Let the Model Decide its Curriculum for Multitask Learning

Anonymous ACL submission

Abstract

Curriculum learning strategies in prior multi-task learning approaches arrange datasets in a difficulty hierarchy either based on human perception or by exhaustively searching the optimal arrangement. However, human perception of difficulty may not always correlate well with machine interpretation leading to poor performance and exhaustive search is computationally expensive. Addressing these concerns, we propose two classes of techniques to arrange training instances into a learning curriculum based on difficulty scores computed via model-based approaches. The two classes i.e Dataset-level and Instance-level differ in the granularity of arrangement. We conduct comprehensive experiments with 12 datasets and show that instance-level and dataset-level techniques lead to an average performance improvement of 4.17% and 3.15% over their respective baseline methods. Furthermore, we find that most of this improvement comes from correctly answering the difficult instances, implying a greater efficacy of our techniques on difficult tasks.

1 Introduction

In recent times, Multi-Task Learning (MTL) (Caruana, 1997) i.e developing one *Generalist* model capable of handling multiple tasks has received significant attention from the NLP community (Aghajanyan et al., 2021; Lu et al., 2020; Sanh et al., 2019; Clark et al., 2019). Developing a single model in MTL has several advantages over multiple *Specialist* models as it (i) can leverage knowledge gained while learning other tasks and perform better in limited-data scenarios (Crammer and Mansour, 2012; Ruder et al., 2017), (ii) prevents overfitting to a single task, thus providing a regularization effect and increasing robustness (Clark et al., 2019; Evgeniou and Pontil, 2004), and (iii) provides storage and efficiency benefits because only one model needs to be maintained for all the tasks (Bingel and Søgaard, 2017).

Prior work has shown that presenting training instances ordered by difficulty level benefits not only humans but also machines (Elman, 1993; Xu et al., 2020). Arranging instances in a difficulty hierarchy i.e Curriculum Learning (easy to hard) and Anti-Curriculum Learning (hard to easy) has been studied in MTL setup (McCann et al., 2018; Pentina et al., 2015). These techniques arrange datasets either based on human perception of difficulty or by exhaustively searching the optimal arrangement. However, both these approaches have several limitations. Firstly, human perception of difficulty may not always correlate well with machine interpretation, for instance, a dataset that is easy for humans could be difficult for machines to learn or vice-versa. Secondly, exhaustive search is computationally expensive and becomes intractable as the number and size of datasets increase.

In this work, we propose two classes of techniques that enable models to form their own learning curriculum in a difficulty hierarchy. The two classes i.e Dataset-level and Instance-level differ in the granularity of arrangement. In dataset-level techniques, we arrange **datasets** based on the average difficulty score of their instances and train the model sequentially such that all the instances of a dataset are learned together. In instance-level techniques, we relax the dataset boundaries and order **instances** solely based on their difficulty scores. We leverage two model-based approaches to compute the difficulty scores (Section 2).

We experiment with 12 datasets covering various sentence pair tasks and show the efficacy of instance and dataset-level techniques with an average performance gain of 4.17% and 3.15% over their respective baseline methods. Furthermore, we analyze model predictions and find that difficult instances contribute most to this improvement implying greater effectiveness of our techniques on difficult tasks. We note that our techniques are generic and can be employed in any MTL setup.

In summary, our contributions are as follows:
(i) **Incorporating Machine Interpretation of Difficulty in MTL**: We introduce a novel framework for MTL that goes beyond human intuition of sample difficulty and provides model the flexibility to form its own curriculum at two granularities: instance-level and dataset-level.
(ii) **Performance Improvement**: We experiment with 12 varied datasets and show that instance and dataset-level techniques lead to a significant performance improvement of 4.17% and 3.15%.
(iii) **Findings and Benefits for the Community**: We conduct experiments in a limited training data regime and find that the proposed techniques are most effective on difficult instances. This finding makes our techniques more applicable for real-world tasks as they are often more difficult than abstract toy tasks and provide limited training instances. Furthermore, we analyze difficulty scores and find that approximately one-third instances of existing datasets get assigned a very low difficulty score i.e very easy-to-lean instances, hinting at presence of dataset artifacts or inherent easiness of a large portion of the datasets. These findings will help the community in developing high-quality and hard datasets.

