Question Mastery Unleashed: Self-Generated Data Augmentation is All You Need for AI to Ace Exams

Aydan Huang, Viola Xu, Tian Zhou
Department of Computer Science,
Johns Hopkins University
Baltimore, MD 21218
{yhuan235, kxu29, tzhou32}@jhu.edu

ABSTRACT

This study explores the effectiveness of QMastery pipeline, which employs self-generated question methods to improve language model performance on math problem-solving across different educational levels and question formats. By implementing a novel data augmentation strategy, we fine-tuned models[1] on two datasets: SAT, featuring high school level multiple-choice questions, and GSM8K, containing grade school level short-answer math problems. Our results indicate significant improvements in model accuracy, demonstrating the potential of self-generated content in training adaptable and robust language models.

1 Motivation

The more data we have, the better performance we can achieve. However, it is very too luxurious to annotate a large amount of training data. Therefore, proper data augmentation is useful to boost up model performance. Data augmentation has shown promising results in computer vision. Image can be augmented easily by flipping, adding salt, etc, and it has been proved that augmentation is one of the anchors to the success of computer vision models.

However, in the field of natural language processing, it is hard to augment text due to the high complexity of language: not every word we can replace it with others such as a, an, the. Also, not every word has a synonym. Even changing a word, the context will be totally different.

Motivated by the Self-instruct[1] framework, we aim to refine language models using self-generated instances. These techniques are often constrained by the inherent complexities of language, where straightforward substitutions or the use of synonyms may not be contextually viable. By utilizing self-generated content, our approach seeks to reduce the need for manually labeling and annotating large datasets, enhancing both task-specific and general capabilities of LMs. This strategy heralds a shift towards more efficient and effective training methodologies in NLP, potentially setting a new standard for model refinement.

2 Related Work

Self-Instruct: Aligning Language Models with Self-Generated Instructions [1] by Wang et al, introduces a framework to improve instruction-tuning's capabilities in large language models by generating new instruction data from the task seeds. The process prompts the model to generate new tasks from seeds with the corresponding input-output examples, and filters out low-quality or similar instructions. The model performs instruction-tuning on this generated dataset. This method resulted in significant

[1]The pipeline and datasets can be found in the following GitHub repository: https://github.com/Viola-kxu/NLP-Self-Supervised-Learning

improvements in following instructions when applied to GPT-3 model, demonstrating the utility of self-generated instruction data for enhancing language model performance without relying on extensive labeled datasets.

Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models[4], by Wang et al, introduces a method that encourages models to develop logical reasoning, leading to more coherent question stems and accurate answers. As a result, the generated questions exhibit higher logical consistency, and question-answer accuracy improves after training. This finding is consistent with prior studies, indicating the potential of this technique in enhancing model performance. We implemented the Chain-of-Thoughts technique into our process of generating new questions and answer instances.

3 Experiment

3.1 Instance Generation Pipeline

We have implemented a pipeline designed to generate new question stems and corresponding answers based on seed questions. The pipeline is structured as follows:

Input: Seed questions are used to generate new question stems. We selected 120 initial questions for this purpose.

Question Generation: The language model (LM) generates new questions based on these seeds. To guide this process, we designed instruction templates to ensure quality and variety. **Instruction:** Instructions were carefully crafted to define the expected outcomes and formats rigorously. Based on the previous study[2], diversifying the data can enhance the generalization ability of LLMs, our approach incorporates a diverse set of instructions. For example, one such instruction is to rewrite a given question into a more complex version by incorporating additional reasoning steps. This method aims to enrich the dataset, making it more robust and effective in training LLMs. We have different template of instructions for different datasets and scenarios (question generating or answer generating).

In each iteration, the process entails selecting and shuffling 3 random questions from a pool of seed questions, combined with 3 questions generated in previous iterations. Then we engage the language model ChatGPT 3.5, providing the written instructions and exemplars, and asks it to generate a diverse and progressively complex questions, drawing inspiration from the provided examples. This iterative approach fosters the development of sophisticated and high-quality questions.

