Exploring Data Augmentation for Question Answering with T5

Omar Kapur, Phillip Ng, Amber Rashid
{omarkapur, phillipng, ajrashid}@berkeley.edu

Abstract

With the goal of improving T5's performance on difficult Question Answering (QA) tasks, we experiment with different fine-tuning approaches, including augmented datasets. Our results show that generating augmented data can provide performance improvements on difficult questions that require reasoning and computation.

1 Introduction

The Question Answering (QA) task typically involves providing a question and a context, and predicting an answer. There are variations of this task that range from extractive, in which the answer to the question (if it exists) is entirely contained within the context, to abstractive, where the model must generate an answer that is inferred from a provided context but not contained directly within the context. While extractive QA has interesting applications, abstractive QA presents a more unique challenge; to a greater degree it often invokes the need for reasoning and the application of context within language.

1.1 Objective

Our primary focus is using T5 (Text-to-Text Transfer Transformer) to perform QA on the DROP Dataset, which contains difficult reasoning-based questions which require the model to provide answers that are often not contained within the context. In a narrow search to improve T5 at the DROP dataset, we experiment with fine-tuning the model on other standardized QA datasets, as well as with generated, augmented labeled data; our goal is to find ways to improve the model's performance on difficult questions.

1.2 Model Selection

The T5 model is a encoder/decoder transformer architecture that provides a simplified text-to-text interface that allows it to seamlessly perform a variety of NLP tasks [1]. The model, accessed through HuggingFace, comes pre-trained using the Colossal Clean Crawled Corpus (C4, a dataset scraped from Common Crawl), first on unsupervised masked language modeling, then fine-tuned on a series of supervised tasks that include QA with SQuAD, summarization with CNN/Daily Mail, and a variety of tasks within GLUE, among others. While transfer learning on different tasks is a broad and fascinating area in which to experiment, we explore additional fine-tuning of T5 only in the vein of QA tasks. While at the time of this writing, the leaderboard for DROP is dominated mostly by variations of BERT, we chose T5 in part due to it's architecture and engineering convenience, but also because of the generative nature of the decoder, and the effects that training different datasets might have on it.

2 Datasets

The public datasets chosen for this experiment include DROP, SQuAD, and Hotpot-QA. These datasets were all imported from the HuggingFace Dataset library [2]. All datasets were processed into a question, context, and answer format. We use two metrics to evaluate performance: F1-span and Exact Match (EM).

2.1 DROP

The DROP (Discrete Reasoning Over Paragraphs) dataset "is a crowdsourced, adversarially-created, 96k-question benchmark, and reading comprehension benchmark that requires discrete reasoning over paragraph. It contains several types of questions which require an 'understanding' of the context in order to find the answer, many times requiring the model to: 1) resolve co-references in a question, 2) consider multiple input positions, and 3) perform mathematical operations over them (such as addition, counting, or sorting)" [3]. Each example provides a paragraph-long context and an accompanying question for the model to answer. Contexts can be re-used for questions.

Table 1: DROP Question Topics

	Training data		Validation data	
Question Topic	Count	% Total	Count	% Total
History	35,825	46.2	6,739	70.7
Sports (NFL)	41,575	53.7	2,796	29.3

Question topic was derived from the section id variable provided with the DROP dataset.

The DROP training and validation dataset questions are provided with an assigned topic of Sport or History (see Table 1). Table 1 shows a breakdown of question types along these topics. DROP contains questions with a variety of phrase beginnings (i.e. who, what, when, where, why, which and how-many,much), and three distinct answer types: number, span (i.e. text), and date (see Table 2).

Table 2: DROP Answer Types

	Training data		Validation data	
Answer Type	**Count**	**% Total**	**Count**	**% Total**
Number	46,973	60.6	5,889	61.8
Span (text)	29,195	37.7	3,503	36.7
Date	1,232	1.6	2,796	1.5

Values for answer type are provided for every record in the DROP dataset. Answers with multiple questions have multiple values for answer type, but there are no cases where a question has multiple types of gold answers.

