(KSC2023 우수발표논문) 대형 언어 모델을 활용한 퓨샷 추론 문제의 데이터 증강

서원규[O1] 심우창[2] 김선동[2ψ]
광주과학기술원 전기전자컴퓨터공학부[1] 광주과학기술원 AI대학원[2]
seowongyu@gm.gist.ac.kr, woochang@gm.gist.ac.kr sundong@gist.ac.kr

Augmenting few-shot demonstrations with Large Language Model

Wongyu Seo[O1] Woochang Sim[2] Sundong Kim[2ψ]
GIST EECS[1] GIST AI[2]

요 약 ARC(Abstraction and Reasoning Corpus)[1]는 인공 일반 지능을 평가하는 벤치마크 데이터 셋으로, 추상화와 추론 능력을 필요로 한다는 점에서 다른 벤치마크와는 차별화된다. 각 ARC 문제는 각기 다른 논리적 관계를 가진 입력과 출력으로 구성되어 있으며, 인공지능이 입출력 간의 논리적 관계를 이해해야만 문제를 해결할 수 있다. 예제의 양이 적은 상황 속에서, 많은 인공지능 모델은 논리적 관계 추론에 어려움을 겪고 있다. 그러므로 논리적 관계 추론의 바탕이 되는 예제를 추가로 증강해야 하며, 본 연구에서는 예제를 추가로 생성하기 위해 대형 언어 모델의 추론 능력을 바탕으로 비용효율적인 데이터 증강 프로세스 설계를 목표로 한다. 실험 결과, 인간이 직접 데이터를 증강하는 것과 비슷한 정도의 비용을 통해 데이터를 증강할 수 있음을 확인할 수 있었다.

키워드: ARC, 일반 인공 지능, 대형 언어 모델, 프롬프트 엔지니어링

Abstract The Abstraction and Reasoning Corpus (ARC)[1] is a benchmark dataset for evaluating general artificial intelligence, distinguished from other benchmarks by requiring abstraction and the reasoning capabilities. Each ARC problem consists of inputs and outputs with distinct logical relationships, necessitating AI to comprehend the logical relationship between input and output to solve the problem. In situations where the number of examples is limited, many AI models struggle with logical inference. Therefore, there is a need to generate additional examples to support logical inference. This research aims to design an cost-effective data augmentation process based on the inferential capabilities of large language models to augment additional examples. Experimental results show that it is possible to augment data at a cost similar to that of human-augmented data.

Key words: ARC, Artificial general intelligence, Large language model, Prompt engineering

1. 서론

인공지능 기술은 특정 도메인에 특화된 모델을 개발하는 방향으로 발전해 왔으나, 이와 같은 발전 방향은 한 번 만든 모델이 다방면으로 사용될 수 없다는 단점을 가지고 있다. 따라서 다목적성을 갖추기 위해 추상화와 추론 능력을 갖춘 일반 인공지능(Artificial General Intelligence, 이하 AGI) 연구의 필요성이 제기되었고, ARC 벤치마크[1]는 AGI 연구의 새로운 이정표를 제시할 수 있다고 생각된다.

ARC 문제 해결을 위해 도메인 특화 언어 등 다양한 접근 방식이 시도되었으며, 최근에는 컴퓨팅 성능의 발전에 따라 딥 러닝 등 데이터에 의존적인 방법을 활용하려는 움직임도 많아지고 있다. [2] 이러한 시도를 뒷받침하기 위해서는 충분한 양의 예제 데이터가 필요하지만, 현재 ARC 데이터 셋은 각 문제 당 2~5개의 예제만을 가지고 있어 충분한 데이터 확보에 난관을 겪고 있다. 따라서 본 연구는 데이터 부족 문제를 극복하기 위해 대형 언어 모델의 추론 능력을 활용하여 효율적으로 ARC 데이터를 증강할 수 있는 방법을 조사하고자 한다.

