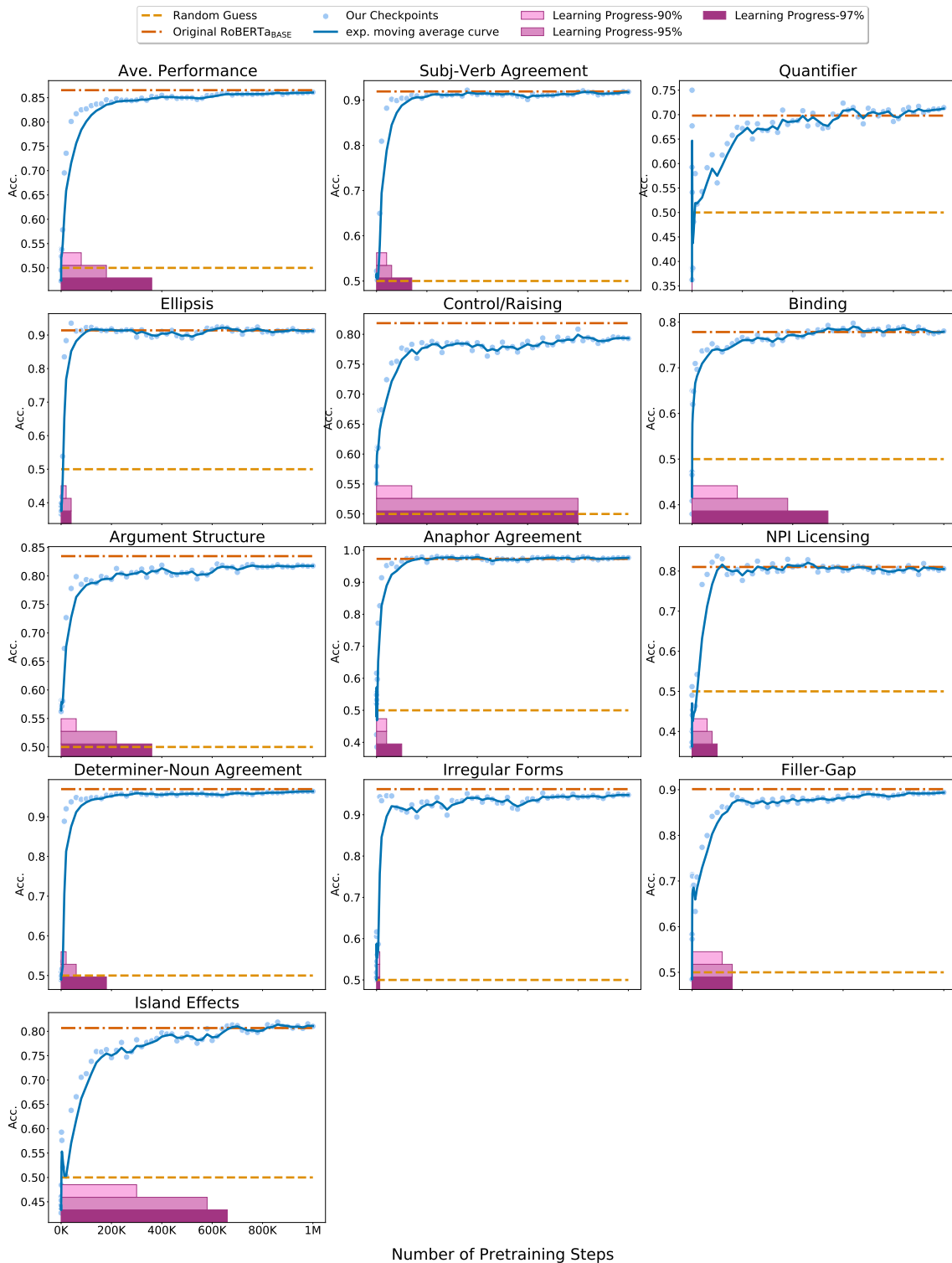


Figure 5: Complete results of **BLIMP (Linguistics)** in Figure 1, plotted in the same format.

