What Would Elsa Do? Freezing Layers During Transformer Fine-Tuning

Jaejun Lee, Raphael Tang, Peng Shi, Ji Xin, and **Jimmy Lin**
David R. Cheriton School of Computer Science
University of Waterloo

Abstract

Pretrained transformer-based language models have achieved state of the art across many tasks in natural language processing. These models are highly expressive, comprising at least a hundred million parameters and a dozen layers. Recent evidence suggests that only a few of the final layers need to be fine-tuned for high quality on downstream tasks. In this work, we analyze this behavior comprehensively across multiple tasks and different languages. For the multi-task experiments, we use MT-DNN trained on RoBERTa; for the crosslingual ones, we evaluate XLM-R. We show that only a fourth of the final layers need to be fine-tuned, on average, to achieve at least 95% of the original model quality. Surprisingly, we also find that fine-tuning all layers does *not* always help.

1 Introduction

Transformer-based pretrained language models are a battle-tested solution to a plethora of natural language processing tasks. XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019c) are two well-known indications, representing the current state of the art in natural language inference, question answering, and sentiment classification, to list a few. The same idea has been transferred to crosslingual tasks, leading to the advent of multilingual pretrained models such as mBERT (Devlin et al., 2019) and XLM-R (Conneau et al., 2019).

However, despite their rich language representation, the extreme size of the models often hinders the use of transformer-based language models in practice; these models consist of at least a hundred million parameters, a hundred attention heads, and a dozen layers. An emerging line of work questions the need for such a parameter-loaded model, especially on a single downstream task. Michel et al. (2019), for example, note that only a few attention heads need to be retained in each layer for acceptable effectiveness. Kovaleva et al. (2019) find that, on many tasks, just the last few layers change the most after the fine-tuning process. We take these observations as evidence that only the last few layers necessarily need to be fine-tuned.

The central objective of our paper is, then, to determine how many of the last layers actually need fine-tuning. Why is this important? Pragmatically, a reasonable cutoff point saves computational memory across fine-tuning multiple tasks, which bolsters the effectiveness of existing parameter-saving methods (Houlsby et al., 2019). Pedagogically, understanding the relationship between the number of fine-tuned layers and the resulting model quality may guide future work in pretrained modeling.

In this work, we provide a thorough evaluation, across multiple pretrained transformers, datasets, and languages, of the number of final layers needed for fine-tuning. From our study, we find that only one fourth of the final layers necessarily need to be fine-tuned to reach 95% of the original quality on average, regardless of the underlying language and target task type; we empirically show that 72% of the overall parameters can be saved across eight tasks in one of our models. Furthermore, we find that on SST-2, a sentiment classification dataset, fine-tuning all of the layers does not always lead to improved quality.

2 Background and Related Work

2.1 Pretrained Language Models

In the pretrained language modeling paradigm, a language model (LM) is trained on vast amounts of text, then fine-tuned on a specific downstream task. The most popular LM is BERT proposed by Devlin et al. (2019), deep 12- and 24-layer bidirectional transformers (Vaswani et al., 2017) pre-trained on the entirety of Wikipedia and BooksCorpus (Zhu

et al., 2015). With BERT, authors achieve state of the art across all tasks in the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018), as well as the Stanford Question Answering Dataset (Rajpurkar et al., 2016).

As a result of this development, a flurry of recent papers has followed this more-data-plus-better-models principle. Two prominent examples include XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019c), both of which contest the present state of the art. XLNet proposes to pretrain two-stream attention-augmented transformers on an autoregressive LM objective, instead of the original cloze and next sentence prediction (NSP) task from BERT. RoBERTa primarily argues for pretraining longer, using more data, and removing the NSP task.

Layerwise interpretability. The prevailing evidence in the literature suggests that earlier layers extract universal features, while later ones perform task-specific modeling (Zeiler and Fergus, 2014). In the NLP literature, similar observations have been made for pretrained language models. Clark et al. (2019a) analyze BERT's attention and observe that the bottom layers attend broadly, while the top layers capture linguistic syntax. Kovaleva et al. (2019) find that the last few layers of BERT change the most after task-specific fine-tuning.

2.2 Representation Learning

Across tasks. Realizing that the bottom layers change little from fine-tuning, Liu et al. (2019b) claim that multi-task learning (MTL) and LM pretraining are complementary, hypothesizing that conflating the two paradigms results in better language representation models. They present MT-DNN, a BERT model fine-tuned for multiple tasks using task-specific output layers. In support of their hypothesis, MT-DNN captures richer language representations and shows superior zero-shot performance for a wide range of natural language processing (NLP) tasks. Clark et al. (2019b) make similar claims, fine-tuning BERT on the outputs of multiple task-specific models.

