How much pretraining data do language models need to learn syntax?

Laura Pérez-Mayos*[1], Miguel Ballesteros[2], Leo Wanner[3,1]

[1] TALN Research Group, Pompeu Fabra University, Barcelona, Spain
[2] Amazon AI
[3] Catalan Institute for Research and Advanced Studies (ICREA), Barcelona, Spain

{laura.perezm|leo.wanner}@upf.edu
ballemig@amazon.com

Abstract

Transformers-based pretrained language models achieve outstanding results in many well-known NLU benchmarks. However, while pretraining methods are very convenient, they are expensive in terms of time and resources. This calls for a study of the impact of pretraining data size on the knowledge of the models. We explore this impact on the syntactic capabilities of RoBERTa, using models trained on incremental sizes of raw text data. First, we use syntactic structural probes to determine whether models pretrained on more data encode a higher amount of syntactic information. Second, we perform a targeted syntactic evaluation to analyze the impact of pretraining data size on the syntactic generalization performance of the models. Third, we compare the performance of the different models on three downstream applications: part-of-speech tagging, dependency parsing and paraphrase identification. We complement our study with an analysis of the cost-benefit trade-off of training such models. Our experiments show that while models pretrained on more data encode more syntactic knowledge and perform better on downstream applications, they do not always offer a better performance across the different syntactic phenomena and come at a higher financial and environmental cost.

1 Introduction

The use of unsupervised pretrained language models in the context of supervised tasks has become a widely spread practice in NLP, with Transformer-based models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019b) achieving outstanding results in many well-known Natural Language Understanding benchmarks such as GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2018). Consequently, several studies investigate the types of knowledge learned by BERT, how and where this knowledge is represented and what the best methods to improve it are; see, e.g., (Rogers et al., 2020). There is evidence that, among other information (e.g., part-of-speech, syntactic chunks and roles (Tenney et al., 2019; Lin et al., 2019; Belinkov et al., 2017), morphology in general (Peters et al., 2018), or sentence length (Adi et al., 2016)), BERT representations implicitly embed entire syntax trees (Hewitt and Manning, 2019b).

Language models are traditionally assessed by information-theoretical metrics such as perplexity, i.e., the probability of predicting a word in its context. The general wisdom is that the more pretraining data a model is fed, the lower its perplexity gets. However, large volumes of pretraining data are not always available and pretraining is costly, such that the following questions need to be answered: **(i)** Do we always need models pretrained on internet-scale corpora? **(ii)** As the models are pretrained on more data, and their perplexity improves, do they encode more syntactic information and offer a better syntactic generalization? **(iii)** Do the models with more pretraining perform better when applied in downstream tasks? To address these questions, we explore the relation between the size of the pretraining data and the syntactic capabilities of RoBERTa by means of the MiniBERTas models, a set of 12 RoBERTa models pretrained from scratch by Warstadt et al. (2020b) on quantities of data ranging from 1M to 1B words. In particular:

- We use the syntactic structural probes from Hewitt and Manning (2019b) to determine whether those models pretrained on more data encode a higher amount of syntactic information than those trained on less data;
- We perform a targeted syntactic evaluation to analyze the generalization performance of the different models using SyntaxGym (Gauthier et al., 2020) and the syntactic tests presented in (Hu et al., 2020);

[*] Work partially done during internship at Amazon AI.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1571–1582
November 7–11, 2021. ©2021 Association for Computational Linguistics

- We compare the performance of the different models on two morpho-syntactic tasks (PoS tagging and dependency parsing), and a non-syntactic task (paraphrase identification);
- We conduct a cost-benefit trade-off analysis (Strubell et al., 2019; Bhattacharjee et al., 2020) of the models training.

We observe that models pretrained on more data encode a higher amount of syntax according to Hewitt and Manning (2019b)'s metrics, but do not always lead to a better syntactic generalization. Indeed, we find that models pretrained on less data perform equally good or even better than those pretrained on more data on 3 out of 6 syntactic test suites. When applied to downstream tasks, the models pretrained on more data perform generally better. However, the analysis of the trade-off between the cost of training a model and its performance shows that small performance gains come at a high economical and environmental cost that should be considered when developing new models.

In what follows, Section 2 provides some background on the syntactic assessment of language models, model costs, and the works related to ours. Section 3 describes our experimental setup, introducing the MiniBERTas models and the syntactic tests as well as the downstream applications we explore. Section 4 presents the outcome of our experiments. Section 5 offers a cost-benefit analysis of the pretraining of the different models, and Section 6 summarizes the implications that our work has for the use of pretrained language models.

