대형 언어 모델을 활용한 퓨샷 추론 문제의 데이터 증강

서원규[1] 심우창[2] 김선동[2ψ]
광주과학기술원 전자전기컴퓨터공학부[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

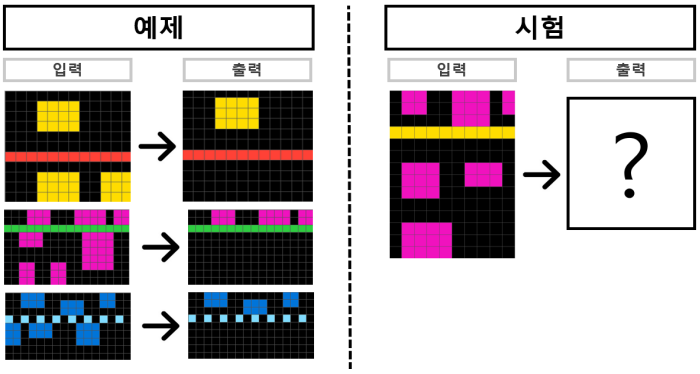
[p] Wongyu Seo[1] Woochang Sim[2] Sundong Kim[2ψ]
GIST EECS[1] GIST AI[2]

요 약

ARC 문제는 예제를 이용하여 문제 입출력 간의 논리적 관계를 추론하는 문제이고, 각 ARC 문제는 각기 다른 논리적 관계를 가지고 있다. 따라서 인공지능이 ARC 문제 해결을 위해서는 입출력 간의 논리적 관계를 이해할 수 있어야 한다. 예제의 양이 적은 상황 속에서, 많은 인공지능 모델은 논리적 관계를 추론하는 것에 어려움을 겪고 있다. 그러므로 논리적 관계 추론의 바탕이 되는 예제를 추가로 생성할 필요가 있다. 본 연구에서는 예제를 추가적으로 생성하기 위해 대형 언어 모델이 가진 추론 능력을 활용하고자 하며, 데이터 증강을 위한 방법으로 「문제 분류 및 입력 예측」을 제안하고 그 결과는 다음 링크에서 확인할 수 있습니다. https://github.com/GIST-DSLab/Augmentation_with_GPT

1[1]. 서 론

2020년에 공개된 ARC(Abstraction and Reasoning Corpus) 데이터셋[1]은 인공지능의 일반화된 지능을 평가하기 위한 것이다. 이 데이터셋은 객체 추론, 기하학적 능력을 측정하기 위한 문제들로 구성되어 있다. 각 문제는 [그림 1]과 같이 3개 내외의 예제와 시험 문제로 이루어져 있으며, 예제 문제를 이용하여 ARC 문제의 논리적 관계를 파악하고, 시험 문제의 출력을 예측하는 문제이다. 하지만 예제의 양이 적어 논리적 관계를 추론하는 것에 어려움을 겪고 있다. 따라서 본 연구에서는 논리적 관계 추론의 바탕이 되는 예제를 추가로 생성하는 효율적인 방법에 대해서 논하고자 한다.



[그림 1] ARC 예제의 입출력 데이터를 바탕으로 유추한 규칙과 시험 입력을 통해 출력을 예상하는 문제

선행 연구[2]에 따르면, GPT-4.0 모델과 같은 대형 언어 모델은 ARC 문제의 논리적 관계를 추론에 어려움을 겪고 있다. 이러한 상황을 고려하면, 대형 언어 모델이 ARC 문제를 해결하는 것을 기대하기는 어렵다. 또한, 복잡한 추론을 이해하고 정확한 연산을 요구하는 상황에서, 트랜스포머 기반의 대형 언어 모델(LLM)의 성능이 급격하게 감소한다고 알려졌다 [3]. 하지만 프롬프팅 기법을 적절하게 활용한다면, ARC 문제의 답과 유사한 결과를 얻을 수 있음을 확인할 수 있었다. [2]. 또한, 유사한 답도 정답으로 인정할 수 있는 관대한 평가 기준을 적용한다면, 대형 언어 모델을 유용하게 활용할 수 있다는 연구 결과가 있다 [3]. 따라서 본 연구에서는 대형 언어 모델을 통한 ARC 예제 증강 가능성에 관해서 확인하고자 한다.

