시각-언어 이동 작업을 위한 장소 미리보기 메모리

오선택, 김인철
경기대학교 컴퓨터과학과
email:choice37@kyonggi.ac.kr, kic@kyonggi.ac.kr

Lookahead Place Memory for Vision-Language Navigation Tasks

Suntaek Oh, Incheol Kim
Department of Computer Science, Kyonggi University

요 약

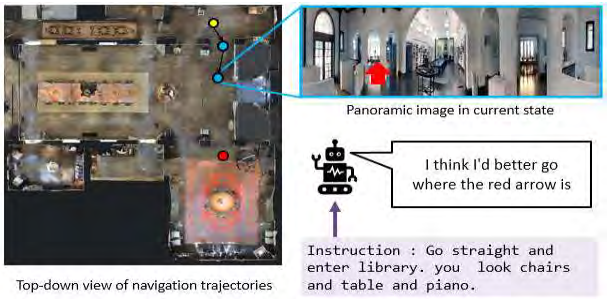
시각-언어 이동 작업은 에이전트가 주어진 지시를 따라 특정 실내 공간 내에서 목적 위치로 이동하는 작업이다. 시각-언어 이동 작업의 특성상 자연어 지시 속에 등장하는 랜드마크인 장소 정보를 인지하는 것은 작업을 수행하는 데 큰 도움이 된다. 본 논문에서는 환경을 구성하는 주요 장소 정보를 저장하기 위한 장소 미리보기 메모리를 제안한다. 에이전트는 장소 미리보기 메모리에 저장된 장소 정보를 고려하여 작업을 수행하게 된다. 본 논문에서는 Matterport3D 시뮬레이션 환경에서의 실험을 통해 R2R 벤치마크 데이터 집합에서 가장 높은 성능을 보였다.

1. 서론

최근 복합 지능형 에이전트에 관한 관심이 높아지면서 VLN(Vision-and-Language Navigation) 작업[1]이 주목받고 있다. VLN 작업의 목표는 실시간으로 주변 환경에 대한 영상(image)을 입력으로 받는 에이전트에게 고-수준의 자연어 지시(natural language instruction)를 주고 이 지시에 따라 자율적으로 기본 이동 동작들을 수행함으로써 성공적으로 목적 위치까지 도달하게 하는 것이다. (그림 1)은 VLN 작업의 한 예를 나타낸다. (그림 1)의 왼쪽은 에이전트가 놓인 환경에 대한 하향식 보기(top-down view)이고 노란색 노드는 시작 위치, 빨간색 노드는 목표 위치, 파란색 노드는 에이전트가 이동한 노드를 각각 나타낸다. 오른쪽은 현재 위치의 입력 파노라마 영상(panoramic image)과 자연어 지시에 기초하여, 에이전트가 선택한 이동 방향/동작을 보여주고 있다.

일반적으로 이와 같은 VLN 작업 에이전트의 효율적인 행동 정책 학습을 위해, R2R[1]과 같이 자연어 지시와 이에 대응하는 최적 경로의 쌍들이 학습 데이터로 제공된다. 하지만 효과적인 VLN 작업 에이전트의 구현을 위해서는 학습 과정에서 보지 못한 미-경험 환경(unseen environment)들에서도 작업을 잘 수행할 수 있도록 설계하는 것이 매우 주요 과제 중 하나이다. 이러한 도전과제를 해결하고자 기존 연구들에서 다양한 시도들이 이루어졌으나, 새로운 환경에서의 사전 탐사(pre-exploration)를 활용한 연구들이 큰 효과를 보였다. 여기서 사전 탐사란 새로운 환경에서 본격적인 행동 정책 학습 또는 검증 과정을 수행하기 전에 에이전트에게 미리 이 환경에 관한 경험 기회를 제공하는 것을 의미한다. 이 사전 탐사 과정 동안에 에이전트는 환경 내부를 스스로 돌아다니면서 입력 영상을 수집할 수는 있으나, R2R 데이터와 같은 자연어 지시와 정답 경로들은 에이전트에게 주어지지 않는다. 사전 탐사의 기회를 활용한 대표적인 기존 연구들로는 [2-4]가 있다. 이 연구들에서는 사전 탐사를 통해 에이전트 스스로 새로운 환경에서 지시와 경로의 쌍으로 구성된 R2R 형태의 학습 데이터를 생성하여 행동 정책 학습에

* 이 연구는 2020년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임('10077538')



