NT5?! Training T5 to Perform Numerical Reasoning

Anonymous ACL-IJCNLP submission

Abstract

Numerical reasoning over text (NRoT) presents unique challenges that are not well addressed by existing pre-training objectives. We explore five sequential training schedules that adapt a pre-trained T5 model for NRoT. Our final model adapted from T5 but further pre-trained on three datasets designed to strengthen skills necessary for NRoT and general reading comprehension before being fine-tuned on Discrete Reasoning over Text (DROP) dataset. We show that our training improves DROP's adjusted F1 performance (a numeracy-focused score) from 45.90 to 70.83. Our model outperforms the best model in the original DROP paper (47.01), and closes in on GenBERT (72.4), a custom BERT-Base model with significantly more parameters.

1 Introduction

Numerical Reasoning over Text (NRoT) is a reading comprehension task that involves producing an answer to numerical question given a short passage as context. Unlike reading comprehension tasks that can be solved by extracting the answer verbatim from the passage, NRoT usually involves using the question to determine the correct mathematical operation(s) while also identifying the correct values from the passage to use.

Research interest in NRoT has grown with the introduction of the Discrete Reasoning Over Paragraphs (DROP) dataset (Dua et al., 2019). The majority of DROP examples are number questions involving arithmetic, which has motivated complex models that combine symbolic and neural processing modules (Andor et al., 2019; Ran et al., 2019; Chen et al., 2020). The best performing DROP models utilize a symbolic arithmetic module in conjunction with a neural network and other techniques such as ensembling.

We demonstrate in this work that manually engineered partitioning of the functionality between distributed and symbol modules is unnecessary for achieving good performance. Rather, the recently introduced Text-to-Text Transfer Transformer (T5) (Raffel et al., 2019) model is able to internalize NRoT without adaptation. We take full advantage of the multitasking ability of T5 to introduce a sequential training pipeline that is low resource, amiable to experimental cycle, and even achieves good performance using smaller scale models.[1]

2 T5 for Numerical Reasoning over Text

We propose five training pipelines for NRoT using T5, each consisting of two stages: pre-training on NRoT and general reading comprehension followed by fine-tuning on DROP and a classification task derived from DROP (Figure 1). Multitask training is used in each stage of training. Unless specified otherwise, we validate on all the datasets in each respective stage. The first stage begins with a pretrained T5-Small model (Raffel et al., 2019). Each following stage, the model begins training on the best performing model from the previous stage. Our first two configurations (Validation Experiments 1&2) are designed to test the performance of selecting the best models using different validation data. We experimented with validating on the DROP dev set versus validating on the dev sets of the synthetic datasets (NUM/TXT), described in detail in Section 3. The next two experiments (RC Experiments 1&2) attempt to strengthen reading comprehension by multitask training using SQuAD. Finally, we attempt multitasking on all datasets simultaneously (Multitask Experiment). Based on original T5 paper, it is reasonable to hypothesize that multitasking without stages would be the best way to achieve optimal performance.

[1] All code and trained models are available on github.

