

ARC 문제에서 그래프를 활용한 객체 탐지 연구

임민택^{2O} 황산하¹ 사비나² 김선동^{1ψ}

광주과학기술원 AI 대학원¹ 광주과학기술원 전자전기컴퓨터공학부²

victorlim@gist.ac.kr, hsh6449j@gm.gist.ac.kr, ualibecovasabina@gm.gist.ac.kr, sundong@gist.ac.kr

Object Detection on ARC Problem Using Graph Abstraction

Mintaek Lim^{2O} Sanha Hwang¹ Ualibecova Sabina² Sundong Kim^{1ψ}

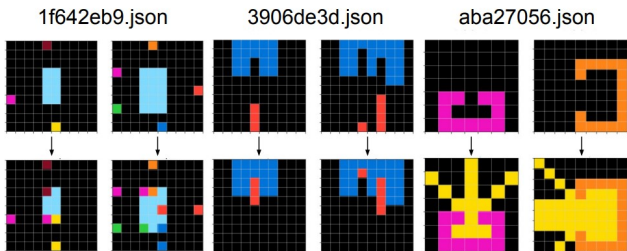
GIST AI¹ GIST EECS²

요약

많은 ARC 문제들이 객체를 중심으로 규칙이 변하기 때문에 일반화된 해결 방법을 위해서는 컴퓨터의 객체 인식 능력이 선행되어야 한다. 객체 중심 문제를 분류하기 위해 객체를 1)색상, 2)픽셀으로 정의하고, ARC 데이터셋에 객체 관련 정보를 추가한 데이터셋을 생성했다. 그 후, ARC 문제를 그래프로 추상화하고 군집화하여 컴퓨터가 객체를 인식할 수 있게 하는 PnP 클러스터링 알고리즘을 개발했다. 알고리즘의 객체 인식 성능은 그래프 클러스터링 방법인 Girvan-Newman과 Greedy Modularity Community Search를 이용한 객체 검출 결과와 비교하는 실험과, ARC 문제를 객체 중심 문제로 분류하는 실험을 진행했다.

1. 서론

2020년 ARC(Abstract and Reasoning Challenge)[1] 데이터셋이 공개되었다. ARC는 컴퓨터의 일반화된 지능을 측정하기 위해 고안된 데이터셋으로 기하학적, 언어적 능력 등이 종합적으로 필요한 문제들로 구성되어 있다. 현재 GPT[2]와 LLaMA[3] 등 거대언어모델(LLM)이 방대한 데이터셋을 학습하여 일반화된 지능을 이루었다고 알려져 있지만, 이들도 ARC 문제를 해결하지 못하고 있다.



[그림 1] (위)입력데이터와 (아래)출력데이터를 보고 규칙을 유추한다.

선행 연구[4]에 따르면 사람은 언어에 기초해 ARC 문제를 바라보는데, 그 중 약 절반은 객체 식별과 관련된 것이었다. 또한 기존의 ARC 문제를 해결하려고 시도했던 모델들은 객체 관련 문제 해결에 어려움을 보였다. 또한 ARC의 약 40%에 해당하는 문제들이 직접적인 객체의 개념을 요구한 것으로 보아 ARC 문제를 해결하는데 있어 객체의 개념을 이용하는 것은 매우 중요하다. 다른 선행 연구[5]에서는 ARC 문제의 해결 과정을 그래프로 추상하는 방법론을 제시했지만 ARC 문제에서 객체를 검출하는 방법은 아직 연구되지 않았다. 우리는 객체의 개념을 직접 채용한 ARC 문제 중 약 71%의 데이터들은, 같은 색으로 구성된 인접 픽셀들의 집합으로 객체를 정의한다는 사실을 확인했고, ARC 데이터셋에 객체 정보를 추가한 데이터셋을 생성했다.

컴퓨터 비전 분야에서는 객체 탐지 연구가 많이 이루어져

왔으며, Slot attention[6], YOLO[7] 등의 기법들은 객체 탐지 문제 성능에서 인간을 능가했다. 그러나 ARC 문제의 데이터 특성 상, 이미지와는 다소 차이가 있기 때문에, 그 기법들을 적용하기에는 어려움이 있다. ARC 문제는 2차원으로 이루어진 배열 데이터와 같으며, 배열의 한 원소는 한 픽셀을 나타낸다. 픽셀이 모여 객체를 이루고 사람은 그 객체가 입출력 데이터(input과 output)에서 변화한 모습을 통해 규칙을 유추하여 문제를 풀게 된다. 따라서 개별 픽셀이 객체 탐지에 있어 중요한 의미를 가진다고 볼 수 있으며, 픽셀들 사이의 관계를 잘 나타내기 위해 그래프 추상화를 진행했다. 우리는 그래프 추상화 단계에서 노드에 색깔, 좌표 등의 정보를 부여하고, DBSCAN[8]을 활용해 객체를 효율적으로 탐지하는 PnP 클러스터링 알고리즘을 개발했다.