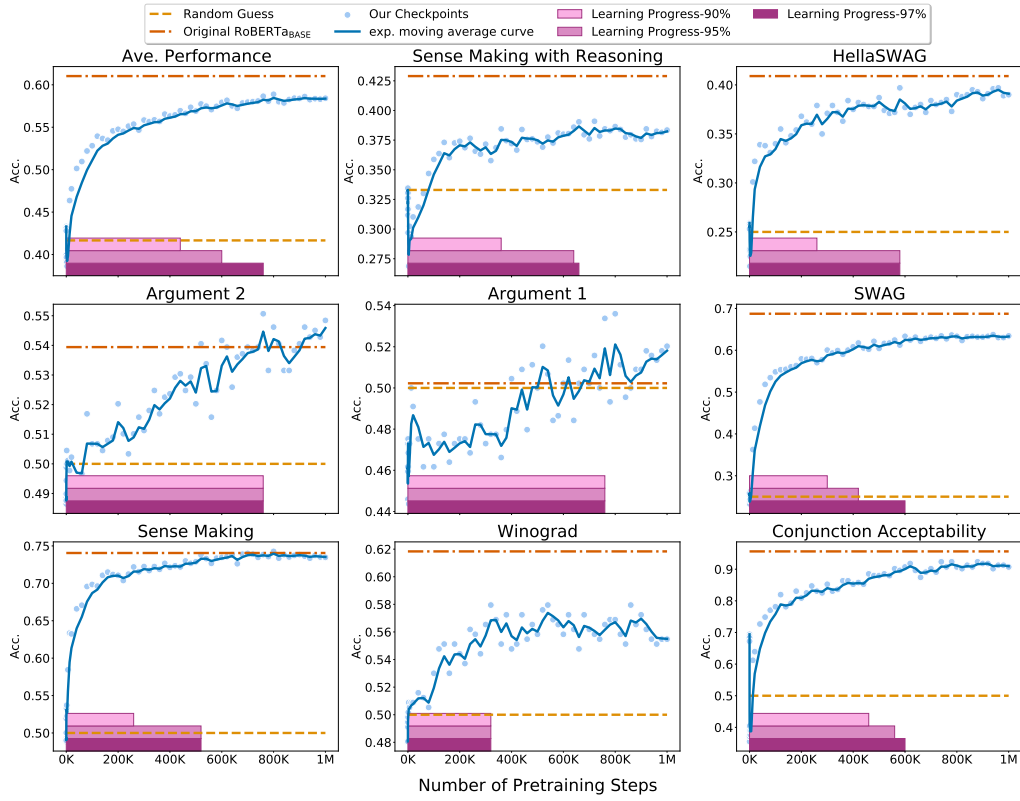


Figure 6: The complete results of **CAT (Commonsense)** in Figure 1, plotted in the same format.

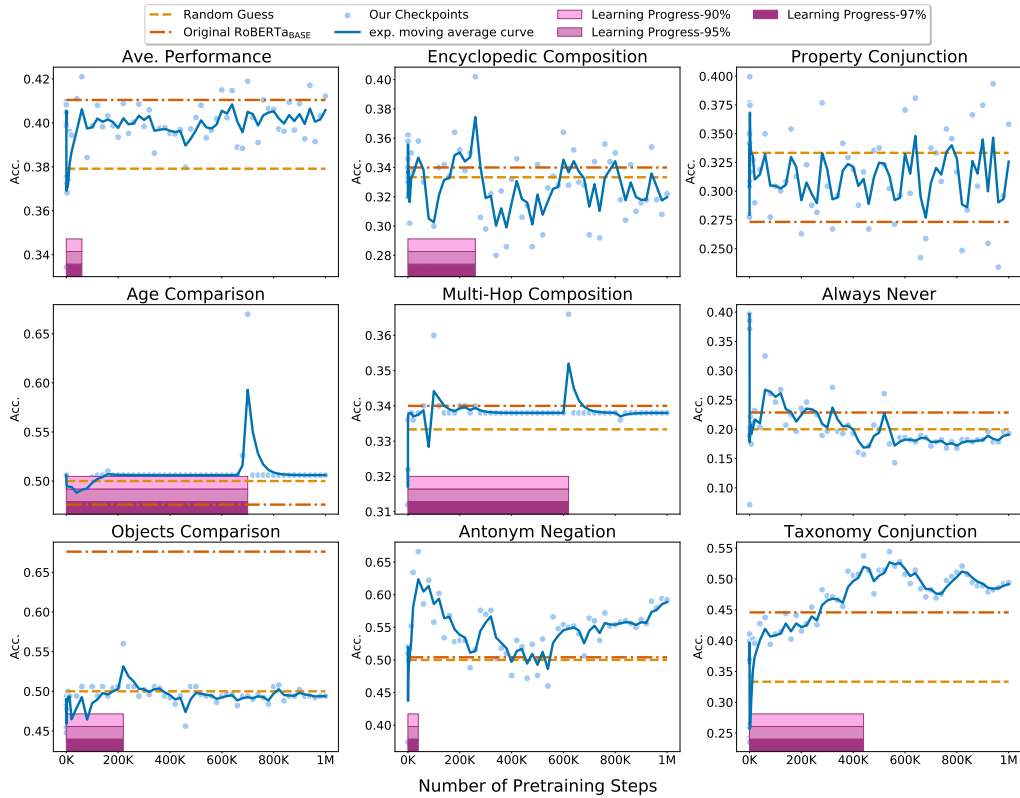


Figure 7: Complete results of **OLMPICS (Reasoning)** in Figure 1, plotted in the same format.



