행동 Scene-Graph 예측기를 활용한 시나리오 생성

임우빈 ○ 김우재 윤성의

한국과학기술원

{iwbn, wkim97, sungeui} @ kaist.ac.kr

Scenario Generation by Action Scene-Graph Prediction

Woobin Im ○ Woo Jae Kim Sung-Eui Yoon

KAIST

요 약

본 연구는 행동 예측기를 통해서 시나리오를 생성하는 문제를 탐구한다. 시나리오란 행동의 시퀀스로서, 소셜로봇이 사용자와 협업을 수행하는 데 이용될 수 있다. 하지만 수집된 데이터만으로는 시나리오의 종류가 한정적이기 때문에 다양한 시나리오를 구성하는 데는 어려움이 있다. 따라서 본 연구에서는 존재하는 시나리오를 바탕으로 새로운 시나리오를 생성하는 모델의 학습 방법론에 대하여 제시한다. 구체적으로는, scene-graph 형태의 행동을 예측하는 모델을 재귀적으로 사용하여 시나리오 생성을 수행한다. 모델의 평가를 위해 ScenarioGenome 데이터셋을 구축하였고 제시하는 모델의 성능을 검증하였다.

1. 서 론

소셜 로봇이 사용자와 깊은 교감을 하기 위해서는 사람의 행동을 이해하는 능력을 함양해야 한다. 사람의 행동을 이해하기 위한 연구는 행동 인지[1] 및 행동 분류[2] 분야에서 활발하게 연구되었다. 이러한 연구의 목적은 사람의 행동을 인지하는 데 있고, 최근 머신러닝 기술의 발전과 함께 높은 성능을 보이고 있다.

하지만, 인지기술만으로는 소셜로봇의 성공을 기대하기 어렵다 [3]. 소셜로봇은 실생활에서 일어나는 다양한 시나리오에서 사람과 교감할 수 있어야 한다. 본 연구에서는 이러한 관점에서 시나리오 생성 연구를 수행한다. 생성된 시나리오는 협업시스템에서 사용될 수 있다. 예를 들어, A→B→C로 구성된 시나리오가 있다면, 사용자가 A 행동을 하고, 로봇이 B, C 행동을 사용자 대신 수행할 수 있다. Task-graph[4] 및 affordance 기술[5]을 활용한다면 보다 고차원의 협업이 가능할 것이다.

시나리오 기반 기술에서 행동 시나리오의 생성기법은 시스템의 성능에 매우 중요하다. 수집된 데이터만으로는 시나리오가 제한적일 수밖에 없고, 사용자 만족도가 떨어지는 결과를 초래한다[3]. 따라서 본 논문에서는 **존재하는 시나리오 데이터셋을 바탕으로 새로운 시나리오를 생성하는 모델의 학습 방법론**에 대하여 다루고자 한다.

본 연구에서는 행동 시나리오 생성과 그 방법론에 대해서 제안한다. 2장에서는 문제 정의를 통해 행동과 시나리오, 그리고 생성 문제를 정의한다. 이어 3장에서는 모델과 실험 데이터의 세부사항을 적시하고 4장에서 실험 결과를 정리한다.

그림 1. 하나의 행동을 표현하는 Scene-graph의 예시. 사람이 책상 앞에서 의자에 앉아 컵으로 무언가를 마시는 장면(왼쪽)을 그래프(오른쪽)로 표현한다.
그림출처: ActionGenome [1]

2. 문제 정의

2.1. Scene-Graph: 행동(Action)의 표현

본 연구에서는 행동을 사람과 물체 사이의 관계로 정의하는 scene-graph 표현을 사용한다[1]. Scene-graph에서는 이러한 행동을 그래프 G로 표현한다. **그림 1**에서 나타나는 것과 같이, G는 <사람-관계-물체>로 구성된 directed graph이며, 이를 통해 각 장면에 해당하는 사람의 행동을 정밀하게 표현할 수 있다. 관계에 해당하는 node는 공간관계 (예: 위, 아래), 접촉관계 (예: 앉아, 먹다), 주의관계 (예: 보다)로 구성되어 있다. Scene-graph에 대한 보다 자세한 정의는 기존 연구에 사용된 것을 참조할 수 있다 [1].

