Augmenting few-shot demonstrations with Large Language Model

Wongyu Seo[1], Advisor: Sundong Kim
[1]Data Science, Gwangju Institute of Science and Technology

1. Introduction

The ARC (Abstraction and Reasoning Corpus) data set[1] is to evaluate the overall intelligence of artificial intelligence systems. ARC is composed of about 3 demonstration problems and 1 test problem.

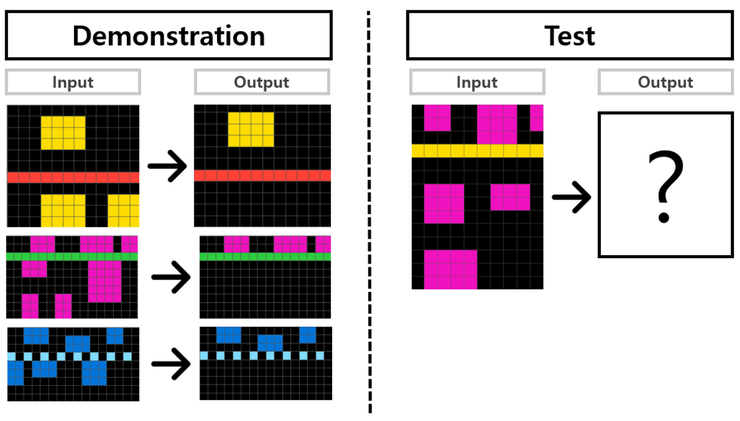

Figure 1: Problem of deducing rules from the input-output data of ARC examples and predicting the output based on test input.

Previous study shows that they have struggled to solve ARC problems with Large Language Models such as GPT-4.0. Furthermore, Transformer-based LLM plunged in performance within complicated logic. However, They've shown that it could have some possibility. We have planned to use these hypothesis.

- Many ARC problems involve a many-to-one correspondence
- Prompt might be helpful to solve ARC

Therefore, we tried to augment ARC Demonstration Data with Chat-GPT.

2. ARC Data Augmentation

2.1 Necessity to Categorize ARC

In this research, we have used a LLM to generated additional few-shot demonstration data for each ARC Problem. To enhance the data augmentation process, the appropriate prompt that notice the transformation from input to output was very essential. In order to make prompt for each problem, we need to classify ARC Problems.

2.2 Data Augmentation Process

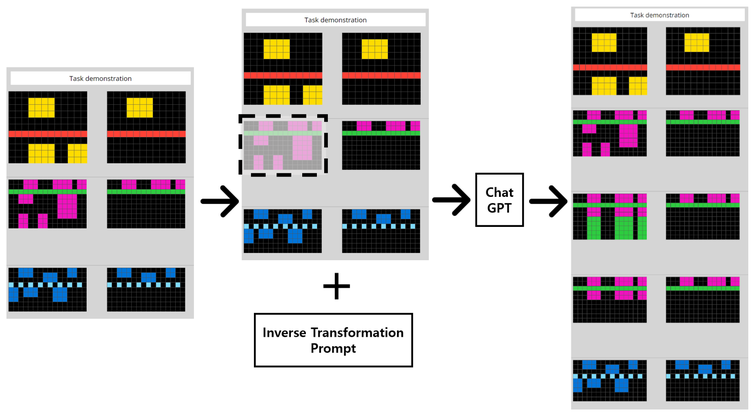

Figure 2: Problem of deducing rules from the input-output data of ARC examples and predicting the output based on test input.

This study utilizes Concept ARC's 16 pre-defined categories to discern meaningful relationships. Understanding the many-to-one correspondence in ARC is crucial for augmenting demonstrations, allowing for diverse answers due to the predictive generation. LLM faces challenges in deducing logical relationships in ARC problems. To assist in this, the study crafted specific prompts for each problem type, providing an inverse transformation prompt for predicting inputs from outputs. I give some demonstrations and one output to infer inputs with inverse transformation prompt, generate new pairs, and repeat the process.

3. Experiment

The experiment was conducted using GPT-4.0 32k with a temperature setting of 1.0 for augmentation. While there were cases, where augmentation was appropriately performed, it is notable, as depicted in Figure 3, that instances of inaccurately predicting inputs occurred frequently. The cases of such inaccuracies will be analyzed in the following section.