1.1 ARC 벤치마크 데이터 셋

2019년에 공개된 ARC 벤치마크는 인공지능의 일반화된 지능을 평가하기 위해 추상화, 객체 추론, 기하학적 추론 등의 개념을 담은 문제로 이루어져 있다. 각 문제는 그림 1과 같이 3개 내외의 예제와 시험 문제로 구성되어 있으며, 예제 문제를 기반으로 시험 문제의 출력을 예측하는 문제이다. 하지만, 현재의

인공지능은 각 문제 입출력 사이의 논리적 관계를 파악하는 것에 어려움을 겪고 있다.

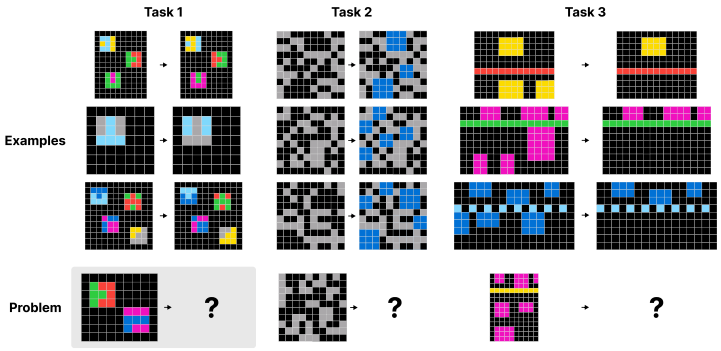

그림 1 ARC 문제의 예시

Fig. 1 Example of ARC Problems

최근에는 ARC를 기반으로 보다 쉽게 접근할 수 있게 변형된 다양한 데이터 셋이 등장하고 있다. 예를 들어, 1-D ARC[3]는 1차원 공간에서의 추상화와 추론 능력을 평가한다, LARC(Language-complete ARC)[4]는 ARC 문제에 언어적 설명이 결합된 형태의 데이터 셋이며, ConceptARC[5]는 ARC에서 사용되는 다양한 개념을 기반으로 분류된 데이터 셋이다. 이러한 다양한 데이터 셋들은 인공지능의 능력을 보다 정교하게 평가할 수 있도록 도와주고 있다.

1.2 관련 연구

선행 연구[3]에 따르면, 대형 언어 모델은 ARC 문제의 논리적 관계 추론에 어려움을 겪고 있다. 또 다른 연구[6]에 따르면, 트랜스포머 기반의 대형 언어 모델(LLM)은 복잡한 추론과 정확한 연산을 요구하는 상황에서 성능이 급격하게 감소한다. 이러한 상황을 고려하면, 대형 언어 모델이 ARC 문제를 해결하는 데 한계가 있다는 사실을 알 수 있다. 그러나 평가 기준을 관대하게 적용할 경우, 대형 언어 모델의 유용성을 극대화하여 사용할 수 있다는 점 역시 지적해 주었다.

이러한 배경을 고려하면, 대형 언어 모델을 활용하여 ARC 예제 증강이 가능하다고 생각되지만, GPT 계열의 모델과 프롬프트를 활용하여 ConceptARC 데이터를 증강한 결과[7], 정확도가 낮다는 문제점을 확인할 수 있었다.

2. ARC 벤치마크 예제 데이터 증강

본 연구는 증강 프로세스의 정확도를 향상시키는 방법을 모색하고자 하였으며, 이전 연구에서 발견된 잘못된 유형의 데이터를 제거하고자 한다. GPT-4.0을 사용하여 [7]의 데이터를 필터링한 결과, 정확도를 17.1%에서 19.5% 정도로 향상시킬 수 있었다. GPT 계열의 모델[8]은 사용되는 토큰에 따라 비용이 발생하므로, 보다 비용효율적으로 데이터를 증강할 수 있는 방법을 찾고자 한다.

2.1 ARC 유형 분류 필요성

1.2절에서 언급한 대로, 대형 언어 모델은 추론 능력이 부족하다는 문제가 있었다. 이를 보완하기 위해 대형 언어 모델의 추론을 돕는 프롬프트를 작성해야 했다. 보다 정확한 프롬프트를 만들기 위해 ARC 문제를 유형별로 분류해야 했고, 본 연구에서는 1.1절에서 언급한 ConceptARC 데이터 셋의 16가지 유형에 따라 프롬프트를 작성하였다.

