MATHWELL: Generating Educational Math Word Problems at Scale

Bryan R. Christ and Jonathan Kropko and Thomas Hartvigsen
brc4cb@virginia.edu, jk8sd@virginia.edu, zcy9jk@virginia.edu[1]

[1]University of Virginia School of Data Science

1 Introduction

Math word problems are a critical part of assessing student learning in K-12 education (Daroczy et al., 2015; Pearce et al., 2013; Schwartz, 2023; Verschaffel et al., 2020). Customizing word problems to student interests can further increase learning, interest in math, and test performance (Bernacki and Walkington, 2018; Walkington, 2013). However, with many students and responsibilities, teachers rarely have time to customize questions for their students. We aim to automatically generate customized math word problems and answers at scale to facilitate fast, interest-guided, math education.

Some previous work has explored generating math word problems with LLMs (Niyarepola et al., 2022; Zong and Krishnamachari, 2023), but not generating question/answer pairs. These approaches also require reference problems (Zong and Krishnamachari, 2023) or the beginning of a word problem (Niyarepola et al., 2022) to guide their output. Other studies use LLMs or deep neural networks to generate math word problems relying on pre-specified equations or reference problems (Jiao et al., 2023; Koncel-Kedziorski et al., 2016; Norberg et al., 2023; Qin et al., 2023; Wang et al., 2021; Zhou and Huang, 2019; Zhou et al., 2023). Therefore, existing methods are *context-dependent*, simply rephrasing or creating new output based on input problems or equations. In practice, curating and choosing specific word problems or equations beforehand is laborious and difficult (Roche, 2013).

To address this limitation, we propose *context-free* educational math word problem generation whereby our model, MATHWELL, generates a grade school (K-8) question/answer pair based solely on a desired student interest. For context-free word problem generators to be effective educational tools, we propose three criteria to evaluate their outputs. 1) Each problem must be *solvable*. 2) Each question's corresponding answer must be *accurate*. 3) Each problem must be *appropriate*, where the question/answer should make logical sense, the math topic should be familiar to a student, and the question's context should be appropriate for a young learner.

To create MATHWELL, we first finetune Llama-2 (70B) (Touvron et al., 2023) on existing math QA data with code-based solutions. Next, we generate synthetic data from this model that have solutions in the form of Python functions and domain experts annotate that data based on our three proposed evaluation metrics of solvability, accuracy and appropriateness to identify high-quality generations. Finally, we further finetune our model on the high-quality generations to create MATHWELL.

We find that not only is MATHWELL effective at context-free word problem generation, with 74% of its question/answer pairs with executable code identified as meeting our evaluation criteria, but also data generated from this model is of high quality for training context-free word problem generators and is comparable to human-written math QA datasets. We release our model, data, and annotations.[1] Our work has the following contributions:

- We release MATHWELL, a context-free word problem generator, and show it is an effective and simpler alternative to traditional context-dependent word problem generators.
- We release the Synthetic Grade School Math (SGSM) dataset and show it is high quality through automatic evaluation metrics. SGSM is the only existing dataset designed to train context-free word problem generators and is the largest English math grade school dataset with code-based solutions.

2 Methods

We re-format MathInstruct GSM8K (Yue et al., 2023) into a dataset of question/answer pairs and

[1] https://github.com/bryanchrist/MATHWELL

Model	Solv.	Acc.	App.	MaC	Top. Spec.	EC	EC/MaC
LLEMMA	48.8 (3.17)	63.9 (4.37)	41.8 (4.48)	15.2 (2.28)	94.8 (1.41)	24.3 (0.70)	3.70 (0.55)
MAmmoTH	86.8 (2.15)	94.9 (1.49)	67.7 (3.18)	56.8 (3.14)	97.6 (0.97)	6.90 (0.36)	3.91 (0.22)
Llama-2	84.0 (2.32)	89.5 (2.12)	81.0 (2.72)	62.4 (3.07)	99.2 (0.56)	55.4 (0.98)	34.6 (1.70)
MATHWELL	**89.2 (1.97)**	**96.9 (1.17)**	**86.5 (2.29)**	**74.8* (2.75)**	**99.6 (0.40)**	**66.4* (1.00)**	**49.6* (1.83)**

Table 1: Average metrics for each model based on 250 generations for human annotated criteria and over 2,000 for assessing the share of questions with executable code (EC). EC/MaC is the estimated share of questions that MaC and have executable code. Bold indicates the best performance in each metric, while a * indicates the difference between the best performance and second best performance is statistically significant at the p<.01 level. Standard errors are in parentheses.

conduct QLoRA finetuning (Dettmers et al., 2023) on Llama-2 70B (Touvron et al., 2023) for 4,250 steps. Because there is no existing age-appropriate math QA dataset with Python function solutions, which we find important for training context-free generators (see Appendix A), we few-shot prompt our finetuned model to generate synthetic data with Python function solutions, resulting in 3,234 question/answer pairs. Domain experts annotate the data for solvability, accuracy, and appropriateness (annotation details are in Appendix H). We denote question/answer pairs that meet each of the three criteria as meeting all criteria (MaC). We use the 1,905 MaC generations to continue finetuning for 1,250 steps to create MATHWELL, which is inspired by recent works that iteratively finetune LLMs (Guo et al., 2024; Wang et al., 2024).

To promote further research on context-free word problem generators, we release SGSM, a dataset of 20,490 question/answer pairs generated by MATHWELL and finetuned Llama-2 consisting of two subsets: SGSM Train (2,093 MaC generations) and SGSM Unannotated (18,397 generations that have executable code but are not labeled). SGSM is the largest available English grade school math QA dataset with code solutions and the only dataset designed specifically to train context-free word problem generators (see Appendix A).

3 Experiments

To evaluate MATHWELL, we sample 250 generations from MATHWELL, LLEMMA (34B) (Azerbayev et al., 2023), MAmmoTH (70B) (Yue et al., 2023), and Llama-2 (70B) (Touvron et al., 2023). We prompt each model using example question/answer pairs from SGSM Train. Domain experts then annotate these data for solvability, accuracy, appropriateness, and topic specificity (e.g., if the question includes the randomly selected topic in the prompt). As shown in Table 1, MATHWELL is the best performing model in each metric of evaluation, with the largest differences being in the share of generations that MaC, have executable code, and have executable code and MaC (see Appendix B for more evaluations). We also automatically evaluate both SGSM and each model's generations using perplexity (PPL), average question length, Flesch-Kincaid grade level (FKGL) (Flesch, 1948; Kincaid et al., 1975), New Dale-Chall (NDC) readability score (Chall and Dale, 1995) and BERTScore (Zhang et al., 2020). Across each metric, our synthetic data is similar to or better than existing human-written datasets, suggesting it is high quality (see Appendix C). Specifically, MATHWELL and SGSM outperform other models and datasets, respectively, in PPL and in generating questions written at an appropriate reading level.