Instruction for Question Generating

Your task is to develop a new question that adheres to the following criteria:
1. Content and Style: …
2. To generate, inspired by the given question but … [diversify & differentiate] …
3. The question must be clear, logical, and answerable.

Figure 1: instruction template example

Filtering: After the questions are generated, they undergo a filtering procedure through ChatGPT 3.5, aimed at assessing and eliminating low-quality instances. Consistent with our methodology, specific instructions are written to direct this phase of evaluation. During each filtration cycle, a subset of 10 questions is randomly sampled, and we asks the language model to identify and remove the worst question from the pool. We have regrex helper method to get rid of the questions which only rephrasing or repeating the seed questions. Our objective is to uphold a standard of quality by retaining only 75% of the generated questions.

In addition to the evaluation metric using existing language model, we plan to use more robust and objective metrics to improve the filtering process in the future. In particular, we plan to utilize REV[3] to sift through the generated questions and answers. Through REV evaluation, we filtered out a

fixed portion of questions and answers with low score after each iteration, thereby streamlining the refinement process and enhancing the overall quality of our dataset.

Answer Generation: During the answer generation phase, we employ ChatGPT 3.5 to review each generated question under the guidance of the written instructions. We use the Chain of Thoughts[6] method to ask LMs to solve the generated question step by step explain the thinking. For every SAT questions, the model is asked to identify the correct answer from the provided options. If the given options do not contain a correct answer, the model is prompted to generate a new set of answer options.

Furthermore, in a effort to enhance accuracy and robustness, we ask the model to justify the selected solution when generating answers. This aligns with our observation that ChatGPT's performance improves when it is prompted to provide an explanation for the chosen solution, corresponding to the "chain of thought" process.

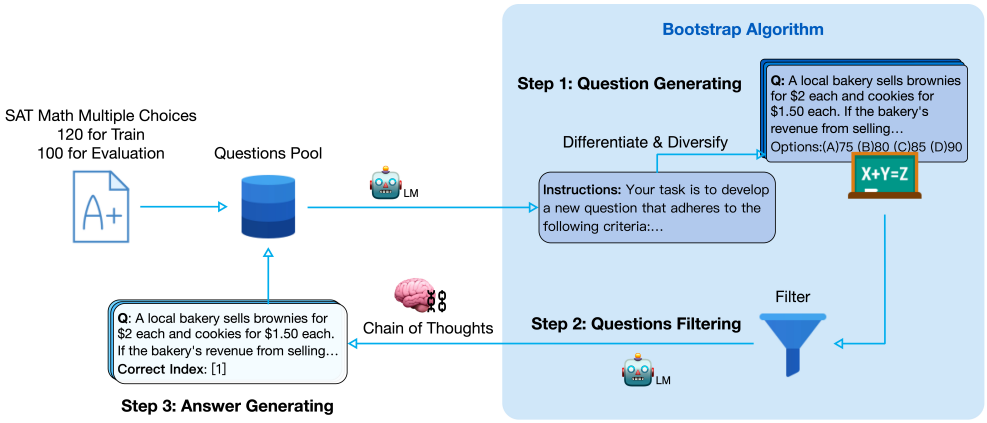


Figure 2: Pipeline for generating instances

3.2 Finetuning

We then apply the obtained dataset to fine-tune the model and examine its ability on the validation set.

The fine-tuning phase involved training on two pretrained models, BERT-small and GPT3.5-turbo, using our generated question-and-answer dataset. We applied full-finetuning, adjusting all model parameters to optimize performance on SAT/GSM8K Math problem solving. This process aimed to improve the models' overall accuracy in answering SAT Math questions. After tuning, we validated the models' performance on a separate test set. The preliminary results (see 6. Results) indicate that both BERT and GPT3.5-turbo showed improvement after fine-tuning. Further optimization and resource availability could yield additional performance gains.