Across languages. Along with an English BERT model, Devlin et al. (2019) release a multilingual model variation, called mBERT, pretrained on monolingual corpora of 104 different languages. In a subsequent study, Pires et al. (2019) find that mBERT capably captures crosslingual language representations, which leads to better zero-shot quality for many languages. Building upon mBERT, Lample and Conneau (2019) suggest pretraining on the cloze task using pairs of sentences in different languages but expressing the same semantics. Their model, called XLM, better captures interlingual relationships and displays superior quality in crosslingual tasks. Recently, Conneau et al. (2019) introduce XLM-R, which is trained under the same objectives as XLM but on richer data.

	BASE (12 layers)		LARGE (24 layers)	
	#	%	#	%
Embedding	39.0M	31.3%	52.0M	14.6%
Per-Layer	7.1M	5.7%	12.6M	3.5%
Output	0.6M	0.5%	1.0M	0.3%
Total	124.6M	100.0%	355.4M	100.0%

Table 1: Parameter statistics for the base and large variants of MT-RoBERTa and XLM-R. Note that "per-layer" indicates the number of parameters in one intermediate layer, which is more relevant to our study.

3 Experimental Setup

In this work, we provide an analysis on the behavior of the model when only a subset of the model is fine-tuned. For each model, we freeze the embeddings and the weights of the first N layers, then fine-tune the rest using the best hyperparameters of the full model. Specifically, if L is the number of layers, we explore $N = \frac{L}{2}, \frac{L}{2}+1, \ldots, L$. Due to computational limitations, we set half as the cutoff point. We also report baseline scores where every layer including the embeddings is fine-tuned.

In this work, we first look at relative model quality, defined as the frozen model scores divided by the corresponding baseline. We also report the savings in parameters: across all the target tasks, we report the actual number of parameters saved in memory, divided by the total number of would-be parameters if each task were given a full model.

Previously, Houlsby et al. (2019) fine-tune the top layers of BERT to evaluate their model compression technique. However, none of the studies thoroughly study the number of necessary final layers across multiple tasks and different languages.

We use the PyTorch Transformers library (v2.1.1; Wolf et al., 2019) to construct our experiments and repeat the training for five times on each configuration. We run the models on NVIDIA Tesla V100 GPUs with CUDA v10.1.

3.1 Fine-Tuning for Different Tasks

First, we evaluate LMs fine-tuned for different tasks. We conduct the experiments with vanilla

Frozen layers	Param. saved	Rel. perf.	CoLA	SST-2	MRPC	STS-B	QQP	MNLI(-mm)	QNLI	RTE
			MCC	Acc.	F_1	ρ	F_1	Acc.	Acc.	Acc.
12/12	0.87	0.60	0.00	80.28	81.22	20.00	62.51	52.60 (53.02)	65.74	57.40
9/12	0.72	0.95	54.76	93.46	88.49	86.99	87.08	84.68 (85.11)	90.77	66.86
Baseline	0.00	1.00	59.85	94.63	92.79	90.76	88.83	87.41 (86.99)	92.75	78.16

Table 2: Development set results of MT-RoBERTa$_{base}$, with all layers plus embeddings frozen (row 1), frozen up to 95% base performance on average (row 2), and all layers plus embeddings fine-tuned (row 3).

Frozen layers	Param. saved	Rel. perf.	CoLA	SST-2	MRPC	STS-B	RTE
			MCC	Acc.	F_1	ρ	Acc.
24/24	0.80	0.48	0.00	79.33	81.22	11.19	48.30
17/24	0.60	0.96	61.82	95.07	91.38	89.58	77.26
Baseline	0.00	1.00	66.04	94.95	93.12	92.01	86.28
MT-BERT$_{KD}$			64.5	94.3	93.3	91.0	88.6

Table 3: Development set results of MT-RoBERTa$_{large}$, with all layers plus embeddings frozen (row 1), frozen up to 95% base performance in average (row 2), and all layers plus embeddings fine-tuned (row 3). We also report scores from MT-BERT$_{KD}$, the current state-of-the-art multi-task LM.