4 Conclusions

We explore context-free word problem generation and create MATHWELL, which generates a question/answer pair based only on an optional topic. To train our model, we generate synthetic data and use expert annotators to identify a high-quality training subset. We release SGSM, a synthetic dataset of 20,490 question/answer pairs for use in future research. Our evaluations show that MATHWELL outperforms other open-source LLMs at context-free word problem generation and that SGSM is of comparable quality to existing math QA data. These findings suggest that context-free word problem generation is a feasible and practical alternative to traditional context-dependent generators. Future research should train context-free word problem generators that can create questions aligned with specific math topics and grade levels.

Limitations

One important limitation of MATHWELL is that it is not designed to generate questions aligned with pre-specified grade levels and mathematical operations/topics, which we chose to leave to future research due to the high cost of annotating questions for these characteristics. For context-free word problem generation models to be most useful in classroom settings, future research should consider how to guide generations to be specific to different grade levels and math operations/topics. Additionally, MATHWELL is trained and evaluated for generating word problems/solutions for K-8 students only; therefore, we do not recommend using it to generate question/answer pairs for other grade levels or for other tasks.

Another limitation of this work is the subjective nature of the appropriateness criteria. While it is critical model-generated questions are appropriate for students, it is hard to fully define all aspects of appropriateness and individuals may have differing opinions on the degree to which a question is appropriate or not. We chose to define several common reasons questions may not be appropriate for students (see Figure 6) and use annotators with K-12 teaching experience/training and who are familiar with what is appropriate in a school setting, but future research should continue to define this criteria and include multiple evaluators.

Ethics Statement

All data used to train MATHWELL come from open-access datasets and, therefore, should not contain any private sensitive information. MATHWELL may generate questions that are inappropriate for use in educational contexts and additional research should be conducted on the model before deploying it in classroom settings. Specifically, future research should continue to improve performance of text classifiers to filter out questions which are not appropriate for students.

Acknowledgements

The authors thank Zooniverse (Zooniverse) for providing a free and user-friendly platform for data annotation as well as our volunteer annotators for providing high-quality labels and feedback on our evaluation criteria/directions.

References

Shivam Bansal Aggarwal, Chaitanya. textstat: Calculate statistical features from text.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An Open Language Model For Mathematics. ArXiv:2310.10631 [cs].

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. ArXiv:2204.05862 [cs].

Matthew L. Bernacki and Candace Walkington. 2018. The role of situational interest in personalized learning. *Journal of Educational Psychology*, 110(6):864–881. Place: US Publisher: American Psychological Association.

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability Revisited: The New Dale-Chall Readability Formula*. Brookline Books. Google-Books-ID: 2nbuAAAAMAAJ.

Gabriella Daroczy, Magdalena Wolska, Walt Detmar Meurers, and Hans-Christoph Nuerk. 2015. Word problems: a review of linguistic and numerical factors contributing to their difficulty. *Frontiers in Psychology*, 6:348.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. ArXiv:2305.14314 [cs].

R. Flesch. 1948. A new readability yardstick. *The Journal of Applied Psychology*, 32(3):221–233.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. PAL: Program-aided Language Models. ArXiv:2211.10435 [cs].

Hongyi Guo, Yuanshun Yao, Wei Shen, Jiaheng Wei, Xiaoying Zhang, Zhaoran Wang, and Yang Liu. 2024. Human-instruction-free llm self-alignment with limited samples.

Ying Jiao, Kumar Shridhar, Peng Cui, Wangchunshu Zhou, and Mrinmaya Sachan. 2023. Automatic Educational Question Generation with Difficulty Level Controls. In *Artificial Intelligence in Education*, Lecture Notes in Computer Science, pages 476–488, Cham. Springer Nature Switzerland.

J. P. Kincaid, Jr. Fishburne, Rogers Robert P., Chissom Richard L., and Brad S. 1975. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel:. Technical report, Defense Technical Information Center, Fort Belvoir, VA.

Rik Koncel-Kedziorski, Ioannis Konstas, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2016. A theme-rewriting approach for generating algebra word problems. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1617–1628, Austin, Texas. Association for Computational Linguistics.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A Diverse Corpus for Evaluating and Developing English Math Word Problem Solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022. LILA: A Unified Benchmark for Mathematical Reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5807–5832, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kashyapa Niyarepola, Dineth Athapaththu, Savindu Ekanayake, and Surangika Ranathunga. 2022. Math Word Problem Generation with Multilingual Language Models. In *Proceedings of the 15th International Conference on Natural Language Generation*, pages 144–155, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Kole Norberg, Husni Almoubayyed, Stephen E Fancsali, Logan De Ley, Kyle Weldon, April Murphy, and Steve Ritter. 2023. Rewriting math word problems with large language models. *Proceedings of the Workshop on Empowering Education with LLMs-the Next-Gen Interface and Content Generation 2023 co-located with 24th International Conference on Artificial Intelligence in Education (AIED 2023)*, 3487:163–172.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. ArXiv:2203.02155 [cs].

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP Models really able to Solve Simple Math Word Problems? ArXiv:2103.07191 [cs].

Daniel Pearce, Faye Bruun, Kim Skinner, and Claricia Lopez-Mohler. 2013. What teachers say about student difficulties solving mathematical word problems in grades 2-5. *International Electronic Journal of Mathematics Education*, 8:3–19.

Longhu Qin, Jiayu Liu, Zhenya Huang, Kai Zhang, Qi Liu, Binbin Jin, and Enhong Chen. 2023. A Mathematical Word Problem Generator with Structure Planning and Knowledge Enhancement. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 1750–1754, New York, NY, USA. Association for Computing Machinery.

Anne Roche. 2013. Choosing, creating and using story problems: Some helpful hints. *Australian Primary Mathematics Classroom*, 18(1):30–35.

Sarah Schwartz. 2023. Why Word Problems Are Such a Struggle for Students—And What Teachers Can Do. *Education Week*.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback. ArXiv:2009.01325 [cs].

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lieven Verschaffel, Stanislaw Schukajlow, Jon Star, and Wim Van Dooren. 2020. Word Problems in Mathematics Education: A Survey. *ZDM: The International Journal on Mathematics Education*, 52(1):1–16. Publisher: Springer ERIC Number: EJ1243930.

