Muppet: Massive Multi-task Representations with Pre-Finetuning

Armen Aghajanyan
Facebook
armenag@fb.com

Anchit Gupta
Facebook
anchit@fb.com

Akshat Shrivastava
Facebook
akshats@fb.com

Xilun Chen
Facebook
xilun@fb.com

Luke Zettlemoyer
Facebook
lsz@fb.com

Sonal Gupta
Facebook
sonalgupta@fb.com

Abstract

We propose pre-finetuning, an additional large-scale learning stage between language model pre-training and fine-tuning. Pre-finetuning is massively multi-task learning (around 50 datasets, over 4.8 million total labeled examples), and is designed to encourage learning of representations that generalize better to many different tasks. We show that pre-finetuning consistently improves performance for pretrained discriminators (e.g. RoBERTa) and generation models (e.g. BART) on a wide range of tasks (sentence prediction, common-sense reasoning, MRC, etc.), while also significantly improving sample efficiency during fine-tuning. We also show that large-scale multi-tasking is crucial; pre-finetuning can hurt performance when few tasks are used up until a critical point (usually above 15) after which performance improves linearly in the number of tasks.

1 Introduction

The recent success of language model pre-training (Devlin et al., 2018; Liu et al., 2019b; Lewis et al., 2019; Raffel et al., 2019; Radford et al., 2019) is remarkable, at least in part, due to the exclusive use of self supervision, without any manually labeled data. For many tasks, however, we already have training examples for related problems, which we should be able to leverage. Recent work has shown gains from fine-tuning schemes that are multi-task (Raffel et al., 2019; Khashabi et al., 2020) and multi-stage (Liu et al., 2019a), but it can be difficult to know which intermediate tasks will best transfer (Raffel et al., 2019). In this paper, we show that multi-task supervised tuning, if done at a sufficiently large scale with many different tasks, can be an effective second stage of task-agnostic pre-training, removing the need to pre-select the best intermediate tasks.

More specifically, in addition to the standard pre-training/fine-tuning methodology of learning language tasks, we introduce a new intermediate stage, *pre-finetuning*. Pre-finetuning involves a massive multi-task learning step (4.8 million total training examples) performed on around 50 classification, summarization, question answering, and common sense reasoning tasks. We believe we are the first to investigate multi-task learning at this scale in terms of both number and types of tasks. We show, in particular, that standard multi-tasking schemes can be unstable and often fail to learn high quality representations. However, we introduce a new training scheme which uses loss scaling and task-heterogeneous batches so that gradient steps are more evenly balanced across multiple different competing tasks, greatly improving training stability and overall performance. We call our pre-finetuned models MUPPET; Massive Multi-task RePresentation with PrE-fineTuning.

Through extensive experiments, we show that incorporating pre-finetuning to RoBERTa (Liu et al., 2019b) and BART (Lewis et al., 2019) models yields consistent improvements, including new state-of-the-art performance for RTE (Bentivogli et al., 2009) and HellaSWAG (Zellers et al., 2019), without having to specify specific intermediate transfer tasks. These gains are particularly strong in the low resource regime, where there is relatively little labeled data for fine-tuning. We also study why pre-finetuning outperforms previous multi-tasking schemes. We first compare different optimization techniques to stabilize training, and find it important to use task-heterogeneous batches with task-rebalancing loss scaling. We also show that scale is crucial for effective multi-task learning. We empirically see a critical point in terms of the number of tasks (usually over 15); having fewer tasks degrades representations, while having more seems to improve performance linearly as far as we were able to scale.

To summarize, our contributions include:

- We show that we can further improve pre-

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 5799–5811
November 7–11, 2021. ©2021 Association for Computational Linguistics

trained representations with an additional stage we call pre-finetuning, which utilizes massively multi-task learning. We show standard pre-trained representations, when further refined with pre-finetuning consistently improve performance on downstream tasks.

- We introduce a new multi-task training scheme for effective learning at scale, which uses loss scaling and task-heterogeneous batches.
- We explore the effects of scale on multi-task learning and show the existence of critical points in multi-task training, beyond which increasing the number of tasks improves generalizable representations.
- We conduct a study surrounding the data efficiency of standard pre-trained representations and their respective pre-finetuned counterparts. We show that the pre-finetuned models consistently require less data for fine-tuning.

2 Related Work

Multi-task learning has been an increasingly active topic in recent literature. Recent advances such as MT-DNN show that by leveraging multi-task learning, we can further improve performance on several language benchmarks on top of traditional pre-training (Liu et al., 2019a). However, T5 (Raffel et al., 2019) shows that incorporating multi-task learning ontop of larger models does not improve upon the standardized pre-training / finetuning. Thus the effect of multi-task learning across different pre-training methods is not fully understood.