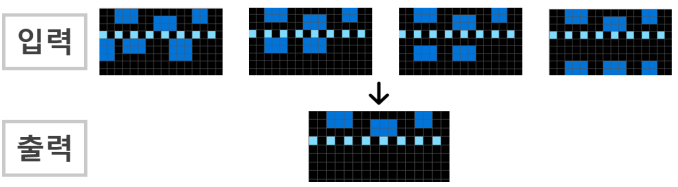
[1] 이 논문은 과학기술정보통신부의 재원으로 한국연구재단과 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2023-00240062, RS-2023-00216011, 2019-0-01842)

2. ARC 문제 데이터 증강

2.1 ARC 유형 분류 필요성

본 연구에서는 각 ARC 문제별로 주어지는 퓨샷 예제 데이터를 추가적으로 생성하기 위해서 대형 언어 모델을 활용하였다. 데이터 생성을 위해서는 대형 언어 모델의 추론을 도울 수 있는 적절한 프롬프트를 작성해야 한다. 더욱 정확한 프롬프트를 작성하기 위해서 ARC 문제를 유형별로 분류한 후, 유형에 따라 프롬프트를 작성하였다. 유용한 프롬프트를 작성하기 위해서는 먼저 합리적인 기준을 통해서 문제 유형을 분류해야 했다. 하지만 각 문제는 서로 다른 논리적 관계를 가지고 있기에 명확한 분류 기준을 찾는 것은 어려웠다. 따라서 본 연구에서는 이미 분류 체계가 확립된 ConceptARC [4] 데이터셋으로 기존의 ARC 문제를 대체하여 실험을 진행하였다. ConceptARC는 분류 체계에 맞추어 새롭게 생성된 ARC 문제들로 이루어져 있으며, 총 16가지 유형이 존재한다. 이렇게 분류된 유형을 바탕으로 프롬프트를 작성하여 ARC 예제 데이터를 증강하였다.

2.2 ARC 예제 증강 방법



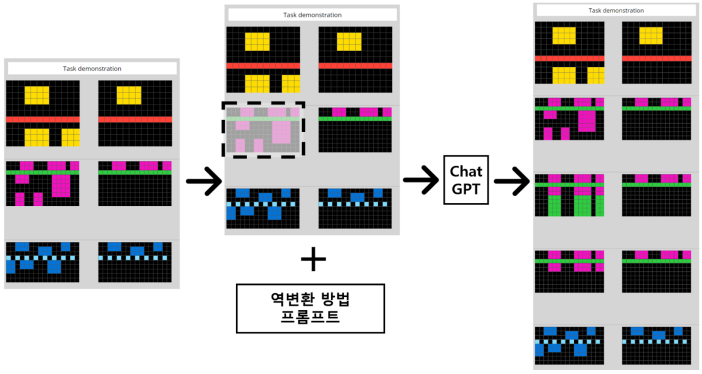
[그림 2] 하늘색 점선을 기준으로 아래쪽 부분을 제거하는 Above and Below 문제로 하나의 출력에 여러 개의 입력 대응 가능

ARC 입출력의 다대일 대응 관계: ARC 예제 증강을 하기에 앞서, ARC 문제의 특성을 파악해야 한다. 많은 ARC 문제는 [그림 2]와 같이 여러 개의 입력과 한 개의 출력이 대응되는 다대일 대응 관계가 성립한다. 따라서 출력을 통해 입력을

예측하는 것은 입출력 쌍을 추가적으로 생성하는 것을 가능하게 하며, 다양한 답변이 가능하므로 1장에서 언급한 관대한 평가 기준을 적용할 수 있다.

논리적 관계 추론을 돕기 위한 프롬프트: 우리는 대형 언어 모델 중 하나인 GPT 계열의 모델[5]을 활용하였다. 현재 GPT 계열의 모델은 ARC 문제의 논리적 관계를 추론함에 있어서 어려움을 겪고 있다. 따라서 대형 언어 모델이 논리적 관계를 유추할 수 있도록 추가적인 프롬프트를 작성해 주어야 했다.