Candace A. Walkington. 2013. Using adaptive learning technologies to personalize instruction to student interests: The impact of relevant contexts on performance and learning outcomes. *Journal of Educational Psychology*, 105(4):932–945. Place: US Publisher: American Psychological Association.

Haoyu Wang, Guozheng Ma, Ziqiao Meng, Zeyu Qin, Li Shen, Zhong Zhang, Bingzhe Wu, Liu Liu, Yatao

Bian, Tingyang Xu, Xueqian Wang, and Peilin Zhao. 2024. Step-on-feet tuning: Scaling self-alignment of llms via bootstrapping.

Zichao Wang, Andrew S. Lan, and Richard G. Baraniuk. 2021. Math Word Problem Generation with Mathematical Consistency and Problem Context Constraints. ArXiv:2109.04546 [cs].

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. MAmmoTH: Building Math Generalist Models through Hybrid Instruction Tuning. ArXiv:2309.05653 [cs].

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Qingyu Zhou and Danqing Huang. 2019. Towards Generating Math Word Problems from Equations and Topics. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 494–503, Tokyo, Japan. Association for Computational Linguistics.

Zihao Zhou, Maizhen Ning, Qiufeng Wang, Jie Yao, Wei Wang, Xiaowei Huang, and Kaizhu Huang. 2023. Learning by analogy: Diverse questions generation in math word problem.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-Tuning Language Models from Human Preferences. ArXiv:1909.08593 [cs, stat].

Mingyu Zong and Bhaskar Krishnamachari. 2023. Solving Math Word Problems concerning Systems of Equations with GPT-3. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(13):15972–15979. Number: 13.

Zooniverse. Zooniverse.

A SGSM Dataset Characteristics

The characteristics we find most important for training context-free word problem generators are PoT solutions written as Python functions and questions with appropriate number ranges for K-8 students. Regarding the former, when we modified our prompt to ask for a Python function solution instead of a Python code solution, the percentage of question/answer pairs with executable code from an early version of MATHWELL increased from 18.9% to 29.0%. Regarding the second characteristic, when we used the GSM-Hard (Gao et al., 2023) dataset as part of MATHWELL's training data, we observed that the model often generated questions with large numbers that are inappropriate for K-8 students. As shown in the Table 2, the SGSM Train subset is the only math QA dataset that has these two characteristics.

Prior work (e.g., Jiao et al. 2023; Norberg et al. 2023; Niyarepola et al. 2022; Zhou and Huang 2019; Zhou et al. 2023) evaluates existing math datasets and/or model-generated word problems based on a measure of question length, Flesch-Kincaid grade level (FKGL) (Flesch, 1948; Kincaid et al., 1975), New Dale-Chall (NDC) readability score (Chall and Dale, 1995) and/or BERTScore (Zhang et al., 2020), so we report these metrics for our synthetic data. As shown in Table 2, while SGSM and its subsets have a shorter average token length than MathInstruct GSM8K (Yue et al., 2023), their token lengths are longer than ASDIV (Miao et al., 2020) and SVAMP (Patel et al., 2021), two other grade school math data sets. This suggests that SGSM's question complexity, as reflected in its average question length, should be similar to other existing datasets. SGSM and its subsets have BERTScores close to those of other existing datasets, suggesting the questions are similar. SGSM and its subsets have the lowest average FKGL and comparable NDC, providing evidence that the questions may be more appropriate for those who struggle to read.

B Additional Human Evaluation

Word problems involving multiplication, division, fractions, and decimals are more complex than those only involving addition and subtraction. In Table 3, we assess whether each model can generate MaC questions when using more complicated operations. Questions may contain more than one operation/topic. Solvable questions may require no math operation if they contain the answer in the question (see Appendix F.4.3), so we also report the share of questions containing no operations. To determine if questions with complex operations are accurate and appropriate, we compare the math operations/topics in solvable questions to those in MaC questions. We also report the average number of distinct operations/topics in MaC questions for each model, which is another way to assess question complexity.

Table 3 shows that MATHWELL is the only model for which the share of MaC questions for each math operation/topic is within two percentage points of that for solvable questions, providing evidence that MATHWELL can generate MaC questions regardless of the complexity of the operation.

Dataset	N	PoT	PF	AD	AL	FKGL	NDC	BF1
GSM-Hard (Gao et al., 2023)	1,319	✓	✓	✗	72.9 (25.6)	4.21 (2.43)	8.20 (1.13)	84.0
MathInstruct GSM8K (Yue et al., 2023)	6,403	✓	✗	✓	66.2 (23.9)	4.25 (2.48)	8.17 (1.13)	84.6
NumGLUE (Mishra et al., 2022)	12,403	✓	✗	✗	144.8 (136.5)	10.04 (6.99)	10.27 (1.51)	81.5
ASDIV (Miao et al., 2020)	2,305	✗	✗	✓	45.1 (15.8)	3.56 (2.40)	7.85 (1.48)	85.5
SVAMP (Patel et al., 2021)	1,000	✗	✗	✓	47.3 (11.7)	3.39 (2.07)	7.84 (1.09)	86.1
SGSM (Ours)	20,490	✓	✓	?	62.0 (15.0)	2.68 (1.97)	7.99 (1.26)	84.8
$SGSM_{Train}$	2,093	✓	✓	✓	57.2 (15.7)	2.50 (1.76)	8.12 (1.25)	85.2
$SGSM_{Unannotated}$	18,397	✓	✓	?	62.5 (14.8)	2.70 (1.99)	7.97 (1.26)	84.9

Table 2: Characteristics of datasets with more than 1,000 examples that can be used to train context-free word problem generators. N is the deduplicated number of questions, PF is Python function solution, AD is appropriate difficulty, AL is average length (in tokens), FKGL is Flesch-Kincaid grade level, NDC is New Dale-Chall readability, and BF1 is BERTScore F1. A "?" denotes we cannot verify whether all questions are written at an appropriate difficulty due to the dataset being unannotated. Standard deviations, where applicable, are in parentheses.

	Solvable Questions							MaC Questions						
Model	Add.	Sub.	Mult.	Div.	Frac.	Dec.	No Ops	Add.	Sub.	Mult.	Div.	Frac.	Dec.	Total Ops
LLEMMA	34.4	27.0	33.6	20.5	6.56	15.6	15.6	36.8	39.5	31.6	15.8	2.63	13.2	1.39
MAmmoTH	39.6	37.8	43.8	19.4	3.69	10.6	2.30	43.0	42.2	40.8	16.9	4.93	9.86	1.58
Llama-2	57.6	58.6	22.9	14.3	8.10	11.4	4.76	59.6	60.3	24.4	12.8	5.77	8.97	1.72
MATHWELL	69.5	69.1	24.7	10.3	5.38	7.62	1.35	71.1	70.6	24.6	8.56	4.81	7.49	1.87

Table 3: Characteristics of model-generated questions. Add., Sub., Div., Frac., Dec., No Ops, Total Ops, and MaC are addition, subtraction, division, fractions, decimals, no operations, total operations, and meets all criteria, respectively. All columns are percentages except total ops, or the average number of distinct operations per question.