Recently Khashabi et al. (2020) showed how doing MTL training on a range of QA tasks can improve the performance of T5 by taking advantage of cross dataset transfer. Unlike our approach, they convert all the data to a seq2seq format, operate on a smaller MTL scale, have a different batching strategy, and focus solely on improving QA tasks. Our work shows how even seemingly very different datasets, for example, summarization and extractive QA, can help each other by improving the model's representations.

Our work aims to explore multi-task learning at a much larger scale; by incorporating a larger number of tasks, we show that we can consistently improve several language benchmarks from several domains. Contrary to T5, we show that incorporating a secondary stage of multi-task learning does lead to better representations. In §5 we demonstrate the effectiveness of multi-task learning to be coming from the large scale of our MTL setup.

Task Type	# Datasets	# Train	# Eval
Classification	26	2.9M	188K
Summarization	4	524K	30K
MRC	6	1.05M	123M
CommonSense	10	360K	49K
Total	46	4.8M	390K

Table 1: Break down of MTL pre-finetuning datasets. The table shows the number of datasets we used per task type and the number of samples in training and evaluation sets.

3 Pre-Finetuning Through Massive Multitask Learning

Previous work has reported mixed results from experiments on multi-task learning (Liu et al., 2019a; Raffel et al., 2019). In general, it can be challenging to balance the losses from different tasks; upsampling can lead to overfitting low resource tasks, and downsampling can lead to improper learning of specific tasks. This difficulty is particularly pronounced when operating at the scale of experiments we show in Section 5.1, where there are more diverse tasks than previously considered. This section presents our pre-finetuning approach that leads to more stable and accurate multi-task training by introducing new optimization, loss scaling, and task sampling schemes to balance each minibatch's updates better.

3.1 Tasks and Losses

Diverse Tasks To learn general language representations, we include a variety of tasks across many domains. We select language tasks across four different domains: *classification*, *commonsense reasoning*, *machine reading comprehension*, and *summarization*. In Table 1, we show the break down of each of the task types along with the number of samples used from each during pre-finetuning. In total our multi-task set up learns over 4.8 supervised samples across 4 families of tasks.

A full list of all of the datasets we leverage for pre-finetuning is described in appendix §A.1.

Task Type	Loss Function
Classification	Cross Entropy (CE)
Summarization	Label Smoothed CE (Szegedy et al., 2015)
MRC	Span Prediction (Seo et al., 2016)
Commonsense	Sentence Ranking Loss (Liu et al., 2019b)

Table 2: Description of loss functions for each task type. Note for summarization the label smoothed cross entropy loss is averaged across tokens.

Standard Losses To train on several datasets, our model contains task-specific heads, each optimizing for a task-specific loss. The loss functions are summarized in table 2. Each loss is scaled with loss scaling described in §3.3. After loss scaling, the gradients from each task are averaged before doing the model update step.

3.2 Optimization

We show two strategies to learn multi-task representations at scale: *Accumulating Gradients Across Tasks (Heterogeneous Batches)* and *Leveraging Better Finetuning*.

Accumulating Gradients Across Tasks Our model is trying to optimize not a single objective but several potentially competing objectives to create a unified representation across several tasks during model training. During gradient descent, moving along the gradient of a single task may not be the optimal direction for the model to move to learn a single unified representation across tasks. To overcome this, we ensure each batch our model optimizes consists of several tasks. Each worker samples a random batch from our set of tasks and computes a gradient, accumulated for the final update. Empirically we use 64 GPUs for pre-finetuning, resulting in each batch consisting of gradients across 64 sampled tasks. In §5.2 we show how such a strategy allows for our model to arrive at a better representation for end task finetuning.

Better Finetuning Instead of starting from scratch, we initialize our model with representations learned from self-supervised pre-training in pre-finetuning. This can inherit the knowledge captured in the pre-trained representations and speed up training. Mosbach et al. (2020) show that standard fine-tuning of pre-trained models can be unstable, which may be aggravated in our case as we are training on a diverse set of tasks simultaneously. Therefore, we employ the R3F/R4F methods (Aghajanyan et al., 2020) to combat this issue. In particular, R3F/R4F consists of an additional loss term, ensuring that small perturbations to the input space result in similar representations, which can be used to learn more robust representations during pre-finetuning.

In early experimentation, we found that R3F was pivotal in getting MUPPET to work for BART. All other fine-tuning and pre-finetuning was done using standard SGD.

3.3 Loss Scaling

Loss scaling methods introduce a multiplicative reweighting of individual losses per data-point. Various loss scaling techniques have been proposed, from dynamic scaling by inverse training loss to simple scaling by the number of data-points in respective datasets (Chen et al., 2018).

As pre-finetuning optimizes several different types of tasks and datasets, each having its own output spaces, loss scaling becomes essential to ensure stable training. We attempted various forms of loss-scaling throughout initial experimentation, but the most effective was the novel method we describe below.