For most questions, the dataset contains multiple acceptable answers for an individual question. However we found numerous instances where the dataset contains erroneous answers within the list of acceptable answers, along with other answers that are correct. We found this by splitting each question into tokens and checking to see if any of the answers matched the tokens within the text; we found this to be the case for 16% of questions in the DROP training data and 19% of questions in the validation data - not all of these instances mark incorrect answers, but searching through positive cases indicated that many DROP questions are not designed to be simple enough to contain the answers to the question within the context, some additional reasoning is typically required. We considered the issue to be potentially problematic as erroneous labels could falsely reward the model for incorrect predictions. To be sure how many of these answers are mislabeled, a more thorough investigation and (potentially a data fix) is required.

2.2 SQuAD

The "Stanford Question Answering Dataset (SQuAD) is a new reading comprehension dataset consisting of 100,000+ questions posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage" [4]. Because the labels are given as start-end indices of the spans, we convert the answer span to text by extracting the span directly from the passage. SQuAD is one of the datasets used to fine-tune the T5 model that we begin our experimentation with. We include it in our datasets in order to have an option that continues the fine-tuning process of T5.

2.3 HotPotQA

HotpotQA is a "dataset with 113k Wikipedia-based question-answer pairs with questions that require finding and reasoning over multiple supporting paragraphs to answer" [5]. This provides for scenarios that require for abstracting information from multiple places to form a correct answer. The HotpotQA dataset is provided with paragraphs separated into multiple strings; to format the dataset into the context, question, answer format, we concatenated all context strings together to form a single context document. The HotpotQA dataset asks about categories of answers that include people, groups, locations, dates, numbers. The most common type of question involves identifying a person (30% of the dataset). While questions with answers of "number" type constitute only 8% of the HotpotQA dataset, the entity recognition, coreference resolution, and reasoning required by this dataset makes it a reasonable contender to enhance T5's performance.

3 Experimentation

3.1 Approach

We establish our experimentation within a limited scope, adopting some approaches based on findings from the T5 paper in our fine-tuning experiments. For our loss metric, we utilize the cross-entropy loss generated internally in the T5 model. All experiments were performed using T5-small, which has approximately 60 million parameters, and using a maximum encoder sequence length of 512 and decoder sequence length of 54. We allow all parameters in the model to be trainable since experimentation results in the T5 paper ([1], table 10) show that fine -tuning with all parameters trainable provided the best results. While the T5 paper recommends the AdaFactor optimizer, we opted to use Adam out of convenience. Our pipeline uses the T5 tokenizer and adds minimal processing to the predicted string, only to remove padding and EOS tokens, and strip unnecessary white space at the beginning, end, or in between numbers of the prediction. We utilize a custom learning rate schedule that includes a linear warm-up and piece-wise decay function. In initial small-scale experiments (3-5 epochs), we explored different hyperparameters; based on our findings, we established these hyperparameters to use across each experiment.