3.3 Dataset

In our evaluation, we aim to demonstrate that our self-generated question method significantly improves the language model's performance across various levels of mathematical problem-solving. Specifically, we compare enhancements in model accuracy on the SAT dataset, featuring high school level math formatted as multiple choice questions, with the GSM8K dataset, which includes grade school level math requiring short-answer responses. This comparison not only showcases the model's ability to adapt to different complexities and question formats but also highlights the effectiveness of our data augmentation approach in training models for a broader spectrum of problem-solving tasks. By using self-generated content, we expect to see a clear improvement in the model's ability to accurately address both structured multiple-choice questions and open-ended mathematical challenges.

3.3.1 Dataset 1: SAT Dataset

The first dataset consists of 10 SAT practice exams for question seeds and an additional 10 for evaluation purposes. From the initial 10 exams, we extracted 120 math questions as seed prompt and generated 1000 questions from the seeds. We extract another 100 question from the remaining 10 practice exams for evaluating the accuracy on the tuned model.

The test data size and the number of generated questions are comparatively small, due to high cost and limited resources. In addition, the high difficulty and complexity of SAT math questions cause both baseline model and the fine-tuned model to have a relatively low performance.

3.3.2 Dataset 2: GSM8K Dataset

GSM8K (Grade School Math 8K) is a dataset of 8.5K high quality linguistically diverse grade school math word problems. The dataset was consisted of short answer questions, created to support the task of question answering on basic mathematical problems that require multi-step reasoning. These problems take between 2 and 8 steps to solve. Solutions primarily involve performing a sequence of elementary calculations using basic arithmetic operations ($+ - \times \div$) to reach the final answer.

By integrating GSM8K, we aim to evaluate the model's ability to generalize beyond SAT-focused questions and apply learned reasoning to broader contexts. This approach also helps to validate the robustness of our fine-tuning strategies by exposing the model to varied question formats and complexity levels found in GSM8K.

Similar to SAT math dataset, we extract 120 questions to be the seed, and we expand the seeds into 2000 questions. Then we employ a set of 1318 instances to evaluate the performance.

Table 1 gives a summary of the two datasets:

Table 1: Comparison of SAT Dataset, GSM8K Dataset, and AquaRat Dataset (validation set)

Attribute	SAT Dataset	GSM8K Dataset	AquaRat Dataset
Size of question seed	120	120	-
Size of generated dataset	1,000	2,000	-
Size of test set	100	1,318	234
Domain	High School Maths	Grade School Maths	Grad School Maths
Question Type	Multiple Choice	Short Answer	Multiple Choice
Source of Questions	SAT Exams	Crowdsourced	GMAT & GRE
Question Complexity	High	Elementary	Very High
Additional Features	Answer Options	Contextual Information	-

4 Distillation Baseline and Hypothesis

4.1 Distillation Baseline

We evaluate our generated dataset by comparing the performance of models fine-tuned with our generated dataset against those fine-tuned with the original dataset. We divide the original dataset into two distinct parts: a training (or fine-tuning) split and a test split. The models' accuracy is then assessed on the test split. This comparison shows the relative improvements from a limited, yet precise, original dataset to a larger, self-generated dataset, and helps us understand the efficacy of the generated dataset in real-world application scenarios.

4.2 Hypothesis

By incorporating the concept of fine-tuning with data augmentation from domain-specific seeds, we expect an improvement in the model's accuracy on mathematical tasks with various question styles, and significantly outperforming the model trained on original data set.

5 Results

5.1 Instance Generation

Overall, the generated questions are logical, clear, and are valid SAT or GSM8K math questions, and the quality and creativity of the question stems is outstanding. An instance of the generations is shown in the Figure 3 below, it is clear that the generated question is different from the seed question, and even more difficult:

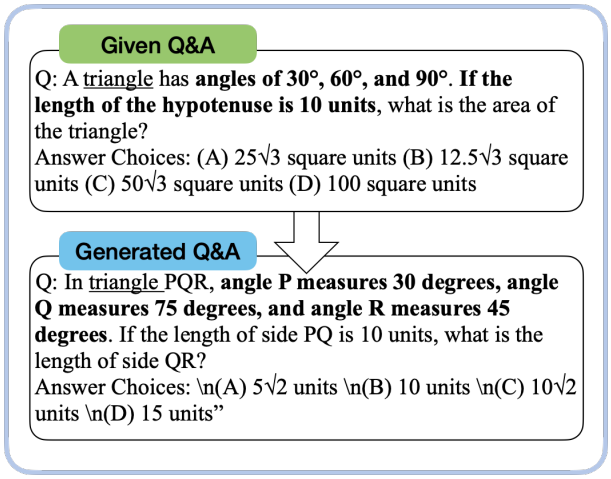


Figure 3: Q&A Generation Example

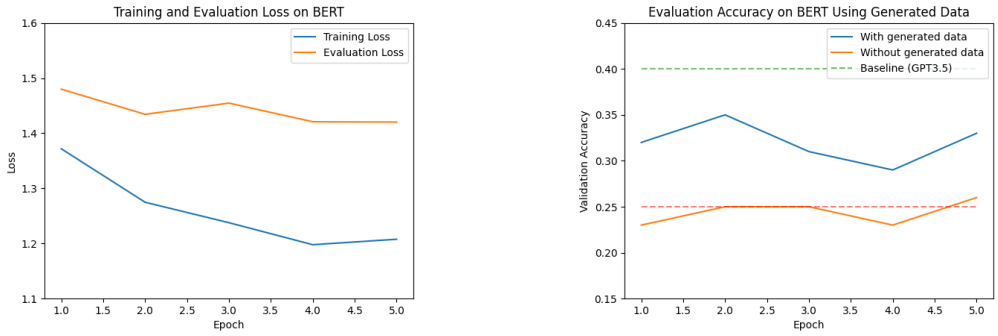
Regardless, we've observed that the overall accuracy and quality of the question is slightly questionable. There are two essential problems. (1) The accuracy of the answer provided by ChatGPT is relative low, this may misleads the model while performing fine-tuning. By testing the capability of ChatGPT 3.5 in solving SAT math through validation set, we observe that the model only attain 40% of accuracy. However, we notice that the key variable that affect the later fine-tuned model performance is the quality of the question, and the accuracy of the answer does not affect much. This might because high-quality questions are likely to contain more relevant and precise information, or they might be structured in a way that aligns well with effective data processing and pattern recognition by the model. This makes it easier for the model to focus on key features and ignore irrelevant information, leading to better generalization even if the answers are not always accurate. Further, LMs such as GPT3.5 are somewhat tolerant to the noise in the output (answers in this case) and they can learn the underlying patterns when the inputs (questions) guide the learning process effectively. (2) The questions generated are not complex enough. They have limited variety of question types and repetitive question patterns and structures. This is likely because of the lack of capability of ChatGPT in understanding mathematics.

5.2 BERT Fine-tuning

By fine-tuning the `BERT-small` model with 1000 generated SAT math instances, we obtain the following results:

The training loss exhibits a gradual decrease, while the validation loss remains relatively stable, as shown in figure (a). This suggests a potential for overfitting, indicating a need for additional generated data.

Figure (b) illustrates the validation accuracy of BERT-small trained with generated data (depicted in blue), BERT-small model trained with the original small dataset (depicted in orange), and the accuracy of ChatGPT 3.5 (indicated by the green dashed line). The model trained solely with the original data achieves an accuracy of 26%, suggesting performance comparable to random guessing, thus highlighting its limitations with a small dataset. In contrast, the model trained with generated data demonstrates a significant improvement, achieving an accuracy of 33%. In addition, the accuracy of our LM on generating data is approximately 0.40, serving as a potential limitation of performance



(a) Training and validation loss

(b) Validation accuracy comparisons

Figure 4: Performance of BERT-small model on SAT math questions

of the model. Although this accuracy remains suboptimal and there persists a notable disparity between the model and ChatGPT 3.5, it underscores the potential of leveraging a pretrained model to assimilate information from the augmented dataset.

We also test our QMasterBert-Small$_{\text{SAT}}$ on aqua-rat, a dataset that is graduate school level multiple choice questions. The result in figure 5, showing that the QMasterBert-Small$_{\text{SAT}}$ outperforms Bert-Small. The accuracy of QMasterBert-Small$_{\text{SAT}}$ stays on 0.24 is because we choose a the best performance model from the finetuned Bert-Small model with generated SAT dataset, and thus we can see the QMasterBert-Small$_{\text{SAT}}$ is consistently better than Bert-Small on aqua-rat dataset. This evaluation shows that the model can generalize well with unforeseen datasets which different from the difficulty levels (even more difficult) of the training dataset.