MATHWELL is also the least likely to generate problems that require no operations and has the highest average total operations, two other pieces of evidence that suggest MATHWELL generates high-quality, complex problems.

In our evaluation, we do not prompt models for specific operations. Under these conditions, MATHWELL generates more problems containing addition and subtraction relative to the other models. In turn, there is a concern that MATHWELL's performance in Table 1 could be due to it generating simple questions for this experiment, which may be more likely to MaC. To address this concern, we conduct two additional analyses reported in Appendix E: 1) logistic regressions showing MATHWELL's higher MaC relative to the other models holds when controlling for math operations and 2) a summary of accuracy by operation showing MATHWELL is the only model for which accuracy does not substantially differ by operation and remains above 90% for each operation.

C Automatic Evaluation

C.1 MATHWELL and SGSM Are Similar to Human-written Data

Like Jiao et al. (2023) and Zhou et al. (2023), we use BERTScore (Zhang et al., 2020) to compute the semantic similarity of questions generated from each model and compare it to existing datasets. A lower BERTScore for a model's questions relative to existing datasets would signal they are less similar to each other than word problems in human-written datasets, while a higher score would suggest that they are more similar. We also compute the BERTScore between all and MaC questions from each model to determine if they are similar. We use BERTScore to compare the SGSM subsets and each model's generations to MathInstruct GSM8K to identify whether they are similar to high-quality, human-written questions. As shown in Table 4, across models, SGSM subsets and comparisons, BERTScores are similar, suggesting the questions we generate and the data we release are similar to human quality.

C.2 MATHWELL and SGSM Have Low Perplexity

Perplexity (PPL), a metric Jiao et al. (2023) also use, is another way to automatically measure the quality of outputs from LLMs, with a lower PPL representing outputs the LLM considers more probable. We calculate PPL using Llama-2 (70B). As shown in Table 4, the SGSM subsets have the lowest PPL of all data sources considered, suggesting the datasets are high quality. MATHWELL's outputs have the lowest PPL among the models considered and lower PPL than MathInstruct GSM8K, providing evidence the model generates high-quality outputs.

C.3 MATHWELL Produces Longer MaC Questions

Longer average token length may signal increased question complexity, as longer word problems often contain more information and mathematical operations than shorter word problems. Comparing the average length of all questions to the length of MaC questions can determine whether MaC questions are shorter or simpler. As shown in Table 4, although MATHWELL's average token length for all questions is slightly shorter than LLEMMA's, its MaC questions are the longest of the models considered, suggesting its MaC problems may be more complex than those from other models. MATHWELL is also the only model whose MaC length is within a token of its overall average, providing evidence that its MaC questions are likely similar in complexity to its average question.

C.4 MATHWELL and SGSM Have Appropriate Readability

Calculating the reading level of math word problems is one way to automatically assess whether they are written at an appropriate level. Like Norberg et al. (2023), we use FKGL and NDC to evaluate reading level. FKGL calculates reading level as a function of the total words, total sentences, and total syllables in a piece of text, with the score representing the U.S. grade level of the text (Aggarwal; Flesch, 1948; Kincaid et al., 1975). Negative FKGL scores are possible and denote text that is easy to read due to having short words and sentences. NDC is computed as a function of sentence length and the number of words in a sentence that are not contained in a list of 3,000 common English words (Aggarwal; Chall and Dale, 1995).

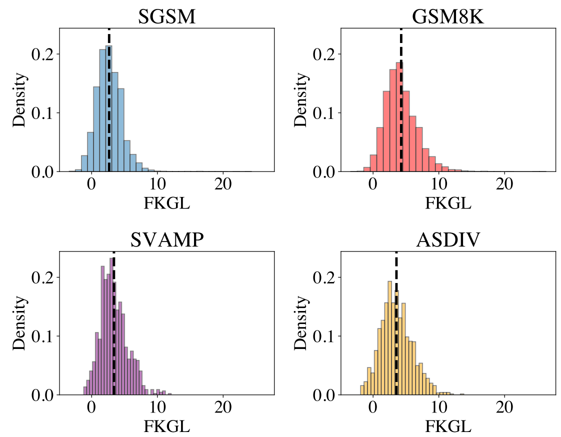

Figure 1: FKGL distribution of training datasets. Dotted lines show the mean for each source.

As shown in Table 4, MATHWELL's NDC is slightly higher than the training datasets but similar to the other models. One reason for this difference is that NDC for model generations might be inflated because they are all topic specific and include proper nouns that may be familiar to a young learner but are not in the list of 3,000 common words (e.g., Fortnite). In turn, MATHWELL's generations, as assessed by NDC, are likely similar in readability to other datasets.

Figure 1 compares the FKGL distribution of SGSM questions to the three other existing datasets that are appropriate for grade school students. Fewer of SGSM's questions than other existing datasets are written at a grade level beyond 8th grade. This suggests that the dataset captures the intended age range and complements existing datasets by including questions more appropriate for struggling readers.

Figure 2 compares the FKGL distribution of MATHWELL generations to the other models considered. While MATHWELL has a similar average FKGL to the other models, it is the only model that does not generate questions beyond an 8th grade reading level. Additionally, its distribution of MaC generations is roughly equivalent to that of all its generations, while the other models' MaC distributions tend to have less density at higher grade levels. These findings suggest that MATHWELL is more likely to generate age-appropriate questions and that its MaC outputs are no simpler than its average output.