또한 기존 그래프 클러스터링 방법인 Girvan-Newman(GN)과 Greedy Modularity Community Search(GM)와의 비교를 수행했다. 이 때, 객체 탐지에 대한 정확도와 실루엣 계수, 그리고 그래프 클러스터링의 모듈성을 함께 비교하여 PnP클러스터링 알고리즘의 성능을 확인했다.

본 연구는 ARC 문제 중에서 객체 기반 문제 해결에 기여할 수 있는 방법론을 제시했고, 기존 방법들과 비교하여 그 성능을 입증했다. 이러한 접근법은 앞으로 일반화된 컴퓨터 인공지능 개발에 있어 중요한 단계가 될 수 있으며, 다양한 객체 인식 기반의 문제 해결에도 활용 가능할 것으로 기대된다.

2. 객체 중심 ARC 문제

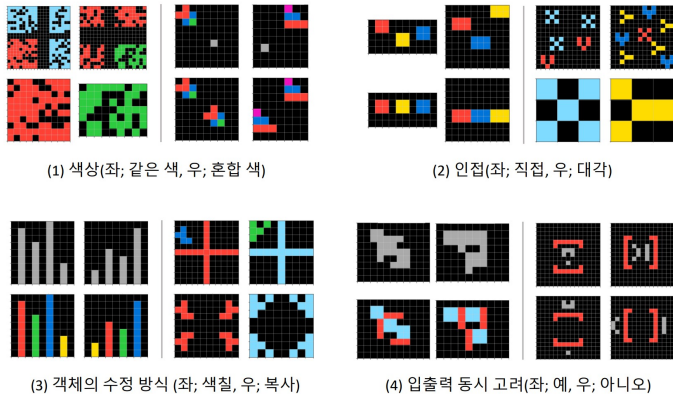
2.1 객체 정의

먼저 ARC 문제에서 객체를 어떻게 정의할 수 있을지 연구했다. ARC 문제는 입력과 출력 데이터가 공유하는 규칙을 파악해야 하는데, 이 과정에서 사람의 직관이 요구되는 경우가 많다. 사람은 주로 입력과 출력데이터의 관계에서 공통된 변화가 있거나, 묶어서 생각할 수 있는 픽셀의 집합을 한 객체라고 인식했다. 이에 따라 객체의 개념을 이용한 모든 ARC 문제는 픽셀의 색상과 픽셀의 위치 관계라는 두 가지 기준을 바탕으로 분류할 수 있었다. 따라서 객체를 정의하는 변수는 1)색상, 2)픽셀 간 위치로 정의했다.

¹ 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2023-00216011, 사람처럼 개념적으로 이해/추론이 가능한 복합인공지능 원천기술 연구)

2.2 객체 중심 문제 분류

2.1 에서 정의한 객체에 따라 우리는 ARC 데이터셋에서 총 128 개의 객체 중심 문제를 선별하고 분류했다. 객체 중심 문제를 선별한 기준은 입출력 데이터에서 같은 방식으로 객체가 정의됨이 직관적으로 확인되어야 하고, 해당 ARC 문제를 해결하는 과정에서 객체의 개념이 활용되어야 한다. 선별된 문제는 4 가지의 세부 속성에 따라 분류했다. 먼저 **(1)객체의 정의 - 색상** 항목은 해당 문제에서 발견된 객체가 한 가지 색상의 픽셀로만 이루어졌는가에 대한 속성이다. 각 원소가 가질 수 있는 값은 '같은 색'과 객체가 두 가지 이상의 색상으로 구성되었을 때 부여한 '혼합 색' 두 가지가 있다. **(2)객체의 정의 - 위치** 항목은 해당 문제에서 발견된 객체를 구성하는 픽셀의 관계에 따라 분류한다. 이 속성에 대한 값은 총 4 가지로, 픽셀이 변을 공유하여 인접(직접인접)한 경우와, 꼭지점을 공유하여 인접(대각인접)하는 경우, 픽셀이 인접하지 않고 특정 범위 내에서 객체를 구성하는 경우(특정범위내), 그리고 다수의 객체가 서로 겹쳐있는 경우(겹침)에 따라 나누었다. **(3)객체의 수정 방식**은 해당 문제의 해결 과정에서 객체가 수정되는 방식을 말한다. 이 속성에는 복사, 색칠, 움직임, 선택, 개수 세기 등이 있다. 마지막으로 **(4)입출력 동시 고려**는 해당 문제에서 객체를 인지할 때 입력과 출력을 동시에 고려해야하는가에 대한 속성이다. ARC 문제의 특성 상 같은 입력이지만 출력이 다르다면 객체가 달라질 수 있기 때문에 해당 항목을 추가했다. 분류를 마친 정보를 원본 ARC 데이터셋에 키 값으로 삽입한 **Augmented ARC 데이터셋**을 생성했다. [그림 2]에서 세부 분류 문제의 예시를 보여준다.