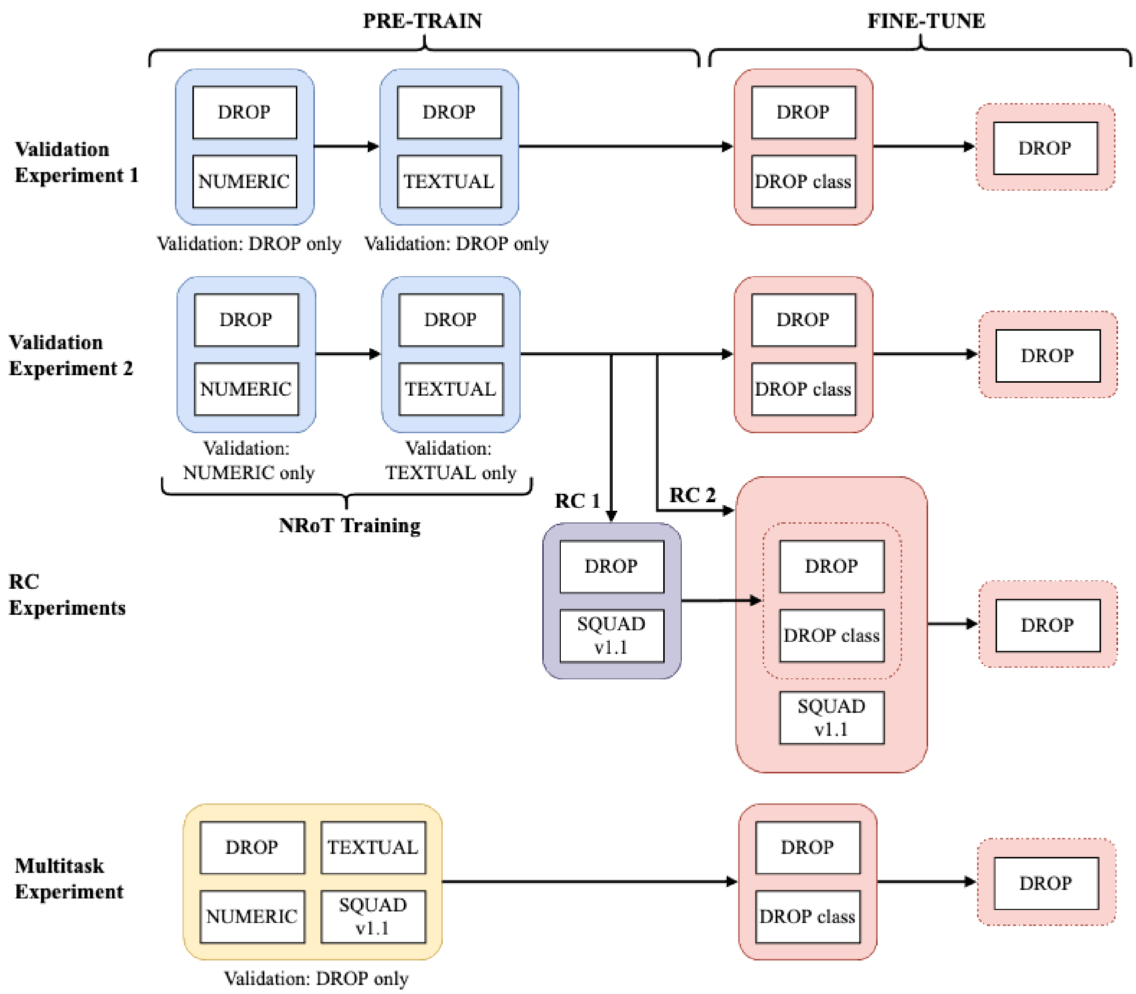

Figure 1: Summary of the five training pipelines. The validation and RC experiments are pre-trained sequentially on NUM, TXT, and SQuAD before fine-tuning on DROP and DROP classification. All experiments begin with pretrained T5-Small. RC experiment 2 moves the RC training from pre-training to fine-tuning by combining SQuAD, DROP classification, and DROP with multitasking. Multitask experiment is our attempt to multitask-train on all datasets prior to fine-tuning.

3 Datasets

DROP Discrete Reasoning Over Paragraphs (DROP), introduced by AllenNLP in 2019 (Dua et al., 2019), is a crowdsourced, adversarially-created 96k question benchmark. The benchmark consists of four types of questions, which can be answered using the context provided. Approximately 61% of the examples in DROP are number questions that involves arithmetic. The other types are “single-span” (32%), “spans” (6%), and “date” (2%). Note that all four question types in DROP can require NRoT skills, as shown in Table 1.

Synthetic Data Two synthetic datasets tailored to boost performance on DROP are developed by Geva et al. (2020). The Numeric dataset (NUM) consists of near 1M synthetically generated questions on seven types of numerical skills. Textual dataset (TXT) builds on NUM, and includes 2M plus synthetically generated examples.

We introduce an additional synthetic task based on the DROP dataset itself, whereby the model learns to predict the DROP question-type. While not provided at test time, we expect that explicit awareness of the question types will aid the model in knowing what reasoning strategies to use.

SQuAD We investigate using SQuAD v1.1 (Rajpurkar et al., 2016) to improve NRoT by strengthening general reading comprehension.

Evaluation DROP employs two metrics for evaluation: an adjusted F1, and Exact-Match (EM). EM uses that same criteria as SQuAD. F1 has additional logic that invalidates all matching material within an answer when there is a numeric mismatch. Overall F1 is computed using macro-averaging over individual answers. In the presence of multiple ground truths, both EM and F1 will take a max over all