2.2 증강된 예제 정의

특정 ARC 문제를 T 라고 하자. T 는 n 개의 예제 $\{d_1, d_2, d_3, \cdots d_n\}$로 구성되어 있다. 각 예제 d는 입력 x와 출력 y로 구성되어 있다. T에 속하는 모든 예제 d는 $\forall(x \rightarrow y)$를 만족시키는 유일한 해결법 f가 존재한다. 문제 T에 대해 생성된 예제 $\tilde{d}$의 집합을 T_G라 하자. T_G의 원소 $\tilde{d}$가 f에 의해 해결된다면, $\tilde{d}$는 유효하게 증강된 예제이다.

2.3 ARC 데이터 증강 프로세스

(1) ARC 입출력의 다대일 대응 관계: ARC 예제 증강을 위해서는 먼저 ARC 문제의 특성을 파악해야 한다. 많은 ARC 문제는 그림 2와 같이 여러 입력이 하나의 출력에 대응되는 다대일 대응 관계가 성립하고, 본 연구에서는 이러한 특성을 활용하여 출력을 통해 입력을 예측하도록 하였다. 이러한 역변환(출력 → 입력) 예측은 오직 하나만 존재하는 출력을 예측하는 것보다 다양한 답변을 가능하게 한다는 점에서 1.2절에서 언급한 관대한 평가 기준이 적용된다.

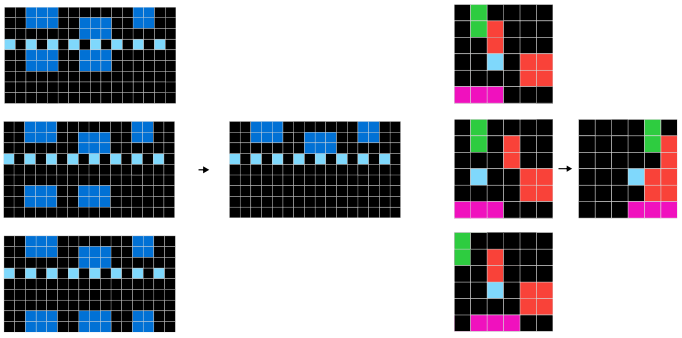

그림 2 ARC 문제의 다대일 관계

Fig. 2 The many-to-one relationship in ARC

(2) 논리적 관계 추론을 돕기 위한 프롬프트: 논리적 관계 추론을 돕기 위한 프롬프트를 작성하기 위해 ConceptARC 데이터 셋의 분류 기준을 따랐다. 프롬프트는 출력에서 입력을 예측하는 것에 도움을 줄 수 있도록 역변환(출력 → 입력) 방법을 설명해 주었다. 예를 들어, Above and Below 유형은 "*수평 기준선의 위아래를 주의 깊게 살핀 후, 그 변화 방법을 적용해 보아라.*", Center 유형은 "*가운데에 있는 것을 움직이거나 제거됐는지 확인하라. 예시를 통해서 패턴을 파악할 수 있다.*"와 같이 역변환 방법을 설명하는

프롬프트를 영문으로 제공해 주었다.

(3) 프롬프트 구체화: (2)에서 생성한 프롬프트는 각 문제가 아닌 카테고리에 따른 프롬프트이다. 하지만, 각 카테고리에는 그림 3처럼 다양한 논리적 관계를 가진 문제가 존재하므로, 각 문제를 세부적으로 설명하는 프롬프트를 추가하여 정확도를 높이고자 하였다.