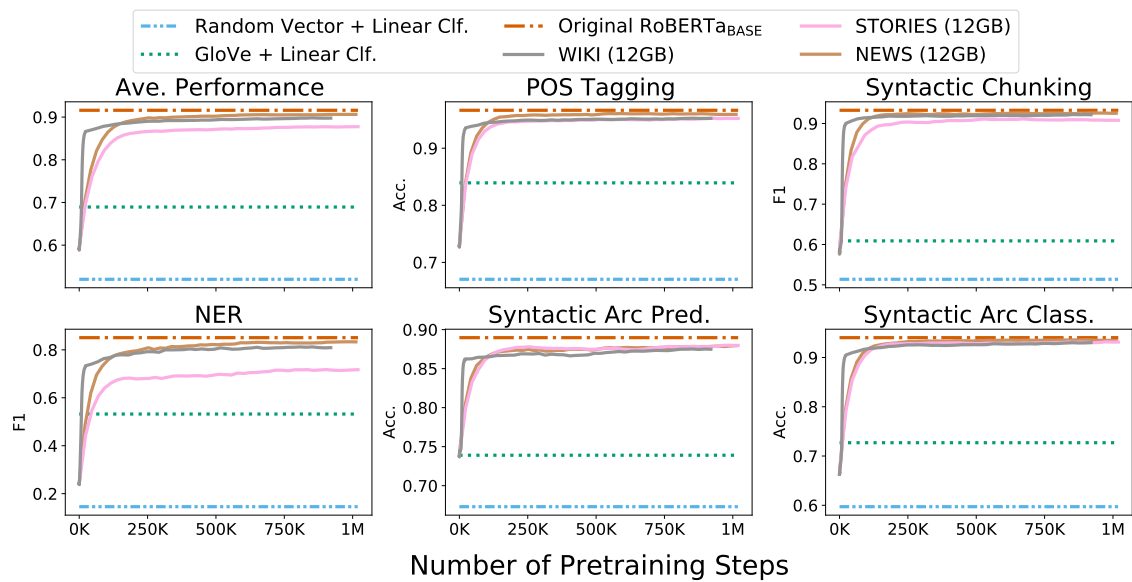


Figure 8: Complete results of **LKT (Linguistics) on different domains**. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

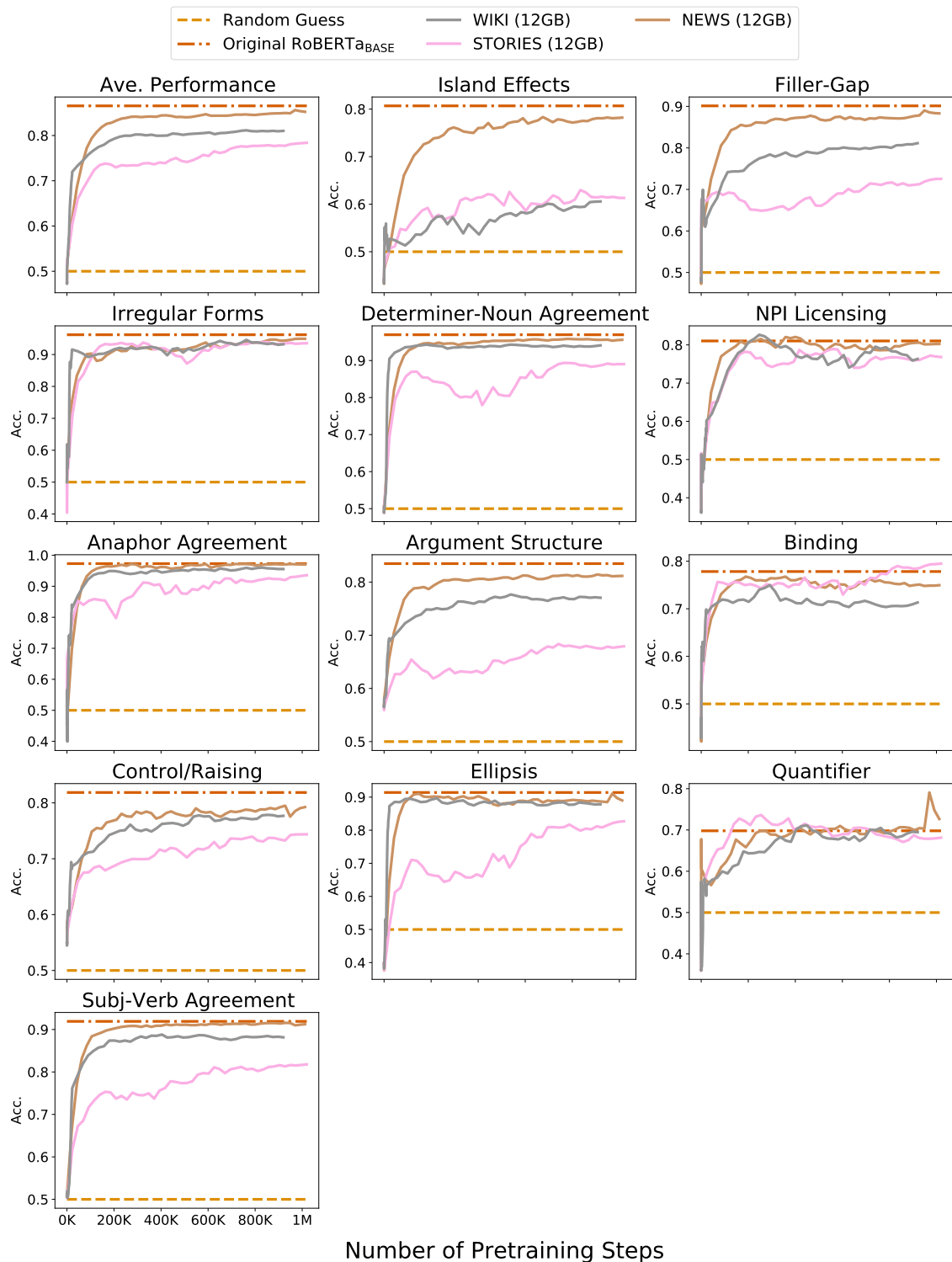


Figure 9: Complete results of **BLIMP (Linguistics)** on different domains. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