Let us denote $\mathcal{L}_i(x_i, y_i; \theta)$ as the loss for data-point i for a model parameterized by θ. Remember that the loss depends on the type of task (commonsense loss is different from binary classification). Furthermore let $n : \mathbb{N} \to \mathbb{N}$ be a function which for each data-point returns the number of predictions $\mathcal{L}$ operates over. For example, for binary classification, n would return two, while for generation, n would return the size of the vocabulary (since we average across loss per token generated). We scale data-point loss so that, if the class distribution were uniformly distributed along with our models predictions, all of our losses would have equivalent values.

$$\mathcal{L}_i^{scaled}(x_i, y_i; \theta) = \frac{\mathcal{L}_i(x_i, y_i; \theta)}{\log n(i)} \tag{1}$$

We found that this static scaling worked incredibly well, outperforming other loss scaling methods in early experimentation.

3.4 Sampling

Another approach to balancing various tasks in a multi-task set up is to up-sample smaller datasets and down-sample larger ones to achieve more uniformity between dataset sizes.

Existing results for dataset sampling methods in multi-task learning are conflicting, but recent work has shown that it does not work well for multi-task learning of pre-trained representations. For example, T5 showed that all various forms of sampling did not improve overusing the natural size of datasets (Raffel et al., 2019).

We also found that sampling datasets were consistently detrimental for multi-task learning over pre-trained representations during initial experimentation. Specifically, we saw unmanageable over-fitting and stability issues. Therefore we opt for maintaining the natural distribution of the datasets throughout all of our experiments.

3.5 Experimental Setup

We selected RoBERTa (Liu et al., 2019b) and BART (Lewis et al., 2019) as our initial pre-trained models to further pre-finetune. For each task type we use a different prediction scheme. Every *Sentence Prediction* dataset gets a separate classification head, for *Commonsense* and *MRC* we utilize a separate unified head for each task. For *Summarization*, we do not add any parameters and use the BART decoder and output layer as is. Experimentally we saw using a different head per individual *Commonsense* and *MRC* datasets lead to severe overfitting.

For both models, we do the pre-finetuning procedure for both the *Base* and *Large* models. We trained each model configuration with 64 GPUs until convergence. Dependent on configuration, this ranged from a day to 4 days. We include the hyper-parameters used per pre-finetuning run in the Appendix in Section §A.2.

4 Empirical Results

We first show that pre-finetuning improves the representations of pre-training models. To do so, we fine-tune our pre-finetuned models on a large set of tasks.

For each of the individual downstream tasks, we use a fixed hyper-parameter search to optimize over simple hyperparameters such as learning rate, Adam ϵ (Kingma and Ba, 2014) and dropout (Srivastava et al., 2014). We present our results in two tables. Table 3 shows our results on the GLUE benchmark (Wang et al., 2018) as well as two *MRC* tasks; SQuAD (Rajpurkar et al., 2016a) and ReCoRD (Zhang et al., 2018). Table 4 reports results on other *Sentence Prediction* tasks as well as *Commonsense* tasks. We also include results from MT-DNN (Liu et al., 2019a), ELECTRA (Clark et al., 2020),[1] and RoBERTa (Liu et al., 2019b) models. For *Summarization* tasks we show that our pre-finetuned BART model outperforms all other summarization baselines. Both of these tables report over data-sets available during the pre-finetuning stage.

Given that our pre-finetuned models now have an understanding of the task at hand through the use of classification heads, we have a choice during finetuning on whether or not to use these heads. In general we found re-using heads to be beneficial for *MRC*, *Commonsense* and *Sentence Prediction* tasks with small dataset size.

Across the board, pre-trained representations that were further refined with pre-finetuning outperformed standard pre-trained representations. We see more modest gains on larger datasets, most likely because we do not need to refine representations beforehand if the fine-tuning dataset is large. On smaller datasets, we see substantial gains. For example, the pre-finetuned RoBERTa-BASE model on RTE improves by close to 9 points, rivaling the RoBERTa-Large accuracy, while the pre-finetuned RoBERTa-Large model gets new state-of-the-art on RTE rivaling models an order of magnitude larger than it.

We do not improve just over sentence prediction tasks but on every set of tasks that we measured. For example, we reach a new state of the art on the HellaSwag dataset previously achieved by utilizing a new fine-tuning approach. Our methods do not increase parameter count or any complexity measures but are quite successful at refining features and preparing them for downstream fine-tuning.

4.1 Finetuning Outside of Pre-Finetuning Domain

We also report the performance on tasks not included in the pre-finetuning data. To do so, we finetune our models on a set of tasks including (1) ANLI (Nie et al., 2019) and Hyperpartisan (Kiesel et al., 2019) for classification, (2) Arxiv (He et al., 2019), PubMed (Cohan et al., 2018), (Sharma et al., 2019) for summarization, and (3) Chunking, Constituency Parsing and Part-Of-Speech tagging for structured prediction from the Penn Treebank dataset (Marcus et al., 1993). We present these results in Table 5 and Table 6.