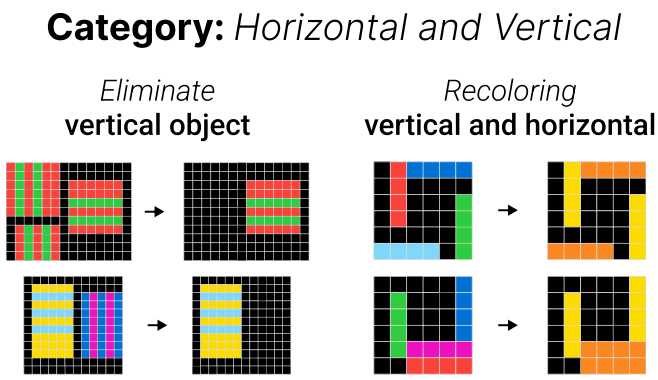

그림 3 같은 카테고리 내의 다양한 논리적 관계

Fig. 3 Diverse relationships within the same category

그림 4는 각 문제에 해당하는 구체화한 프롬프트를 생성하는 방법을 설명한 그림이다. GPT가 ARC 예제 입출력과 (2)의 카테고리 프롬프트를 바탕으로, 문제 입출력 사이의 논리적 관계를 설명하는 구체화한 프롬프트를 추가로 생성하였다.

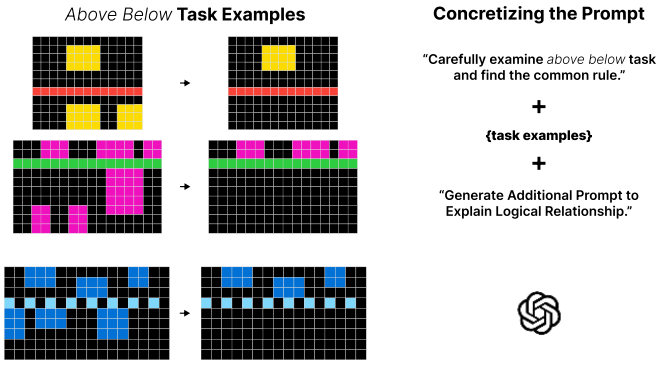

그림 4 프롬프트 구체화 과정

Fig. 4 Process of generating appropriate prompts

(4) 예제 생성: 그림 5는 Above and Below 문제의 예제를 생성하는 과정을 보여준다. 문제의 입출력을 2차원 배열로 대형 언어 모델에 전달해 주었다. 세 번째 예제는 출력만 보여주었고, 나머지 예제(이 경우에는 첫 번째와 두 번째 예제)는 입출력 쌍을 보여주었다.

입출력의 논리적 관계 추론을 돕기 위한 프롬프트는 카테고리별로 사람이 만든 것과 문제에 맞춰 구체화한 것을 합쳐서 만들었다. 이 프롬프트를 바탕으로 세 번째 예제의 입력을 추론하도록 하여 새로운 입출력 쌍을 생성하였고, 이 과정을 나머지 예제(이 경우에는 첫 번째와 두 번째 예제)에도 반복하였다.

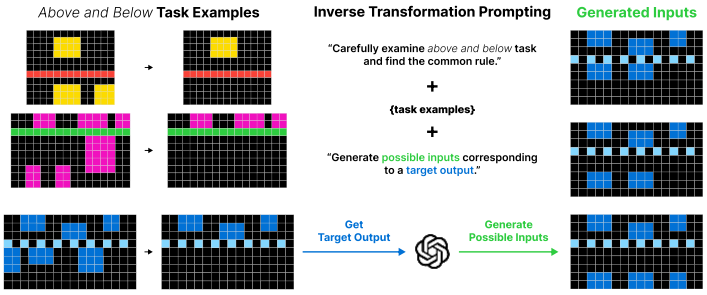

그림 5 예제 데이터 생성 과정

Fig. 5 Process of generating example data

(5) 생성된 예제 필터링: 네거티브 프롬프트[9]를 활용하였음에도 대형 언어 모델이 생성한 데이터 중 2.2절에서 정의한 기준에 적합하지 않게 생성된 것들도 많았다. 따라서 추가적인 선택 과정을 거칠 필요가 있었고, 본 연구에서는 대형 언어 모델을 활용하여 필터링하였다. 그림 6은 예제를 선택하는 방법을 보여준다. ARC 문제의 원본 예제 데이터와 예제에 대한 논리적 관계를 설명해 준 후, 생성된 예제가 원본 예제 데이터와 유사한 논리적 관계를 가지고 있는지 여부를 GPT가 판별하도록 하였다.