2 Background

2.1 Syntactic assessment of language models

The targeted syntactic evaluation incorporates methods from psycholinguistic experiments, focusing on highly specific measures of language modeling performance and allowing to distinguish models with human-like representations of syntactic structure (Linzen et al., 2016; Lau et al., 2017; Gulordava et al., 2018; Marvin and Linzen, 2018; Futrell et al., 2019). Regarding the evaluation of modern language models, Warstadt et al. (2020a) present a challenge set that isolates specific phenomena in syntax, morphology, and semantics, finding that state-of-the-art models struggle with some subtle semantic and syntactic phenomena, such as negative polarity items and extraction islands. Hu et al. (2020) test 20 model type combinations and data sizes on 34 English syntactic test suites, finding substantial differences in syntactic generalization performance by model architecture.

Supervised probing models have also been used to test for the presence of a wide range of linguistic phenomena (Conneau et al., 2018; Liu et al., 2019a; Tenney et al., 2019; Voita and Titov, 2020; Elazar et al., 2020), and it has been shown that entire syntax trees are embedded implicitly in BERT's vector geometry (Hewitt and Manning, 2019b; Chi et al., 2020). However, other works have criticized some probing methods, claiming that classifier probes can learn the linguistic task from training data (Hewitt and Liang, 2019), and can fail to determine whether the detected features are actually used (Voita and Titov, 2020; Pimentel et al., 2020; Elazar et al., 2020).

2.2 Costs of modern language models

While modern language models keep growing in orders of magnitude, so do the resources necessary for their development and, consequently, also the inclusivity gap. The financial cost of the required hardware and electricity favors industry-powered research, and harms academics, students, and non-industry researchers, particularly those from emerging economies. Moreover, the training of such models is not only financially expensive, but has also a large carbon footprint. Schwartz et al. (2019) propose to report the financial cost of developing, training, and running models in order to provide baselines for the investigation of increasingly efficient methods. Along the same lines, Strubell et al. (2019) offer an analysis of the computation required for the research, development and hyperparameter tuning of several recently successful neural network models for NLP, and propose actionable recommendations to reduce costs and improve equity, namely 1) reporting training time and sensitivity to hyperparameters; 2) a government-funded academic compute cloud to provide equitable access to all researchers; and 3) prioritizing computationally efficient hardware and algorithms.

2.3 Related work

Several studies investigate the relation between pretraining data size and linguistic knowledge in language models. van Schijndel et al. (2019); Hu et al. (2020); Micheli et al. (2020) find out that, given a relatively large data size (e.g., 10M words), models with less pretraining perform similarly to models with much more pretraining, concluding that model architecture plays a more important role than training data scale in yielding correct syntactic generalizations (Hu et al., 2020). Complementary, Raffel et al. (2020) shows that performance can degrade when an unlabeled data set is small enough that it is repeated many times over the course of pretraining. In contrast, Zhang et al. (2020) argue that while relatively small datasets suffice to reliably encode most syntactic and semantic features, a much larger quantity of data is needed to master conventional NLU tasks. This discrepancy may be due to the difference in model architectures, pretraining techniques and the scaling and nature of the difference datasets.

Our work differs significantly from recent works. We make use of a single architecture and data source, and focus exclusively on the syntactic capabilities of the models, offering an in-depth analysis that includes structural syntactic probing, detailed syntactic generalization, and downstream applications performance. Moreover, we also provide a cost-benefit analysis of the models.

3 Experimental setup

3.1 The MiniBERTas models

The MiniBERTas are a set of 12 RoBERTa models pretrained from scratch by Warstadt et al. (2020b) on 4 datasets containing 1B, 100M, 10M and 1M tokens, available through HuggingFace Transformers.[1] The datasets are sampled from Wikipedia and Smashwords – the two datasets that make up the original pretraining dataset of BERT and that are included in the RoBERTa pretraining data. For each dataset size, pretraining is run 25 times (10 times for 1B) with varying hyperparameter values; the three models with the lowest development set perplexity are released. For the smaller dataset, a smaller model size is used to prevent over-fitting. We refer to models trained on the same amount of data as a *family* of models, and models inside a family as *intra-family members* (e.g.,the *roberta-base-100M-1* model is a member of the *roberta-base-100M* family). Table 1 offers an overview of the hyperparameters per model size.

3.2 Structural probing

Hewitt and Manning (2019b)'s structural probes assess how well syntax trees are embedded in a linear transformation of the network representation space applying two different evaluations: Tree distance evaluation, in which squared L2 distance encodes the distance between words in the parse tree, and Tree depth evaluation, in which squared L2 norm encodes the depth in the parse tree.

Model Size	L	AH	HS	FFN	P
BASE	12	12	768	3072	125M
MED-SMALL	6	8	512	2048	45M

Table 1: Hyperparameters per model sizes. AH = number of attention heads; HS = hidden size; FFN = feed-forward network dimension; P = number of parameters.

Tree distance evaluation. Evaluates how well the predicted distances between all pairs of words in a model reconstruct gold parse trees by computing the Undirected Unlabeled Attachment Score (*UUAS*). It also computes the Spearman correlation between true and predicted distances for each word in each sentence, averaging across all sentences with lengths between 5 and 50 (we refer to as *DSpr.*).