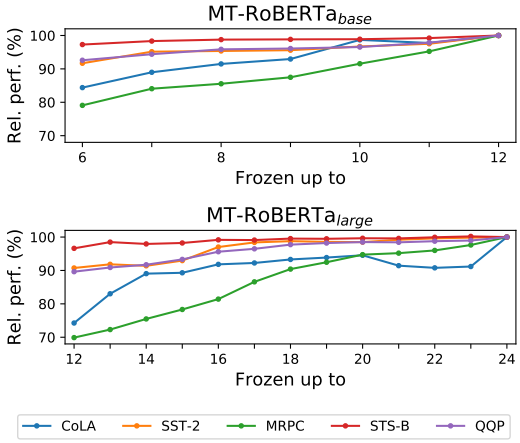

Figure 1: Relative performance of MT-RoBERTa across different degree of freezing collected for CoLA, SST-2, MRPC, STS-B, and QQP.

BERT and RoBERTa, along with their MT-DNN variants. Each LM is fine-tuned on eight tasks of the GLUE benchmark (Wang et al., 2018), which comprises tasks in natural language inference, sentiment classification, linguistic acceptability, and semantic similarity: CoLA, SST-2, MRPC, STS-B, QQP, MNLI, QNLI, and RTE.[1] We restrict our comprehensive all-datasets exploration to the base variant, since the large model is much more computationally intensive. On the smaller CoLA, SST-2, MRPC, STS-B and RTE datasets, we comprehensively evaluate both models. These choices do not substantially affect our analysis.

To distinguish our new RoBERTa-based MT-DNN from BERT-based MT-DNN of Liu et al. (2019b), we call them MT-RoBERTa and MT-BERT, respectively. Due to the limited space, we relegate experimental details to the appendix and focus on MT-RoBERTa in the following section, which achieves the best results on most tasks.

3.2 Fine-Tuning for Different Languages

Next, we fine-tune XLM-R (Conneau et al., 2019), a crosslingual language model that achieves state of the art on a wide range of multilingual tasks, such as named entity recognition (NER) and natural langauge inference. In our experiments, we fine-tune the model for NER and part-of-speech (POS) tagging in different languages. We focus on the NER task in the following section and discuss POS tagging in the appendix.

NER is the task of assigning correct categories to entities in a given sentence. We have selected the CoNLL 2002 and 2003 datasets (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003), which consist of four Romance and Germanic languages—English (EN), Spanish (ES), Dutch (NL), German (DE)—with the standard, self-explanatory four entity types of PERSON, LOCATION, ORGANIZATION and MISC.

4 Experimental Results

4.1 Across Tasks

First, we note that large variant of MT-RoBERTa achieves higher scores on CoLA, SST-2, and STS-B than the current state of the art, BERT fine-tuned with multi-task knowledge distillation (Liu et al., 2019a) (see the last row of Table 3).

When every component except the last task-specific layer is frozen, the fine-tuned base model achieves only 60% of the original quality, on average. The quality of the large model is worse,

[1] WNLI is excluded due to known dev set issues.

Frozen layers	Param. saved	Rel. F_1	EN	ES	NL	DE
			F_1	F_1	F_1	F_1
12/12	0.75	0.07	0.07	0.06	0.06	0.06
8/12	0.58	0.96	0.87	0.85	0.86	0.79
Baseline	0.00	1.00	0.91	0.87	0.90	0.82

Table 4: Test set results of XLM-R$_{base}$, with all layers plus embeddings frozen (row1), frozen up to 95% base performance on average (row 2), and all layers plus embeddings fine-tuned (row 3).

Frozen layers	Param. saved	Rel. F_1	DE	EN	ES	NL
			F_1	F_1	F_1	F_1
24/24	0.75	0.06	0.04	0.08	0.06	0.03
19/24	0.61	0.95	0.81	0.88	0.84	0.88
Baseline	0.00	1.00	0.85	0.92	0.89	0.93

Table 5: Test set results of XLM-R$_{large}$, with all layers plus embeddings frozen (row1), frozen up to 95% base performance on average (row 2), and all layers plus embeddings fine-tuned (row 3).

attaining only 48% of the original quality—see rows 1 and 3 in Tables 2 and 3.

As more layers are fine-tuned, the model effectiveness often improves drastically (see Figure 1). This demonstrates that gains decompose nonadditively with respect to the number of frozen initial layers. Fine-tuning subsequent layers shows diminishing returns, with every model rapidly approaching the baseline quality at fine-tuning half of the network; the base models, for example, need fine-tuning of only 3 layers out of the 12 to reach 95% of the original quality on average (see Table 2). Similarly, fine-tuning only 7 layers out of 24 is sufficient for the large models (see Table 3).