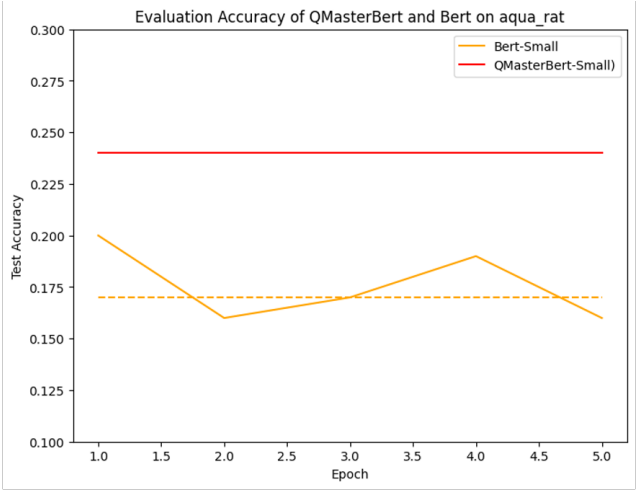


Figure 5: Performance on aqua_rat dataset

5.3 GPT-3.5-turbo Fine-tuning

We fine-tuned the GPT3.5 model with 1000 self-generated SAT data and 2000 self-generated GSM8K data, called QMasterGPT3.5$_{\text{SAT}}$ and QMasterGPT3.5$_{\text{gsm8k}}$. We obtained the result in figure 6, revealing that QMasterGPT3.5$_{\text{SAT}}$ and QMasterGPT3.5$_{\text{gsm8k}}$ both have better performance than GPT3.5 fine-tuned with original dataset. Notably, the average accuracies tested on the GSM8K dataset were higher than those on the SAT dataset. This discrepancy is due to the nature of the problems: GSM8K are grade-school level math questions, which are inherently simpler than the high school-level problems in the SAT. Further, the two datasets have different question style, while SAT is multiple choices, the GSM8K is short answers. This means the generated dataset through our pipeline has good

generalization ability on different styles of questions, and potential of enhancing model performance on all varieties of dataset.

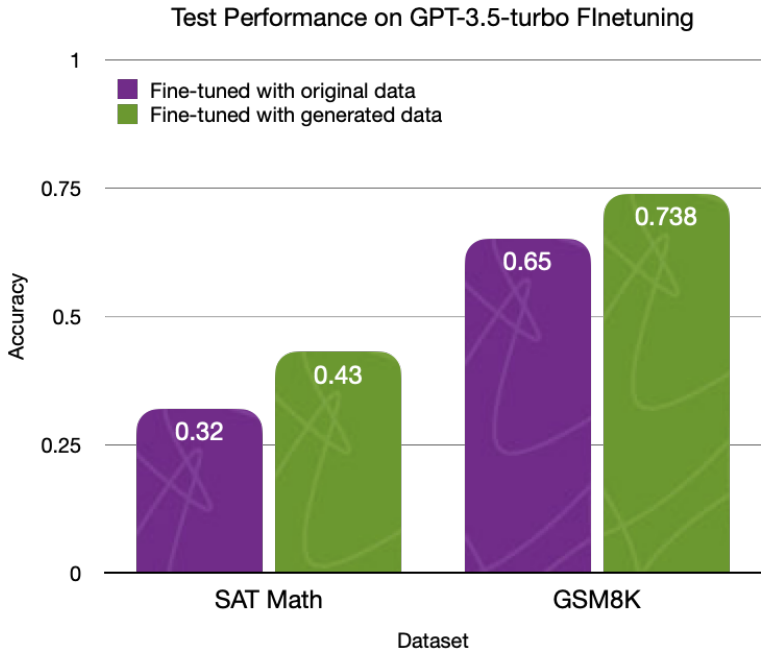


Figure 6: Performance of GPT-3.5-turbo on SAT and GSM8K

5.4 Summary

The results demonstrate that the process is effective across various styles of questions, including multiple-choice and short-answer formats. The process is generalizable to different models, such as GPT-3.5 and BERT. We observe improved performance in these models across various datasets after fine-tuning with the generated dataset, indicating that the data produced by the pipeline is both usable and meaningful.