[1] For ELECTRA results we leverage the results presented in the ELECTRA github `https://github.com/google-research/electra#expected-results`

	GLUE						MRC
	MNLI	QQP	RTE	QNLI	MRPC	SST-2	SQuAD
RoBERTa-B	87.6	**91.9**	78.7	92.8	90.2	94.8	82.6
+ MUPPET	**88.1**	**91.9**	**87.8**	**93.3**	**91.7**	**96.7**	**86.6**
RoBERTa-L	90.2	**92.2**	88.1	94.7	90.9	96.4	88.7
+ MUPPET	**90.8**	**92.2**	**92.8**	**94.9**	**91.4**	**97.4**	**89.4**
BART	**89.9**	92.5	87.0	**94.9**	90.4	96.6	
+ MUPPET	**89.9**	**92.7**	**92.4**	94.6	**92.2**	**96.9**	
ELECTRA-B	88.8	91.5	82.7	93.2	89.5	95	80.5
ELECTRA-L	90.9	92.4	88.0	95.0	90.8	96.9	88.1
MT-DNN	87.1	91.9/89.2	83.4	92.9	91.0/87.5	94.3	-

Table 3: We present results for the GLUE benchmark task and a MRC dataset. Bolded numbers show the MUPPET vs. base model, underline marks the best number. If not explicitly stated, the results are showing the accuracy of the evaluation set. For the MRC tasks, we report both exact match (EM) and F1 as is standard in the literature. For SQuAD, we reused the task head from pre-finetuning.

	SP	Commonsense			Summarization		
	BoolQ	CQA	HellaSwag	OpenQA	CNN/DailyMail	Gigaword	Reddit TIFU
RoBERTa-B	82.0	66.2	65.1	63.8	-	-	-
+ MUPPET	**83.8**	**69.4**	**69.0**	**64.6**	-	-	-
RoBERTa-L	86.4	78.1	83.4	73.6	-	-	-
+ MUPPET	**87.5**	**79.2**	**86.4**	**74.4**	-	-	-
BART	86.2	**78.1**	**84.1**	**71.4**	44.16/21.28/40.90	39.29/20.09/35.65	24.19/8.12/21.31
+ MUPPET	**86.9**	74.8	75.9	70.8	**44.45/21.25/41.4**	**40.40/20.54/36.21**	**30.30/11.25/24.92**
T5-L	86.2	75.6	83.9	70.4	42.50/20.68/39.75	-	-
T5-11B	86.8	78.9	85.8	75.4	43.52/21.55/40.69	-	-
PEGASUS	-	-	-	-	44.17/**21.47**/41.11	39.12/19.86/36.24	26.63/9.01/21.60
ERNIE-GEN	-	-	-	-	44.02/**21.17**/**41.26**	39.25/ 20.25/**36.53**	-
ProphetNet	-	-	-	-	44.20/21.17/**41.30**	39.51/20.42/**36.69**	-

Table 4: We present results for the non-GLUE Sentence Prediction tasks as well as a set of standard Commonsense tasks. Bolded numbers signify MUPPET vs. base model, while an underline signifies the best number. If not explicitly stated, the results are showing the accuracy of the evaluation set. For commonsense tasks, we re-use the task head from pre-finetuning.

We see that the MUPPET variants of our models out-perform the baselines consistently across task type and dataset. As a special case we do an in depth analysis of the MUPPET variant of RoBERTa on the notoriously tough ANLI dataset and see the same pattern. Pre-finetuned models consistently outperform their base counterparts.

5 Understanding Multi-Task at Scale

5.1 Importance of Scale

The first axis we would like to explore is the scale on which multi-task learning is done. Previous work, such as T5 and MT-DNN, focused on the MTL scale of around a dozen datasets. To the best of our knowledge, our paper has the largest MTL set up to date. Accordingly, we are interested in empirically exploring the effects of scaling up the number of datasets to the representations learned during MTL.

We pre-finetune a collection of *RoBERTa-Base* models with varying numbers of datasets. We train seven models, six uniformly chosen between 10 and 40, ensuring that at each point, the selected datasets are a superset of the datasets from prior points. The last model is fully trained on all datasets. Concretely given two models trained with a different number of datasets $a, b : a > b$, model a will contain all datasets used to train model b and more.