Finally, for MT-RoBERTa$_{large}$ fine-tuned on SST-2, we observe a surprisingly consistent increase in quality when the first 17 layers are frozen. This finding suggests that these models may be overparameterized for SST-2.

4.2 Across Languages

In the case of crosslingual NER, fine-tuning only the task-specific output layer fails to effectively model the task, where, regardless of the underlying language, the F_1 score falls below 0.1. On the other hand, when all layers are fine-tuned, it exceeds 0.8. Fortunately, as in the multi-task case, the scores improve drastically as more layers are fine-tuned and show diminishing returns, asymptoting at the base scores—see Tables 4 and 5 and Figure 2.

For the base model, we find that the last four

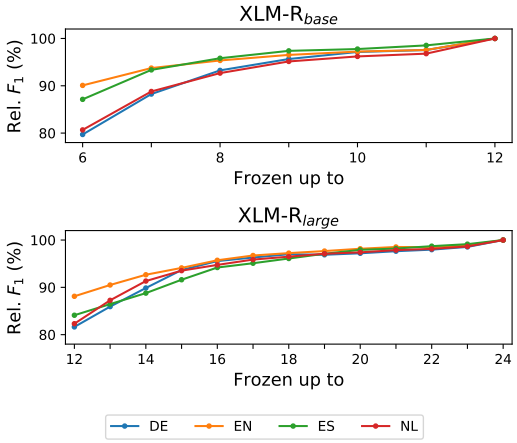

Figure 2: Relative F_1 of XLM-R across different degree of freezing on four languages.

layers are sufficient for achieving 96% of the original quality on average. Similiarly, the large model attains 95% of the baseline quality with only five of the final layers fine-tuned.

4.3 Overall Parameter Savings

When transformer-based LMs are fine-tuned for a target task, regardless of the underlying language, we demonstrate that about one fourth of the final layers necessarily need to be fine-tuned. In other words, ignoring the parameters of the output layer, we need to keep only one copy of the pretrained model and one fourth of the parameters for layers fine-tuned for each task. As the number of tasks increases, the parameter savings converge to the percentage of frozen layers. In fact, for XLM-R and MT-RoBERTa$_{large}$, we achieve about 60% savings in parameters with 4 and 5 tasks, respectively. On the other hand, MT-RoBERTa$_{base}$ achieves 72% savings with 8 tasks, which is closer to 75%, the percentage of frozen layers. These findings further suggest that the fine-tuning time can also be reduced, since we need to backpropagate through the nonfrozen layers only.

5 Conclusions and Future Work

In this paper, we present a comprehensive evaluation of the number of final layers that need to be fine-tuned for pretrained transformer-based language models. We find that only a fourth of the layers are sufficient to achieve at least 95% of the original quality, regardless of the underlying language and the type of target task. One line of future work is to conduct a similar, more fine-grained analysis on the contributions of the attention heads.

References

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019a. What does BERT look at? An analysis of BERT's attention. *arXiv:1906.04341*.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019b. BAM! Born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv:1911.02116*.

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*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. *arXiv:1908.08593*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv:1901.07291*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv:1904.09482*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. *arXiv:1901.11504*.

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

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *arXiv:1905.10650*.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? *arXiv:1906.01502*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art natural language processing. *arXiv:1910.03771*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: generalized autoregressive pretraining for language understanding. *arXiv:1906.08237*.

Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*.

A Fine-Tuning for Different Tasks

A.1 GLUE Benchmark

For the fine-tuning experiments, we use eight natural language understanding tasks from the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). For natural language inference (NLI), it provides the Multigenre NLI (MNLI; Williams et al., 2018), Question NLI (QNLI; Wang et al., 2018), and Recognizing Textual Entailment (RTE; Bentivogli et al., 2009) datasets. For semantic textual similarity and paraphrasing, it contains the Microsoft Research Paraphrase Corpus (MRPC; Dolan and Brockett, 2005), the Semantic Textual Similarity Benchmark (STS-B; Cer et al., 2017), and Quora Question Pairs (QQP; Iyer et al.). Finally, its single-sentence tasks consist of the binary-polarity Stanford Sentiment Treebank (SST-2; Socher et al., 2013) and the Corpus of Linguistic Acceptability (CoLA; Warstadt et al., 2018). We exclude the Winograd NLI (Levesque et al., 2012) dataset due to the known development set issue.[1]

Task	# Training	# Dev
CoLA	8,550	1,041
SST-2	67,349	872
MRPC	3,668	408
STS-B	5,749	1,500
QQP	363,870	40,432
MNLI	392,702	9,815
MNLI-mm	392,702	9,832
QNLI	104,743	5,463
RTE	2,490	278

A.2 Fine-tuning Configuration

Our fine-tuning procedure closely resembles those of BERT and RoBERTa. We choose the Adam optimizer (Kingma and Ba, 2014) with a batch size of 16 and fine-tune BERT for 3 epochs and RoBERTa for 10, following the original papers. For hyperparameter tuning, the best learning rate is different for each task, and all of the original authors choose one between 1×10^{-5} and 5×10^{-5}; thus, we perform line search over the interval with a step size of 1×10^{-5}.

