

# To Pretrain or Not to Pretrain: Examining the Benefits of Pretraining on Resource Rich Tasks

Sinong Wang, Madian Khabsa, Hao Ma

Facebook AI, Seattle, USA

{sinongwang, mkhabasa, haom}@fb.com

## Abstract

Pretraining NLP models with variants of Masked Language Model (MLM) objectives has recently led to a significant improvements on many tasks. This paper examines the benefits of pretrained models as a function of the number of training samples used in the downstream task. On several text classification tasks, we show that as the number of training examples grow into the millions, the accuracy gap between finetuning BERT-based model and training vanilla LSTM from scratch narrows to within 1%. Our findings indicate that MLM-based models might reach a diminishing return point as the supervised data size increases significantly.

## 1 Introduction

Language modeling has emerged as an effective pretraining approach in wide variety of NLP models. Multiple techniques have been proposed, including bi-directional language modeling (Peters et al., 2018), masked language models (Devlin et al., 2018), and variants of denoising auto-encoder approaches (Lewis et al., 2019; Raffel et al., 2019; Joshi et al., 2019). Today, it is rare to examine a leaderboard without finding the top spots occupied by some variant of a pretraining method.<sup>1</sup>

The future of NLP appears to be paved by pretraining a universal contextual representation on wikipedia-like data at massive scale. Attempts along this path have pushed the frontier to up 10× to the size of wikipedia (Raffel et al., 2019). However, the success of these experiments is mixed: although improvements have been observed, the downstream task is usually data-limited. There is evidence that large-scale pretraining does not always lead to state-of-the-art results (Raffel et al., 2019), especially on tasks such as machine translation, where abundance of training data, and the

existence of strong augmentation methods such as back translation might have limited the benefit of pretraining.

This paper examines the pretraining benefits of downstream tasks as the number of training samples increases. To answer this question, we focus on multi-class text classification since: (i) it is one of most important problems in NLP with applications spanning multiple domains. (ii) large sums of training data exists for many text classification tasks, or can be obtained relatively cheaply through crowd workers (Snow et al., 2008). We choose three sentiment classification datasets: Yelp review (yel, 2019), Amazon sports and electronics review (Ni et al., 2019), ranging in size from 6 to 18 million examples.<sup>2</sup>

We finetune a RoBERTa model (Liu et al., 2019) with increments of the downstream dataset, and evaluate the performance at each increment. For example, on the Yelp dataset whose size is 6 million, we train the models on subsets of the data with each subset size being in the sequence (60k, 600K, 1.8M, 3M ..., 6M). For comparison, we also train a vanilla BiLSTM, and another BiLSTM which uses pretrained Roberta token embeddings. We observe that when both models are trained on 1% of the data, the gap between BiLSTM and RoBERTa models is at its peak, but as the training dataset size increases, the BiLSTM model accuracy keeps on increasing whereas RoBERTa’s accuracy remain mostly flat. As the dataset size increases, the accuracy gap shrinks to within 1%.

Our study suggests that collecting data and training on the target tasks is a solution worth considering, especially in production environments where accuracy is not the only considered factor, rather inference latency is often just as crucial. We benchmarked the inference latency of the these models on

<sup>2</sup>These datasets are the largest publicly available classification datasets that we are aware of.

<sup>1</sup><https://super.gluebenchmark.com/leaderboard>

both CPU and GPU for different batch sizes, and as expected, we observe at least  $20\times$  speedup for the BiLSTM compared to the RoBERTa. This paper provides new experimental evidence and discussions for people to rethink the MLM pre-training paradigm in NLP, at least for resource rich tasks.

## 2 Related Works

Scaling the number of training examples has long been identified as source of improvement for machine learning models in multiple domains including NLP (Banko and Brill, 2001), computer vision (Deng et al., 2009; Sun et al., 2017) and speech (Amodei et al., 2016). Previous work has suggested that deep learning scaling may be predictable empirically (Hestness et al., 2017), with model size scaling sub-linearly with training data size. (Sun et al., 2017) concluded that accuracy increases logarithmically with respect to training data size. However, these studies have focused on training models in the fully supervised setting, without pretraining.