2 Difficulty Score Computation

In this section, we describe two model-based difficulty computation methods based on recent works.

2.1 Cross Review Method

Xu et al. (2020) proposed a method that requires splitting the training dataset D into N equal meta-datasets (M_1 to M_N) and training a separate model on each meta-dataset with identical architecture. Then, each training instance is inferred using the models trained on other meta-datasets and the average prediction confidence is subtracted from 1 to get the difficulty score. Mathematically, score of instance i ($\in M_k$) is calculated as,

$$s_i = 1 - \frac{\sum_{j \in (1,...,N), j \neq k} c_{ji}}{N-1}$$

where c_{ji} is prediction confidence on instance i given by the model trained on M_j.

2.2 Average Confidence Across Epochs

In this method, the difficulty score is computed by simply averaging the prediction confidences across epochs of a single model and subtracting it from 1.

$$s_i = 1 - \frac{\sum_{j=1}^{E} c_{ji}}{E}$$

where the model is trained till E epochs and c_{ji} is prediction confidence of the correct answer given by the model at j^{th} checkpoint. This method is based on a recent work (Swayamdipta et al., 2020) that analyses the behavior of model during training i.e "training dynamics".

Algorithm 1: General Training Structure

Input:
D: the training dataset,
$\{S_1, ..., S_K\}$: splits created from D
$frac$: fraction of previous split
Initialization: Model M
for $i \leftarrow 1$ **to** K **do**
    $train\_data$ = S_i
    for $j \leftarrow 1$ **to** $i-1$ **do**
        $sampled\_S_j$ = Sampler($S_j, frac$)
        $train\_data$ += $sampled\_S_j$
    end
    Train M with $train\_data$
end
Train M with D

3 Proposed Techniques

Addressing the limitations of current approaches highlighted in Section 1, we propose two classes of techniques to arrange training instances that allow models to form the learning curriculum based on their own difficulty interpretation. The technique classes i.e Dataset-Level and Instance-Level leverage difficulty scores computed using methods described in section 2 and follow the general training structure shown in Algorithm 1. The training dataset D is divided into K splits ($S_1, ..., S_K$) based on the difficulty score, and model M is trained sequentially on these ordered splits. Furthermore, while training the model on split S_i, a fraction ($frac$) of instances from previous splits ($S_j (j < i)$) is also included in training to avoid catastrophic forgetting (Carpenter and Grossberg, 1988) i.e forgetting the previous splits while learning a new split. Note that D is a collection of multiple datasets in the MTL setup. The final step requires training on the entire dataset D as the evaluation sets often contain instances of all tasks and difficulty levels. Dataset-level and Instance level techniques vary in the way splits ($S_1, ..., S_K$) are created as described below:

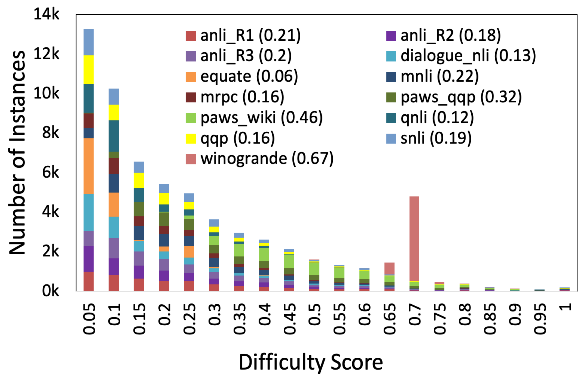


Figure 1: Distribution of instances based on difficulty score computed using Average Confidence method. Difficulty score of datasets are shown in the legends.

Dataset-level techniques: In this technique class, each **dataset** represents a split and is arranged based on the average difficulty score of its instances i.e score of a dataset D_k is calculated as:

$$d_k = \frac{\sum_{i \in D_k} s_i}{|D_k|}$$

where, s_i is the difficulty score of instance $i \in D_k$.

Instance-level techniques: Here, we relax the dataset boundaries and arrange **instances** solely based on their difficulty scores. We study two approaches of dividing instances into splits $(S_1, ..., S_K)$: Uniform and Distribution-based splitting. In the former, we create K uniform splits from D, while in the latter, we divide based on the distribution of scores such that instances with similar scores are grouped in the same split[1]. The latter approach can result in unequal split sizes as we show in Figure 3 that the number of instances varies greatly across difficulty scores.