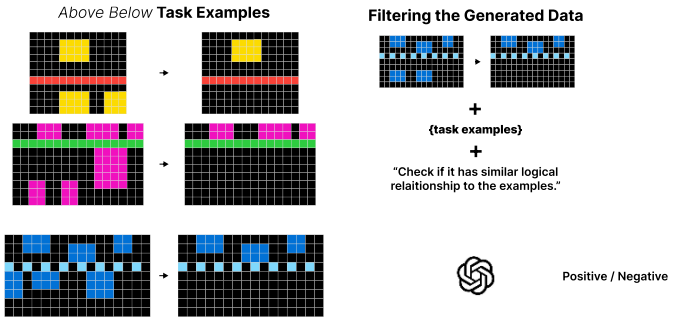

그림 6 생성 데이터 필터링 과정

Fig. 6 Process of filtering generated data

3. 실험

3.1 실험 결과

실험에서는 GPT-3.5 turbo(GPT-3.5)와 GPT-4.0 turbo(GPT-4.0) 두 모델을 활용했으며, 모델 외의 temperature 같은 조건은 모두 같았다. 이를 통해 프롬프트 구체화 여부와 데이터 필터링의 유무에 따른 정확도와 비용을 비교했다. 이러한 접근을 통해 두 모델과 프로세스 간의 성능과 비용을 비교함으로써, 어떤 모델을 어떻게 활용해야 비용 효율적으로 ARC 예제 데이터를 생성할 수 있는지를 확인하였다.

(1) 프롬프트 구체화 여부

프롬프트 구체화에 따른 실험 결과는 표 1을 통해 확인할 수 있다. GPT-4.0을 프롬프트를 구체화한 경우, 구체화하지 않은 경우보다 정확도가 약 15.4% 낮았음을 확인할 수 있었다. 반면, GPT-3.5를 이용하여 프롬프트를 구체화한 경우에는 구체화하지 않은 경우 대비 약 120%의 정확도 상승을 관찰할 수 있었다.

표 1의 Generated Data 열은 중복을 제거한 결과인데, GPT-3.5가 GPT-4.0보다 중복되는 답변을 더 많이 생성했다는 점을 확인할 수 있었다.

표 1 프롬프트 구체화에 따른 실험 결과
Table 1 Experimental results of the prompt concretization

		Generated Data	Valid Data	Validity
GPT-3.5	Generic Prompt	346	24	6.94%
	Prompt Concretization	335	51	15.22%
GPT-4.0	Generic Prompt	411	40	9.73%
	Prompt Concretization	413	34	8.23%

(2) 필터링 모델 유무

표 2 LLM 필터링 실험 결과
Table 2 Experimental results of the LLM filtering

		Status w/o Filter	Status with Filter	False-Negative Rate
GPT-3.5	Generic Prompt	23/346	20/294	5.77%
	Prompt Concretization	51/335	40/280	20.00%
GPT-4.0	Generic Prompt	40/411	18/129	7.80%
	Prompt Concretization	34/413	10/98	7.62%

표 2의 Status는 필터링 과정을 거친 결과로, "*(유효 데이터)/(생성 데이터)*"의 형태로 표시한 것이다. 생성 데이터의 변화량에서 알 수 있듯, GPT-4.0은 GPT-3.5보다 훨씬 적극적으로 데이터를 제거하였음을 확인할 수 있다.

필터링 과정에서 유효 데이터를 제거한 위음성(False-Negative) 사례가 발생했음을 확인할 수 있었다. 우리는 위음성 비율을 식(1)과 같이 정의하여 필터링 과정에서 유효 데이터가 얼마나 손실되었는지 측정하는 지표로 사용하였다. GPT-4.0은 약 7.7% 내외의 균일한 위음성 비율을 가짐을 확인할 수 있었다.

$$(\text{위음성 비율}) = \frac{(\text{유효 데이터 제거량})}{(\text{생성 데이터 제거량})} \qquad (1)$$

표 3 LLM 필터링에 따른 정확도
Table 3 Experimental validity results of LLM filtering

		Validity w/o Filter	Validity with Filter	Validity Improvement
GPT-3.5	Generic Prompt	6.94%	6.80%	-2.02%
	Prompt Concretization	15.22%	14.29%	-6.11%
GPT-4.0	Generic Prompt	9.73%	13.95%	55.67%
	Prompt Concretization	8.23%	10.20%	23.94%

GPT-3.5와 GPT-4.0 모두 프롬프트 구체화하지 않은 경우, 프롬프트를 구체화한 경우보다 필터링의 효과가 더 높았다. 또한, GPT-4.0의 필터링 성능이 GPT-3.5보다 우수한 성능을 가짐을 알 수 있었다.