One closer work is (He et al., 2019) where it is shown that randomly initialized standard computer-vision models perform no worse than their *ImageNet* pretrained counterparts. However, our work focuses on text classification. We do not examine the benefit of pretraining, at large, rather we focus on the benefit of pretraining for resource rich tasks. Another concurrent work that is still under review, in (Nakkiran and Sutskever, 2020) observes that, in some translation task such as IWSLT14, small language models exhibit even lower test loss compared to the large transformer model when the number of training samples increases.

## 3 Experiments

### 3.1 Task and Data

We focus on a multi-class sentiment classification task: given the user reviews, predict the rating in five points scale  $\{1, 2, 3, 4, 5\}$ . The experiments are conducted on the following three benchmark datasets.

- **Yelp Challenge** (yel, 2019) contains text reviews, tips, business and check-in sets in Yelp. We use the 6.7m user reviews with ratings as our dataset.
- **Amazon Reviews** (Ni et al., 2019) contains product reviews (ratings, text, helpfulness

votes) from Amazon. We choose two categories: *sports / outdoors*, and *electronics* as two separate datasets. We only use the review text as input features.

The distribution across five ratings of each dataset is illustrated in Table 1. In our experiment, all the above data is split into 90% for training and 10% for testing.

Dataset	Size	1	2	3	4	5
Yelp	6.69M	15%	8%	11%	22%	44%
Sports	11.9M	7%	5%	7%	16%	65%
Electronics	18.6M	11%	5%	7%	16%	61%

Table 1: Data size and percentage of samples in each (n)-star category

### 3.2 Models

We choose the following three types of pretrained and vanilla models:

- **RoBERTa** (Liu et al., 2019) RoBERTa is a transformer-based model pretrained with masked language modeling objectives on a large corpus. We finetune our classification task on both Roberta-Base (12 layers, 768 hidden, 12 heads) and Roberta-Large (24 layers, 1024 hidden, 16 heads).
- **LSTM** (Hochreiter and Schmidhuber, 1997) We use a bidirectional LSTM with a max-pooling layer on top of the hidden states, followed by a linear layer. Token embeddings of size 128 are randomly initialized.
- **LSTM + Pretrained Token Embedding** Similar to the previous setup, except we initialized the token embeddings with Roberta pretrained token embedding (Base: 768-dimensional embedding, Large: 1024-dimensional embedding). The embeddings are frozen during training.

For fair comparison, all the above models share the same vocabulary and BPE tokenizer (Sennrich et al., 2015).

### 3.3 Experimental Setup

We use the Adam optimizer and the following hyperparameter sweep for each model. (i) RoBERTa is finetuned with the following learning rates  $\{5e-6, 1e5, 1.5e-5, 2e-5\}$ , with linear warm up in the first 5% of steps followed by a linear