4 Experiments

Datasets: We experiment with 12 datasets covering various sentence pair tasks, namely, Natural Language Inference (SNLI (Bowman et al., 2015), MultiNLI (Williams et al., 2018), Adversarial NLI (Nie et al., 2020)), Paraphrase Identification (QQP (Iyer et al., 2017), MRPC (Dolan and Brockett, 2005), PAWS (Zhang et al., 2019)), Commonsense Reasoning (Winogrande (Sakaguchi et al., 2020)), Question Answering NLI (QNLI (Wang et al., 2018)), Dialogue NLI (DNLI (Welleck et al., 2019)), and Numerical Reasoning (Stress Test of Equate (Ravichander et al., 2019)). For evaluation on robustness and generalization parameters, we use HANS (McCoy et al., 2019) and Stress Test (Naik et al., 2018) datasets.

Setup: We experiment in a low-resource regime limiting the number of training instances of each dataset to 5000. This enables evaluating our techniques in a fair and comprehensive manner as transformer models achieve very high accuracy when given large datasets. Furthermore, inspired by decaNLP (McCann et al., 2018), we reformulate all the tasks in our MTL setup as span identification Question Answering tasks[1]. This allows creating a single model to solve the tasks that originally have different output spaces.

Implementation Details: We use three values of $frac$: 0, 0.2, and 0.4 (refer Algorithm 1), $N = 5$ (in Cross Review method), and $E = 5$ (in Average Confidence method). For distribution-based splitting, we experiment by dividing D into 3 and 5 splits[1]. These hyper-parameters are selected based on development dataset performance.

Baseline Methods: In MTL, *heterogeneous* batching where all the datasets are combined and a batch is randomly sampled has been shown to be much more effective than *homogeneous* and *partitioned* batching strategies (Gottumukkala et al., 2020). Therefore, we use it as the baseline for instance-level techniques. For dataset-level techniques, we generate multiple dataset orders and take the average performance as the baseline. We average these baseline scores across 3 different runs.

5 Results:

Table 1 shows the efficacy of our proposed curriculum learning techniques.

Performance Improvement: Instance and Dataset-level techniques achieve an average improvement of 4.17% and 3.15% over their respective baseline methods. This improvement in consistent across all the datasets and also outperforms single-task performance in most cases. Furthermore, we find that models leveraging Average Confidence method (2.2) outperform their counterparts using the Cross Review method (2.1)[1] rendering Average Confidence approach as more effective both in terms of performance and computation as Cross Review requires training multiple models (one for each meta-dataset).

Uniform Vs Distribution based splitting: In instance-level experiments, distribution-based splitting shows slight improvement over uniform splitting. We attribute this to the superior inductive bias

[1]Refer Supplementary for details

Datasets	Single-Task		Instance-Level								Dataset-Level			
			Heterogeneous(B)		Uniform		Distribution (D)		D with $frac$=0.4		Random Order(B)		Proposed Order	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
SNLI	77.26	77.42	74.55	74.62	77.79	**77.79**	77.64	77.7	77.65	77.65	77.7	77.75	78.94	**79.05**
MNLI Mismatched	65.98	66.12	62.07	62.14	66.14	66.3	66.71	**66.78**	66.6	66.66	66.29	66.4	69.15	**69.28**
MNLI Matched	65.33	65.45	61.23	61.36	65.85	65.96	66.91	**67.01**	66.82	66.85	65.96	66.09	69.18	**69.33**
Winogrande	50	50	47.34	50	50.24	**50.27**	50	50.12	49.82	49.85	47.99	49.85	48.37	**50.3**
QNLI	74.21	74.23	66.78	66.81	70.42	70.44	71.81	**71.81**	71.38	71.38	70.35	70.39	73.75	**73.79**
EQUATE	98.99	98.99	98.99	98.99	99.14	99.21	99.57	**99.57**	99.28	99.28	99.57	99.57	99.57	**99.57**
QQP	80.04	80.06	75.34	75.35	78.89	78.9	79.23	**79.25**	79.11	79.12	79.23	79.26	80.27	**80.29**
MRPC	80.98	80.98	74.42	74.45	74.05	74.05	75.95	**75.98**	75.4	75.4	75.73	75.77	79.08	**79.08**
PAWS Wiki	52.45	52.49	55.92	56.01	53.15	53.16	54.39	54.47	70.59	**70.62**	56.44	56.51	80.33	**80.34**
PAWS QQP	68.25	68.41	73.03	73.03	69	69	71.83	71.83	78.84	**78.84**	73.08	73.12	83.46	**83.46**
ANLI R1	42.2	42.57	38.1	38.28	42.1	42.13	45.7	**45.7**	43.2	43.33	42.9	**43.04**	42.3	42.58
ANLI R2	38.1	38.78	35	35	39.8	**39.9**	38.9	39.05	37.2	37.25	38.4	**38.5**	36.8	36.97
ANLI R3	39.25	39.38	36.17	36.24	38.5	**38.62**	38.17	38.24	36.5	36.56	37.92	**38.03**	37.25	37.4
DNLI	84.68	84.83	80.36	80.48	83.51	**83.57**	83.15	83.2	82.09	82.12	82.52	82.59	82.67	**82.73**
HANS	-	-	49.06	49.07	48.95	49.01	48.3	48.38	49.39	**49.45**	48.22	**48.27**	48	48.09
Stress Test	-	-	55.28	55.44	56.2	56.31	58.66	**58.77**	57.7	57.75	56.74	56.84	59.94	**60.15**