Tree depth evaluation. Evaluates the ability of models to recreate the order of words specified by their depth in the parse tree, assessing their ability to identify the root of the sentence as the least deep word (*Root %*) and computing the Spearman correlation between the predicted and the true depth ordering, averaging across all sentences with lengths between 5 and 50 (we refer to as *NSpr*).

3.3 Targeted syntactic evaluation

We test the MiniBERTas on the syntactic tests assembled by Hu et al. (2020), accessible through the SyntaxGym toolkit (Gauthier et al., 2020). The tests are divided into 6 syntactic circuits, introduced below, based on the type of algorithm required to successfully process each construction.

1. Agreement: Tests a language model for how well it predicts the number marking on English finite present tense verbs. It is composed of 3 Subject-Verb Number Agreement tests from Marvin and Linzen (2018),

2. Center Embedding: Tests the ability to embed a phrase in the middle of another phrase of the same type. Subject and verbs must match in a first-in-last-out order, meaning models must approximate a stack-like data-structure in order to successfully process them. The circuit is composed of 2 tests from Wilcox et al. (2019a).

3. Garden-Path Effects: Measures the syntactic phenomena that result from tree structural ambiguities that give rise to locally coherent but globally

[1] https://huggingface.co/nyu-mll

implausible syntactic parses. The circuit is composed of 2 Main Verb / Reduced Relative Clause (MVRR) tests and 4 NP/Z Garden-paths (NPZ) tests, all from Futrell et al. (2018).

4. Gross Syntactic Expectation: Tests the ability of the models to distinguish between coordinate and subordinate clauses: introducing a subordinator at the beginning of the sentence should make an ending without a second clause less probable, and should make a second clause more probable. The circuit is composed of 4 Subordination tests from Futrell et al. (2018).

5. Licensing: Measures when a particular token must exist within the scope of an upstream licensor token. The circuit is composed of 4 Negative Polarity Item Licensing (NPI) tests and 6 Reflexive Pronoun Licensing tests, all from Marvin and Linzen (2018).

6. Long-Distance Dependencies: Measures covariations between two tokens that span long distances in tree depth. The circuit is composed of 6 Filler-Gap Dependencies (FGD) tests from Wilcox et al. (2018) and Wilcox et al. (2019b), and 2 Cleft tests from (Hu et al., 2020).

3.4 Encoding unidirectional context with bidirectional models

The tests in SyntaxGym evaluate whether models are able to assign a higher probability to grammatical and natural continuations of sentences. As RoBERTa is a bidirectional model, to be able to ask it to predict the probability of a token given the context of previous tokens we test it in a left-to-right generative setup, as done in (Rongali et al., 2020; Zhu et al., 2020). More precisely, we follow Wang and Cho (2019)'s sequential sampling procedure, which is not affected by the error that was reported in equations 1-3, related to the Non-sequential sampling procedure. To compute the probability distribution for a sentence with N tokens, we start with a sequence of *begin_of_sentence* token plus N *mask* tokens plus an extra *mask* token to account for the *end_of_sentence* token. For each masked position in $[1, N]$, we compute the probability distribution over the vocabulary given the left context of the original sequence, and select the probability assigned by the model to the original word. Note that this setup allows the models to know how many tokens there are in the sentences, and therefore the results are not directly comparable with those of unidirectional models, that do not have any information regarding the length of the sequence.

For example, in a Subordination test with the examples 'Because the students did not like the material.' and 'The students did not like the material.', we expect the model to assign a higher surprisal (Wilcox et al., 2019c) to the first example, because the initial "Because" implies that the immediately following clause is not the main clause of the sentence, but instead is a subordinate that must be followed by the main clause. However, instead of finding the main clause, the model encounters a dot indicating the end of the sentence. To test whether the model has learned about subordination, we feed the models the tokens sequences [*begin_of_sentence*, *Because*, *the*, *students*, *did*, *not*, *like*, *the*, *materials*, *mask*, *mask*] and [*begin_of_sentence*, *The*, *students*, *did*, *not*, *like*, *the*, *materials*, *mask*, *mask*], and compare the surprisal of the model predicting a dot '.' for the first masked position in each case.