For each version of the model, we fine-tune five datasets and plot the results in Figure 1. Specifically we finetune STS-B (Cer et al., 2017), BoolQ

	SP	Structured Prediction (Penn)			Summarization		
	Hyperpartisan	Chunking	Parsing	POS	Arxiv	PubMed	BigPatent
RoBERTa-B	84.2	93.4	**95.1**	**93.7**	-	-	-
+ MUPPET	**85.8**	**95.5**	94.5	93.2	-	-	-
RoBERTa-L	90.4	95.1	94.5	93.4	-	-	-
+ MUPPET	**92.5**	**96.9**	**95.7**	**97.9**	-	-	-
BART	85.1	92.1	91.1	91.8	41.20/9.20/32.45	39.87/16.43/35.56	48.54/29.35/39.42
+ MUPPET	**87.2**	**96.1**	**94.5**	**97.2**	**43.90/14.50/40.10**	**45.13/19.80/39.90**	**52.34/33.50/42.80**
Pegasus	-	-	-	-	43.85/16.83/39.17	44.53/19.30/40.70	52.25/33.04/41.80

Table 5: We present results on a large set of different tasks across datasets that are not available to the model during the pre-finetuning stage. Bolded numbers signify MUPPET vs. base model, while an underline signifies the best number. For Chunking/Parsing, we use F1, while for Part-Of-Speech tagging, we use accuracy.

Model	**Training Data**	A1	A2	A3	ANLI
RoBERTa	S,M	47.6	25.4	22.1	31.1
	+F	54.0	24.2	22.4	32.8
	+F+A1$^{\star 2}$	68.7	19.3	22.0	35.8
	+F+A1+A2$^{\star 3}$	71.2	44.3	20.4	43.7
	S,M,F,ANLI	73.8	48.9	**44.4**	53.7
RoBERTa-MUPPET	S,M	**49.9**	**28.2**	**24.2**	**33.3**
	+F	**55.2**	**26.8**	**24.6**	**33.9**
	+F+A1$^{\star 2}$	**70.9**	**22.5**	**25.1**	**36.7**
	+F+A1+A2$^{\star 3}$	**74.3**	**48.2**	**22.8**	**45.9**
	S,M,F,ANLI	**76.9**	**52.3**	44.2	**56.9**
InfoBERT (Wang et al., 2021)	S,M,F,ANLI	76.4	51.6	48.6	58.3
ALUM (Liu et al., 2020)	S,M,F,ANLI	73.3	53.4	48.2	57.7
XL-NET (Yang et al., 2019)	S,M,F,ANLI	67.6	50.7	48.3	55.1

Table 6: We show the performance of the RoBERTa model and the pre-finetuned RoBERTa-MUPPET model on the ANLI benchmark. Bolded numbers signify MUPPET vs base model, underline signifies best number. 'S' refers to SNLI, 'M' to MNLI dev (-m=matched, -mm=mismatched), and 'F' to FEVER; 'A1–A3' refer to the rounds respectively and 'ANLI' refers to A1+A2+A3.

(Clark et al., 2019), RACE (Lai et al., 2017), SQuAD (Lai et al., 2017), and MNLI (Williams et al., 2018a). We include these five datasets in the first MTL run (10 datasets) to remove any bias from adding them in a later stage.

We see a couple of interesting patterns. First, for individual tasks such as RTE (Bentivogli et al., 2009), increasing the pre-finetuning scale monotonically improves performance. This is aligned with other papers that have seen benefits from first training on MNLI (Williams et al., 2018a) and then fine-tuning on RTE (Liu et al., 2019b). For other datasets, we see that doing MTL in the < 15 datasets regime is detrimental for end-task fine-tuning. This is also aligned with other empirical observations, i.e., T5 reported that doing MTL did not improve over only fine-tuning. Nevertheless, it seems that as we increase the number of tasks past some critical point, our pre-trained representations become more generalizable. Furthermore, although dependent on the dataset, this critical point is roughly between 10 and 25 tasks.

This suggests that previously observed MTL limitations were not fundamental and can instead be attributed to the lack of sufficient scale.

5.2 Importance of Heterogenous Batches

Another critical factor to getting MTL to learn generalizable representations is the method through which MTL is implemented, specifically the selection of batches. To better quantify this trend, we experimented with three balancing schemes: dataset homogenous, batch homogenous and batch heterogenous.

We refer to dataset homogenous as selecting batches from datasets sequentially. So we first train on dataset A, then train on dataset B, etc. On the other hand, batch homogenous refers to selecting batches containing only data from the same task; therefore, all gradients are from the