A.3 Experimental Results

Following tables summarize how the scores on the development sets change with respect to the amount of frozen layers. To provide more insights about the model sensitivity, we also report relative performance scores averaged across the tasks.

For some tasks, we realize that fine-tuning all layers does not always help. We highlight the finding by underlining the scores that are higher than the corresponding baseline score.

- BERT$_{base}$

Task (metric)	Baseline	Frozen layers						
		6/12	7/12	8/12	9/12	10/12	11/12	12/12
CoLA (MCC)	58.26	53.27	51.85	50.59	47.49	44.01	42.10	29.59
SST-2 (Acc.)	92.69	92.06	91.75	91.15	90.84	90.89	91.33	85.03
MRPC (F_1)	90.30	87.70	86.90	86.65	85.45	84.33	82.31	81.42
STS-B (ρ)	88.87	88.47	88.34	88.36	87.97	87.38	86.18	77.97
QQP (F_1)	87.89	86.92	86.44	85.97	85.33	84.30	82.17	71.98
MNLI (Acc.)	84.34	83.91	83.60	83.30	81.98	79.94	76.22	56.36
MNLI-mm (Acc.)	84.81	84.25	84.02	83.64	82.44	80.22	76.81	57.14
QNLI (Acc.)	91.38	91.14	90.77	90.34	89.52	88.04	85.32	74.46
RTE (Acc.)	67.55	<u>68.71</u>	66.67	63.85	62.32	59.93	57.87	57.82
Rel. perf. (%)	100.00	98.51	97.58	96.57	94.98	92.85	90.26	78.18

[1] See the 12th answer in https://gluebenchmark.com/faq.

- BERT$_{large}$

Task (metric)	Baseline	Frozen layers												
		12/24	13/24	14/24	15/24	16/24	17/24	18/24	19/24	20/24	21/24	22/24	23/24	24/24
CoLA (MCC)	61.86	59.12	58.91	57.96	56.33	56.01	54.35	51.56	49.48	47.97	45.69	42.14	39.10	24.91
SST-2 (Acc.)	93.41	93.33	93.33	93.45	93.25	93.15	93.02	92.67	92.40	91.95	91.47	91.12	90.93	87.77
MRPC (F_1)	90.34	88.90	88.31	87.49	86.52	85.97	85.44	85.43	84.86	84.85	84.56	83.80	82.78	81.33
STS-B (ρ)	89.77	89.03	89.06	89.01	89.02	88.59	88.32	88.00	87.29	86.81	86.27	86.00	85.28	71.77
RTE (Acc.)	72.27	71.60	71.76	69.15	68.41	68.38	66.90	65.05	64.30	60.69	61.16	60.97	58.05	56.55
Rel. perf. (%)	100.00	98.42	98.28	97.08	96.09	95.74	94.59	93.03	91.81	90.11	89.22	87.72	85.50	76.49

- RoBERTa$_{base}$

Task (metric)	Baseline	Frozen layers						
		6/12	7/12	8/12	9/12	10/12	11/12	12/12
CoLA (MCC)	59.53	58.54	58.57	55.19	54.26	52.62	51.53	0.00
SST-2 (Acc.)	94.31	94.00	93.28	93.49	93.58	93.18	92.02	80.17
MRPC (F_1)	92.28	90.78	89.61	88.86	88.74	87.98	85.27	81.22
STS-B (ρ)	90.64	88.85	87.65	87.17	86.84	85.49	83.94	19.68
QQP (F_1)	88.84	87.70	87.47	87.44	87.08	85.62	83.11	62.51
MNLI (Acc.)	87.36	85.71	84.93	85.04	84.59	82.51	76.22	52.61
MNLI-mm (Acc.)	87.05	85.88	84.99	85.62	85.07	83.27	78.00	53.03
QNLI (Acc.)	92.75	91.71	91.35	90.99	90.70	89.24	84.93	65.71
RTE (Acc.)	77.51	75.45	72.45	68.63	66.97	64.95	61.66	57.54
Rel. perf. (%)	100.00	98.45	97.37	96.12	95.45	93.70	90.07	59.03