Source	PPL ↓	M PPL ↓	BF1	M BF1	A/M BF1	G BF1	AL	M AL	NDC
MathInstruct GSM8K	3.07 (0.691)	–	84.6	–	–	–	**65.7** (23.7)**	–	8.15 (1.12)
SGSM Train	2.44 (0.439)	–	85.2	–	–	84.4	57.3 (15.7)	–	8.13 (1.25)
SGSM Unannotated	**2.33 (0.679)**	–	84.9	–	–	84.0	62.2 (14.9)	–	7.98 (1.28)
LLEMMA	3.79 (1.60)	3.12 (0.615)	84.3	85.3	84.6	84.4	56.5 (22.6)	51.1 (17.3)	8.39 (1.37)
MAmmoTH	2.75 (0.526)	2.74 (0.517)	85.9	86.4	86.1	84.8	46.1 (17.9)	44.0 (13.4)	8.25 (1.26)
Llama-2	2.48 (0.527)	2.47 (0.512)	85.4	85.8	85.6	84.5	53.1 (15.6)	51.5 (14.4)	8.17 (1.13)
MATHWELL	2.45 (0.439)	**2.46 (0.427)**	85.6	85.7	85.6	84.3	55.2 (13.8)	**54.5* (13.8)**	8.27 (1.25)

Table 4: Automatic evaluation metrics for each training dataset or model. BF1 is BERTScore F1, M is MaC, A/M BF1 compares all to MaC questions, G BF1 compares each source's questions to MathInstruct GSM8K, and AL is average token length. Bold indicates the lowest PPL and longest AL in each column. A * or ** indicates the difference between the longest AL and the second longest AL is statistically significant at the p<.1 or p<.01 level, respectively. A – in dataset rows indicates the dataset is either fully MaC (MathInstruct GSM8K/SGSM Train) or does not have a MaC subset (SGSM Unannotated). Standard deviations, where applicable, are in parentheses.

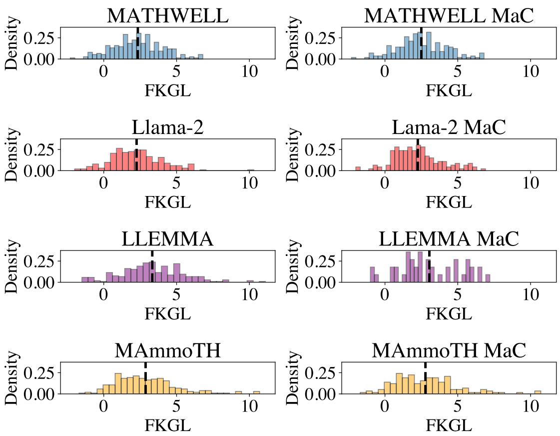

Figure 2: FKGL distribution of model generations for all versus MaC questions. Dotted lines show the mean for each source.

D Annotated Data Characteristics

Our annotated dataset consists of 4,234 question/answer pairs with human annotations for solvability, accuracy, appropriateness, and MaC. The data are comprised of the 3,234 word problem dataset used to generate training data for MATHWELL in addition to the 250 evaluation set for each model described in Section 3. Based on our annotations, 81.1% of the question/answer pairs are solvable, 86.5% have accurate solutions, 67.3% are appropriate, and 57.4% meet all criteria.

E Additional Analyses

E.1 Logistic Regression for Predicting MaC

As shown in Table 5, the coefficients for all models for MaC remain negative and statistically significant relative to MATHWELL, even when controlling for the type of mathematical operation. This finding supports the assertion that MATHWELL is more capable of generating MaC questions regardless of the operation considered, even if it is less likely to generate questions from the more complex mathematical operations.

Predictor	Coefficient	SE	Z	p
Constant	1.648	0.182	9.053	0.000**
LLEMMA	-2.441	0.267	-9.138	0.000**
MAmmoTH	-1.009	0.231	-4.363	0.000**
Llama-2	-0.587	0.241	-2.435	0.015*
Constant	1.237	0.286	4.327	0.000**
LLEMMA	-2.215	0.278	-7.954	0.000**
MAmmoTH	-0.855	0.241	-3.548	0.000**
Llama-2	-0.492	0.245	-2.006	0.045*
Addition	0.187	0.195	0.956	0.339
Subtraction	0.439	0.207	2.124	0.034*
Multiplication	0.215	0.212	1.014	0.311
Division	-0.155	0.260	-0.594	0.552
Fractions	-0.347	0.361	-0.960	0.337
Decimals	-0.211	0.281	-0.752	0.452

Table 5: Logistic regression results for meets all criteria (MaC), with and without controlling for the impact of question type. These results only consider questions which are labeled as solvable. The reference model for the constant is MATHWELL. A * or ** indicates statistical significance at the p<0.05 or p<0.01 level, respectively.

E.2 Accuracy by Question Type

As shown in Table 6, MATHWELL's accuracy does not differ significantly or substantively by operation, while the other models have a significant and/or substantive gap in their accuracy for the operation they perform best on relative to the operation they perform worst on. While MAmmoTH outperforms MATHWELL for addition, multiplication, and division, MATHWELL performs better in the other three operations and in overall accuracy.

Model	Add.	Sub.	Mult.	Div.	Frac.	Dec.
LLEMMA	76.2	72.3	63.4	56.0	50.0	63.2
MAmmoTH	**96.5**	96.3	**96.8**	**97.6**	87.5	91.3
Llama-2	89.3	91.1	87.5	80.0	82.4	75.0
MATHWELL	96.1	**97.4**	94.5	91.3	**100.0**	**94.1**

Table 6: Accuracy by operation. Add., Sub., Div., Frac., and Dec. are addition, subtraction, division, fractions, and decimals, respectively. Bold indicates the best performance in each operation. A bold model name indicates the difference between that model's operation with the highest accuracy and lowest accuracy is statistically significant at the p<0.05 level.

F Example Generations

The sections below display randomly sampled generations from each model, including examples of both questions that do not meet evaluation criteria and those that do.

F.1 LLEMMA

F.1.1 Solvability

Positive Example Steph Curry makes 2133 free throws for the year. He hits 342 more free throws than he misses. How many free throws does Steph Curry miss?

Negative Example There are 5175 Pokémon available to battle in Pokémon Sword and Shield. There are 310 ground-type Pokémon and 182 water-type Pokémon. There are 610 Pokémon that are not ground nor water type. How many Pokémon are ground-type or water-type?

Negative Example Reasoning The question contains conflicting information in that it defines the number of ground and water type Pokémon twice.

F.1.2 Accuracy

Positive Example Naruto eats 4 bananas a day. How many bananas does he eat in a month if there are 30 days in a month?

Solution:
```
def solution():
    #Naruto eats 4 bananas a day
    bananas_per_day = 4
    #There are 30 days in a month
    days_in_month = 30
    #The answer is
    result = bananas_per_day * days_in_month
    return result
```

Negative Example 12 cats eat 400 pounds of cat food every Saturday morning. 15 cats eat 500 pounds of cat food on Sunday morning. How many pounds of cat food are eaten in total?

Solution:
```
def solution():
    #12 cats eat 400 pounds of cat food every
    #Saturday morning
    cat_food_satuday_morning = 12*400
    #15 cats eat 500 pounds of cat food on Sunday
    #morning
    cat_food_sunday_morning = 15*500
    #In total,
    result = cat_food_satuday_morning +
        cat_food_sunday_morning
    return result
```

Negative Example Reasoning The model multiplies the number of cats by the amount of cat food eaten, rather than adding the two sums of cat food together.