2.1장에서 소개된 ConceptARC 유형에 따라 프롬프트를 다르게 작성해 주었다. 프롬프트는 출력에서 입력을 예측하는 것에 도움을 줄 수 있는 프롬프트(역변환 방법, 출력 → 입력)를 작성했다. 예를 들어, Above and Below 유형은 "수평 기준선의 위아래를 주의 깊게 살핀 후, 그 변화 방법을 적용해 보아라.", Center 유형은 "가운데에 있는 것을 움직이거나 제거됐는지를 확인하라. 예시를 통해서 확인할 수 있다."와 같은 방법으로 역변환 방법에 대한 프롬프트를 영문으로 제공했다.



[그림 3] 예제 데이터 증강 과정이다. 두 번째 예제를 증강하는 과정

예제 증강 과정: [그림 3]은 Above and Below 문제의 예제를 증강하는 과정을 보여준다. 우리는 문제의 입출력을 2차원 배열로 대형 언어 모델에 전달해 주었다. 두 번째 예제는 출력만 보여주었고, 나머지 예제(이 경우에는 첫 번째와 세 번째 예제)는 입출력 쌍을 보여주었다. ARC 문제의 논리적 관계를 추론하기 위한 역변환 프롬프트는 Above and Below 유형에 해당하는 "수평 기준선의 위아래를 주의 깊게 살핀 후, 그 변화 방법을 적용해 보아라."를 사용했다. 이 정보들을 토대로 두 번째 예제의 입력을 추론토록 하여 새로운 입출력 쌍을 생성하였고, 이 과정을 나머지 예제(이 경우에는 첫 번째 예제와 세 번째 예제)에도 반복하였다.

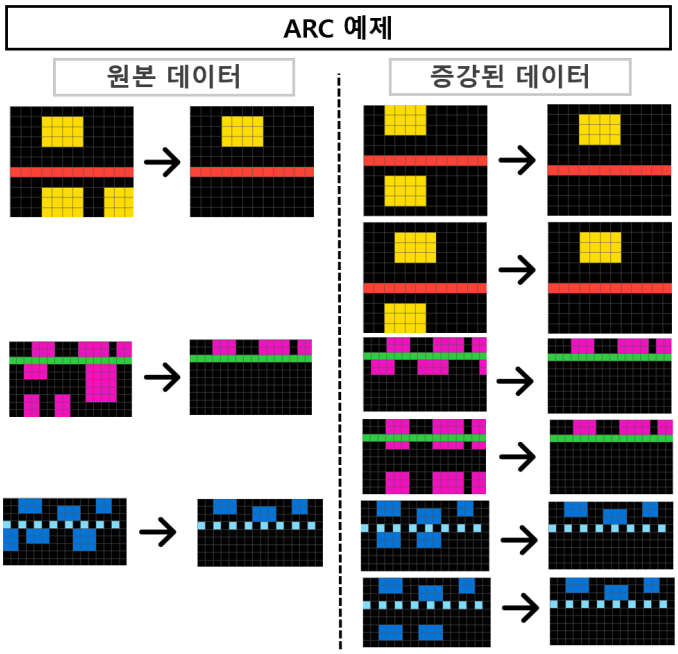
2.3 증강된 예제 정의

특정 ARC 문제를 T라고 하자. T는 n개의 예제 $\{d_1, d_2, \cdots, d_n\}$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 2.2장에 소개된 방법으로 생성된 예제 $\tilde{d}$의 집합을 T_G라 하자. T_G의 원소 $\tilde{d}$가 f에 의해 해결된다면, $\tilde{d}$는 유효하게 증강된 예제이다.

3. 실험

3.1 실험 결과

실험은 GPT-4.0 32k (temperate 1.0)를 사용하여 증강하였다. [그림 4]와 같이 적절하게 증강한 경우도 있었지만, [표 1]에서 알 수 있듯 509개의 데이터를 증강했다. 하지만 입력을 정확하게 예측하지 못하는 경우도 빈번하게 발생하였다. 이런 부정확한 예측 사례에 대해서는 다음 장에서 분석하고자 한다.