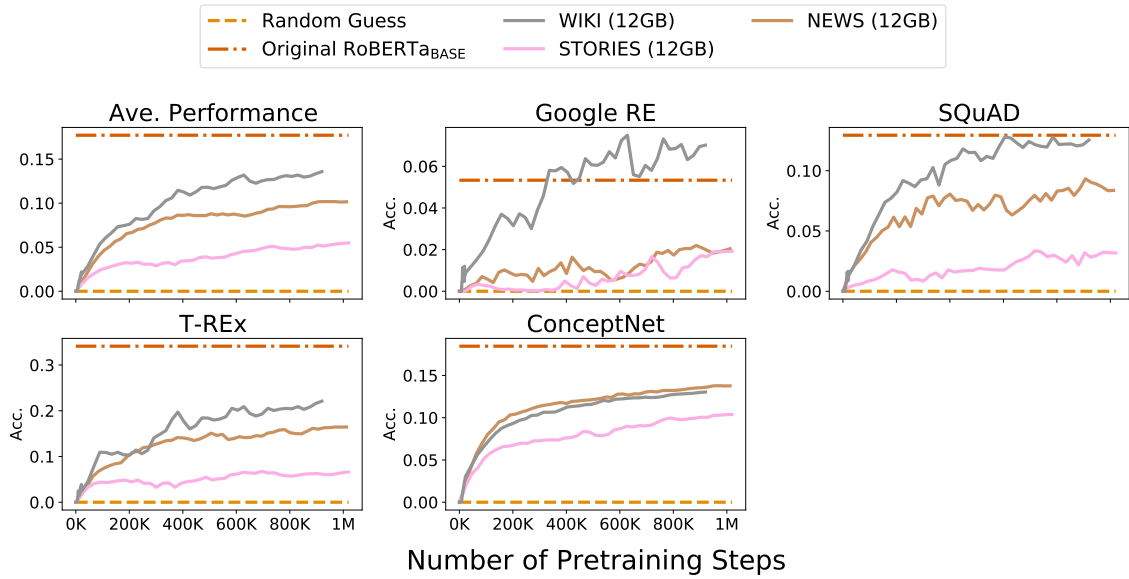


Figure 10: Complete results of **LAMA (Factual & Commonsense)** on different domains. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