(3) 각 실험 별 유효 데이터 증강 비용 비교

본 연구는 비용 효율적인 데이터 증강 방법을 찾는 것을 주요 목표로 하므로 실험의 비용을 분석하였다. 앞에서 언급했듯 모든 실험에서 정확도가 높지 않았다. 이에 따라, 증강된 데이터를 사용하기 위해서 사람이 직접 데이터를 선택하여 사용해야 함을 알 수 있다.

각 실험 당 데이터 증강 비용을 비교하기 위하여 동일한 상황으로 통일해야 했고, 본 연구에서는 증강된 데이터를 바로 사용할 수 있는 상황으로 통일하기 위해 그림 7처럼 사람이 직접 데이터를 선택하는 과정을 추가하였다.

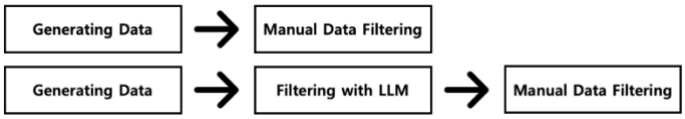

그림 7 정확도 100%로 변환하는 과정
Fig. 7 Method of converting to 100% validity

이 과정에서 데이터 선택 수작업 비용[1]은 시간당 ₩ 10,000으로 책정하였고, 달러로 지불하는 GPT 비용[2]은 원화로 환산한 후, 증강된 데이터 수로 나눠 유효한 데이터 1개를 생성하는 데 드는 비용을 계산하였다.

각 실험의 비용을 비교한 결과, 표 4와 같은 결과를 얻었다. 4가지 실험 모두 필터링했을 때의 비용이 더 많이 소요되었음을 알 수 있다. 특히, GPT-4.0을 사용하여 필터링하는 경우, 사람이 직접 데이터를 선택하는 것보다 5배 내외의 비용이 부담된다는 점을 확인할 수 있다. 따라서, 비용 효율적으로 데이터를

[1] 시간당 500개의 예제 데이터 검토로 가정하였다.
[2] GPT-3.5 turbo (입력: $0.5/1M tokens, 출력: $1.5/1M tokens), GPT-4.0 turbo (입력: $10/1M tokens, 출력: $30/1M tokens)이며, USD/KRW 환율은 '24년도 3월의 평균인 ₩ 1,333.78/$ 를 사용하였다.

증강하기 위해서 GPT-3.5를 활용하여야 함을 알 수 있다. 1개를 생성하는 데 드는 비용이 가장 적은 경우를 선택하더라도 예제 1개당 약 150원이 든다는 것을 확인할 수 있었다.

표 4 각 실험 유형별 비용 비교

Table 4 Cost comparison for each experiment

		Without Filter	With Filter
GPT-3.5	Generic Prompt	₩ 325	₩ 403
	Prompt Concretization	₩ 151	₩ 199
GPT-4.0	Generic Prompt	₩ 730	₩ 3,181
	Prompt Concretization	₩ 950	₩ 6,090

3.2 실험 결과 분석

현재의 실험 방법의 가장 큰 문제점은 본 데이터 증강 프로세스보다 GPT의 성능에 크게 의존한다는 점에서 문제가 있다. 비용 역시 GPT 비용에 크게 의존한다. 현재 GPT-4.0의 비용은 GPT-3.5보다 20배 비싸기에, GPT-4.0의 성능이 높더라도 GPT-3.5보다 비용 효율적인 데이터 증강 결과를 기대하기는 힘들다. 하지만, 대형 언어 모델을 활용한다면, 프롬프트 개선, 도메인 특화 언어 적용 등의 확장 가능성을 가지고 있으므로 대형 언어 모델을 활용하여 데이터를 증강하는 것은 충분히 고려할 만한 가치가 있다고 생각한다.