3.5 Downstream applications

To compare the performance of the models on downstream applications, we analyze their learning curves along the fine-tuning process on two morpho-syntactic tasks (PoS tagging and dependency parsing) and a non-syntactic task (paraphrase identification). Each task is fine-tuned for 3 epochs, with the default learning rate of 5e^{-5}. To mitigate the variance in performance induced by weight initialization and training data order (Dodge et al., 2020; Reimers and Gurevych, 2017), we repeat this process 5 times per task with different random seeds and average results.[2] For **PoS tagging**, we fine-tune RoBERTa with a linear layer on top of the hidden-states output for token classification.[3] Dataset: Universal Dependencies Corpus for English (UD 2.5 English EWT (Silveira et al., 2014)). For **Dependency parsing**, we fine-tune a Deep Biaffine neural dependency parser (Dozat and Manning, 2016). Dataset: UD 2.5 English EWT (Silveira et al., 2014). For **Paraphrase identification**, we fine-tune RoBERTa with a linear layer on top of the pooled sentence representation.[4] Dataset: Microsoft Research Paraphrase Corpus (MRPC)

[2] The implementation relies in the Transformers library (Wolf et al., 2020) and AllenNLP (Gardner et al., 2018). For implementation details, pretrained weights and hyperparameter values, cf. the documentation of the libraries.

[3] Source: `https://github.com/Tarpelite/UniNLP/blob/master/examples/run_pos.py`

[4] Source: `https://github.com/huggingface/transformers/blob/master/examples/text-classification/run_glue.py`.

(Dolan and Brockett, 2005).

4 Results

In this section, we explore the impact of the size of pretraining data on the syntactic information encoded by RoBERTa from three different angles.

4.1 Structural probing

We use Hewitt and Manning's syntactic structural probes to determine whether the MiniBERTa models pretrained on more data encode a higher amount of syntactic information than those trained on less data. Following the original work, we probe layer 7 of all models, as it was shown to encode most of the syntax. Results are shown in Table 2.

	Tree distance eval.		Tree depth eval.	
Model	**UUAS**	**Dspr.**	**Root %**	**Nspr.**
1b-1	70.75	78.82	83.92	85.38
1b-2	72.93	79.86	83.53	85.92
1b-3	77.23	82.66	85.13	86.87
100m-1	68.46	76.95	81.21	84.06
100m-2	70.02	78.11	81.25	84.53
100m-3	69.35	78.73	79.88	84.59
10m-1	61.48	73.19	70.88	81.65
10m-2	62.01	73.78	70.07	81.89
10m-3	60.12	72.58	67.14	80.62
1m-1	56.96	71.70	57.12	74.16
1m-2	55.78	71.33	56.56	74.74
1m-3	55.84	71.33	57.41	74.46

Table 2: Structural probing with Hewitt and Manning's syntactic structural probes. '1b-*' corresponds to the family roberta-base-1B, '100M-*' to roberta-base-100M, '10M-* to roberta-10M, and '1M-*' to roberta-med-small-1M.

Tree distance evaluation. The models trained with more data encode better syntactic information (as measured by the probe metrics). While *DSpr.* shows a less pronounced variability between family members, and smaller differences across families, *UUAS* shows a higher intra-family variability and bigger differences between families. Noticeably, for the *roberta-base-1B* family, there is a 7 points difference in *UUAS* between model 1 and model 3, which have a difference of only 0.09 points in perplexity, highlighting the importance of training hyperparameters for the performance of the models.

Tree depth evaluation. As for the distance metrics, the models trained on more data show a better encoding of syntactic information. Again, the correlation shows less variability between family members and smaller differences between families, while *Root %* shows a higher intra-family variability (especially noticeable for *roberta-base-10M*).

4.2 Syntactic generalization evaluation

We assess the syntactic generalization performance of the different MiniBERTas models using Hu et al. (2020)'s test suites (cf. Subsection 3.3) to answer the following questions: Do models pretrained on more data generalize better? Do models with lower perplexity perform better in the syntactic tests? Do models with more pretraining or better perplexity perform better in all circuits?

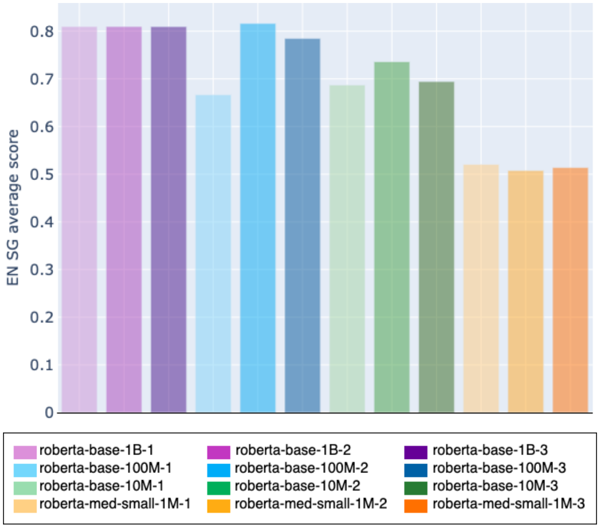

Figure 1: Syntactic generalization evaluation. Average SyntaxGym score.

Average SG Score. Figure 1 shows the performance of each model averaged across all 6 circuits. We observe a variability between family members, especially for *roberta-base-100M*, with a difference of 15 points between models 1 and 2. As intuitively expected, the smallest family of models, *roberta-med-small-1M*, performs clearly worse than the other families. However, it is interesting to observe that more training data does not always imply better syntactic generalization: model *roberta-base-100M-1* performs worse than the whole *roberta-base-10M* family, and model *roberta-base-100M-2* performs better than the whole *roberta-base-1B* family.