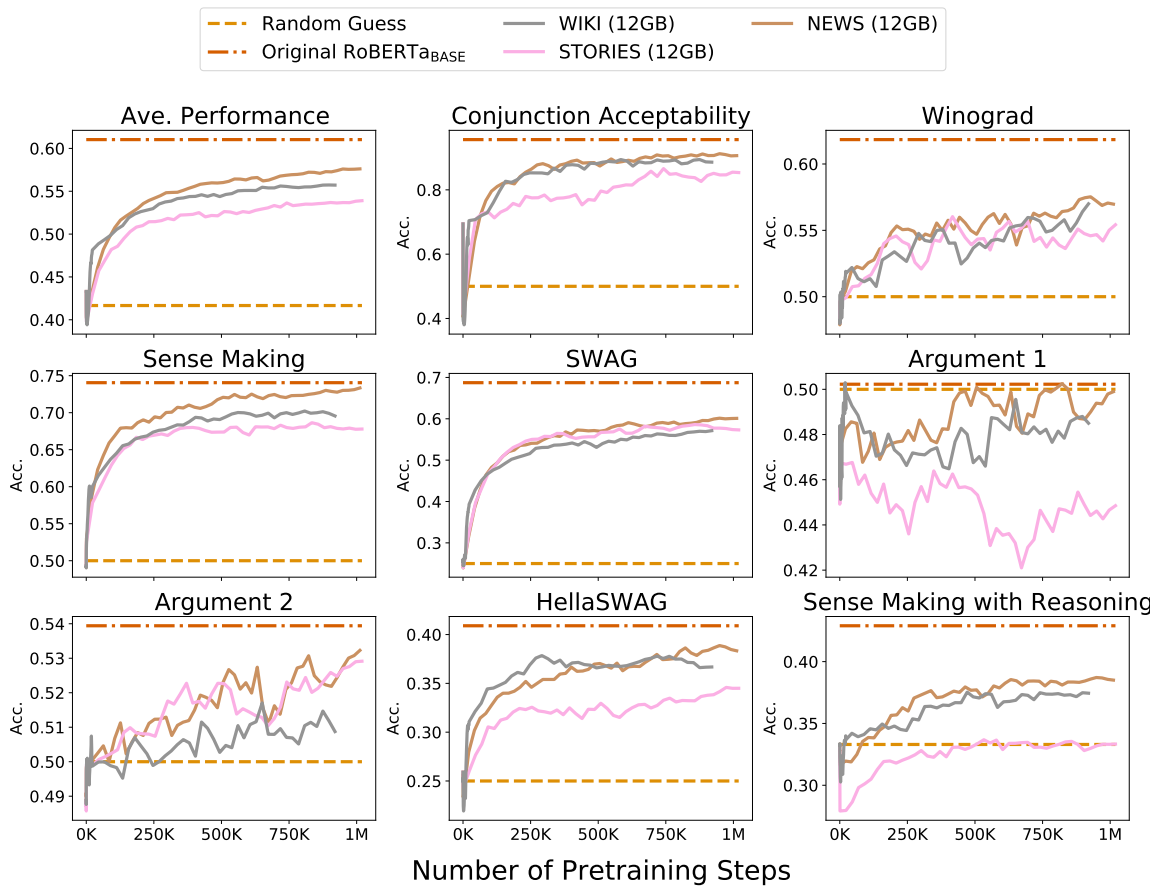


Figure 11: Complete results of **CAT (Commonsense)** on different domains. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

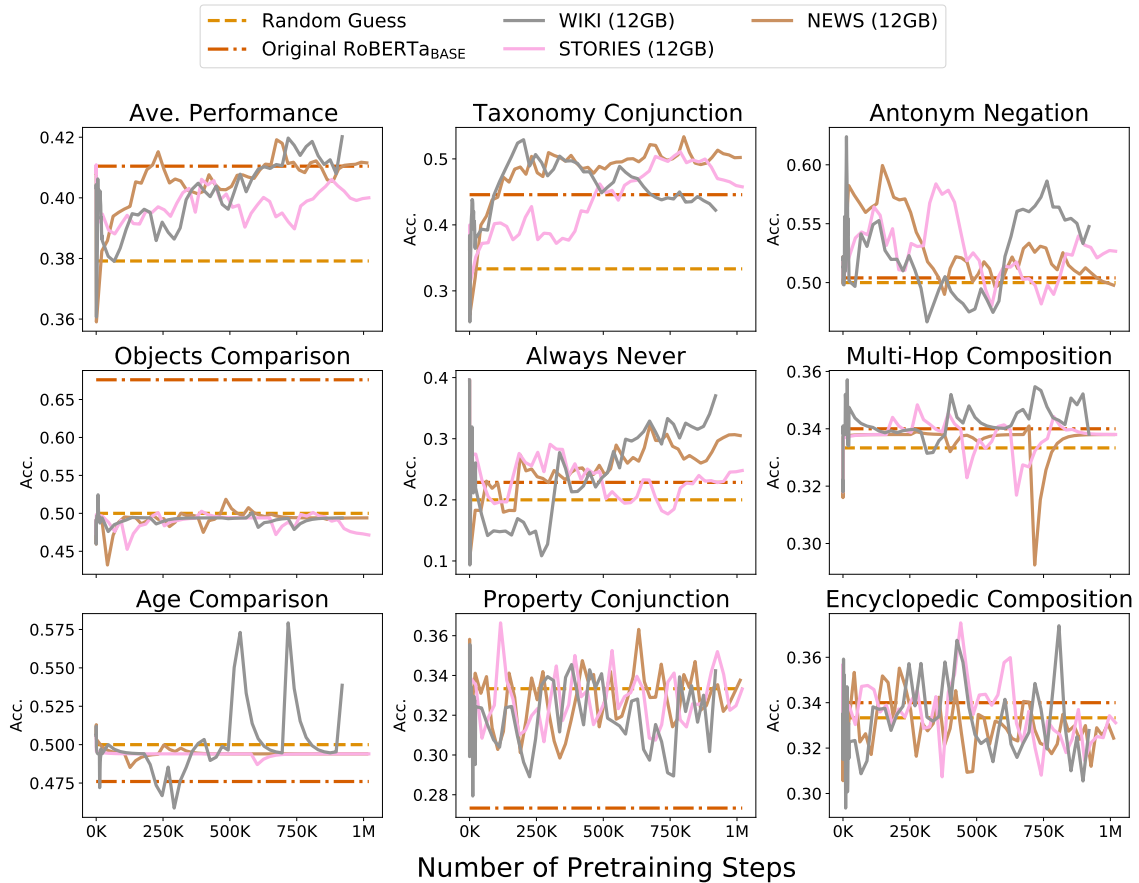


Figure 12: Complete results of OLMPICS (**Reasoning**) on different domains. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