[그림 2] 세부 분류 예시, 위 그림은 각 세부 속성의 대표 문제인데 Augmented ARC 데이터는 위 4 가지 세부 속성에 해당하는 값을 하나씩 가진다.

3. PnP(Push and Pull) 클러스터링 알고리즘

3.1 그래프 추상화

ARC 데이터셋은 2 차원 배열로 이루어져 있으며, 배열의 각 원소는 이미지화 된 ARC 문제의 한 픽셀의 정보(색상)를 담고 있다. 따라서 먼저 배열을 그래프로 추상화 했다. [그림 3]에 나타나 있는 것처럼, 아래의 규칙에 따라 추상화 함수 f 는 2 차원 형식의 배열을 입력 받아, 배열과 일대일 대응하는 그래프를 반환한다.

- (1) ARC 문제의 한 픽셀은 그래프의 한 노드로 변환한다.
- (2) 각 노드에는 해당 대응하는 픽셀의 색, 좌표, 노드 번호(index)의 속성을 부여한다. 좌표 정보는 원본 배열의 색인에 3 씩 곱하여 순차적으로 부여한다.
- (3) 검은색 픽셀의 노드는 배경으로 취급한다.
- (4) 배경이 아닌 인접한 두 노드 사이에 엣지를 그린다.
- (5) 각 엣지는 그 속성에 따라 [표 1]과 같이 가중치를 가진다.

[표 1] 엣지의 가중치, 인접 픽셀 사이의 관계에 따라 노드 간 상대적 거리감을 가중치로 표현 했다

분류	가중치
같은 색, 직접인접	1
같은 색, 대각인접	2
혼합 색, 직접인접	4
혼합 색, 대각인접	5

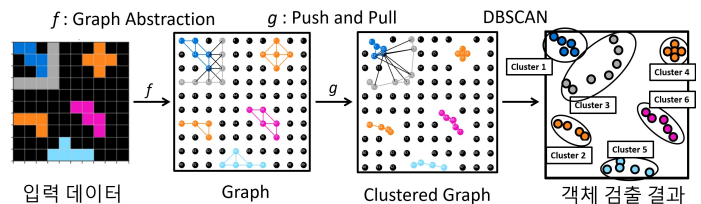
3.2 인/척력에 기반한 클러스터링과 객체 검출

엣지의 가중치는 연결된 두 노드 간의 상대적 거리감을 의미한다. 엣지의 가중치가 낮을 수록 노드들이 같은 객체로 묶일 가능성이 높아진다. 따라서 노드가 직접인접하거나 같은 색일 경우 작은 가중치가 부여되며, 반대로 노드가 대각인접하거나 다른 색일 경우 큰 가중치가 부여된다. Push and Pull은 [그림 3]과 같이 함수 g 에 의해 진행되는데, 이 함수는 추상화 함수 f 의 출력값(그래프)을 입력 받아 수정된 그래프를 반환한다. 구체적으로, 함수는 각 노드를 순회하며 인/척력을 계산한다. 인/척력은 각 엣지의 가중치에 따라 정의하는데 각 엣지가 노드에 작용하는 힘의 크기는

$$\frac{\text{엣지의 가중치} - 3}{2} \quad (1)$$

으로 계산하고, 음수일 경우 인력, 양수일 경우 척력으로 작용한다. 힘의 방향은 각 엣지가 노드에 연결된 방향으로 지정하여 인/척력에 따라 좌표가 이동한다. 결과적으로 같은 객체를 구성하는 노드들은 가까워지는 경향을 보이고, 다른 객체를 구성하는 노드들은 서로 멀어지는 경향을 보인다.