2.2. 시나리오 (Scenario): 행동의 시퀀스

시나리오는 개별 행동 G_i의 시퀀스로 구성되어 있다. 즉, 시나리오 S는 다음과 같이 정의된다:

$$S = [(G_1, T_1), (G_2, T_2), (G_3, T_3), \ldots, (G_N, T_N)],$$

이 논문은 2022년도 KT의 재원으로 “AI One Team – 로봇 소셜인터랙션 기술개발” 의 지원을 받아 수행된 연구 결과입니다. (KT과제고유번호: B220002198)

각 행동은 행동이 일어난 시간 $T_i \in \mathbb{R}$를 함께 기록한다. 이하에서 간결성을 위해 행동이 일어난 시간 T_i는 G_i로 축약된 것으로 한다.

2.3. 행동 예측을 통한 시나리오 생성

행동 예측. 행동 예측모델 f_θ 정의하고 그것을 통해 시나리오를 생성하는 방식을 제안한다. f_θ는 이전 행동의 시퀀스 $(G_1, G_2, \ldots, G_{n-1})$가 주어졌을 때, 다음 행동 $(\mathrm{G_n})$을 예측하는 모델이다 (**그림 2**). 즉, 다음 행동을 예측하는 모델 f_θ는 조건부 확률 $P(G_n|G_{n-1}, \ldots, G_1)$을 학습하는 생성 모델이다.

시나리오 생성. 행동 예측모델 f_θ이 $P(G_{n+1}|G_n, \ldots, G_1)$를 나타내기 때문에, 예측된 $n+1$ 번째 행동 $\hat{G}_{n+1} = f_\theta(G_n, G_{n-1}, \ldots, G_1)$을 재귀적으로 활용하여 $n+2$번째 행동 $\hat{G}_{n+2} = f_\theta(\hat{G}_{n+1}, G_n, \ldots, G_1)$을 예측할 수 있다. 이러한 구조는 **그림 3a**에 묘사되어 있는 것처럼, 재귀적으로 계속되는 행동 시퀀스를 만들어낼 수 있다. 주어진 행동 시퀀스에서 예측된 행동을 추가함으로써 기존 행동으로 조합되지 않는 새로운 시나리오를 생성할 수 있다. 이에 적합한 모델 f_θ로서는 RNN (**그림 3b**)를 활용할 수 있다.

3. 시나리오 데이터셋과 예측 모델

3.1. ScenarioGenome (SG): 실험 데이터셋

본 연구는 비디오 기반 scene-graph 데이터셋인 ActionGenome (AG) 데이터셋[1]을 기반으로 본 저자들이 가공한 데이터셋, ScenarioGenome (SG) 데이터셋을 기반으로 한다.

SG 데이터셋은 AG데이터셋의 그래프를 기반으로 시나리오 예측을 수행할 수 있도록 가공되었다. 우선, 행동 G는 36개의 객체($|O| = 36$)와 26개의 행동($|A| = 26$)을 조합해 $|O| \times |A| = 936$ 차원의 vector로 표현한다. 각 차원은 사람과 객체가 어떤 관계로 연결되었는지 여부를 나타낸다. 이는 AG데이터셋의 일반적인 프로토콜과 일치한다[1]. 다음으로, 시나리오 S는 다음과 같이 행동의 시퀀스와 함께 이전 행동으로부터의 상대 시간 $T_i - T_{i-1}$을 정의한다:

$$S = \{(G_1, 0), (G_2, T_2 - T_1), (G_3, T_3 - T_2), \ldots, (G_N, T_N - T_{N-1})\}.$$

데이터셋은 학습/테스트가 구분되어 있고, 학습데이터는 218,413개의 행동 및 7,787개의 시나리오, 테스트 데이터는 70,369개의 행동 및 1,814개의 시나리오로 구성되어 있다. 각 시나리오를 구성하는 평균 행동의 개수는 23개이다. 학습 및 테스트시에는 2개 이상의 행동 시퀀스를 추출 후 그 다음의 행동을 예측한다.

3.2. 예측 모델

예측모델 f_θ는 RNN모델의 일종인 GRU[6]로 정의한다. **그림 3b**에서와 같이, f_θ는 다음 행동 $\hat{G}_{n+1}$을 예측한다: $\hat{G}_{n+1} = f_\theta(G_n, \ldots, G_1)$. 본 논문에서는 8레이어 Stacked GRU에 3레이어 MLP를 붙인 간단한 구조를 활용하였다. 이 구조는 시퀀스 길이가 다른 시나리오를 하나의 구조로 처리할 수 있어 유리하다. 내부적으로 Stacked GRU와 MLP의 차원은 128차원으로 고정하였다.