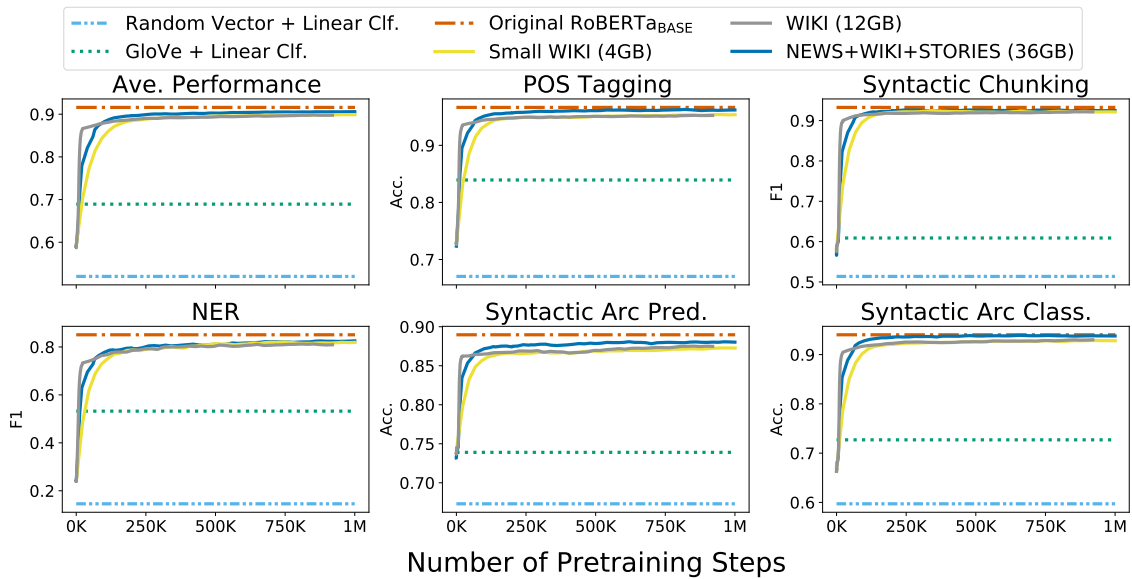


Figure 13: Complete results of LKT (**Linguistics**) on different corpus sizes. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