(그림 1) 시각-언어 이동(VLN) 작업 예시

이용하는 접근 방식을 제안하였다. 이를 위해 사전 탐사로 4~6걸음(step)의 경로를 경험하고 [5]에서 제안한 발화자(speaker) 모델을 사용하여 그 경로를 설명하는 지시를 생성하는 방식을 사용하였다. 하지만 이러한 선행 연구는 사전 탐사 과정에서 얻을 수 있는 장소에 대한 정보를 불완전한 발화자 모델에 의존하고 있다. 또한, 환경에서 장소 정보를 파악하고 활용하는 방법이 정책 네트워크의 종단간 학습(end to end learning)에 의존하기 때문에 제대로 이루어지는지 알 수 없다는 문제가 존재한다.

본 논문에서는 이러한 문제점을 해결하기 위해서 레이블링(labeling)된 장소 데이터를 활용하여 에이전트의 이동 방향에 따라 마주할 가능성이 있는 장소들을 장소 미리보기 메모리 LPM(Lookahead Place Memory)에 저장하고 이를 활용하는 방법을 제안한다. LPM은 이동 방향의 부분 영상과 해당 방향으로 이동했을 때 마주할 미리보기 장소 lp(lookahead place)를 키와 값으로 갖는 메모리이다. 에이전트는 환경마다 특정 상태에서 특정 방향으로 이동했을 경우 마주할 가능성이 있는 장소를 계산하여 LPM에 저장하게 된다. 이때 학습 데이터에 없는 미-경험 환경은 사전 탐사를 통해 경험할 수 있도록 하였다. 이후 에이전트의 행동 결정 과정에서 저장된 미리보기 장소 정보를 고려하여 더 나은 행동 결정을 내릴 수 있도록 하였다. 본 논문에서는 Matterport3D[6] 시뮬레이션 환경에서 R2R 데이터 집합을 이용한 다양한 실험들을 통해, 제안 모델의 우수한 성능을 입증한다.

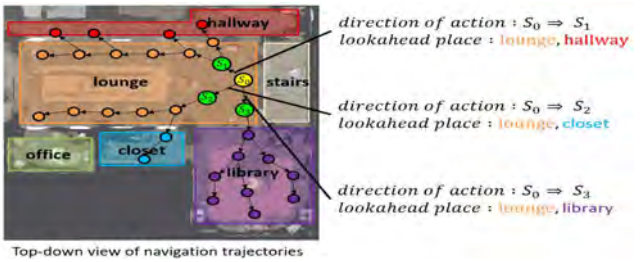
2 시각과 언어기반의 이동

2.1 문제 정의

VLN 작업은 실시간으로 영상을 입력받는 에이전트가 경로를 설명하는 지시를 따라 목적지까지 이동하는 작업이다. 지시 $I=\{u_0,u_1,..,u_l\}$는 최대 l개의 단어 u_i들로 구성되어 있고 n개의 순차적인 상태로 이루어진 정답 경로 $R = <s_1, s_2, \ldots, s_{n-1}, s_n>$을 설명하고 있다. 이때 상태 $s \in S$는 특정 순간에 에이전트가 얻을 수 있는 영상정보(v)와 방향 정보(ψ, θ)로 나타낼 수 있다. v는 특정 위치에서 에이전트에게 입력되는 360° 파노라마 영상이다. 파노라마 영상은 가로로 30°씩 12개, 위아래 30°씩 3개로 총 36개의 부분 영상으로 이루어져 있다. ψ와 θ는 각각 수평(elevation), 수직(heading) 방향(orientation)을 의미한다. 에이전트는 매 순간 지시와 상태 정보를 고려하여 행동을 선택하게 된다. 가능한 행동 $a \in A$는 36개 이내의 이동 가능한 방향(navigable directions) 중 하나로 직진 또는 정지이다.

2.2 장소 미리보기 메모리

시각-언어 이동 작업을 수행하기 전, 환경에서의 사전 탐사를 통해 환경 정보를 활용할 수 있다. 사전 탐사 과정에서 에이전트는 환경에 대한 시각정보만을 경험할 수 있다. 본 논문에서는 사전 탐사를 통해 미리보기 장소 정보를 별도의 메모리에 저장하고 활용하는 방법을 제안한다. 장소 정보는 Matterport3D에서 제공하는 정답 데이터(ground truth)를 활용했다. 이 데이터는 환경과 상태에 따라 30개의 장소 클래스로 구분된다. (그림 2)는 미리보기 장소를 설명하기 위한 예를 나타낸다.