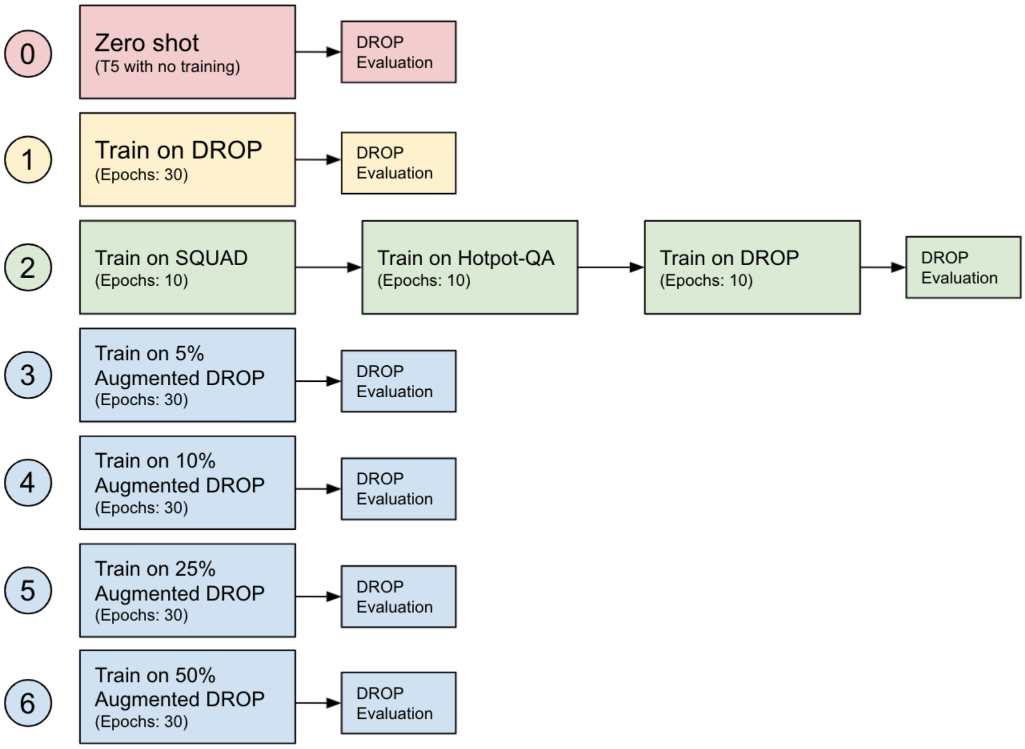

FIGURE 1: Experiment schedule; all experiments were validated using the DROP validation set. Where DROP was used for training, only the DROP training dataset was used.

The first experiment (Experiment 0) constitutes a true baseline - we utilize the pre-trained T5 model with no training on DROP, and predict on the validation set. Experiment 1 involves training the model on the DROP training data for 30 epochs, this provides us with a useful comparison to also compare against subsequent experiments. Our third experiment tests the effect of fine-tuning on two other datasets, SQuAD and Hotpot-QA, for 10 epochs each, before also training on the DROP training data for 10 epochs. We established this experiment to see how more QA training on these datasets would affect performance on DROP, a particular contrast given they contain questions that are relatively easier. The full experiment schedule is found above (see Figure 1).

After fine-tuning on standardized QA datasets, we wanted to explore the effects of augmented data on our model. In our initial exploration and small-scale experiments, we attempted to train T5 solely on the generated augmented data, as well as transfer learning experiments between the augmented data and drop. While we hoped to see interesting results, we found that the model was severely over-fitting to the generated data, and even performed worse on the evaluation set than the zero-shot experiment. The approach we settled on involves blending the augmented data with the DROP dataset and conducting Experiments 3-6 by progressively increasing the proportion of augmented data in the training dataset.

3.2 Data Augmentation

Data augmentation can help the model understand variations of the data and generalize to a broader range of inputs. One key advantage of using this technique is that it is an automated process, and can scale to records exceeding the original number of training records. Automatic data generation is particularly useful in task that require numerical reasoning, which is why DROP is a good candidate for this. Previous work has been done to generate synthetic data for training on numerical tasks on GenBERT, another sequence-to-sequence model [6]. Another study has fine-tuned T5 on this synthetic dataset, which consists of near 1M synthetically generated questions on seven types of numerical skills [7]. This research stems from attempts to solve numerical word problems using verb categorization [8]. Our data generation and augmentation process deviates somewhat from these approaches; we also focus on numerical questions, but begin by identifying targeted question types and forming question and context templates with phrase permutations. Because it is possible to modulate the context and answer for each example, there can be a template to generate examples, substituting these tokens for other tokens within a set range of values. For example, numerical reasoning questions, where we can perform mathematical operation based on the data based in the context. This allows us to create more examples based on a single example.

The DROP data-set has a unique split of questions, where 66.1% of the data requires a numerical answer [9]. Because the majority of questions are numerical in nature and easily modulated, we focus our attention in generating data in this category. In the case below, the passage provided the fact that "46.28% were Marriage living together"[1]; the answer requires taking that proportion and performing the arithmetic operation $P_{NotMarried} = 1 - P_{Married}$.

> Question: How many percent are not Marriage couples living together?
>
> Passage: There were 664,594 households out of which 24.35% had children under the age of 18 living with them, 46.28% were Marriage living together, 11.68% had a female householder with no husband present, and 37.40% were non-families. 30.11% of all households were made up of individuals and 14.70% (4.02% male and 10.68% female) had someone living alone who was 65 years of age or older. The average household size was 2.39 and the average family size was 2.97.
>
> Correct Answer: 53.72

3.3 Generating Augmented Data

In order to better differentiate the DROP dataset, we further classify questions based on the syntax of the question; note that these classifications are not mutually exclusive (see Table 3).