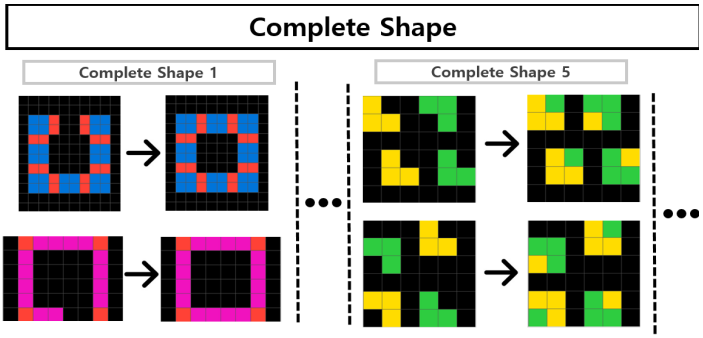
[그림 4] ARC 문항의 예제를 적절하게 증강한 사례에 해당

ConceptARC 분류	기존 데이터 수	유효한 데이터 수	전체 데이터 수
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

[표 1] 생성된 데이터와 유효하게 증강된 데이터의 양

실험을 통하여 생성되는 데이터 대비 약 17.1%의 비율로 유효한 데이터가 증강됨을 확인할 수 있었다. LLM에 몇 개의 데이터를 요구하느냐에 따라 증강되는 데이터의 수는 변화될 수 있으므로 이러한 성질을 잘 활용한다면, 원하는 양의 데이터를 증강할 수 있으리라 생각한다.

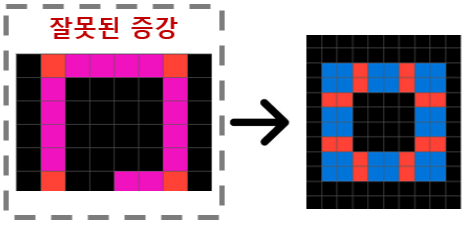
3.2 실험 결과 분석



[그림 5] 표기된 두 문제 모두 ConceptARC Complete Shape에 해당하는 문제로 좌측(Complete Shape 1번 문제)은 상하 좌우 대칭에 해당하는 객체를 완성하는 것이지만, 우측(Complete Shape 5번 문제)에 있는 건 2 x 2 정사각형의 색깔을 추론하여 완성하는 문제

[그림 5]에서 알 수 있듯, 같은 Complete Shape 유형의 문제임에도, 입출력 쌍의 문제 해결법은 모두 상이함을 알 수 있다. [그림 5]에서 좌측(Complete Shape 1)에 있는 문제에 대한 역변환 프롬프트를 작성하게 되면, "상하 좌우 대칭에 해당하는 객체 일부분을 삭제하라." 이고, 우측(Complete Shape 5)에 해당하는 문제는 "2 x 2 크기의 정사각형에서 색이 다른 한 부분을 검은색으로 바꿔라."와 같은 프롬프트가 필요하다. 이처럼 상이한 프롬프트를 보편적으로 설명할 수 있는 하나의 프롬프트를 작성하니 프롬프트가 추상화되어 역변환 방법을 적절하게 설명할 수 없었다.

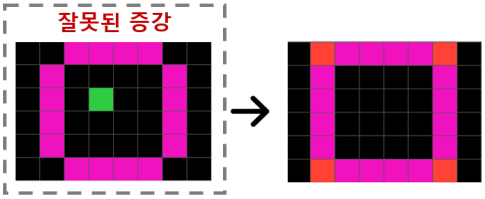
결과 1. 예시 입력을 이용한 추론 결과



[그림 6] Complete Shape 1번을 증강한 사례 중 예시 입력을 이용하여 추론한 경우

예시를 그대로 가져오는 현상을 방지하기 위해 네거티브 프롬프팅 기법[6]을 적용했음에도, [그림 6]처럼 예시의 입출력 쌍의 입력을 변형하여 출력하는 사례도 다수 확인할 수 있었다.