객체 검출 단계에서는 수정된 그래프를 바탕으로 DBSCAN 알고리즘을 이용하여 같은 객체에 속한 노드를 추측하고 분류된 정보(클러스터)를 각 노드의 정보에 반영한다. 검정을 색상 정보로 가지는 노드는 배경을 나타내는 노드이므로 객체 분류에서 배제하고, 객체 정보에 -1 을 부여했다. 이렇게 인/척력에 기반한 클러스터링과 객체 검출 과정을 통해, ARC 문제의 객체 추출이 가능해진다.



[그림 3] PnP clustering 알고리즘 도식도, 그림은 ARC 문제의 각 픽셀을 그래프로 추상화하고, 인력과 척력에 의해 같은 객체끼리 근접화(Push and Pull)하고 객체를 식별하는 과정을 보여준다.

4. 실험

본 연구에서는 PnP클러스터링 알고리즘이 객체 중심 ARC 문제를 입력으로 받아 올바른 객체의 정의를 유추하고 분류하는 지 실험했다. 성능평가를 위해 정확도와 실루엣 계수(silhouette index)를 사용했으며, 엣지를 기준으로 클러스터링을 진행하는 그래프 클러스터링과 결과를 비교했다. 이 때 그래프 알고리즘의 클러스터링 정도를 평가하기 위해 모듈성(modularity)을 사용했다.

4.1 비교 대상- 그래프 클러스터링

Girvan-Newman(GN) 알고리즘은 그래프의 사이 중심성을 기반으로 커뮤니티를 탐지한다. 사이중심성은 노드들 간의 최단 경로에 엣지가 포함되는 횟수를 나타낸다. GN 알고리즘은 사이 중심성이 높은 순서로 엣지를 제거하며 계층적으로 커뮤니티를 분리할 수 있다. 이러한 과정을 반복하면서 최적의 커뮤니티

구조를 찾아낸다. 반대로 **Greedy Modularity community search(GM)** 알고리즘은 모듈성 최적화를 기반으로 커뮤니티를 탐지한다. GM 알고리즘은 초기에 각 노드를 독립적인 커뮤니티로 간주하고, 그 이후에 반복적으로 커뮤니티를 병합하면서 전체 모듈성이 최대화되는 방향으로 커뮤니티를 형성한다. [그림 4]에서 나타나있는 것처럼 그래프로 나타내어진 ARC 문제를 그래프 클러스터링을 통해 객체를 탐지하여 PnP클러스터링 알고리즘과 비교했다.

4.2 성능 지표

각 ARC 문제에 대한 각 알고리즘의 **재현율(Recall)**는

$$\frac{\text{올바르게 식별한 객체의 개수}}{\text{해당 문제에서 관측 가능한 총 객체의 개수}} \quad (2)$$

로 계산하여 실수(float)로 부여했다.

실루엣 계수는 클러스터링의 품질을 평가하는 지표로, 개별 데이터 포인트들이 얼마나 잘 클러스터에 할당 되었는지를 나타낸다. 실루엣 계수는 -1 에서 1 의 범위를 가지며, 값이 클수록 객체가 해당 클러스터에 올바르게 포함되었다는 것을 의미한다.

모듈성은 그래프의 클러스터링을 평가할 수 있는 지표로, 커뮤니티 간의 연결에 대한 정보를 제공한다. 모듈성은 0 부터 1 까지의 값을 가지며, 값이 클수록 같은 커뮤니티 간 노드의 연결성이 큰 것을 의미한다.

[표 2] 모든 문제 분류에서 PnP클러스터링 알고리즘이 그래프 클러스터링보다 우수한 것을 정확도를 통해 알 수 있다.

구분	재현율			실루엣 계수			모듈성		
	PnP	GN	GM	PnP	GN	GM	PnP	GN	GM
같은 색, 직접인접	0.96	0.64	0.70	0.63	0.56	0.59			
같은 색, 대각인접	0.81	0.55	0.47	0.54	0.49	0.51			
혼합 색, 직접인접	0.81	0.55	0.59	0.54	0.54	0.56			
혼합 색, 대각인접	0.62	0.50	0.47	0.40	0.53	0.57			
같은 색, 겹침	0.65	0.03	0.03	0.48	0.48	0.51			
같은 색, 특정범위내	0.80	0.20	0.30	0.59	0.71	0.29			
종합 점수	0.88	0.57	0.60	0.57	0.54	0.56			

입출력 동시 고려	재현율			실루엣 계수			모듈성		
	PnP	GN	GM	PnP	GN	GM	PnP	GN	GM
예	0.85	0.57	0.49	0.46	0.49	0.51			
아니오	0.89	0.60	0.65	0.60	0.56	0.58			