TABLE 3: DROP QUESTION CLASSIFICATION

Classification Type	Equation Representation	Examples
Math Easy	Y=J	Q: How many apples did Jack buy?
Math Hard	Y=J-(N+S)	Q: How many more apples did Jack buy than Nancy and Sarah combined?
Sort	Y=max(J,N,S)	Q: Who bought the most apples?
Or	Y=OR(J>N,J<N)	Q: Who bought more apples, Jack or Nancy?

Consider the question types available given the example context: *A store has 50 apples. Jack bought 15 apples, Nancy bought 10 apples, and Sarah bought 5 apples .*

If a question contains a number in the answer that is found in the context, we classify it an "Easy Math" question. If a question contains a number in the answer that is **not** found in the context, we classify it a "Hard Math" question. If a question contains any words or sub-words from a ordered (such as "first", "last", "more", "less", etc.), we classify it as a "Sort" question. If a question contained the word "or" or had a comparative task, we classify it as a "Or" type question. The table below provides a breakdown of these categories in the DROP validation dataset (See Table 4).

TABLE 4: DROP Validation Question Categories

Category	Count
Math (Hard)	5,528
Sort	3,987
Sort (Hard)	1,807
Or	1,805
Or + Sort	1,322
Math (Easy)	361
Or + Math (Hard)	279
Sort + Math (Easy)	147
Or + Sort + Math (Hard)	100
Or + Math (Easy)	19
Or + Sort + Math (Easy)	11

[1] Counts provided are for the DROP validation dataset. Question categories are not mutually exclusive, one question can count in multiple categories.

In order to generate data, we chose example DROP dataset questions that could easily morph into a Math-Easy, Math-Hard, Sort, or Or question. From there, we altered the numeric value, such as *A stores has x apples. Jack bought y apples and Nancy bought z apples*, and used a combination of filler phrases, sentences, pronouns, places, actions, and other grammatical elements to accompany different, such as "*A store* has *x object*. *person*1 bought *y object* and *person*2 bought *z object*". Our generated examples were approximately 5-sentence paragraphs that had a more complicated structure and had more combination of elements that could be switched out. We create the data based on the topics history and sports; in addition, we added the topic animals to diversify the context. Depending on the experiment, we added in different amounts of augmented data. We had shuffled the augmented data in with the original DROP training dataset (see Table 5).

TABLE 5: AUGMENTED DATA COUNT

Experiment	Count	Percentage
4	3870	5%
5	7740	10%
6	19350	25%
7	38700	50%

The huggingface Dataset DROP training dataset has 77,400 examples.

[1] DROP Dataset, Query ID: 86dd1721-6bf4-45fa-b01e-de47e4f7301d

4 Results

Our results consist of the EM and F1 scores for each of the 7 experiments run. The score are broken down to provide the overall scores, scores by answer type, and scores by questions type. A summary of the scores are found in Table 6 below (for accuracy by 1,2,3-tag questions, see Tables 9,10, and 11 in the Appendix).

Table 6: Overall Accuracy & by Answer Type

	Overall		**Date**		**Number**		**Span**	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Zero Shot	0.0996	0.1367	0.0909	0.1004	0.0436	0.056	0.1941	0.274
DROP (30)	0.4229	0.473	0.3077	0.3552	0.4062	0.4093	0.4556	0.585
Squad (10) + Hotpot (10) + DROP (10)	0.4014	0.4507	0.3427	0.3917	0.3656	0.3693	**0.4639**	**0.5898**
DROP Augmented 5% (30)	0.4216	0.4732	**0.3776**	**0.4187**	0.408	0.4111	0.4462	0.5799
DROP Augmented 10% (30)	0.4255	0.4745	0.3077	0.3567	0.4064	0.4097	0.4625	0.5881
DROP Augmented 25% (30)	0.4437	0.4923	0.3427	0.3957	0.4401	0.4432	0.4539	0.5789
DROP Augmented 50% (30)	**0.4535**	**0.5031**	0.3497	0.3678	**0.4624**	**0.466**	0.4428	0.571

F1 score is the harmonic mean of precision and recall and exact matches measure the percentage of predictions that match exactly the ground truth answers.