Reasoning	Passage (shorten)	Question	Answer
Count and Sort	Denver would retake the lead with kicker Matt Prater nailing a 43-yard field goal, yet Carolina answered as kicker John Kasay ties the game with a 39-yard field goal. . . . Carolina closed out the half with Kasay nailing a 44-yard field goal. . . . In the fourth quarter, Carolina sealed the win with Kasay's 42-yard field goal.	Which kicker kicked the most field goals?	John Kasay
Subtraction	That year, his Untitled (1981), a painting of a haloed, black-headed man with a bright red skeletal body, depicted amid the artists signature scrawls, was sold by Robert Lehrman for 16.3 million, well above its 12 million high estimate.	How many more dollars was the Untitled (1981) painting sold for than the 12 million dollar estimation?	4300000
Addition	Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSK in the village of Nos Kalik, located in a pink zone near Sibenik, and captured the village at 4:45 p.m. on 2 March 1992. The JNA formed a battlegroup to counterattack the next day.	What date did the JNA form a battlegroup to counterattack after the village of Nos Kalik was captured?	3 March 1992
Other Arith˙	Although the movement initially gathered some 60,000 adherents, the subsequent establishment of the Bulgarian Exarchate reduced their number by some 75%.	How many adherents were left after the establishment of the Bulgarian Exarchate?	15000

Table 1: Examples of QA pairs found in DROP. The question types and distribution in DROP are subtraction (28.8%), comparison (18.2%), selection (19.4%), addition (11.7%), count (16.5%), sort (11.7%), coreference resolution (3.7%), other arithmetic (3.2%), set of spans (6.0%), other (6.8%). As shown, combinations of reasoning skills are possible.

computed scores.

Model	Development EM	Development F_1	Test EM	Test F_1
Baseline (T5-Small)	41.12	44.64	41.97	45.90
Validation Experiment 1	65.00	68.53	-	-
Validation Experiment 2	66.04	69.60	-	-
RC Experiment 1	**66.87**	**70.31**	**67.00**	**70.83**
RC Experiment 2	66.41	69.80	-	-
Multitask Experiment	63.10	66.47	-	-
NAQANet	46.20	49.24	44.07	47.01
GenBert	68.8	72.3	68.6	72.4
QDGAT	-	-	87.04	90.10

Table 2: Performance summary for our baseline, training experiments, and select benchmarks. NAQANet is the highest-performing model proposed in DROP's original paper. GenBERT is a modified BERT-base model fine-tuned on the same synthetic datasets. QDGAT is the current state-of-the-art.

4 Results

The overall results of our five training experiments are summarized in Table 2, and decomposed in Table 3. Our best model achieves 66.8 EM and 70.3 F1 on the dev set, and a 67.0 EM and 70.8 F1 on test. This puts it on the 21st place on the DROP leaderboard, right behind GenBERT. While we underperform QDGAT, the current state-of-the-art that makes use of both neural and symbolic modules, we perform well for a purely neural based method. Notably, our models use T5-Small with significantly fewer parameters than GenBERT. The encoder-decoder T5-small model has 60 million parameters, compared to the 110 million parameters of GenBERT in its encoder alone.

Overall, Table 3 shows that pre-training with DROP, synthetic datasets and SQuAD, and fine-tuning on DROP and DROP classification sequentially is able to significantly boost the performance on number questions, an increase of F1 from 31.83 to 70.39, while maintaining or improving performance on other types of questions. Additionally, when testing out the baseline, we found T5-Base increase F1 score over T5-Small by 11 points.

4.1 Difference in validation dataset

A surprising finding here is that saving models while validating on the synthetic dev sets outperforms saving models while validating on the DROP dev sets after the first stage. Specifically, this achieves a F1 score (50.32) that is 3.37 points higher (46.95) without sacrificing performance on span/spans questions. We reason that this performance gap is caused by the difference between the loss on development and DROP's evaluation metrics, as detailed in Section 3.

4.2 Strengthening Reading Comprehension

Performance on extractive RC tasks is boosted with the addition of SQuAD v1.1 in pre-training. We further test if this performance change persist when multitask training SQuAD v1.1 together with DROP and DROP classification in the fine-tuning stage. The resulting model sees improvement across all question types at the end of the