Stability with respect to modifiers. Five of the test suites (Center Embedding, Cleft structure, MVRR, NPZ-Verb, NPZ-Object) include tests with and without modifiers, i.e,. intervening content inserted before the critical region. These additional clauses or phrases increase the linear distance between two co-varying items, making the task more difficult, and sometimes they also include a distractor word in the middle of a syntactic dependency,

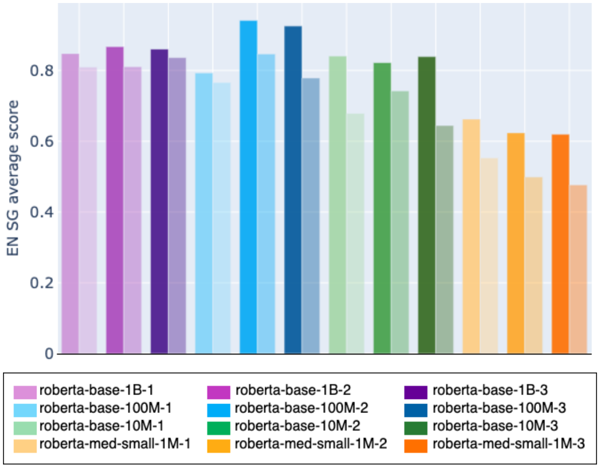

Figure 2: Syntactic generalization evaluation. SyntaxGym score on Center Embedding, Cleft structure, MVRR, NPZ-Verb, and NPZ-Object, without (dark bars) and with (light bars) modifiers.

which can lead the models to misinterpret the dependency. Figure 2 shows the models' average scores on these test suites, without modifiers (dark bars) and with modifiers (light bars), evaluating how robust each model is with respect to the intervening content. We observe that all models are affected by the presence of modifiers, but the difference is narrower for *roberta-base-1b*, which offers the best stability.

Perplexity vs. SG Score. Figure 3 shows the relation between the average score across all circuits (*SG score*) and the perplexity of the models. As previously observed in (Hu et al., 2020), even though there is a (not perfect) negative correlation between the two metrics when comparing different families, when comparing points corresponding to the same family of models (with equal architecture and training data size, points of the same color in Figure 3), there is no clear relation between them. This suggests that both metrics capture different aspects of the knowledge of the models.

Syntactic generalization of the models. Figure 4 offers an overview of the syntactic capabilities of all the models on the different syntactic circuits. The family with more pretraining data, *roberta-base-1B*, outperforms all other families in 3 out of 6 circuits, but offers a surprisingly low performance in Gross Syntactic State, clearly outperformed by *roberta-base-100M* and *roberta-base-10M*, and matched by the *roberta-med-small-1M*. Again, the smallest family offers the lowest performance across all circuits, with individual models outperforming isolated models of other families in Center Embedding, Gross Syntactic State and Long Distance Dependencies. There is a high variability between the scores achieved by the models of the same family in the same circuit, with the exception of *roberta-base-1B* in Licensing, where all models offer a similar performance. Interestingly, there is not a single model for any family that performs best (nor worst) across all tests.

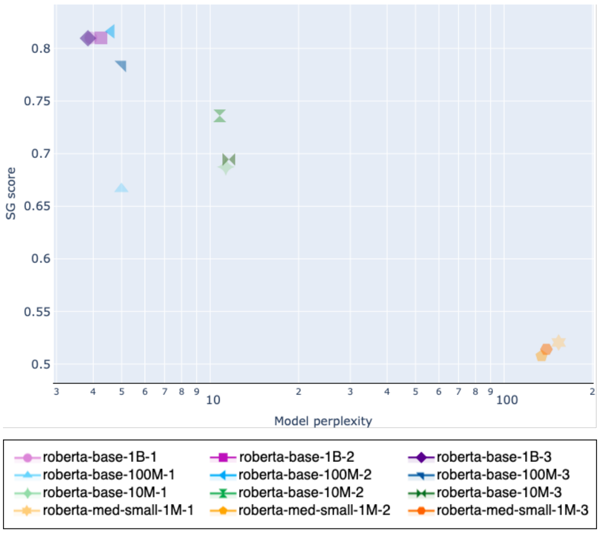

Figure 3: Relationship between average SyntaxGym score and model perplexity.