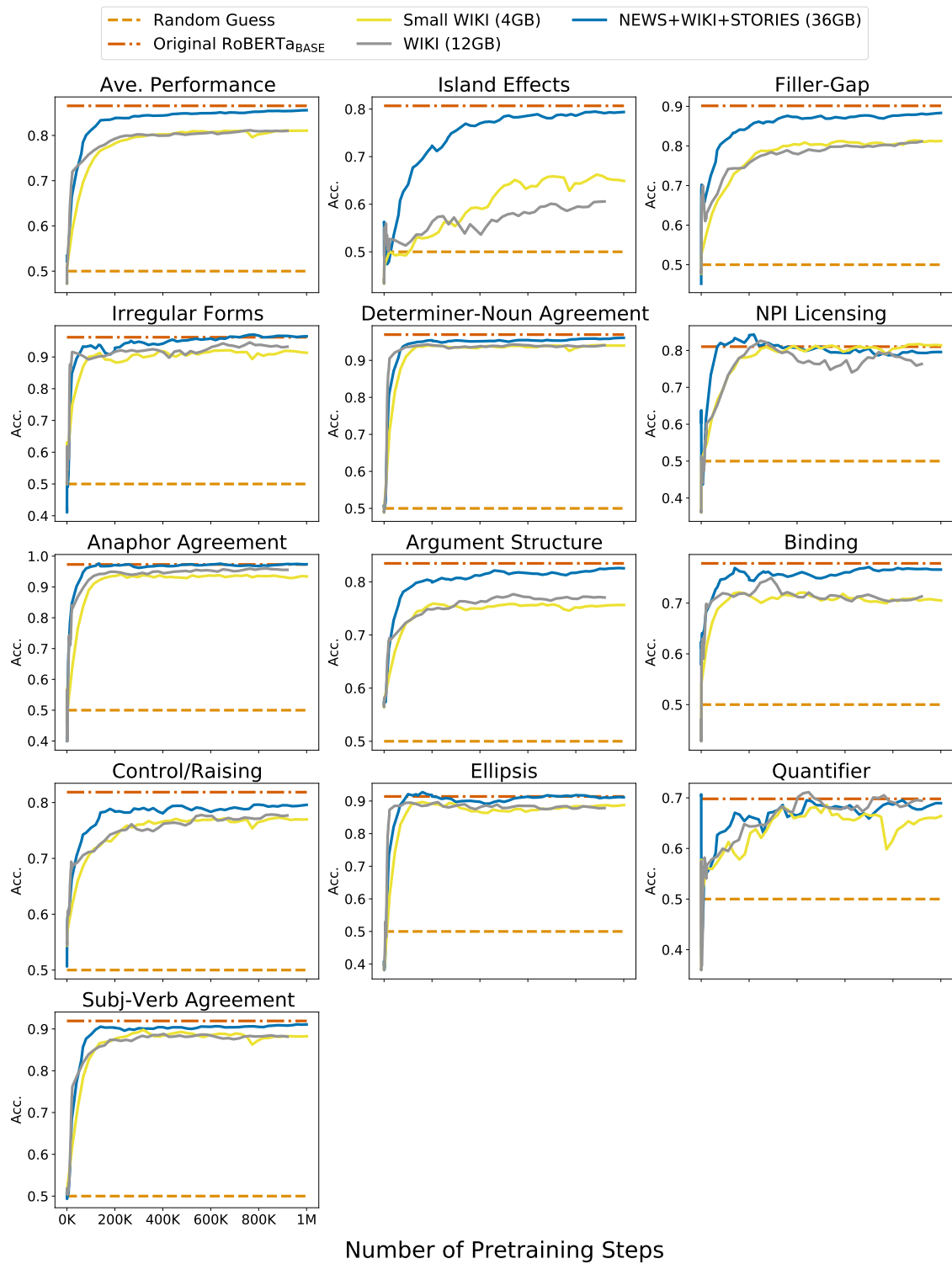


Figure 14: Complete results of **BLIMP (Linguistics)** on different corpus sizes. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

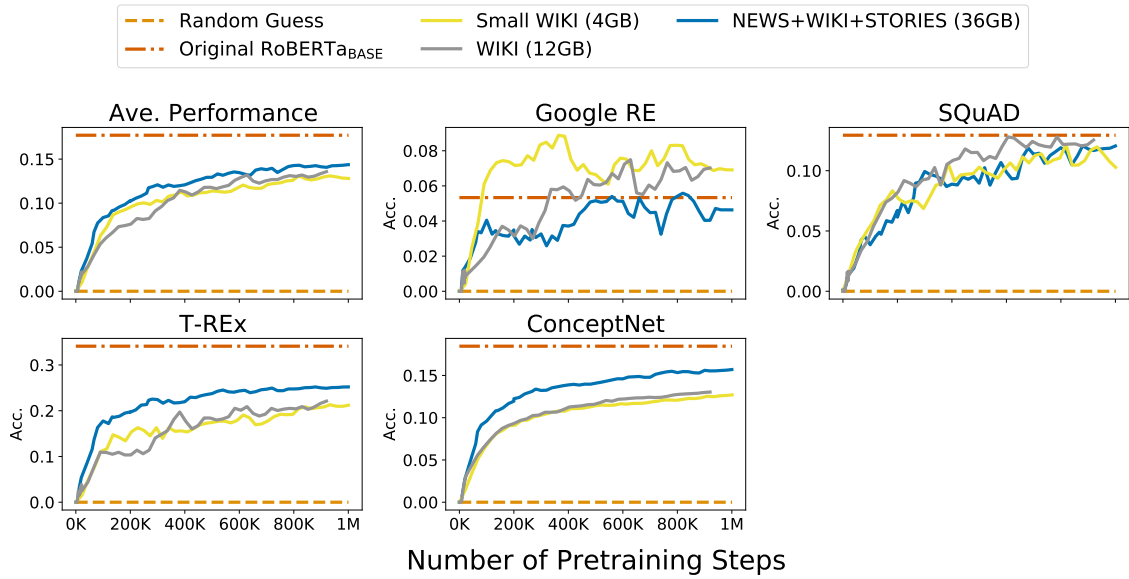


Figure 15: Complete results of **LAMA (Factual & Commonsense)** on different domains. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