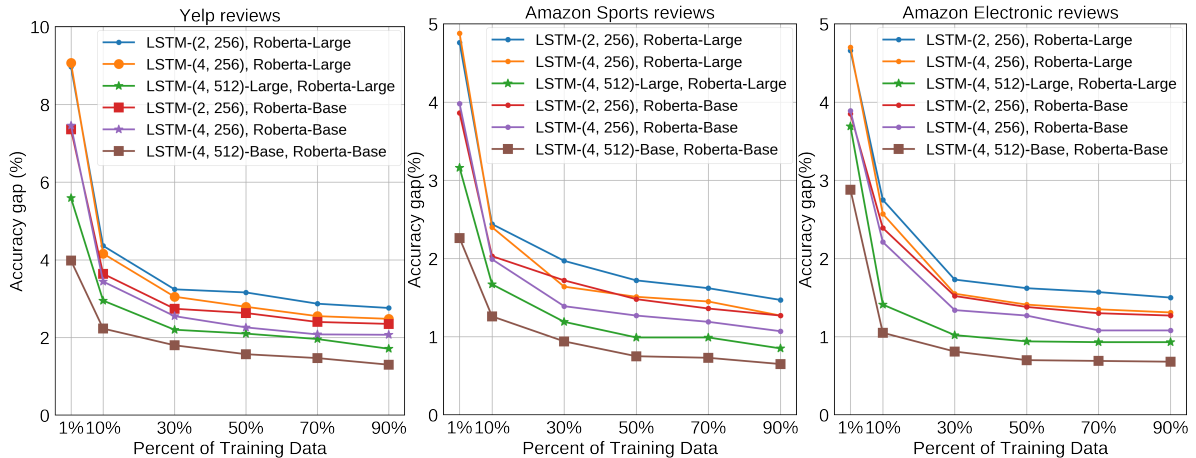


Figure 1: Accuracy Gap of Roberta, BiLSTM trained on different amount of data

Models	Yelp		Sports		Electronics		Params
	Accuracy	$\Delta$	Accuracy	$\Delta$	Accuracy	$\Delta$	
Roberta-Large	78.85	-	79.65	-	79.07	-	304M
Roberta-Base	78.44	0.41	79.45	0.20	78.84	0.23	86M
LSTM-4-512 + Large	77.14	1.71	78.80	0.85	78.16	0.92	25M
LSTM-4-512 + Base	77.07	1.78	78.72	0.93	78.07	1.0	24M
LSTM-4-256 + Large	77.02	1.83	78.76	0.89	78.12	0.95	7.4M
LSTM-4-256 + Base	77.03	1.82	78.62	1.03	77.98	1.09	6.8M
LSTM-4-256	76.37	2.48	78.38	1.27	77.76	1.31	4.8M
LSTM-2-256	76.09	2.76	78.18	1.47	77.57	1.5	2.4M

Table 2: Test Accuracy of Roberta-base, BiLSTM, and BiLSTM with Roberta Pretrained Token Embedding when trained on the full dataset. The  $\Delta$  column shows the difference between each model’s accuracy and that of Roberta-Large. For LSTM models, LSTM-n-k denotes an LSTM model with n layers and k cells. + Large or + Base indicate the use of Roberta Large or Roberta Base token embeddings, respectively. The number of parameters does not count the size of embedding table.

decay to 0. The batch size is set to 32, with dropout being 0.1. (ii) For the LSTM, it is trained with a constant learning rate from the sequence:  $\{2.5e - 4, 5e - 4, 7.5e - 4, 1e - 3\}$ . The batch size is set to 64. We train each model on 8 GPUs for 10 epochs and perform early stopping based on accuracy on the test set. The maximum sequence length of input was set to 512 for all models.

## 4 Results

### 4.1 Impact of Data Size

We first investigate the effect of varying the number of training samples, for fixed model and training procedure. We train different models using  $\{1\%, 10\%, 30\%, 50\%, 70\%, 90\%\}$  amount of data to mimic the “low-resource”, “medium-resource” and “high-resource” regime. Figure 1 shows that the accuracy delta between the LSTM and RoBERTa models at different percentages of the

training data. From the plot, we observe the following phenomena:

(i) Pretrained models exhibit a diminishing return behavior as the size of the target data grows. When we increase the number of training examples, the accuracy gap between Roberta and LSTM shrinks. For example, when both models are trained with 1% of the Yelp dataset, the accuracy gap is around 9%. However, as we increases the amount of training data to 90%, the accuracy gap drops to within 2%. The same behaviour is observed on both Amazon review datasets, with the initial gap starting at almost 5% for 1% of the training data, then shrinking all the way to within one point when most of the training data is used.

(ii) Using the pretrained RoBERTa token embeddings can further reduce the accuracy gap especially when training data is limited. For example, in the Yelp review data, a 4-layers LSTM with pretrained embeddings provides additional 3 per-

cent gain compared to its counterparts. As Table 2 shows, an LSTM with pretrained RoBERTa token embeddings always outperforms the ones with random token initialization. This suggests that the embeddings learned during pretraining RoBERTa may constitute an efficient approach for transfer learning the knowledge learned in these large MLM.