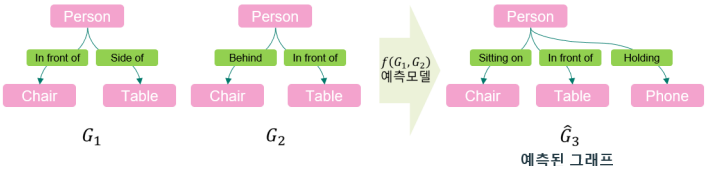
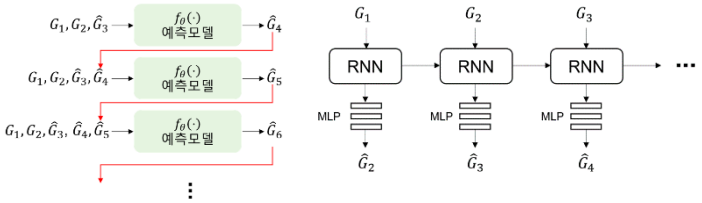


그림 2. 행동 예측 모델의 예시



(a) 시퀀스 생성 (b) RNN 구조의 적용

그림 3. 시나리오 예측 모델의 개요도

3.3. 손실 함수 (Loss Function)

3.1절에서 언급하였듯이, 행동 G의 표현은 객체와 행동의 조합으로 나타낼 수 있다. 여기서 각 차원은 어떤 객체 i가 사람과 어떤 관계 j를 맺고 있는지에 대한 여부 y_{ij} (0 또는 1)로 나타내어진다. 각 i, j를 독립적으로 보고 우도를 최대화하는 방식으로 모델을 학습한다. 즉, 다음과 같이 정의되는 binary classification loss를 활용할 수 있다:

$$L_{BCE} = -\frac{1}{\mathrm{N}} \sum_{n}^{N} \sum_{i,j} y_{nij} \log(\hat{y}_{nij}) + (1 - y_{nij}) \log(1 - \hat{y}_{nij}),$$

식은 N개의 데이터, 사람-관계의 여부의 참값 y_{nij}, 그리고 예측확률 $\hat{y}_{nij}$로 정의된다.

하지만, SG데이터셋의 행동 데이터 특성상, 대부분의 참값이 0인 sparse 특성을 보인다 (26개의 관계중 1~2개만이 활성화). 이 경우, 일반적인 BCE 함수로 학습할 경우 모델이 바람직하지 않은 방향으로 학습될 염려가 있다. 따라서, 다음과 같이 정의되는 balanced binary classification loss를 활용한다:

$$L_{BBCE} = -\frac{1}{\mathrm{N}} \sum_{n}^{N} \sum_{i,j} \frac{Z_0}{Z_1} y_{nij} \log(\hat{y}_{nij}) + \frac{Z_1}{Z_0} (1 - y_{nij}) \log(1 - \hat{y}_{nij}),$$

여기서 $\mathrm{Z}_1 = \sum y_{ij}$, $\mathrm{Z}_0 = \sum 1 - y_{ij}$ 로 정의되며, 각각 참

값에서 1의 개수와 0의 개수를 의미한다. 이는 class 불균형 문제를 해결하는 방식으로 적은 클래스에 대해서 과대샘플링(oversampling)하는 방식과 같다.

4. 실험 결과

4.1. 실험세부사항

모델의 학습은 SG데이터셋의 학습 데이터를 이용해 학습하고 테스트 데이터를 이용해 평가하였다. 학습 세팅은 다음과 같다. Batch size는 2048, learning rate는 0.001에서 2,000스텝마다 0.5비율로 감소하였다. 최대 스텝수는 10,000스텝으로 고정하였다.

베이스라인으로 Replicate, MLP를 사용하였다:

Replicate. 시나리오의 특성상 어떤 행동 (예: 사람이 책상 앞에 있다)이 일어났다면, 다음 행동도 앞선 행동을 포함할 가능성이 높다 (예: 사람이 책상 앞에서 책을 본다). 즉, 앞선 행동을 그대로 반복하였을 때의 성능을 가장 기초적인 성능지표로 활용할 수 있다.