F.1.3 Appropriateness

Positive Example Hello Kitty makes 18 bracelets in 4 hours. How many bracelets per hour does she make?

Negative Example Batman caught a baddie with his trademark punch. Each punch knocks out 7 baddies. If Batman has thrown 60 punches, how many baddies has he knocked out?

Negative Example Reasoning While this question is comical, it is not appropriate for a K-8 student because it involves physically harming another person.

F.2 MAmmoTH

F.2.1 Solvability

Positive Example The 5 Power Rangers, Red, Blue, Black, Yellow, and Pink, are eating pizza. Red ate 31 slices, Blue ate 28 slices, Black ate 23 slices, Yellow ate 21 slices, and Pink ate 15 slices. How many slices of pizza did they eat in total?

Negative Example In Fortnite, the player has 5 health points. When the player is hit by an enemy, they lose 2 health points. How many health points does the player have left?

Negative Example Reasoning This question defines how many health points a player loses when they get hit by an enemy, but it does not define how many times the player got hit, so it is impossible to determine how many health points they have left.

F.2.2 Accuracy

Positive Example Harry Potter and his friends have just finished their exams and are looking forward to a well-deserved break. They decide to go on a camping trip together. They have 120 Galleons between them. They spend 30 Galleons on food in the morning and 20 Galleons on food in the afternoon. They have 20 Galleons left. How many Galleons did they spend in the evening?

```
Solution:
def solution():
    # total amount of Galleons they had
    total_galleons = 120
    # amount spent on food in the morning
    morning_food = 30
    # amount spent on food in the afternoon
    afternoon_food = 20
    # amount left
    left_galleons = 20
    # calculate the amount spent in the evening
    evening_food = total_galleons - morning_food -
        afternoon_food - left_galleons
    return evening_food
print(solution())
```
[2]

Negative Example Captain Marvel has 100 friends on Facebook. She has 40 more friends than the average number of friends her friends have. How many friends does the average friend of Captain Marvel have?

```
Solution:
def solution():
    #Captain Marvel has 100 friends on Facebook
    captain_marvel_friends = 100
    #She has 40 more friends than the average
    #number of friends her friends have
    captain_marvel_friends_more = 40
    #The average number of friends her friends have
    average_friends = (captain_marvel_friends -
        captain_marvel_friends_more) / 2
    return average_friends
```

Negative Example Reasoning The model calculates an average rather than subtract the average number of friends Captain Marvel's friends have from her number of friends.

F.2.3 Appropriateness

Positive Example Taylor Swift has 11 Grammys, 29 AMAs, 12 CMAs, 8 ACMs and 35 BMAs. How many awards has she won in total?

Negative Example The soccer team has 32 players. Each player has 2 legs. How many legs does the team have?

Negative Example Reasoning While this question is solvable, it is not appropriate because it is strange to ask how many legs a soccer team has.

F.3 Llama-2

F.3.1 Solvability

Positive Example The Pittsburgh Steelers have 11 players on the field. 3 of the players are from the 10th grade, 6 are from the 11th grade, and the rest are from the 12th grade. How many players are from the 12th grade?

Negative Example The cheerleaders at the school are planning a pep rally. They need to make 200 pom-poms. Each pom-pom requires 20 minutes to make. How many hours will it take to make 200 pom-poms?

Negative Example Reasoning The question does not define how many cheerleaders there are, so it is impossible to determine how long it will take them to make the pom-poms.

F.3.2 Accuracy

Positive Example Barbie has 100 pairs of shoes. She gave 20 pairs of shoes to Ken, 30 pairs of shoes to Skipper, and 40 pairs of shoes to Stacie. How many pairs of shoes does Barbie have left?

```
Solution:
def solution():
    #Barbie started with 100 pairs of shoes
    shoes_initial = 100
    #She gave 20 pairs of shoes to Ken
    shoes_ken = 20
    #She gave 30 pairs of shoes to Skipper
    shoes_skipper = 30
    #She gave 40 pairs of shoes to Stacie
    shoes_stacie = 40
    #The answer is
    result = shoes_initial - shoes_ken -
```

[2] MAmmoTH is prone to ending its generations with print(solution()) because its PoT training data always ends with a print statement (Yue et al., 2023).

```
    shoes_skipper - shoes_stacie
  return result
```

Negative Example A basketball team scored 120 points in a game. The team scored 30 points in the first quarter, 35 points in the second quarter, 20 points in the third quarter, and 35 points in the fourth quarter. How many points did the team score in the second half of the game?

```
Solution:
def solution():
  #The team scored 120 points in a game
  total_points = 120
  #30 points in the first quarter
  points_first_quarter = 30
  #35 points in the second quarter
  points_second_quarter = 35
  #20 points in the third quarter
  points_third_quarter = 20
  #35 points in the fourth quarter
  points_fourth_quarter = 35
  #The answer is
  result = points_second_quarter +
    points_third_quarter + points_fourth_quarter
  return result
```

Negative Example Reasoning The model correctly defines the variables, but it incorrectly adds the points scored in the second quarter to the total for the points scored in the second half.

F.3.3 Appropriateness

Positive Example Patrick Mahomes is a professional football player. He is 6 feet and 3 inches tall. 1 inch is 2.54 centimeters. How many centimeters tall is Patrick Mahomes?

Negative Example A cat has 100 kittens. 20 of them are calico, 30 are tabby, and the rest are siamese. How many kittens are siamese?

Negative Example Reasoning This question is not based in reality, as it is not possible for one cat to birth 100 kittens, nor is it possible for them to be different breeds.

F.4 MATHWELL

F.4.1 Solvability

Positive Example A soccer team has 11 players on the field at a time. They have 22 more players on the bench. How many players are on the team in total?

Negative Example Steph Curry, the NBA star, is 6.2 feet tall. He wears shoes that are 1.5 inches thick. How tall is he in inches? (1 foot is 12 inches)

Negative Example Reasoning This question is not solvable because it is vague: we do not know if we are calculating his height in inches with or without shoes on.