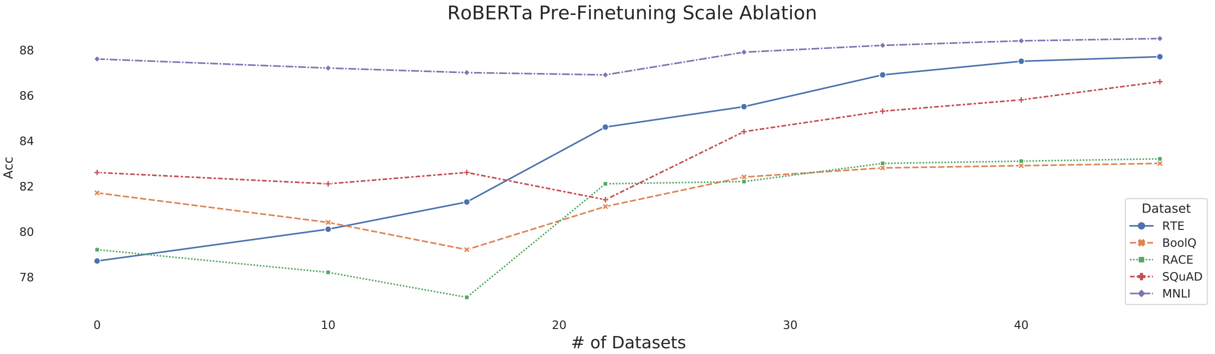

Figure 1: We plot the RoBERTa evaluation accuracy of five datasets: RTE, BoolQ, RACE, SQuAD, and MNLI, across various scales of multi-task learning measured in the number of datasets. We notice that performance initially degrades until a critical point is reached regarding the number of the datasets used by the MTL framework for all but one dataset. Post this critical point; our representations improve over the original RoBERTa model.

same dataset. This is implemented by selecting all datasets, batching on a dataset level, and selecting those same batches randomly during training. Finally, batch heterogeneous refers to a single update containing a batch from multiple different datasets spanning different tasks. We implemented this by first creating homogenous sub-batches, calculating loss per sub-batch per GPU, and then aggregating across GPUs manifesting in a gradient update that contains various datasets and, therefore, tasks.

To dissect the importance of heterogeneous batches, we train a RoBERTa-Base model on 35 randomly selected tasks using the three data selection methodologies outlined above. We then fine-tune these three models on the same five data-sets mentioned in the previous section.

We present our results in Figure 2. We see the importance of properly defining a batching strategy for effective multi-task learning. Our findings are also consistent with (Aghajanyan et al., 2020) which saw that sequential training of data-sets degrades generalizable representations.

5.3 Low Resource Experiments

We noticed in Section §4 that data-sets with smaller data-set sizes tended to improve more from MTL training. To strengthen this hypothesis, we look at two factors: the scale of pre-finetuning and the scale of fine-tuning (size of fine-tuning data-set).

We select three data-sets that were not used in pre-finetuning in Section §5.1. We also select nine partitions per fine-tuning data-set, which is sampled uniformly between 10% of the data-set and 100% of the data-set. Selecting the low-resource splits was done through random sampling.

We then fine-tune every low-resource split with every pre-finetuning checkpoint from Section §5.1. We plot the heatmaps generated from these runs in Figure 3.

Multiple patterns emerge. First, we see a clear visualization of the critical point mentioned when doing pre-finetuning. As we increase the scale of MTL, better representations are available for downstream finetuning. Furthermore, we see that pre-finetuned models at a larger scale are much more data-efficient than standard pre-trained models.

Specifically looking at the 34/40 pre-finetuning scale on Figure 3 we see that we reach higher evaluation accuracies much sooner than the base RoBERTa model (row 0).

6 Conclusion

In this work, we propose *pre-finetuning*, a stage after pre-training to further refine representations before end-task finetuning. We show that we can effectively learn more robust representations through multi-task learning (MTL) at scale. Our MTL models outperform their vanilla pre-trained counterparts across several tasks. Our analysis shows that properly scaling MTL with heterogeneous batches and loss scaling is critical to leveraging better representations. We also show a critical point regarding the number of tasks when doing multi-task learning, where fewer tasks degrade representations compared to the pre-trained model, but more tasks than this point improve representations.

We discussed a practical setting in which doing this massive multi-task learning is stable and effective through simple loss scaling and heterogeneous batches. With our method, we improve upon prior state of the art methods for RTE (Bentivogli et al., 2009) and HellaSWAG (Zellers et al.,