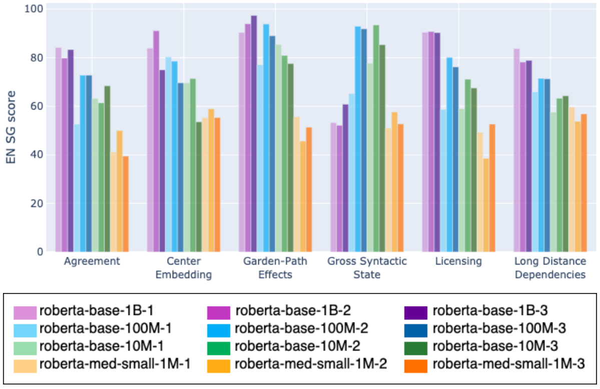

Figure 4: SyntaxGym evaluation across circuits.

4.3 Targeted downstream tasks evaluation

We compare the performance of the different models on three different downstream tasks: PoS tagging (Figure 5), dependency parsing (Figure 6) and paraphrase identification (Figures 7) to determine if models pretrained on more data perform better on downstream applications. We observe the same tendency for all tasks: models with more training data perform better, and the model with the

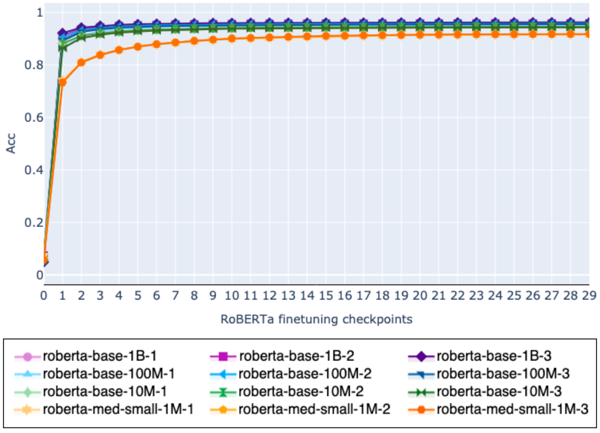

Figure 5: Targeted downstream task evaluation. PoS tagging accuracy evolution.

smaller architecture (*roberta-med-small-1M*) performs remarkably worse. Although note that while the increase of training data between families is exponential (1M, 10M, 100M, 1B), the performance grows at a slower rate. This observation suggests that there may be a limit to the amount of data that we can feed into a RoBERTa model and the knowledge that the model can acquire.

5 Cost-benefit analysis

For the sake of a more holistic view on the quality of the models, we perform a cost–benefit analysis of the performance gains in the different tasks, with an estimate of the financial and environmental cost of developing the models. As the resources used to train the MiniBERTas are not publicly available, we rely on the data provided in (Strubell et al., 2019) to estimate the cost of developing each individual model based on the costs of RoBERTa, trained on 30B words, in proportion to the amount of words used to train each family of models.

Financial cost. As RoBERTa base was trained on 1024 Nvidia V100 GPUs for 24 hours (i.e., 24,576 GPU hours), and the price per hour of Nvidia V100 (on-demand) is $2.48 (Strubell et al., 2019), the cost of training RoBERTa base amounts to $60,948, and the cost of training a MiniBERTas model can be estimated to be $60,948 / 30B words * #TrainingWords. E.g., for the *roberta-base-1b* model: $60,948 / 30B words * 1B words = $2,032.

CO$_2$ Emissions. Using Strubell et al. (2019), we extrapolate that Nvidia V100 GPUs emit 0.28441456 lbs of CO$_2$ per GPU per hour, which means that the training of RoBERTa base emitted

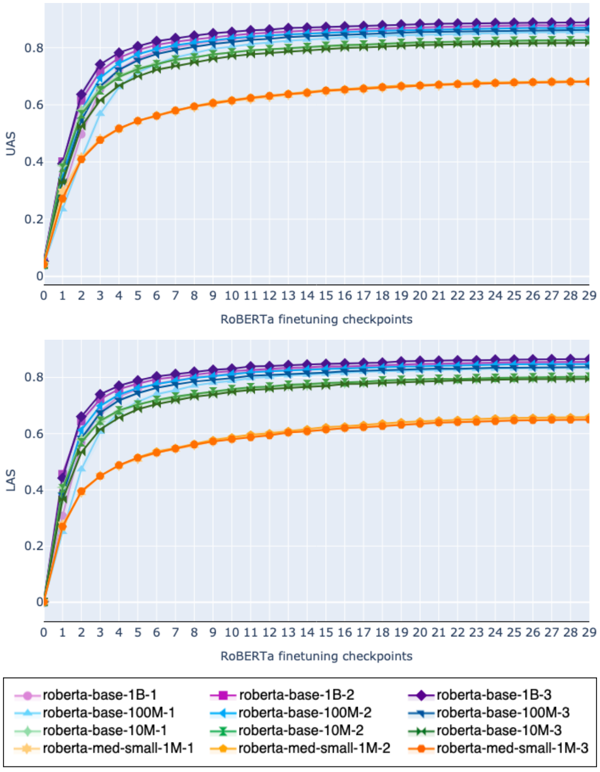

Figure 6: Targeted downstream tasks evaluation. Dependency parsing UAS and LAS evolution.