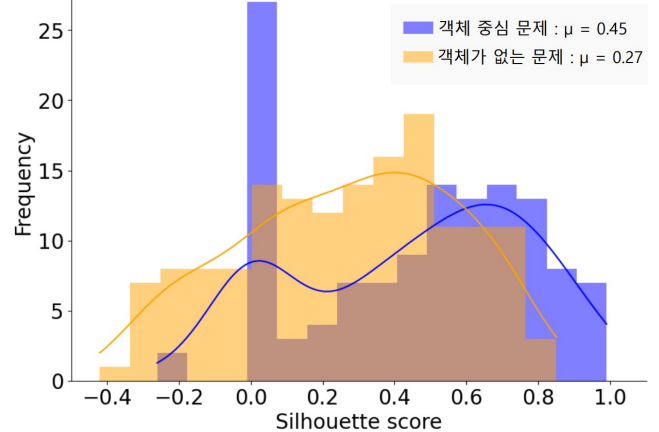
4.3 성능 비교

객체 중심 문제로 선별된 128 개 문제의 학습 데이터에 대해 PnP클러스터링, GN 알고리즘, GM 알고리즘을 적용하여 성능을 비교했다. [표 2]는 모듈성과 실루엣 계수, 그리고 사람이 직접 평가한 정확도로 구성되어 있다. 2.2 에서 나눈 문제의 속성에 따라 평균 성능을 분류 별로 나타냈다. PnP클러스터링은 모든 문제의 평균 정확도가 0.88 로 객체를 잘 판별 하는 것을 보여주었다. 하지만 그래프 클러스터링 알고리즘은 각각 GN 알고리즘은 0.57, GM 알고리즘은 0.60 으로 상대적으로 낮은 성능을 보였다.

4.4 객체 중심 문제 분류

두번째 실험에서는 PnP 클러스터링 알고리즘이 객체 중심 문제와 객체가 없는 문제를 잘 구분할 수 있을지 통계적으로 검토했다. [그림 4]를 살펴보면, 객체 중심 문제의 실루엣 계수 평균은 0.45 고, 객체가 없는 문제의 실루엣 계수 평균은 0.27 로 두 그룹 간 명확한 차이가 있음을 확인할 수 있었다. 결과적으로, 정보 이득(Information Gain)을 이용하여 실루엣 계수의 최적 임계값을 0.27 로 설정했다.

임계값을 적용한 분류 결과, 정확도는 0.9 로 높았으나, 민감도는 0.55, 정밀도는 0.52 로 나타났다. 이는 객체 중심 문제를 잘 분류 해내지만, 객체가 없는 문제를 올바르게 판별하는 성능은 아직 미흡함을 의미한다.



[그림 4] 객체 중심 문제와 객체가 없는 문제 그룹의 실루엣 계수 히스토그램과 평균

5. 결론

이 연구는 ARC 문제에서 객체를 추출하는 방법을 제시하였다. 또한 ARC 문제에서 객체 지향 연구에 활용 될 수 있는 분류 기준을 정하고 이를 적용한 새로운 데이터셋을 제안했다. PnP 클러스터링을 이용해 객체 탐지에 높은 성능을 달성하였으며, 객체 중심 문제와 객체가 없는 문제를 해당 알고리즘을 통해 분류할 수 있는 가능성을 확인할 수 있었다. 이를 통해 객체 중심의 ARC 문제에서 규칙을 추론하고 해결하는 데에 효율적이고 빠른 인코더를 구축할 수 있을 것으로 기대된다. 추후 연구에서는 객체 식별에 있어 한계점을 보완하여 사람의 직관을 완벽하게 반영한 객체 식별 알고리즘을 구현하는 것을 목표로 할 것이다.

참고문헌

[1] Francois Chollet, ARC challenge, Kaggle, 2022
 [2] Open AI, Language Models are Few-Shot Learners, arxiv, 2020
 [3] Meta, LLaMA: Open and Efficient Foundation Language Models, arxiv, 2023
 [4] Sam Acquaviva et al., Communicating Natural Programs to Humans and Machines, NeurIPS, 2022
 [5] William Xu, et al., Graphs, Constraints, and Search for the Abstraction and Reasoning Corpus, AAAI, 2023
 [6] Francesco Locatello et al., Object-Centric Learning with Slot Attention, NeurIPS, 2020
 [7] Joseph Redmon et al., You Only Look Once: Unified, Real-Time Object Detection, arxiv, 2016
 [8] Martin Ester et al., A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, KDD, 1996