For overall EM and F1 score, the best performing model was the DROP Augmented 50% model, which improved the F1 score by 0.36 points over the DROP baseline model and 0.03 points over the DROP-only model. The Augmented 50% model also provided the best scores across the "Sort", "Easy Math", and "Hard Math" categories, but not the "Or" category. The DROP-only model remained the best in the "Or" category. The DROP Augmented 50% model provided the best scores for all question types that contained "Hard Math." When we look further into combinations of the "Or" question types, the augmented data models only improved the "Or + Hard Math" and "Or + Sort + Hard Math" question types. The Squad + Hotpot + DROP model performed the best for "Or + Sort" and "Or + Easy Math" questions, followed by the baseline DROP model. Neither the augmented data models nor the Squad + Hotpot + Drop model provided any improvement in F1 or EM scores for "Or + Sort + Easy Math" questions from the baseline DROP model. For scores based on answer type, Drop Augmented 5% performed the best for "Date" types, Squad + Hotpot + DROP for "Span" types, and Drop Augmented 50% for "Number" types. These can be easily compared in Figure 2.

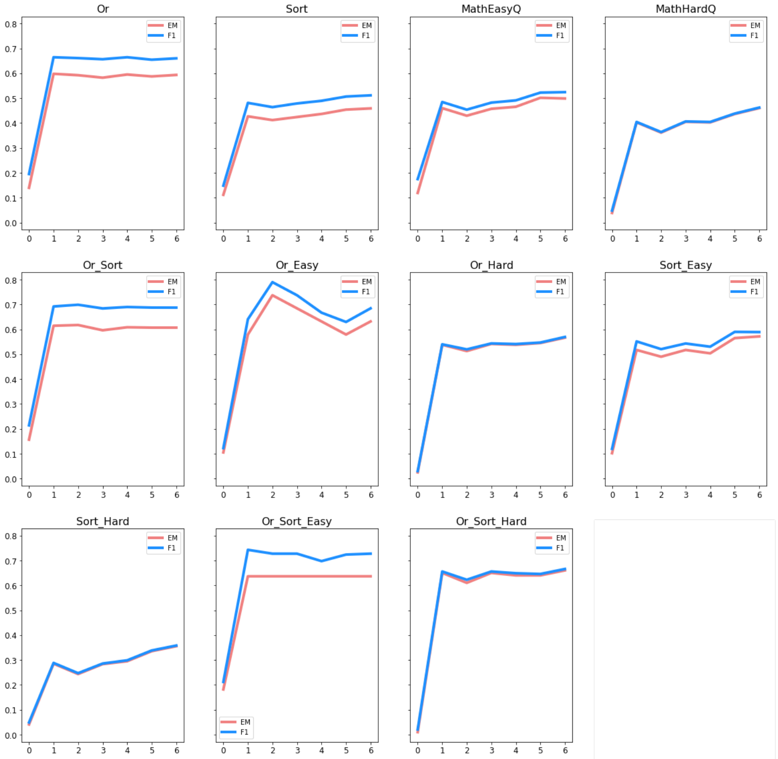

Figure 2: The F1 and EM Scores tended to increase as we progressed through each of our experiments.

5 Error Analysis

Table 7: Hotpot + Squad + DROP vs. DROP Aug50%

Context	Question	Answer	DROP Aug 50%	Hotpot + Squad + DROP
The 2010 United States Census reported that Huntington Beach had a population of 189,992. The population density was 5,959.1 people per square mile (2,300.8/km²). The racial makeup of Huntington Beach was 145,661 (76.7%) White...	Were more people Asian or from two or more races in 2010?	Asian	9	two or more races
Charles V returned to Spain on July 16, 1522. Acts of repression and retaliation against former comuneros did occur, but only sporadically. Embarrassingly large numbers of important people had supported the comuneros, or at least were suspiciously slow to declare allegiance to the king...	How many of the 293 prisoners were not executed, died in prison, purchased amnesty or were pardoned?	100	33	100

DROP Query Ids: '42988045-9e2d-4ae5-a9b2-db5de5c18f96', 'cf71e91c-4819-41ce-9283-9924d94b9354'

Interesting aspects of model behaviors can be found in two question types in particular: Sort questions and Or + Easy Math Questions.

5.1 Sort Questions

The main effects of our data augmentation can be seen in the sorting type questions as well as the hard type math questions. We found that the DROP Augmented 50% model was able to answer more questions that required multiple instances of reasoning and computation than the DROP-only model (see Table 7).

5.2 Or + Easy Questions

The stronger performance of the Squad + Hotpot + DROP model on the "Or + Easy" subset of questions, led us to dig deeper into the questions the model could and could not answer as compared to the DROP Augmented 50% model. We found that the Squad + Hotpot + DROP model performed poorly on even simple reasoning questions, but was able to parse text computational heavy questions, which the DROP Augmented 50% model could not (see Table 8).