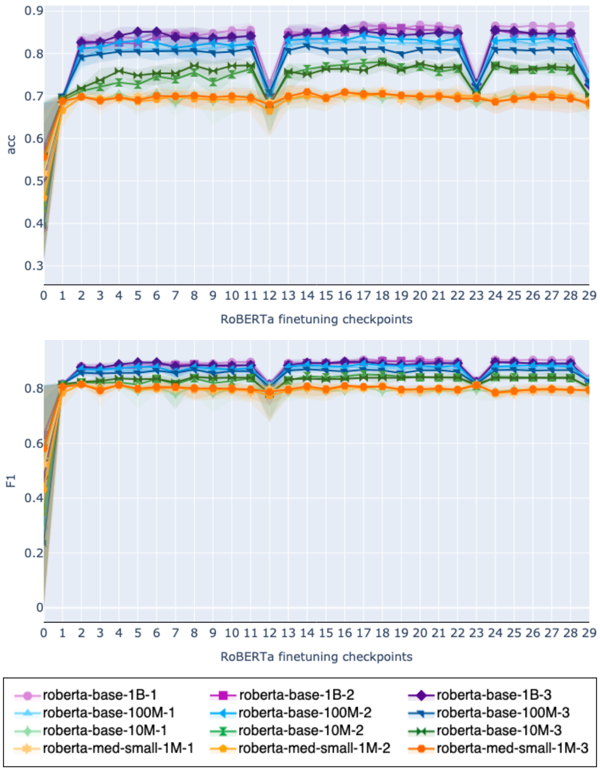

Figure 7: Targeted downstream tasks evaluation. Paraphrase identification accuracy and F1 evolution.

Model family	Cost	CO_2e (lbs)	PoS	Dep. parsing	Paraphrase id.
roberta-base-1B	\$20320	2330	96.03 (+0.5%)	85.73 (+1.76%)	89.59 (+2.02%)
roberta-base-100M	\$5075	582.5	95.53 (+1.11%)	83.97 (+4.04%)	87.57 (+2.79%)
roberta-base-10M	\$500	58.25	94.42 (+2.73%)	79.93 (+14.48%)	84.78 (+5.34%)
rob-med-small-1M	\$50	5.825	91.69 (base)	65.45 (base)	79.44 (base)

Table 3: Comparison of the estimated cost of developing the different MiniBERTas families in terms of cloud compute cost (USD) and CO_2 emissions (lbs) and their averaged performances on PoS tagging (acc), Dep. Parsing (LAS), and Paraphrase identification (F1). In parentheses, we show the increment with respect to the previous smaller model.

6,990 lbs of CO_2. We estimate the emissions of the training of each MiniBERTas model as 6,990 lbs / 30B * #TrainingWords.

To develop each MiniBERTas models, Warstadt et al. run the pretraining 10 times for the bigger family (roberta-base-1B), and 25 times for the other three families (roberta-base-100M, roberta-base-10M and roberta-med-small-1M) with varying hyperparameters. Therefore, to compute the cost of developing each family of models, we multiply the cost of training a single model by the number of pretraining runs needed to obtain it. Table 3 lists the estimated costs and CO_2 emissions of the development of each MiniBERTas family, along with their averaged performance on the three studied downstream applications. We see that small performance gains come at high financial and environmental costs. E.g., for *roberta-base-1B*, a performance increase of 0.5%–2.02% on downstream applications has a cost of \$20K in computing resources and significant carbon emissions, higher than the estimated 1984 lbs generated by a single passenger flying between New York and San Francisco (Strubell et al., 2019).

6 Discussion and conclusions

Our experiments shed light on the impact of pretraining data size on the syntactic capabilities of RoBERTa. Our results indicate that models pretrained with more data encode better syntactic information (as measured by Hewitt and Manning's structural probes) and offer a higher syntactic generalization over the different syntactic phenomena covered by the tests assembled in (Hu et al., 2020). Moreover, models pretrained with more data seem to be more robust to the presence of modifiers in the syntactic tests, i.e,. intervening content inserted before the critical region. As was already observed in (Hu et al., 2020), there is no simple relationship between the perplexity of the models and the SyntaxGym score: the variance in intra-family SG score is not explained by the perplexity differences. When zooming in on the different test circuits, probing different linguistic phenomena, we observe that there is a high variability between the scores achieved by the models of the same family, with no single model for any family performing best across all tests. While the family pretrained with more data outperforms all the models of the other families on 3 out of 6 circuits, it offers a surprisingly low performance in Gross Syntactic State, clearly outperformed by the smaller models.

We also compare the performance of the different models fine-tuned on PoS tagging, dependency parsing and paraphrase identification, observing that models with more training data offer a better performance, and the model with the smaller architecture (roberta-med-small-1M) performs remarkably worse. However, while the amount of training data between families grows exponentially, we observe that the performance grows at a much slower rate, suggesting that there may be a limit to the knowledge that a RoBERTa model can acquire solely from raw pretraining data.