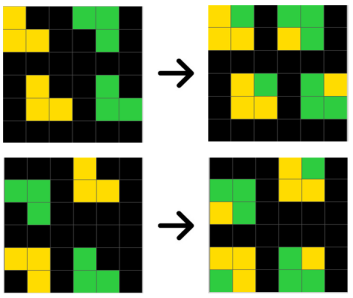
결과 2. 대형 언어 모델의 잘못된 추론 결과



[그림 7] Complete Shape 1번을 잘못된 추론의 결과다. 이 경우에는 입력 이미지를 통해 사각형의 꼭짓점 색을 추론 불가능

기존의 d를 해결하는 방법인 f를 통해서 문제를 해결할 수 없는 경우도 찾을 수 있었다. 이러한 경우는 f로 증강된 예제 $\tilde{d}$를 해결할 수 있는지 확인하는 모델이 등장하기 전까지는 사람이 직접 개입하여 선택할 수밖에 없다.

결과 3. 일대일 대응으로 인한 증강 불가



[그림 8] ConceptARC의 Complete Shape 5번 문제

우리는 데이터 증강을 위해 입출력 쌍의 다대일 관계를 활용했다. 하지만 문항에 따라서 입출력 관계가 [그림 8]처럼 일대일 대응인 경우에는 본 연구에서 제안한 방법을 통하여 예제를 증강하는 것이 불가능하다.

4. 결론 및 향후 연구

본 연구는 입출력이 다대일 대응에 해당하는 ARC 문제는 대형 언어 모델을 통해서 데이터 증강이 가능함을 확인하였다. 프롬프트와 대형 언어 모델을 활용한 본 증강 방법을 추가로 개선한다면, 더욱 다양한 데이터를 얻을 수 있으리라 생각한다. 추후에는 ConceptARC을 벗어나, 모든 ARC의 예제를 증강하고자 하며, 이를 위해서는 아래와 같이 세 부분에서 개선이 필요하다. 먼저 이 연구에서는 생성된 데이터의 사용 가능성 유무를 사람이 직접 수행하는 방법을 사용했다. 그러나 이러한 수작업은 분류하는 사람에 따라서 주관적으로 선택될 우려가 있다. 따라서 향후 연구에서는 대형 언어 모델이 잘못 추론한 결과를 명확한 기준을 통해서 식별할 필요가 있으며, 수작업을 줄이기 위해서 자동으로 필터링할 수 있는 모델을 설계할 필요가 있다. 다음으로는, 명확한 분류 체계가 갖추어져 있지 않은 ARC 문제에 제안한 방법을 적용하기 위해서는 ARC 문제의 분류 체계를 확립해야 한다. 따라서 ARC 문제 분류를 위한 연구가 추가적으로 필요하다. 마지막으로, 현재 연구는 각 문항에 대한 출력이 고정된 상태로 증강된다는 한계점을 가지고 있다. 향후 연구에서는 새로운 출력을 생성할 수 있는 방법에 관해 연구할 필요가 있다. 새로운 출력의 생성은 새로운 입출력 쌍의 생성 가능성을 의미하므로, 훨씬 더 다양한 예제를 생성할 거로 생각한다.

참고문헌

[1] Francois Chollet, "On the Measure of Intelligence", arXiv:1911.01547, 2019.

[2] Xu, Yudong, et al. "LLMs and the Abstraction and Reasoning Corpus: Successes, Failures, and the Importance of Object-based Representations." arXiv preprint arXiv:2305.18354 (2023).

[3] Dziri, Nouha, et al. "Faith and Fate: Limits of Transformers on Compositionality." arXiv preprint arXiv:2305.18654 (2023).

[4] Moskvichev, Arseny, Victor Vikram Odouard, and Melanie Mitchell. "The ConceptARC Benchmark: Evaluating Understanding and Generalization in the ARC Domain." arXiv preprint arXiv:2305.07141 (2023).

[5] OpenAI, "GPT-4 Technical Report", arXiv:2303.0877, 2023.

[6] 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).