F.4.2 Accuracy

Positive Example In the game, Super Mario can stomp on 30 Goombas, 20 Koopas, and 10 Piranha Plants. How many enemies does he stomp on total?

```
Solution:
def solution():
  #Super Mario can stomp on 30 Goombas
  goombas = 30
  #20 Koopas
  koopas = 20
  #and 10 Piranha Plants
  piranha_plants = 10
  #The answer is
  result = goombas + koopas + piranha_plants
  return result
```

Negative Example A Minecraft player has 100000 blocks. 20000 of the blocks are dirt, 30000 of the blocks are stone, 20000 of the blocks are wood, and 30000 of the blocks are diamond. How many of the blocks are not dirt, stone, wood, or diamond?

```
Solution:
def solution():
  #100000 blocks
  blocks_total = 100000
  #20000 of the blocks are dirt
  blocks_dirt = 20000
  #30000 of the blocks are stone
  blocks_stone = 30000
  #20000 of the blocks are wood
  blocks_wood = 20000
  #30000 of the blocks are diamond
  blocks_diamond = 30000
  #The answer is
  result = blocks_total - blocks_dirt - blocks_stone
    - blocks_wood
  return result
```

Negative Example Reasoning The model correctly defines the necessary variables, but does not subtract the number of diamond blocks from the total number of blocks.

F.4.3 Appropriateness

Positive Example LeBron James has 12000 points. He is 4000 points away from the all-time scoring record. How many more points does he need to average per game for the next 20 games to break the record?

Negative Example A field hockey team has 11 players. 3 of them are forwards, 3 of them are midfielders, 3 of them are defenders, and 2 of them are goalies. How many forwards are there?

Negative Example Reasoning This question is inappropriate to give to a student because it does not require any mathematical operations to solve. It directly defines the number of forwards on the team.

G Prompting Process

G.1 Standard Prompt

Our standard prompt for interacting with the MATHWELL model is, "Write a grade school math word problem about {topic} and Python function with a commented out step-by-step solution to solve the word problem." In this prompt, topic is an optional argument, which we randomly select from a list of topics relevant to K-8 students. We begin every prompt with a random selection of 8-shot examples from SGSM Train.

G.2 Suggested Prompting Strategies

We find MATHWELL is more likely to generate executable code when given a topic than when a topic is not specified. For example, when prompting our finetuned Llama-2 model before further training it on the SGSM Train data, we found the model generated executable code 63.1% of the time when given a topic, and only 32.7% of the time when a topic was not specified. As a result, for evaluating models in this paper, we provide them with a randomly selected topic, which also gives us the ability to assess their ability to effectively generate topic-specific word problems. Additionally, this evaluation strategy is aligned with how a teacher or student would use the model in practice, as they would want the generated questions to align with a particular topic. Qualitative evaluations of model generations also revealed that MATHWELL is more likely to generate executable code when the topic is more specific. For example, if their desired topic is superheroes, a user would have a higher likelihood of receiving a generation with executable code by prompting with a specific superhero (e.g., Superman) than leaving the topic general (e.g., superheroes).

H Annotation Process

H.1 Annotators

All annotators had previous K-12 teaching experience or training, including a research team member who annotated every question. We had three primary annotators who reviewed at least 200 questions each in addition to our research team member.

H.2 Inter-Annotator Agreement

For annotating synthetic data to train MATHWELL, 998 questions were annotated by two people and 232 were annotated by three people. Annotators agreed on solvability $84.6 \pm 2.0\%$ of the time, accuracy $92.0 \pm 1.5\%$ of the time, appropriateness $74.6 \pm 2.4\%$ of the time, all three labels $66.3 \pm 2.6\%$ of the time, and MaC $76.1 \pm 2.4\%$ of the time. The agreement rates for accuracy and solvability are higher than reported in recent human evaluation studies that analyze human preferences in LLM outputs, and the agreement rates for appropriateness and MaC are on par with these studies (Bai et al., 2022; Ouyang et al., 2022; Stiennon et al., 2022; Ziegler et al., 2020). As a result, we feel confident in the quality of our labels.

H.3 Handling Annotator Disagreement

For annotating synthetic data to train MATHWELL, if the question was reviewed by two annotators and they disagreed on one of the criteria, we labeled the example as not having the desired criteria. If the question was reviewed by three annotators and there was a disagreement on one of the criteria, we assigned the label with the majority vote.

H.4 Validating Final Evaluation Labels

For annotating the 250 samples from each model for our experiments reported in Section 3, we randomized questions from each model and had them blindly reviewed by one of our highly trained annotators with K-12 teaching experience. To evaluate the quality of these labels, we had 285 randomly reviewed by one additional annotator and 60 random reviewed by two additional annotators. The annotators agreed on solvability $88.2 \pm 3.4\%$ of the time, accuracy 94.8 ± 2.3% of the time, appropriateness 81.0 ± 4.1% of the time, all three labels 67.1 ± 4.9% of the time, and MaC 79.3 ± 4.3% of the time. Similar to above, the agreement rates for accuracy and solvability are higher than reported in recent studies that explore human alignment of LLM outputs, and the agreement rates for appropriateness and MaC are on par with these studies (Bai et al., 2022; Ouyang et al., 2022; Stiennon et al., 2022; Ziegler et al., 2020).

Additional analysis reveals that most annotator disagreement (81.6%) was due to the primary annotator being more conservative than the additional annotators by labeling questions as not having the desired criteria when the additional annotators rated them as having the desired criteria. As a result, we chose to use the labels from the primary annotator when reporting final results to be conservative, though we also found the results do not vary when switching labels based on annotator disagreement. Annotators were least likely to disagree on labels for MATHWELL outputs and our primary annotator was not more likely to rate MATHWELL outputs as having the desired criteria than the additional annotators. Taken together, this evidence suggests our final labels are highly accurate.

H.5 Annotation Interface

We used Zooniverse (Zooniverse) to collect our human annotation data. Figures 3, 4, 5 and 6 show the instructions each annotator was given for each of our evaluation criteria.

I Early MATHWELL Experimentation

In addition to training context-free question/answer pair generators, we also experiment with training context-free question generation models. Our theory is that if we could train a model to generate questions effectively, we could pass those questions to a math QA model to retrieve answers automatically. To test this theory, we finetune both Llama-2 and MAmmoTH as question generators using MathInstruct GSM8K, excluding the solution for each question and modifying the standard prompt to ask the model to generate a question only. We then sample and evaluate 100 generations from each model. We find that MAmmoTH performs better than Llama-2 at this task, but neither model performs optimally. For example, only 19% of the MAmmoTH generations include the requested topic and 52.6% are solvable. Therefore, based on the results we report in Table 1, we conclude that it is more efficient to train a context-free question/answer pair generator than question generator.