Table 8: DROP vs DROP Aug50%

Context	Question	Answer	DROP Aug 50%	DROP
Coming off their road loss to Green Bay, Washington returned home for a duel at FedExField with the Detroit Lions, matching up with them for the first time since 2010, but the first time in DC since 2007...	After Akers 32-yard field goal, how many points behind was Washington?	9	9	19
Hoping to rebound from their loss to the Patriots, the Raiders stayed at home for a Week 16 duel with the Houston Texans. Oakland would get the early lead in the first quarter as quarterback JaMarcus Russell completed a 20-yard touchdown pass to rookie wide receiver Chaz Schilens...	How many yards longer was the longest passing touchdown than the shortest?	3	3	17

DROP Query Ids: '43810a23-d66e-49bb-816b-16f5a331fbb0', '35ee11f4-a4dd-4cd6-8907-54b034077342'

6 Conclusion

Augmented data shows promising potential for finetuning models to answer difficult questions that require reasoning and computation. Increasing data augmentation had a net positive effect for F1 and EM metrics, but had a notable impact for select question categories such as hard math questions, for which the generated dataset had been specifically targeted. The overfitting issue initially observed in our small-scale experimentation was accounted for in our experimentation plan by blending the augmented data in with the DROP dataset. This worked because the data generation approach that was taken did not provide a sufficiently diverse vocabulary on it's own to improve the model's ability to generalize. By concatenating it with the DROP dataset, we were able to increase the number of high quality, difficult questions to train on without overfitting.

While the blended data models in Experiments 3-6 showed progressive improvements at difficult questions, the SQuAD + HotpotQA + DROP model's better performance on easier questions and span questions illustrate why an augmented dataset should also be balanced. The additional training performed by that model on datasets that contained greater emphasis on non-numerical questions led it to perform slightly better. Since these question types are not the primary focus of DROP, the augmented data models were still able to outperform the SQuAD + HotpotQA + DROP model.

For future work, we recommend focusing on optimizing the data augmentation process. This should involve developing a more comprehensive augmentation approach to question and context generation that can create greater lexical diversity while retaining meaning; augmentation frameworks such as Snorkel [[10]] or Checklist [[11]] would be appropriate. Since generated data was based on hand-picked examples with grammatical structure and elements based on a set of common phrases or words, the scope of generated data is limited to the scope of the authors of each example. This modification could potentially prevent the over-fitting issue and allow for training on only generated datasets, as other synthetic data experiments have done. We also would like to experiment more with hyperparameters and architecture decisions that we kept constant due to time and resource constraints.

7 References

[1] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *CoRR* abs/1910.10683 (2019). arXiv: 1910.10683. URL: http://arxiv.org/abs/1910.10683.

[2] *HuggingFace Datasets*. URL: https://huggingface.co/docs/datasets/.

[3] URL: https://allennlp.org/drop.

[4] Pranav Rajpurkar et al. "SQuAD: 100, 000+ Questions for Machine Comprehension of Text". In: *CoRR* abs/1606.05250 (2016). arXiv: 1606.05250. URL: http://arxiv.org/abs/1606.05250.

[5] Zhilin Yang et al. "HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering". In: *CoRR* abs/1809.09600 (2018). arXiv: 1809.09600. URL: http://arxiv.org/abs/1809.09600.

[6] Mor Geva, Ankit Gupta, and Jonathan Berant. "Injecting Numerical Reasoning Skills into Language Models". In: *CoRR* abs/2004.04487 (2020). arXiv: 2004.04487. URL: https://arxiv.org/abs/2004.04487.

[7] Peng-Jian Yang et al. "NT5?! Training T5 to Perform Numerical Reasoning". In: *CoRR* abs/2104.07307 (2021). arXiv: 2104.07307. URL: https://arxiv.org/abs/2104.07307.

[8] Mohammad Javad Hosseini et al. "Learning to Solve Arithmetic Word Problems with Verb Categorization". In: (Oct. 2014), pp. 523–533. DOI: 10.3115/v1/D14-1058. URL: https://aclanthology.org/D14-1058.