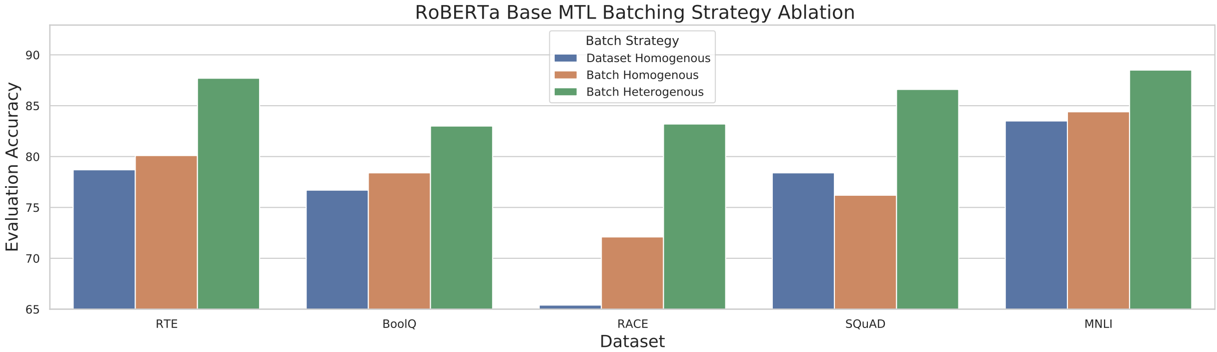

Figure 2: We plot the evaluation accuracy of RoBERTa across five datasets: RTE, BoolQ, RACE, SQuAD, and MNLI, using our three batching strategies for multi-task: Dataset Homogeneous, Batch Homogeneous, Batch Heterogeneous. The use of heterogenous batches outperforms other batching strategies by a significant margin and highlights the importance of implementing MTL with the correct batching strategy.

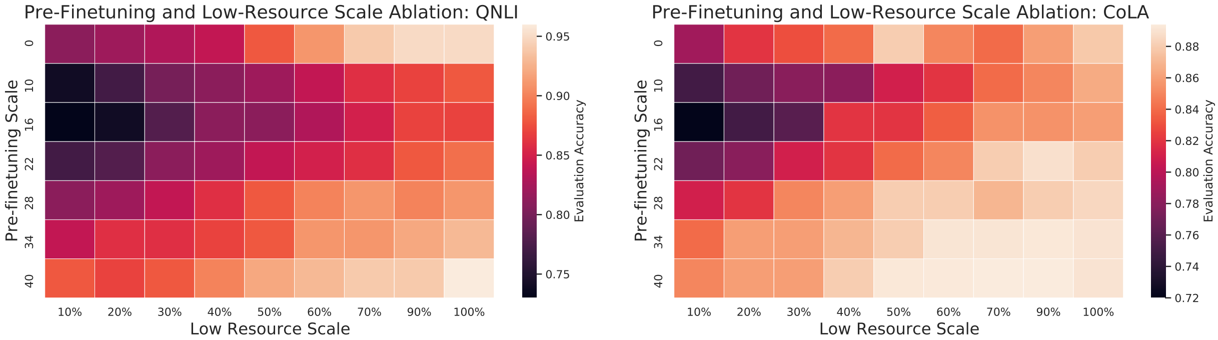

Figure 3: We fine-tune every low-resource split with every pre-finetuning checkpoint from Section §5.1 for two datasets not available in any of the pre-finetuning MTL datasets; QNLI (Rajpurkar et al., 2016b) and CoLA (Warstadt et al., 2018). The pre-finetuning scale is reported in terms of the number of datasets.

2019), as well as improve upon vanilla pre-trained representations for MNLI (Williams et al., 2018a), SQuAD (Rajpurkar et al., 2016a), BoolQ (Clark et al., 2019), and Common Sense QA (Talmor et al., 2018). We also our MTL model performance with low resource experiments. We show that on held-out datasets, leveraging representations from our pre-finetuned models with 34-40 tasks, we reach higher evaluation accuracies with much less data than the RoBERTa model.

References

Armen Aghajanyan, Akshat Shrivastava, Anchit Gupta, Naman Goyal, Luke Zettlemoyer, and Sonal Gupta. 2020. Better fine-tuning by reducing representational collapse. *arXiv preprint arXiv:2008.03156*.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International Conference on Machine Learning*, pages 794–803. PMLR.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of NAACL-HLT 2019*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The Commitment-Bank: Investigating projection in naturally occurring discourse. To appear in proceedings of Sinn und Bedeutung 23. Data can be found at https://github.com/mcdm/CommitmentBank/.

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*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. Eraser: A benchmark to evaluate rationalized nlp models.

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

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proc. of NAACL*.

Vladimir Eidelman. 2019. Billsum: A corpus for automatic summarization of us legislation. In *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, pages 48–56.

Alexander R Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir R Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. *arXiv preprint arXiv:1906.01749*.

Jun He, Liqun Wang, Liu Liu, Jiao Feng, and Hao Wu. 2019. Long document classification from local word glimpses via recurrent attention learning. *IEEE Access*, 7:40707–40718.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in neural information processing systems*, pages 1693–1701.

Eduard Hovy, Ulf Hermjakob, Chin-Yew Lin, et al. 2001. The use of external knowledge in factoid qa. In *TREC*, volume 2001, pages 644–52.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. *arXiv preprint arXiv:1909.00277*.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. 2017. First quora dataset release: Question pairs.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, page arXiv:1705.03551.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system.

Tushar Khot, Ashish Sabharwal, and Peter Clark. 2018. SciTail: A textual entailment dataset from science question answering. In *AAAI*.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. Semeval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839.

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

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

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

Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2011. The Winograd schema challenge. In *AAAI Spring Symposium: Logical Formalizations of Commonsense Reasoning*, volume 46, page 47.