Model	Dataset	Accuracy
`Bert-Small`	SAT	0.26
`QMasterBert-Small`	SAT	0.33
`Bert-Small`	Aqua-rat	0.18
`QMasterBert-Small`	Aqua-rat	0.24
`GPT3.5`	SAT	0.32
`QMasterGPT3.5`	SAT	0.43
`GPT3.5`	GSM8K	0.65
`QMasterGPT3.5`	GSM8K	0.74

Table 2: Model performance comparison

6 Contributions

This project was a collaborative effort, with each author contributing in the following areas:

- Aydan Huang: led the development of the instance-generation pipeline, including question generation, question filtering, and answer generation

- Viola Xu: write the instructions for generating instances, conducted GPT-3.5-turbo model fine-tuning, and perform data visualization
- Tian Zhou: gather and clean datasets, and conducted BERT model fine-tuning.

7 Conclusion & Future Works

7.1 Conclusion

The QMastery pipeline has demonstrated significant improvements in language model performance across a variety of mathematical problem-solving scenarios, confirming the effectiveness of our self-generated question approach in enhancing model adaptability and accuracy. Notably, this method offers a cost-effective and highly efficient strategy to enhance model performance, particularly in situations where data resources are scarce. By leveraging our innovative approach, we effectively address data limitations, ensuring robust model training and improved reliability in real-world applications.

7.2 Future Works

While the QMastery approach has proven beneficial, several avenues for further enhancement remain. Future efforts will focus on:

(a) *Improving Question Quality*: Current limitations in question generation quality, particularly with ChatGPT 3.5, restrict model performance. We plan to explore alternative data generation techniques that could produce more accurate and complex questions, thereby enhancing the training dataset's quality.

(b) *Exploring Larger Models*: The potential of larger models like GPT4 and BERT-large in the question generation process remains untapped. We intend to assess their capabilities once resources permit, expecting these models to better capture nuances in question complexity.

(c) *Expanding Validation Sets*: To increase the robustness and validity of our models, we are targeting the inclusion of 500 additional validation questions from diverse SAT resources, enabling a more comprehensive evaluation of model performance.

(d) *Enhancing Fine-tuning Techniques*: We will investigate advanced fine-tuning strategies to optimize model training, focusing on techniques that might reduce overfitting and improve accuracy across different types of math problems.

Ablation Study: Future research will also include:

(a) *Seed Selection Analysis*: To mitigate biases introduced by seed selection, we will generate training instances from seeds of varying difficulties and categories to understand their impact on model training.

(b) *Domain Expansion*: Exploring model performance across various domains will help us assess how knowledge in one area, such as math, impacts reasoning in other areas like history or programming, potentially leading to a broader application of our models.

References

[1] Wang et al., "Self-Instruct: Aligning Language Models with Self-Generated Instructions." https://arxiv.org/pdf/2212.10560.pdf

[2] YuLan-Chat-Team. 2023. Yulan-chat: An open- source bilingual chatbot. https://github. com/RUC-GSAI/YuLan-Chat.

[3] Chen, H., Brahman, F., Ren, X., Ji, Y., Choi, Y. and Swayamdipta, S., 2022. REV: information-theoretic evaluation of free-text rationales. arXiv preprint arXiv:2210.04982.

[4] Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., & Lim, E.-P. (2023). Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models. arXiv preprint arXiv:2305.04091. Retrieved from https://arxiv.org/abs/2305.04091

[5] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Car- roll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instruc- tions with Human Feedback. In Advances in Neural Information Processing Systems (NeurIPS).

[6] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., & Zhou, D., 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv preprint arXiv:2201.11903. Retrieved from https://arxiv.org/abs/2201.11903