Table 1: Results on performing curriculum learning using the proposed techniques with difficulty scores computed via Average Confidence approach. Note that $frac$ is 0 unless otherwise mentioned. B means baseline and D with $frac$=0.4 column represents Distribution based splitting with value of $frac$ as 0.4.

resulting from the collation of instances with similar difficulty scores to the same split.

Effect of adding instances from previous splits: For dataset-level techniques, we find that it does not provide any improvement. This is because all the instances of a dataset are grouped in a single split therefore, adding instances from other splits doesn't contribute much to the inductive bias. Furthermore, in the case of instance-level, it leads to a performance improvement because previous splits contain instances of the same dataset hence, providing the inductive bias.

Difficulty Scores Analysis: Figure 3 shows the distribution of training instances of all datasets with difficulty scores computed using Average confidence (2.2) method. This distribution reveals that instances across datasets and within every dataset vary greatly in difficulty as they are widely spread across the difficulty scores. Comparing the average difficulty score of all datasets (shown in legends of Figure 3) shows that Equate and QNLI are easy-to-learn while PAWS and Winogrande are relatively difficult-to-learn. Furthermore, around 32% of the training instances get assigned a difficulty score of ≤ 0.1 hinting at either the presence of dataset artifacts or the inherent easiness of these instances. A similar observation is made with Cross Review method with the percentage being 37%.

Test Set Analysis: We also compute difficulty scores of test instances and plot the performance improvement achieved by our approach over the baseline method for every difficulty score bucket in Figure 2. We find that our technique is effective especially on instances with high difficulty scores.

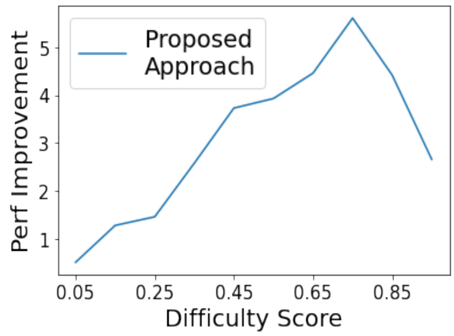


Figure 2: Performance improvement vs Difficulty score for dataset level techniques.

This implies a greater efficacy of our techniques on tasks that contain difficult instances.

6 Conclusion

In this paper, we proposed two classes of techniques for MTL that allow models to form the learning curriculum based on their own interpretation of difficulty. Comprehensive experiments with 12 datasets showed that our techniques lead to a performance improvement of 4.17% and 3.15%. Furthermore, we found that difficult instances contribute most to this improvement, implying a greater efficacy of our techniques on difficult tasks. We also analyzed the difficulty scores computed using two model-based approaches and showed that almost one-third of the training instances get assigned a score of ≤ 0.1, hinting at presence of dataset artifacts or inherent easiness of a large portion of the existing datasets. We hope that our techniques and findings will foster development of stronger MTL models and high-quality hard datasets.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain. Association for Computational Linguistics.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Gail A. Carpenter and Stephen Grossberg. 1988. The art of adaptive pattern recognition by a self-organizing neural network. *Computer*, 21(3):77–88.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

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*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

Koby Crammer and Yishay Mansour. 2012. Learning multiple tasks using shared hypotheses. *Advances in Neural Information Processing Systems*, 25:1475–1483.