		Number		Date		Span		Spans		**Overall**	
Model	Schedule	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
baseline	DROP	31.79	31.83	43.95	53.28	62.09	67.42	26.98	55.44	41.12	44.64
Validation Experiment 1	DROP + NUM (validate on DROP)	36.97	36.99	43.95	51.63	59.45	64.67	27.69	55.79	43.52	46.95
	DROP + TXT (validate on DROP)	63.25	63.27	42.04	51.42	63.03	68.29	29.63	56.97	60.83	64.26
	DROP + DROP class	67.03	67.07	42.68	51.43	65.19	70.80	31.57	58.89	63.95	67.49
	DROP	68.72	68.78	43.31	50.36	65.09	70.62	32.10	60.05	65.00	68.53
Validation Experiment 2	DROP + NUM (validate on NUM)	41.37	41.38	40.76	48.94	61.31	66.45	29.81	58.73	46.86	50.32
	DROP + TXT (validate on TXT)	63.16	63.18	44.59	52.76	63.23	68.56	29.81	58.85	60.90	64.42
	DROP + DROP class	67.73	67.75	45.86	53.80	65.16	70.61	33.69	61.18	64.54	68.02
	DROP	69.78	69.83	42.68	51.23	66.00	71.47	34.22	62.53	66.04	69.60
RC Experiment 1	DROP + SQuAD⋆	65.61	65.68	45.22	55.38	65.94	71.23	34.39	62.75	63.52	67.06
	DROP + DROP class	68.65	68.69	45.22	53.87	66.54	72.01	36.68	63.47	65.71	69.17
	DROP	**70.34**	**70.39**	45.22	53.85	66.75	72.35	**37.74**	63.43	**66.87**	**70.31**
RC Experiment 2	DROP + DROP class + SQuAD⋆⋄	65.90	65.94	45.22	54.31	66.48	71.73	35.80	62.55	63.95	67.35
	DROP	69.44	69.47	45.22	53.17	**67.45**	**72.60**	35.63	63.08	66.41	69.80
Multitask Experiment	DROP + TXT + NUM + SQuAD	56.84	56.86	42.68	50.44	64.58	69.72	33.51	61.78	57.62	61.04
	DROP + DROP class	63.73	63.81	**49.04**	**56.28**	65.97	71.24	36.16	63.65	62.54	65.99
	DROP	64.43	64.49	45.86	52.99	66.48	71.61	36.51	**63.80**	63.10	66.47

Table 3: The decomposed and overall EM and F_1 scores on different answer types in the development set of DROP for each experiment. High scores for each type are in bold. ⋆Notice that the RC experiments begin training using the weights learned in validation experiment 2. ⋄RC Experiment 2 fine-tune with SQuAD in addition to DROP and DROP classification.

training on SQuAD v1.1. Crucially, performance on RC tasks (date, span, and spans) sees an average improvement of 3.06 points in F1 over the previous result. However, this came at the expense of minor deteriorated performance on numeric questions.

4.3 All datasets multitasking

Fine-tuning simultaneously on all datasets underperforms our best model by nearly 6-point on F1.

5 Error Analysis

To better understand the achievement and limitations of the best model, we analyzed its errors on the dev set. In 38 of 100 errors sampled from number questions, the model has made at least one partial digit match. Of the total 86 errors on date questions, 39 questions require arithmetic calculations. In 9 of these 86 errors, the model wrongly performs numerical calculations, instead of simply extracting answers. With a sample of 100 span and spans errors, 49 of the questions contain reasoning skills not covered in the pre-training datasets. This is compared to the 43% shown by Dua et al. (2019)[2]. Many of these errors can be addressed with pre-training datasets that cover more complicated calculations and reasoning skills.

[2]Criteria might vary due to human evaluation

6 Related Works

Introduced by (Geva et al., 2020), GenBERT is a BERT-Base model customized with specialized heads for handling discrete reasoning. It is also the main inspiration for our approach. The current state-of-the-art model on DROP, QDGAT, uses a directed graph attention network between a BERT based representation extractor and a prediction module for discrete reasoning (Chen et al., 2020). A tangential line of work exists for analyzing the mathematical reasoning ability of models over text (Wallace et al., 2019; Ravichander et al., 2019).

7 Conclusion

We introduced a sequential pre-training framework for numeracy with T5. Our method demonstrates strong improvements on NRoT over a baseline vanilla T5 model. Although current state of the art, QDGAT, which makes use of a hybrid of a neural and symbol modules, and human performance on DROP are better performing, our approach touts both simplicity and low resource usage, achieving strong performance using only T5-small.

References

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving BERT a calculator: Finding operations and arguments with reading comprehension. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5947–5952, Hong Kong, China. Association for Computational Linguistics.

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020. Question directed graph attention network for numerical reasoning over text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6759–6768, Online. Association for Computational Linguistics.

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 *North American Association for Computational Linguistics*.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Association for Computational Linguistics*, volume arXiv:2004.04487.

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. volume arXiv:1910.10683.

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*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2474–2484, Hong Kong, China. Association for Computational Linguistics.

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*.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do nlp models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.