- RoBERTa$_{large}$

Task (metric)	Baseline	Frozen layers												
		12/24	13/24	14/24	15/24	16/24	17/24	18/24	19/24	20/24	21/24	22/24	23/24	24/24
CoLA (MCC)	66.56	61.93	60.43	60.44	60.77	61.58	60.74	60.39	60.39	59.67	58.79	54.63	49.18	0.00
SST-2 (Acc.)	95.47	95.81	95.59	95.45	95.10	95.28	94.98	95.08	94.71	93.71	93.86	94.01	92.23	79.26
MRPC (F_1)	92.27	92.54	92.55	92.40	91.77	92.12	91.30	91.18	89.92	86.07	84.89	85.15	84.28	81.22
STS-B (ρ)	91.94	91.06	90.73	90.60	90.59	90.43	89.69	88.74	88.05	85.71	84.31	83.59	82.29	11.11
RTE (Acc.)	84.73	82.64	80.91	80.65	80.43	79.31	77.66	74.22	70.40	66.39	63.72	61.23	59.25	49.17
Rel. perf. (%)	100.00	98.05	97.08	96.93	96.76	96.82	95.78	94.65	93.25	90.53	89.11	87.20	84.25	48.23

- MT-BERT$_{base}$

Task (metric)	Baseline	Frozen layers						
		6/12	7/12	8/12	9/12	10/12	11/12	12/12
CoLA (MCC)	57.38	55.69	56.25	54.28	54.11	54.22	54.22	53.64
SST-2 (Acc.)	92.78	92.55	92.59	92.34	92.27	92.22	92.04	91.86
MRPC (F_1)	92.28	92.28	91.85	91.83	91.49	91.50	90.61	85.40
STS-B (ρ)	90.88	90.64	90.54	90.47	90.30	90.15	89.97	89.54
QQP (F_1)	88.38	87.76	87.70	87.63	87.54	87.50	87.40	87.18
MNLI (Acc.)	84.44	84.02	84.11	84.19	83.91	83.72	83.05	76.65
MNLI-mm (Acc.)	84.67	84.46	84.30	84.25	84.16	83.90	83.34	77.74
QNLI (Acc.)	91.11	90.94	90.87	90.66	90.39	90.18	89.25	85.89
RTE (Acc.)	78.41	79.49	79.13	78.48	76.82	75.60	75.23	71.05
Rel. perf. (%)	100.00	99.59	99.56	99.02	98.59	98.32	97.83	94.41

- MT-BERT$_{large}$

Task (metric)	Baseline	Frozen layers												
		12/24	13/24	14/24	15/24	16/24	17/24	18/24	19/24	20/24	21/24	22/24	23/24	24/24
CoLA (MCC)	59.55	58.17	57.82	57.97	57.21	56.90	57.60	57.71	57.97	58.03	57.50	57.51	57.32	57.82
SST-2 (Acc.)	92.87	92.68	92.91	92.78	92.52	92.57	92.91	92.84	92.91	92.71	93.07	93.14	93.10	93.10
MRPC (F_1)	91.08	90.52	90.62	90.41	90.18	90.35	90.05	89.91	89.50	88.86	88.50	88.19	87.65	87.73
STS-B (ρ)	91.11	90.65	90.47	90.31	90.20	90.11	90.00	89.84	89.57	89.37	89.20	89.11	88.92	88.52
RTE (Acc.)	80.43	81.81	81.08	81.30	81.16	80.87	80.07	79.78	78.27	74.01	73.43	71.05	69.53	70.18
Rel. perf. (%)	100.00	99.61	99.35	99.34	98.92	98.77	98.80	98.68	98.25	96.99	96.63	95.97	95.36	95.62

- MT-RoBERTa$_{base}$

Task (metric)	Baseline	Frozen layers						
		6/12	7/12	8/12	9/12	10/12	11/12	12/12
CoLA (MCC)	59.85	58.51	59.06	55.63	54.76	53.26	50.51	0.00
SST-2 (Acc.)	94.63	93.90	93.58	93.53	93.46	93.05	92.06	80.28
MRPC (F_1)	92.79	90.49	89.74	88.71	88.49	88.29	85.08	81.22
STS-B (ρ)	90.76	88.80	87.62	87.21	86.99	85.67	84.03	20.00
QQP (F_1)	88.83	87.69	87.47	87.47	87.08	85.52	83.12	62.51
MNLI (Acc.)	87.41	85.68	84.94	84.92	84.68	82.45	77.09	52.60
MNLI-mm (Acc.)	86.99	85.93	85.17	85.60	85.11	83.24	77.58	53.02
QNLI (Acc.)	92.75	91.59	91.33	90.95	90.77	89.15	84.97	65.74
RTE (Acc.)	78.16	74.44	71.55	68.38	66.86	65.70	61.81	57.40
Rel. perf. (%)	100.00	97.99	97.15	95.90	95.29	93.70	89.73	58.91