MLP. RNN대신, MLP를 사용하여 다음 행동을 예측하는 모델이다. 이 경우, GRU대신에 8레이어 MLP를 사용하고 average pooling을 통해 정보를 합쳐준 다음 3레이어 MLP를 통과하여 다음 행동을 예측한다.

4.2. 정량적 비교

평가척도 (metric). 생성된 시나리오가 얼마나 현실적인지는 각 객체(예: phone)에 대하여 관계(예: hold)를 얼마나 잘 예측했는지로 평가할 수 있다. 본 논문에서는 multi-label classification에서 사용하는 mAP (mean average precision)와 Recall@K 방식을 사용한다. 한 물체에 대한 관계가 여러 종류로 동시에 나타날 수 있기 때문에, classification accuracy는 적합하지 않다.

표 1. 모델 종류에 따른 성능

Model	mAP	R@1	R@10	R@20
Replicate	0.746	0.242	0.829	0.936
MLP	0.702	0.202	0,978	0.999
GRU	**0.806**	**0.266**	**0.991**	**1.000**

모델 종류에 따른 성능. 표 1에서는 여러 모델의 성능을 비교한다. 평가는 SG데이터의 1,814개의 시나리오에서 2개 이상의 모든 행동 시퀀스에 대해 예측 및 평가를 수행하였다. 먼저, GRU모델이 다른 baseline에 비해 모든 평가지표에서 높은 성능을 보이는 것을 확인할 수 있다. 주목할 점은, Replicate 모델도 상당히 높은 성능을 보이는데, 이는 바로 이전 행동을 포함하는 다음 행동이 많이 존재한다는 것을 보여준다. MLP 모델은 단순한 Replicate 모델보다 mAP와 Recall@1에서 낮은 성능을 보이는데, 이는 MLP모델이 시나리오 예측에 적합하지 않다는 것을 보여준다.

표 2. 손실함수에 따른 성능

Model	mAP	R@10	TPR ↑	FNR ↓
MLP + BCE	**0.713**	**0.978**	0.235	0.765
MLP + BBCE	0.702	**0.978**	**0.738**	**0.262**
GRU + BCE	**0.809**	0.989	0.094	0.906
GRU + BBCE	0.806	**0.991**	**0.935**	**0.065**

손실함수에 따른 성능. 표 2에서는 기본적인 손실함수인 BCE와 제안한 BBCE 손실함수를 비교한다. mAP와 Recall 성능에서는 두 손실함수가 유사한 성능을 보이고 있지만, True Positive Rate (TPR)과 False Negative Rate (FNR)에서 BBCE 손실함수가 시나리오 생성에서 중요한 역할을 하는 true positive를 늘리고 false negative를 줄이는 역할을 한다는 것을 확인할 수 있고, 모델은 이에 따른 이점을 누린다.

5. 결 론

본 연구에서는 행동 예측기를 통한 시나리오 생성 방법에 대해서 탐구하였다. 시나리오의 정의와 행동 예측기를 활용한 시나리오 생성의 방법론을 정의하고 시나리오 데이터셋(SG 데이터셋)을 구축하였다. 행동 예측기는 RNN을 기반으로 모델을 설계하여, MLP모델 대비 큰 성능 향상을 보였다. 또한 class의 불균형한 문제를 해결하기 위해서 balanced binary cross entropy 손실함수를 적용하였다. 제안 모델과 손실함수의 효용을 분석하기 위해서 SG 테스트셋에서 검증하였다.

6. 참고 문헌

[1] Ji, Jingwei, et al. "Action genome: Actions as compositions of spatio-temporal scene graphs. CVPR. 2020.

[2] Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." NIPS. 2014.

[3] Hoffman, Guy. “Anki, Jibo, and Kuri: What We Can Learn from Social Robots That Didn't Make It”, IEEE Spectrum, 2019.

[4] Darvish, Kourosh, et al. "A hierarchical architecture for human-robot cooperation processes." IEEE T-RO, 2020

[5] Do, Thanh-Toan, et al.. "Affordancenet: An end-to-end deep learning approach for object affordance detection." ICRA, 2018.

[6] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." arXiv:1409.1259. 2014.