[9] Dheeru Dua et al. "DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs". In: *CoRR* abs/1903.00161 (2019). arXiv: 1903.00161. URL: http://arxiv.org/abs/1903.00161.

[10] Alexander Ratner et al. "Snorkel: Rapid Training Data Creation with Weak Supervision". In: *Proc. VLDB Endow.* 11.3 (Nov. 2017), pp. 269–282. ISSN: 2150-8097. DOI: 10.14778/3157794.3157797. URL: https://doi.org/10.14778/3157794.3157797.

[11] Marco Túlio Ribeiro et al. "Beyond Accuracy: Behavioral Testing of NLP models with CheckList". In: *CoRR* abs/2005.04118 (2020). arXiv: 2005.04118. URL: https://arxiv.org/abs/2005.04118.

8 Appendix

Table 9: Accuracy by 1-tag questions

	Or		**Sort**		**Easy Math**		**Hard Math**	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Zero Shot	0.1396	0.1951	0.1116	0.1483	0.1191	0.1747	0.0387	0.0482
DROP (30)	**0.5978**	**0.6644**	0.4269	0.481	0.4598	0.4845	0.4027	0.4044
Squad (10) + Hotpot (10) + DROP (10)	0.5922	0.6609	0.4118	0.4639	0.4294	0.4537	0.3614	0.3638
DROP Augmented 5% (30)	0.5823	0.6564	0.4241	0.4787	0.4571	0.4819	0.4048	0.4065
DROP Augmented 10% (30)	0.595	0.6644	0.4364	0.4893	0.4654	0.491	0.4025	0.4044
DROP Augmented 25% (30)	0.5873	0.6545	0.454	0.5064	**0.5014**	0.5222	0.4361	0.438
DROP Augmented 50% (30)	0.5934	0.6599	**0.4587**	**0.5114**	0.4986	**0.524**	**0.46**	**0.4622**

F1 score is the harmonic mean of precision and recall and exact matches measure the percentage of predictions that match exactly the ground truth answers.

Table 10: Accuracy by 2-tag Questions

	Or + Sort		**Or + Easy Math**		**Or + Hard Math**		**Sort + Easy Math**		**Sort + Hard Math**	
	EM	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**	**EM**	**F1**
Zero Shot	0.1566	0.214	0.1053	0.1226	0.0251	0.03	0.102	0.1197	0.041	0.049
DROP (30)	0.6142	0.6918	0.5789	0.6405	0.5376	0.5397	0.517	0.5514	0.285	0.2882
Squad (10) + Hotpot (10) + DROP (10)	**0.6172**	**0.6987**	**0.7368**	**0.7895**	0.5125	0.5194	0.4898	0.5203	0.2435	0.2471
DROP Augmented 5% (30)	0.5961	0.6838	0.6842	0.7368	0.5412	0.5433	0.517	0.5429	0.2833	0.2858
DROP Augmented 10% (30)	0.6082	0.6897	0.6316	0.6668	0.5376	0.5407	0.5034	0.5301	0.295	0.2987
DROP Augmented 25% (30)	0.6067	0.6872	0.5789	0.6295	0.5448	0.5468	0.5646	**0.5896**	0.3354	0.3383
DROP Augmented 50% (30)	0.6067	0.6871	0.6316	0.6842	**0.5663**	**0.5692**	**0.5714**	0.5888	**0.3553**	**0.3581**

F1 score is the harmonic mean of precision and recall and exact matches measure the percentage of predictions that match exactly the ground truth answers.

Table 11: Accuracy by 3-tag Questions

	Or + Sort + Easy Math		**Or + Sort + Hard Math**	
	EM	**F1**	**EM**	**F1**
Zero Shot	0.1818	0.2118	0.01	0.0207
DROP (30)	**0.6364**	**0.7427**	0.65	0.6557
Squad (10) + Hotpot (10) + DROP (10)	**0.6364**	0.7273	0.61	0.6224
DROP Augmented 5% (30)	**0.6364**	0.7273	0.65	0.6557
DROP Augmented 10% (30)	**0.6364**	0.6973	0.64	0.6486
DROP Augmented 25% (30)	**0.6364**	0.7236	0.64	0.6457
DROP Augmented 50% (30)	**0.6364**	0.7273	**0.66**	**0.6657**

F1 score is the harmonic mean of precision and recall and exact matches measure the percentage of predictions that match exactly the ground truth answers.