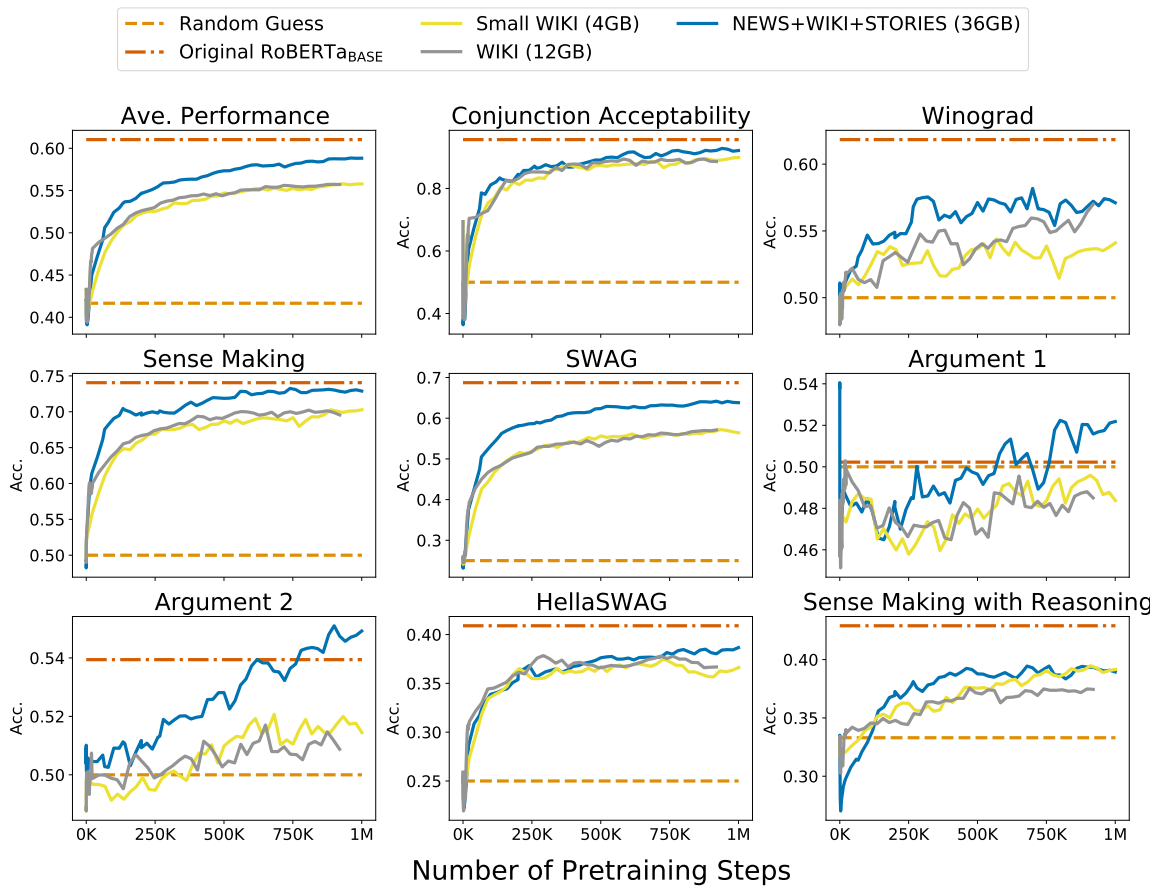


Figure 16: Complete results of **CAT (Commonsense)** on different corpus sizes. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.

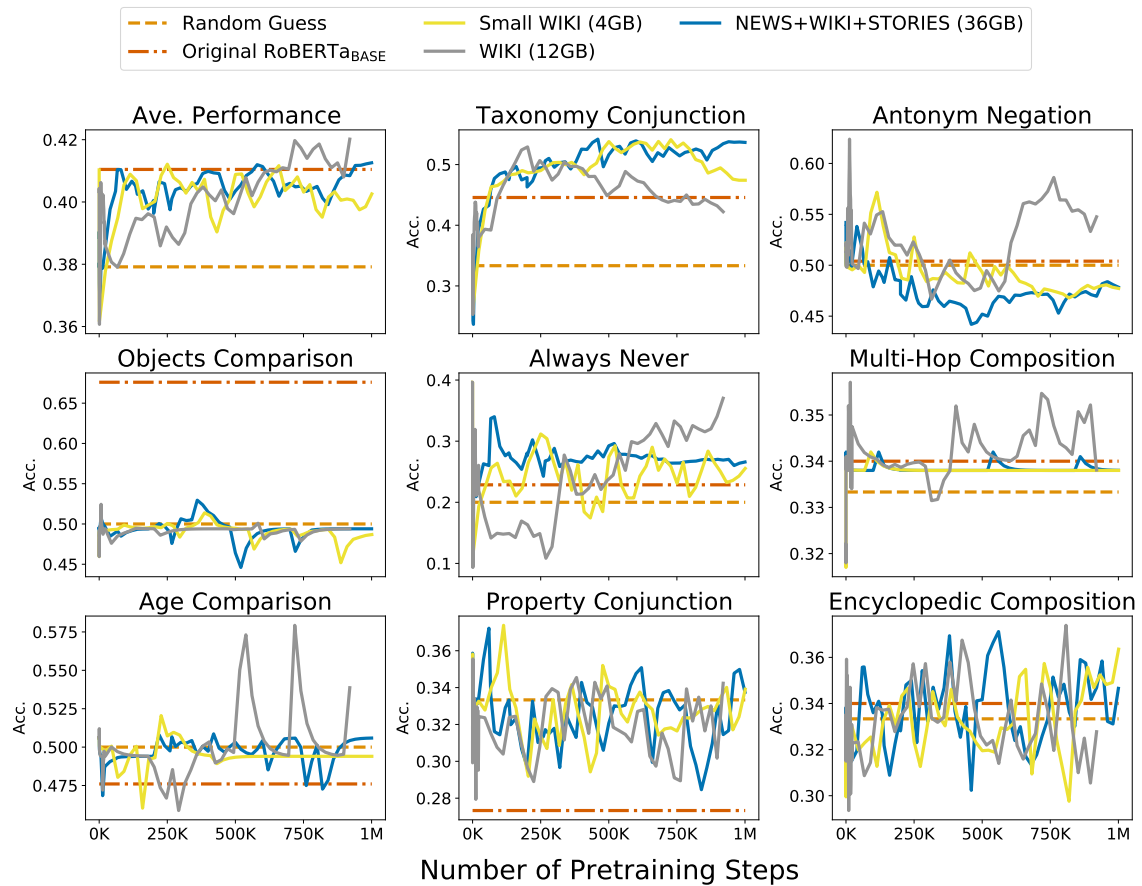


Figure 17: Complete results of OLMPICS (Reasoning) on different corpus sizes. Following Fig. 1, every line in plots are smoothed using exponential moving average with coefficient 0.5. Average performance is calculated from all included tasks in this plot.