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*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

X. Liu, Hao Cheng, Pengcheng He, W. Chen, Yu Wang, Hoifung Poon, and Jianfeng Gao. 2020. Adversarial training for large neural language models. *ArXiv*, abs/2004.08994.

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

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*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *CoRR*, abs/1902.01007.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2020. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. *arXiv preprint arXiv:2006.04884*.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2019. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv:1910.14599*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the ACL*.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: The word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of NAACL-HLT*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

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*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016a. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016b. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. *arXiv preprint arXiv:1611.01603*.

Eva Sharma, Chen Li, and Lu Wang. 2019. Bigpatent: A large-scale dataset for abstractive and coherent summarization. *arXiv preprint arXiv:1906.03741*.

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*, pages 1631–1642.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2015. Rethinking the inception architecture for computer vision. corr abs/1512.00567 (2015).

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint 1905.00537*.

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.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2021. Info{bert}: Improving robustness of language models from an information theoretic perspective. In *International Conference on Learning Representations*.

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

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018a. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018b. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Yang Yi, Yih Wen-tau, and Christopher Meek. 2015. WikiQA: A Challenge Dataset for Open-Domain Question Answering. page 2013–2018.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. *arXiv preprint arXiv:1808.05326*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Rui Zhang and Joel Tetreault. 2019. This email could save your life: Introducing the task of email subject line generation. In *Proceedings of The 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015a. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015b. Character-level convolutional networks for text classification. In *NIPS*.

A Appendices

A.1 Datasets Used

1. CoLA (Warstadt et al., 2018)
2. SST-2 (Socher et al., 2013)
3. MRPC (Dolan and Brockett, 2005)
4. QQP (Iyer et al., 2017)
5. MNLI (Williams et al., 2018a)
6. QNLI (Rajpurkar et al., 2016b)
7. RTE (Bentivogli et al., 2009)
8. WNLI (Levesque et al., 2012)
9. SuperGLUE (Wang et al., 2019)
10. Bool Q (Clark et al., 2019)
11. MultiRC (Khashabi et al., 2018)
12. WIC (Pilehvar and Camacho-Collados, 2019)
13. WSC (Levesque et al., 2011)
14. CB (De Marneffe et al., 2019)
15. COPA (Roemmele et al., 2011)
16. AG News (Zhang et al., 2015b)
17. IMDB (Maas et al., 2011)
18. MultiNLI (Williams et al., 2018b)
19. SNLI (Bowman et al., 2015)
20. HANS (McCoy et al., 2019)
21. Rotten Tomatoes (Pang and Lee, 2005)
22. Yelp Polarity (Zhang et al., 2015a)
23. Eraser Multi RC (DeYoung et al.)
24. Wiki QA (Yi et al., 2015)
25. Trec (Li and Roth, 2002; Hovy et al., 2001)
26. SciTail (Khot et al., 2018)
27. CNN Daily Mail (Hermann et al., 2015)
28. Billsum (Eidelman, 2019)
29. XSUM (Narayan et al., 2018)
30. Aeslc (Zhang and Tetreault, 2019)
31. Multinews (Fabbri et al., 2019)
32. Math QA (Amini et al., 2019)
33. Openbook QA (Mihaylov et al., 2018)
34. SWAG (Zellers et al., 2018)
35. HellaSWAG (Zellers et al., 2019)
36. RACE (Lai et al., 2017)
37. CommonSense QA (Talmor et al., 2018)
38. Cosmos QA (Huang et al., 2019)
39. AI2 ARC - Easy (Clark et al., 2018)
40. AI2 ARC - Challenge (Clark et al., 2018)
41. SCIQ (Welbl et al., 2017)
42. SQUAD (Rajpurkar et al., 2016a)
43. NQ (Kwiatkowski et al., 2019)
44. DROP (Dua et al., 2019)
45. RECORD (Zhang et al., 2018)
46. Hotpot (Yang et al., 2018)
47. TriviaQA (Joshi et al., 2017)

A.2 Hyperparameters

Hyper parameter	Value
Optimizer	Adam
Adam-betas	(0.9, 0.999)
Adam-eps	1e-8
LR	6e-05
LR Scheduler	polynomial decay
Dropout	0.1
Weight Decay	0.01
Warmup Updates	5000
Total Updates	200000
Max Tokens	1024

Table 7: Hyperparameters for pre-finetuning BART

Hyper parameter	Value
Optimizer	Adam
Adam-betas	(0.9, 0.999)
Adam-eps	1e-8
LR	3e-05
LR Scheduler	polynomial decay
Dropout	0.1
Weight Decay	0.01
Warmup Updates	5000
Total Updates	200000
Max Tokens	4400

Table 8: Hyperparameters for pre-finetuning RoBERTa-B and RoBERTa-L