- MT-RoBERTa$_{large}$

Task (metric)	Baseline	Frozen layers												
		12/24	13/24	14/24	15/24	16/24	17/24	18/24	19/24	20/24	21/24	22/24	23/24	24/24
CoLA (MCC)	65.91	60.10	59.84	60.27	62.32	61.87	61.49	60.80	60.53	58.86	58.69	54.72	48.95	0.00
SST-2 (Acc.)	95.55	95.74	95.49	95.16	95.20	95.05	95.11	94.69	94.75	93.85	93.58	94.11	92.32	79.27
MRPC (F_1)	92.79	92.64	92.43	92.05	91.39	91.46	91.66	91.31	90.02	86.25	84.80	85.22	84.20	81.22
STS-B (ρ)	91.91	90.95	90.77	90.46	90.53	90.26	89.83	88.67	87.88	85.76	84.29	83.54	82.38	11.19
RTE (Acc.)	85.05	83.03	81.65	80.96	80.58	78.64	76.90	73.65	69.25	66.61	64.20	61.49	59.45	48.74
Rel. perf. (%)	100.00	97.56	97.02	96.77	97.18	96.51	95.95	94.56	93.01	90.42	89.11	87.31	84.23	47.99

A.4 Multi-Task Model Comparison

The following table compares MT-RoBERTa with existing multi-task models. The reported scores are from large variants evaluated on development sets. We also include BERT and RoBERTa as baselines (first two rows) and the best score for each task is bolded.

Model	CoLA MCC	SST-2 Acc.	MRPC F_1	STS-B ρ	RTE Acc.
BERT (Devlin et al., 2019)	61.9	93.4	90.3	89.8	72.3
RoBERTa (Liu et al., 2019c)	66.6	95.5	92.3	91.9	84.7
BAM-BERT (Clark et al., 2019)	61.8	93.6	89.3	89.7	82.8
MT-BERT (Liu et al., 2019b)	63.5	94.3	91.0	90.7	83.4
MT-BERT$_{KD}$ (Liu et al., 2019a)	64.5	94.3	**93.3**	91.0	**88.6**
MT-RoBERTa	**66.0**	**95.0**	93.1	**92.0**	86.3

B Fine-Tuning for Different Languages

B.1 Named Entity Recognition

For crosslingual named entity recognition (NER), we have selected the CoNLL 2002 and 2003 datasets (Tjong Kim Sang, 2002; Tjong Kim Sang and De Meulder, 2003) which consist of four Romance and Germanic languages—English, Spanish, Dutch, German—with the standard, self-explanatory four entity types of PERSON, LOCATION, ORGANIZATION and MISC.

Language	# Training	# Dev	# Test
German	15,823	2,868	3,005
English	14,041	3,250	3,453
Spanish	8,335	1,915	1,518
Dutch	12,152	2,895	5,199

Each XML-R model is fine-tuned with the Adam optimizer for 3 epochs. We set the learning rate to 1×10^{-5} and the batch size to 8. To evaluate the model quality, we report F_1 scores.

- XLM-R$_{base}$

Language	Baseline	Frozen layers						
		6/12	7/12	8/12	9/12	10/12	11/12	12/12
German	0.82	0.80	0.80	0.79	0.77	0.73	0.66	0.06
English	0.91	0.88	0.88	0.87	0.86	0.85	0.82	0.07
Spanish	0.87	0.86	0.85	0.85	0.84	0.81	0.76	0.06
Dutch	0.90	0.87	0.87	0.86	0.84	0.80	0.73	0.06
Rel. F_1 (%)	100.00	97.61	97.09	96.18	94.28	91.04	84.40	7.15