또한, 입출력 간의 다대일 대응 관계를 바탕으로 추론한다는 점에서 일대일 대응 관계에 해당하는 경우에는 사용할 수 없다는 분명한 한계가 존재한다. 하지만, 본 연구는 프롬프트 구체화를 통해 그림 3에 따른 문제를 해결하고, 데이터 필터링 과정을 추가하여 대형 언어 모델의 잘못된 추론 결과를 줄일 수 있는 방법을 제시했다는 데에 의의가 있다.

4. 결론 및 향후 연구

본 연구에서는 입출력이 다대일 대응에 해당하는 ARC 문제는 대형 언어 모델을 통해서 데이터 증강이 가능함을 확인할 수 있다. 기존 연구들이 활용했던 생각의 트리 등의 프롬프트 기법을 활용한다면, 데이터 증강의 정확도를 보다 높일 수 있으리라 기대된다. 특히, 현재의 프롬프트 생성 및 필터링 과정에서는 비용 효율성을 극대화하기 위하여 필터링 모델의 답변은 단답형으로 제한하였다. 하지만, 답변을 보다 자세하게 하도록 하여 생각의 사슬 기법을 적용한다면, 더 나은 성능을 보여줄 것으로 기대한다.

하지만, 데이터 증강 방법이 GPT의 성능과 비용에 크게 의존한다는 문제점은 부정할 수 없으며, 오직 다대일 대응 문제에만 증강할 수 있는 것도 명백한 한계점이다. 이렇게 증강된 데이터는 편향성의 문제를 가질 수 있으므로, 보다 다양한 출력을 가진 데이터를 증강할 필요가 있다. 이러한 난관을 극복하기 위해 랜덤하게 생성된 출력과 필터링 기법을 활용하여 예제의 출력 부분만을 새롭게 생성하는 방법에 대한 논의가 필요하다. 출력 부분의 다양성을 확보할 수 있다면, 본 연구의 데이터 증강 방법과 결합을 통해 보다 다채로운 예제 데이터를 얻을 수 있으리라 생각한다.

6. 참고 문헌

[1] François Chollet, “On the measure of intelligence,” arXiv:1911.01547, 2019.

[2] Michael Hodel. (2023, July 7). [Online]. Availabe: https://lab42.global/arc/article/

[3] Xu, Yudong, et al. "LLMs and the Abstraction and Reasoning Corpus: Successes, Failures, and the Importance of Object-based Representations." *Transactions on Machine Learning Research* (2023).

[4] Acquaviva, Sam, et al. "Communicating natural programs to humans and machines." Advances in Neural Information Processing Systems 35 (2022): 3731-3743.

[5] Moskvichev, Arsenii Kirillovich, Victor Vikram Odouard, and Melanie Mitchell. "The ConceptARC Benchmark: Evaluating Understanding and Generalization in the ARC Domain." *Transactions on machine learning research* (2023).

[6] Dziri, Nouha, et al. "Faith and fate: Limits of transformers on compositionality." *Advances in Neural Information Processing Systems* 36 (2024).

[7] W. Seo, W. Sim, S. Kim, “Augmenting few-shot demonstrations with Large Language Model,”, Proc. Of the KIISE Korea Software Congress 2023, pp. 341-343, 2023. (in Korean)

[8] OpenAI, “GPT-4 Technical Report”, arXiv:2303.0877, 2023.

[9] Oppenlaender, Jonas, Rhema Linder, and Johanna Silvennoinen. "Prompting ai art: An investigation into the creative skill of prompt engineering." arXiv preprint arXiv:2303.13534(2023).

[그림 1, 3, 5] Lee, Seungpil, et al. "Reasoning Abilities of Large Language Models: In-Depth Analysis on the Abstraction and Reasoning Corpus." *arXiv preprint arXiv:2403.11793* (2024).