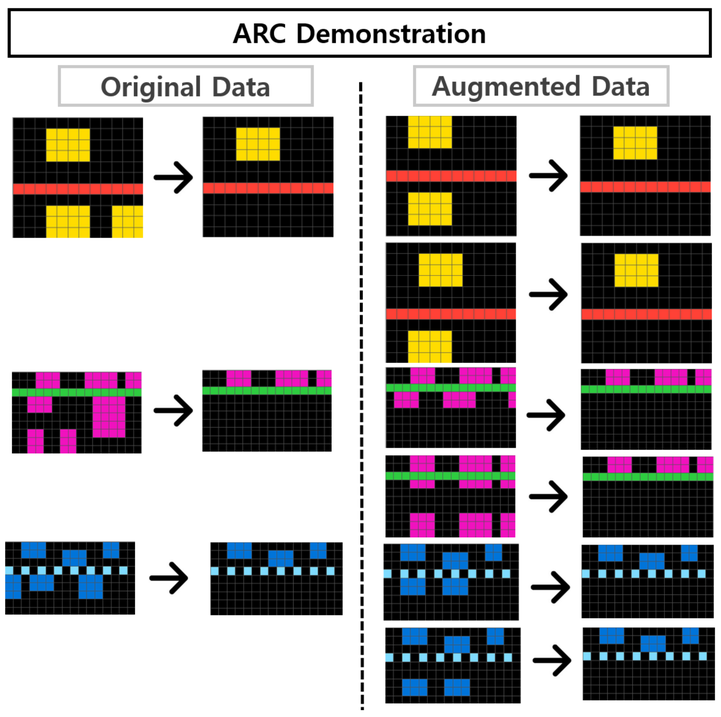

Figure 3: Problem of deducing rules from the input-output data of ARC examples and predicting the output based on test input.

Table 1: The quantity of generated data and valid data

Category	Original Data	Valid Data	Data
Above Below	24	34	58
Center	30	35	65
Clean Up	23	83	106
Complete Shape	21	37	58
Copy	23	4	27
Count	27	29	56
Extend To Boundary	29	8	37
Extract Objects	23	21	44
Filled Not Filled	29	29	58
Horizontal Vertical	25	7	32
Inside Outside	29	24	53
Move To Boundary	25	12	37
Order	21	26	47
Same Different	33	76	109
Top Bottom 2D	34	59	93
Top Bottom 3D	31	25	56
Total	427	509	936

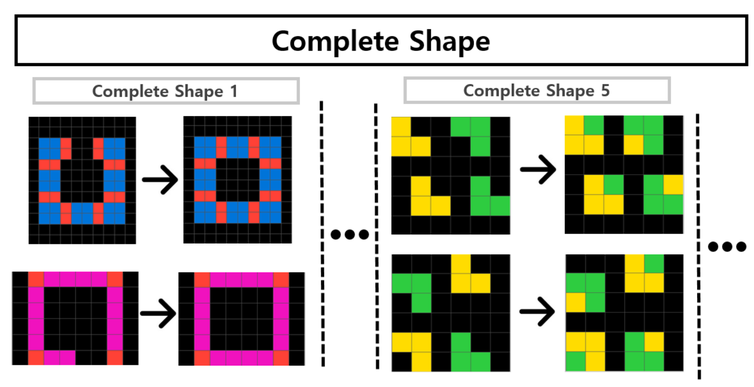

Figure 4: Both problems indicated in the diagram belong to the ConceptARC Complete Shape category.

This upper figure As evident from the diagram, despite being of the same Complete Shape type, the problem-solving approaches vary significantly. For the left problem, a suitable reverse prompt might be "Remove a portion of the object corresponding to the symmetry in all directions." On the other hand, for the right problem, a prompt like "Change one part with a different color in the 2 x 2 square to black" is needed. Attempting to create a universal prompt to describe such diverse prompts proved challenging and abstracting the prompts couldn't adequately explain the inverse transformation. Thus, I got 2 kinds of wrong generation.

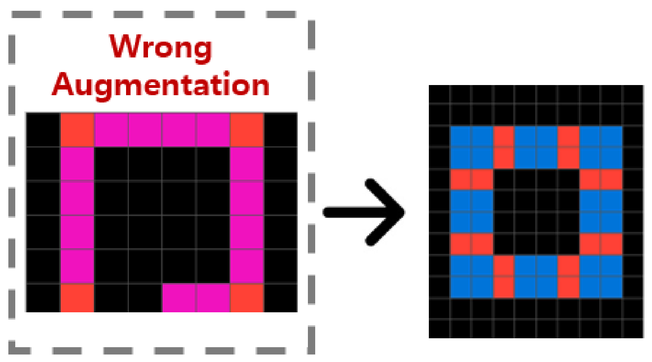

Figure 5: Problem of deducing rules from the input-output data of ARC examples and predicting the output based on test input.

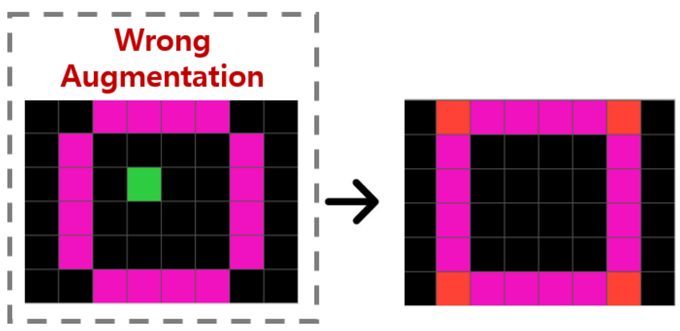

Figure 6: Problem of deducing rules from the input-output data of ARC examples and predicting the output based on test input.

4. CONCLUSIONS

This study confirmed the feasibility of data augmentation with large language model for ARC problems. If improvements are made to the augmentation method using prompts, it is believed that more diverse data can be obtained. In the future, the goal is to go beyond ConceptARC and augment examples for all ARC problems. To achieve this, I think Large-Language Model is supposed to make proper prompt itself.