- XLM-R$_{large}$

Language	Baseline	Frozen layers											
		12/24	13/24	14/24	15/24	16/24	17/24	18/24	19/24	20/24	21/24	22/24	23/24	24/24
German	0.85	0.84	0.83	0.83	0.83	0.82	0.82	0.82	0.81	0.80	0.76	0.73	0.69	0.04
English	0.92	0.91	0.91	0.91	0.91	0.90	0.90	0.89	0.88	0.87	0.86	0.84	0.81	0.08
Spanish	0.89	0.88	0.88	0.87	0.87	0.86	0.85	0.84	0.84	0.81	0.79	0.77	0.75	0.06
Dutch	0.93	0.92	0.91	0.91	0.91	0.90	0.90	0.89	0.88	0.87	0.85	0.81	0.77	0.03
Rel. F_1 (%)	100.00	98.80	98.32	98.06	97.67	97.19	96.71	96.00	95.05	93.23	90.66	87.54	84.04	5.78

B.2 Part-of-speech Tagging

We also fine-tune XLM-R for part-of-speech (POS) tagging with Universal Dependencies POS tags, following Petrov et al. (2011). For this experiment, we use CoNLL 2006 and 2007 (Buchholz and Marsi, 2006; Nivre et al., 2007) which consist of seven languages—Italian, Greek, Danish, German, Spanish, Dutch, and Slovenian.

Language	# Training	# Dev	# Test
Italian	3,110	293	249
Greek	2,705	267	197
Danish	5,190	491	322
German	39,216	3,875	357
Spanish	3,306	300	206
Dutch	13,349	1,401	386
Slovenian	1,534	207	402

Same as in the previous NER experiments, each model is fine-tuned with the Adam optimizer for 3 epochs. The learning rate is set to 1×10^{-5}, and the batch size set to 8. The metric of interest for POS tagging is accuracy.

- XLM-R$_{base}$

Language	Baseline	Frozen layers						
		6/12	7/12	8/12	9/12	10/12	11/12	12/12
Italian	98.26	97.73	97.58	97.49	97.17	96.51	95.68	41.98
Greek	96.43	95.57	95.46	95.17	94.85	93.85	92.97	36.64
Danish	98.63	98.19	98.06	97.82	97.47	96.92	96.15	51.71
German	99.21	99.17	99.08	99.05	98.92	98.80	98.35	82.20
Spanish	98.97	98.84	98.66	98.44	98.24	97.72	96.86	35.18
Dutch	95.85	95.18	95.03	94.87	94.66	94.14	93.43	63.20
Slovenian	98.03	97.01	96.92	96.39	95.65	94.34	93.15	27.19
Rel. acc. (%)	100.00	99.46	99.33	99.10	98.77	98.09	97.25	49.32

- XLM-R$_{large}$

Language	Baseline	Frozen layers												
		12/24	13/24	14/24	15/24	16/24	17/24	18/24	19/24	20/24	21/24	22/24	23/24	24/24
Italian	98.56	98.34	98.34	98.20	98.28	98.12	98.09	98.04	97.89	97.68	97.38	96.76	96.12	53.98
Greek	96.62	96.31	96.23	96.12	96.12	96.02	95.92	95.82	95.61	95.31	94.89	94.19	93.11	53.81
Danish	98.89	98.64	98.59	98.55	98.53	98.48	98.39	98.35	98.18	97.93	97.43	97.01	96.18	62.51
German	99.24	99.22	99.21	99.15	99.16	99.14	99.14	99.11	99.10	99.09	98.94	98.66	98.45	83.46
Spanish	99.21	98.99	98.90	98.88	98.79	98.82	98.76	98.65	98.41	98.25	97.96	97.46	96.95	60.06
Dutch	96.03	95.53	95.37	95.20	95.23	95.08	95.12	95.01	94.88	94.58	94.39	93.97	93.14	66.72
Slovenian	98.52	97.98	97.89	97.74	97.69	97.58	97.46	97.24	96.94	96.14	95.45	94.40	92.93	44.48
Rel. acc. (%)	100.00	99.70	99.63	99.53	99.52	99.44	99.39	99.29	99.11	98.82	98.45	97.87	97.06	61.85

References

Luisa Bentivogli, Ido Kalman Dagan, Dang Hoa, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC 2009 Workshop*.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, CoNLL-X '06.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation*.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D. Manning, and Quoc V. Le. 2019. BAM! Born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the 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*.

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

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First Quora dataset release: Question pairs.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980.*

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019a. Improving multi-task deep neural networks via knowledge distillation for natural language understanding. *arXiv:1904.09482*.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019b. Multi-task deep neural networks for natural language understanding. *arXiv:1901.11504*.

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

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 915–932, Prague, Czech Republic. Association for Computational Linguistics.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2011. A universal part-of-speech tagset. *arXiv:1104.2086*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Erik F. Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*.

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

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2018. Neural network acceptability judgments. *arXiv:1805.12471*.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.