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

Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Theodoros Evgeniou and Massimiliano Pontil. 2004. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117.

Ananth Gottumukkala, Dheeru Dua, Sameer Singh, and Matt Gardner. 2020. Dynamic sampling strategies for multi-task reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 920–924.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs. *data. quora. com*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2020. 12-in-1: Multi-task vision and language representation learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Aakanksha Naik, Abhilasha Ravichander, Norman Sadeh, Carolyn Rose, and Graham Neubig. 2018. Stress test evaluation for natural language inference. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2340–2353, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial NLI: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4885–4901, Online. Association for Computational Linguistics.

Anastasia Pentina, Viktoriia Sharmanska, and Christoph H Lampert. 2015. Curriculum learning of multiple tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5492–5500.

Abhilasha Ravichander, Aakanksha Naik, Carolyn Rose, and Eduard Hovy. 2019. EQUATE: A benchmark evaluation framework for quantitative reasoning in natural language inference. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 349–361, Hong Kong, China. Association for Computational Linguistics.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Sluice networks: Learning what to share between loosely related tasks. *ArXiv*, abs/1705.08142.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8732–8740.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

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

Sean Welleck, Jason Weston, Arthur Szlam, and Kyunghyun Cho. 2019. Dialogue natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3731–3741, Florence, Italy. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel 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, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase adversaries from word scrambling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1298–1308, Minneapolis, Minnesota. Association for Computational Linguistics.

Dataset	Size	Dataset	Size
SNLI	9824	MNLI	19645
Winogrande	1654	QNLI	5650
PAWS qqp	671	PAWS wiki	7987
MRPC	1630	ANLI R1	1000
ANLI R2	1000	ANLI R3	1000
DNLI	16408	HANS	30000
Equate	696	QQP	40371
Stress Test	136464		

Table 2: Statistics of our test set.

A Test set Statistics

Table 2 shows the statistics of the test sets used in our experiments.

B Implementation Details:

We use the huggingface implementation of BERT-Base model, batch size 16, learning rate $5e-5$ for our experiments. We use three values of $frac$: 0, 0.2, and 0.4 (refer Algorithm 1), $N=5$ (in Cross Review method), and $E=5$ (in Average Confidence method). For distribution based splitting, we experiment by dividing D into 3 and 5 splits. The results reported in the paper are for 3 splits. These hyper-parameters are selected based on performance on the dev dataset. We adjust the per gpu training batch size and gradient accumulation accordingly to fit in our 4 Nvidia V100 16GB GPUs. We keep the maximum sequence length of 512 for our experiments to ensure that the model uses the full context.

C Dataset Examples

Table 3 shows examples of datasets used in this work.

D Difficulty Scores

Figure 3 shows the distribution of difficulty scores computed using Cross Review and Average Confidence approach.

E Results

Table 4 shows the results of instance-level and dataset-level techniques.

F Analysis

Table 5 shows the comparison of comparison of performance across difficulty scores for instance-level approaches.

G Limitations

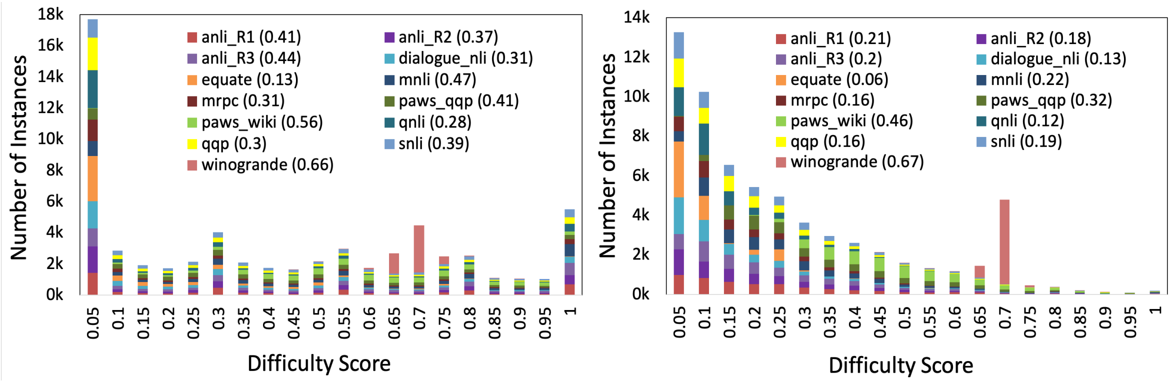
Our method involves computing the difficulty scores of training instances which requires additional computation. However, this computation is only required during training and not required during inference. Hence, it does not add any computational overhead when deployed in an application.