We complement our findings with a financial and environmental cost–benefit analysis of pretraining models on different amounts of data. We show that while models pretrained on more data encode more syntactic information and perform generally better on downstream applications, small performance gains come at a huge financial and environmental cost. Thus, when developing and training new models we should weigh between the benefit of making models bigger and pretraining them on huge datasets and the costs this implies, prioritizing computationally efficient hardware and algorithms.

A question that still needs to be addressed by future work is whether it is possible to complement information-theoretical metrics such as perplexity with metrics measuring specific types of knowledge, e.g., syntax, in order to develop and select more robust and efficient models to solve Natural Language Understanding tasks.

Acknowledgments

This work has been partially funded by the European Commission via its H2020 Research Program under the contract numbers 786731, 825079, 870930, and 952133. This work has been partially supported by the ICT PhD program of Universitat Pompeu Fabra through a travel grant.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Kasturi Bhattacharjee, Miguel Ballesteros, Rishita Anubhai, Smaranda Muresan, Jie Ma, Faisal Ladhak, and Yaser Al-Onaizan. 2020. To BERT or not to BERT: Comparing task-specific and task-agnostic semi-supervised approaches for sequence tagging. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7927–7934, Online. Association for Computational Linguistics.

Ethan A. Chi, John Hewitt, and Christopher D. Manning. 2020. Finding universal grammatical relations in multilingual BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5564–5577, Online. Association for Computational Linguistics.

Alexis Conneau, German 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, Melbourne, Australia. Association for Computational Linguistics.

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

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Timothy Dozat and Christopher D. Manning. 2016. Deep biaffine attention for neural dependency parsing.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. 2020. When bert forgets how to pos: Amnesic probing of linguistic properties and mlm predictions.

Richard Futrell, Ethan Wilcox, Takashi Morita, and Roger Levy. 2018. Rnns as psycholinguistic subjects: Syntactic state and grammatical dependency.

Richard Futrell, Ethan Wilcox, Takashi Morita, Peng Qian, Miguel Ballesteros, and Roger Levy. 2019. Neural language models as psycholinguistic subjects: Representations of syntactic state. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 32–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. AllenNLP: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Jon Gauthier, Jennifer Hu, Ethan Wilcox, Peng Qian, and Roger Levy. 2020. SyntaxGym: An online platform for targeted evaluation of language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 70–76, Online. 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.

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

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

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

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger Levy. 2020. A systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1725–1744, Online. Association for Computational Linguistics.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019a. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Vincent Micheli, Martin d'Hoffschmidt, and François Fleuret. 2020. On the importance of pre-training data volume for compact language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7853–7858, Online. Association for Computational Linguistics.

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, New Orleans, Louisiana. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Subendhu Rongali, Luca Soldaini, Emilio Monti, and Wael Hamza. 2020. Don't parse, generate! a sequence to sequence architecture for task-oriented semantic parsing. *Proceedings of The Web Conference 2020*.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green ai. *arXiv preprint arXiv:1907.10597*.

Natalia Silveira, Timothy Dozat, Marie-Catherine de Marneffe, Samuel Bowman, Miriam Connor, John Bauer, and Chris Manning. 2014. A gold standard dependency corpus for English. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2897–2904, Reykjavik, Iceland. European Language Resources Association (ELRA).

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. 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*.

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

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Alex Wang and Kyunghyun Cho. 2019. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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

Alex Warstadt, Yian Zhang, Xiaocheng Li, Haokun Liu, and Samuel R. Bowman. 2020b. Learning which features matter: RoBERTa acquires a preference for linguistic generalizations (eventually). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 217–235, Online. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, and Richard Futrell. 2019a. Hierarchical representation in neural language models: Suppression and recovery of expectations. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 181–190, Florence, Italy. Association for Computational Linguistics.

Ethan Wilcox, Roger Levy, Takashi Morita, and Richard Futrell. 2018. What do RNN language models learn about filler–gap dependencies? In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 211–221, Brussels, Belgium. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019b. Structural supervision improves learning of non-local grammatical dependencies. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3302–3312, Minneapolis, Minnesota. Association for Computational Linguistics.

Ethan Wilcox, Peng Qian, Richard Futrell, Miguel Ballesteros, and Roger Levy. 2019c. Structural supervision improves learning of non-local grammatical dependencies.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yian Zhang, Alex Warstadt, Haau-Sing Li, and Samuel R Bowman. 2020. When do you need billions of words of pretraining data? *arXiv preprint arXiv:2011.04946*.

Qile Zhu, Haidar Khan, Saleh Soltan, Stephen Rawls, and Wael Hamza. 2020. Don't parse, insert: Multilingual semantic parsing with insertion based decoding. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 496–506, Online. Association for Computational Linguistics.