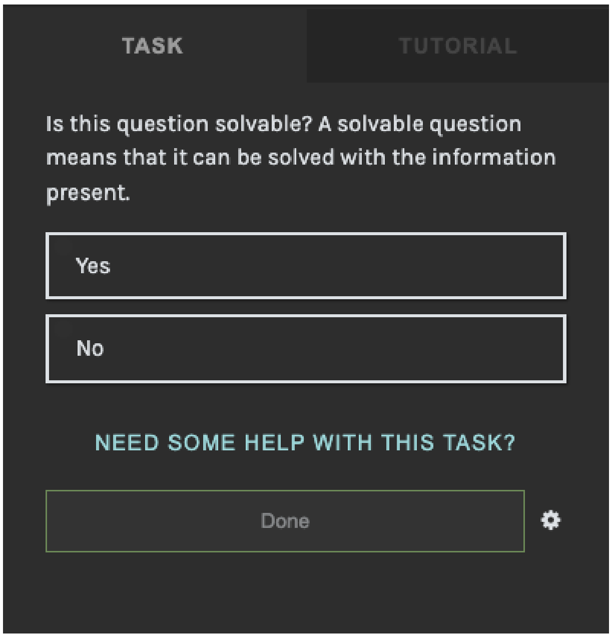

Figure 3: Solvability directions.

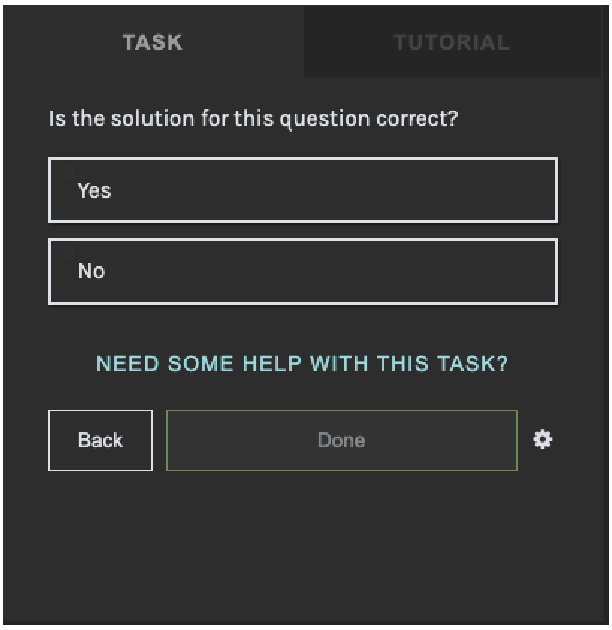

Figure 4: Accuracy directions.

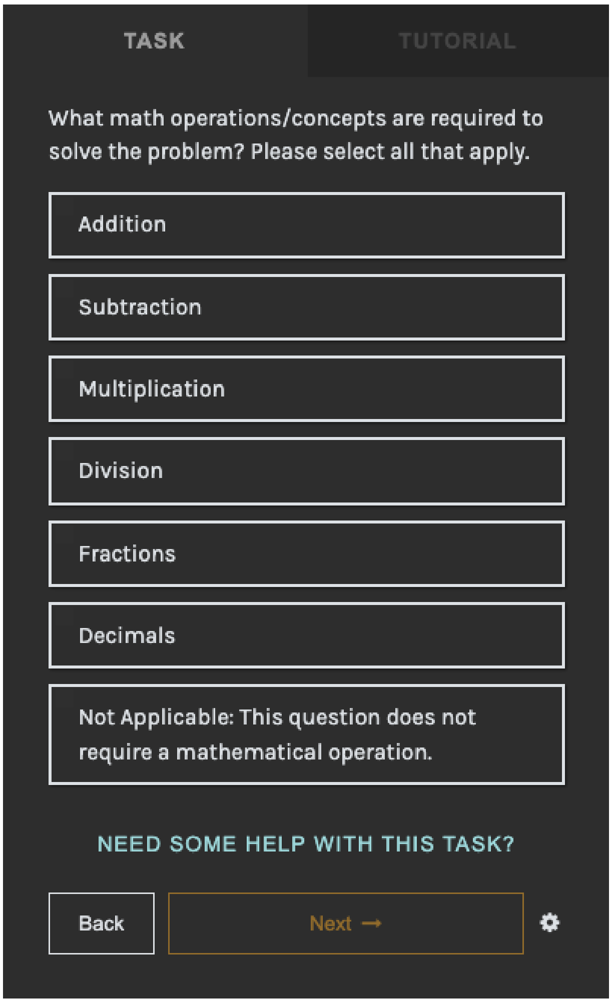

Figure 5: Labeling operations directions.

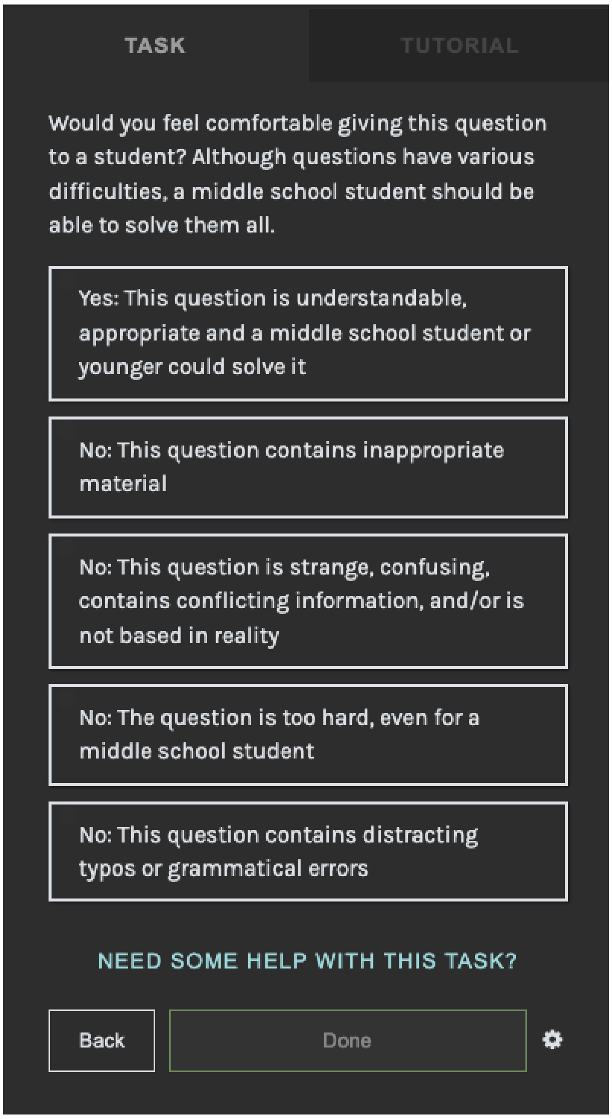

Figure 6: Appropriateness directions.