Context – Question	**Datasets**
C: Kyle doesn't wear leg warmers to bed, while Logan almost always does. he is more likely to live in a colder climate. **false**, or true ? **Q:** Kyle is more likely to live in a colder climate.	Winogrande
C: In order for an elevator to be legal to carry passengers in some jurisdictions it must have a solid inner door. **false**, or true ? **Q:** What is another name for a freight elevator? Does the context sentence contain answer to this question ?	QNLI
C: What makes a great problem solver? false, or **true**? **Q:** How can I be a fast problem solver? Are the two sentences semantically equivalent?	QQP, MRPC, PAWS
C: i sell miscellaneous stuff in local fairs . **contradiction**, or neutral, or entailment ? **Q:** i used to work a 9 5 job as a telemarketer . Consistency of the dialogues ?	DNLI
C: 205 total Tajima' s are currently owned by the dealership. contradiction, or neutral, **entailment** ? **Q:** less than 305 total Tajima' s are currently owned by the dealership.	Equate
C: Two collies are barking as they play on the edge of the ocean contradiction, or neutral, or **entailment** ? **Q:** Two dogs are playing together.	SNLI, MNLI, ANLI

Table 3: Examples context-question pairs of various types of training datasets used in our experiments. Answers are highlighted in bold.

Datasets	**Instance-Level** Uniform Splitting + Prev		**Dataset-Level** Proposed Order with $frac$=0.4		AC on Proposed Order	
	EM	F1	EM	F1	EM	F1
SNLI	76.19	76.2	77.09	77.11	77	77.02
MNLI Mismatched	64.54	64.55	65.83	65.85	65.36	65.41
MNLI Matched	63.63	63.64	66.06	66.08	64.72	64.77
Winogrande	50.48	50.48	50.6	50.94	48.43	49.79
QNLI	68.16	68.17	71.24	71.25	72.23	72.26
EQUATE	99.71	99.71	99.43	99.43	99.57	99.57
QQP	77.61	77.61	79.32	79.32	79.68	79.71
MRPC	72.15	72.15	76.07	76.07	77.55	77.55
PAWS Wiki	52.11	52.13	69.48	69.48	52.92	52.95
PAWS QQP	68.7	68.7	69.75	69.75	66.62	66.69
ANLI R1	41.9	41.93	43.8	43.88	44.7	44.8
ANLI R2	37.8	37.85	36.8	36.83	37.4	37.5
ANLI R3	37.58	37.62	36.5	36.53	36.83	36.83
DNLI	82.55	82.58	83.64	83.66	81.83	81.93
HANS	49.76	49.77	48.24	48.28	50.25	50.26
Stress Test	56.07	56.09	57.55	57.57	58.79	58.87
Average	62.43	62.45	64.46	64.5	63.37	63.49

Table 4: Results on test sets.



(a) Cross Review approach

(b) Average Confidence approach

Figure 3: Distribution of instances based on difficulty score.

Difficulty Score	**Instances**	**Random Order**	**Proposed Order**
0.1	63736	94.86	93.77
0.2	18703	87.8	85.55
0.3	28035	81.85	79.85
0.4	17238	74.5	72.81
0.5	21502	65.03	65.84
0.6	17338	57.69	57.94
0.7	21255	46.75	48.92
0.8	18058	38.36	44.05
0.9	22327	26.8	33.07
1	46008	9.17	14.05

Table 5: Performance comparison across difficulty scores for instance level techniques.