We further report the accuracy metric of each model using all the training data. The full results are listed in Table 2. We observe that the accuracy gap is less than 1% on the Amazon datasets. even compared to 24 layers RoBERTa-large model. As for the Yelp dataset, the accuracy gap is within 2 percent from the RoBERTa-large model, despite an order of magnitude difference in the number of parameters.

## 4.2 Inference Time

We also investigate the inference time of the three type of models on GPU and CPU. The CPU inference time is tested on Intel Xeon E5-2698 v4 with batch size 128. The GPU inference time is tested on NVIDIA Quadro P100 with batch size  $\in \{128, 256, 384\}$ . The maximum sequence length is 512. We run 30 times for each settings and take the average. The results are listed in TABLE 3.

Model	CPU	GPU		
		128	256	384
Roberta-Base	323	16.1	16.1	16.1
Roberta-Large	950	55.5	55.5	-
LSTM-2-256	15.2	0.47	0.43	0.42
LSTM-4-256	28.1	1.17	0.94	0.86
LSTM-4-256+Base	35.2	1.33	1.09	1.02
LSTM-4-256+Large	37.5	1.33	1.17	1.07
LSTM-4-512+Base	64.8	3.52	3.20	3.13
LSTM-4-512+Large	64.8	3.36	3.32	3.26

Table 3: Inference time (ms) of Roberta, BiLSTM on CPU and GPU

Not surprisingly, the LSTM model is at least 20 time faster even when compared to the Roberta-Base. Note that the P100 will be out of memory when batch size is 384 for Roberta-Large. Another observation is that although using the Roberta pretrained token embedding introduces 10 times more model parameters compared to vanilla BiLSTM, the inference time only increases by less than 25%. This is due to the most additional parameters are from a simple linear transformation.

## 5 Discussion

Our findings in this paper indicate that increasing the number of training examples for ‘standard’ models such as LSTM leads to performance gains that are within 1 percent of their massively pretrained counterparts. Due to the fact that there is no good large scale question answering dataset, it is not clear if the same findings would hold on this type of NLP tasks, which are more challenging and semantic-based. In the future work, we will run more experiments if there are some other large scale open datasets. Despite sentiment analysis being a crucial text classification task, it is possible, though unlikely, that the patterns observed here are limited to sentiment analysis tasks only. The rationale behinds that is that pretrained LSTMs have kept up very well with transformer-based counterparts on many tasks (Radford et al.).

One way to interpret our results is that ‘simple’ models have better regularization effect when trained on large amount of data, as also evidenced in the concurrent work (Nakkiran and Sutskever, 2020). The other side of the argument in interpreting our results is that MLM based pretraining still leads to improvements even as the data size scales into the millions. In fact, with a pretrained model and 2 million training examples, it is possible to outperform an LSTM model that is trained with  $3\times$  more examples.

## 6 Conclusion

Finetuning BERT-style models on resource-rich downstream tasks is not well studied. In this paper, we reported that, when the downstream task has sufficiently large amount of training examples, i.e., millions, competitive accuracy results can be achieved by training a simple LSTM, at least for text classification tasks. We further discover that reusing the token embeddings learned during BERT pretraining in an LSTM model leads to significant improvements. The findings of this work have significant implications on both the practical aspect as well as the research on pretraining. For industrial applications where there is a trade-off typically between accuracy and latency, our findings suggest it might be feasible to gain accuracy for faster models by collecting more training examples.

## References

2019. Yelp dataset challenge. <https://www.yelp.com/dataset/challenge>.
- Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182.
- Michele Banko and Eric Brill. 2001. Scaling to very very large corpora for natural language disambiguation. In *Proceedings of the 39th annual meeting on association for computational linguistics*, pages 26–33. Association for Computational Linguistics.
- Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4918–4927.
- Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. 2017. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.
- Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2019. Spanbert: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Kaplan G. Bansal Y. Yang T. Barak B. Nakkiran, P. and I. Sutskever. 2020. Deep double descent: Where bigger models and more data hurt. *ICLR 2020*.
- Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197.
- Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Rion Snow, Brendan O’Connor, Daniel Jurafsky, and Andrew Y Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *Proceedings of the conference on empirical methods in natural language processing*, pages 254–263. Association for Computational Linguistics.
- Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852.