(그림 2) 미리보기 장소 예시

(그림 2)의 왼쪽은 하향식 보기이고 노란색 노드는 에이전트의 현재 위치, 초록색 노드는 현재 위치에서 이동 가능한 위치, 나머지 노드는 에이전트가 움직인 경로상의 노드를 의미한다. 이때 경로는 현재 위치에서 특정 위치까지의 최단 경로만 포함한다. 최단 경로만은 고려한 이유는 에이전트가 최적 행동을 취했을 때 마주할 수 있는 장소 정보를 알기 위함이다. (그림 2)에서 에이전트가 현재 위치인 S_0에서 특정 위치까지 최단 경로로 가기 위한 첫 번째 행동의 결과가 S_1이라면 미리보기 장소는 특정 위치의 장소가 될 수 있는 장소 집합이 된다.

미리보기 장소 정보를 저장하는 별도의 메모리는 장소 미리보기 메모리 LPM(Lookahead Place Memory)라고 정의하였다. LPM은 키-값(key-value) 형태의 메모리로 <표 1>과 같은 형태를 보인다. LPM의 목적은 현재 위치에서 목적 위치까지 최단 경로로 이동한다고 가정했을 때, 현재 위치에서 이동 가능한 방향 중 하나로 이동함에 따라 목적 위치가 될 수 있는 장소 정보를 저장하는 것이다. 따라서 LPM의 키로써 이동 가능한 방향의 행동 영상 av(action view)가 사용된다. av는 '환경(env), 상태(state), 행동(action)'에 따라 고유한 값을 갖는다. 그리고 해당 방향으로 이동 시 얻을 수 있는 미리보기 장소 lp(lookahead place)가 키에 대한 값으로 사용된다. 이때 lp는 장소를 구분하는 30개의 클래스마다 등장 여부에 따라 0과 1로 이루어진 배열로 표현한다.

<표 1> LPM 구조

LPM	
KEY(Candidate View)	VALUE(Lookahead Place)
 (e_0, s_0, a_0)	[0, 1, 0 … 1, 0, 0]
 (e_0, s_0, a_1)	[0, 0, 0 … 1, 1, 0]
⋮	
 (e_{89}, s_i, a_j)	[0, 0, 1 … 0, 0, 0]

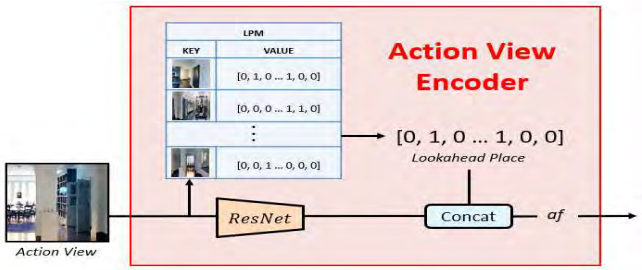
LPM에 필요한 정보를 저장하기 위해 사전 탐사 과정에서 얻어야 할 정보는 특정 상태에서 이동 가능한 행동 영상 av, 해당 방향으로 이동 시 얻는 미리보기 장소 정보이다. 본 논문에서 사용한 사전 탐사 알고리즘은 <표 2>와 같은 의사 코드(pseudo code)로 표현될 수 있다. 우선 1-5번 줄을 통해 환경마다 가능한 모든 경로를 추출한다. 이때 경로는 시작 상태에서 목적 상태까지 다익스트라 알고리즘(Dijkstra algorithm)을 이용한 최단 거리로 이동한 경로이다. 예를 들어 5번 줄의 dijkstra(s1, s2)는 상태 s1에서 s2로 가는 최단 경로를 의미한다. 그리고 6번 줄을 통해 길이가 2~7걸음으로 이루어진 경로만을 고려하도록 경로 길이를 제한하였다. 제한된 경로 길이는 3.2절의 실험에 따라 결정하였다. 이후 7번 줄에서 경로의 첫 번째 상태에서 두 번째 상태로 가기 위한 행동 영상 av를 구하고, 8번 줄에서 경로상에 존재하는 상태가 갖는 장소 정보를 통해 lp를 구한다. 9번 줄에서는 7-8번 줄에서 구한 정보를 LPM의 키와 값으로 저장한다. 이때 같은 키의 값이 저장되어있는 경우 기존 값에 새로운 값을 추가하게 된다.

<표 2> 사전 탐사 알고리즘

Algorithm 1 Algorithm in Pre-exploration

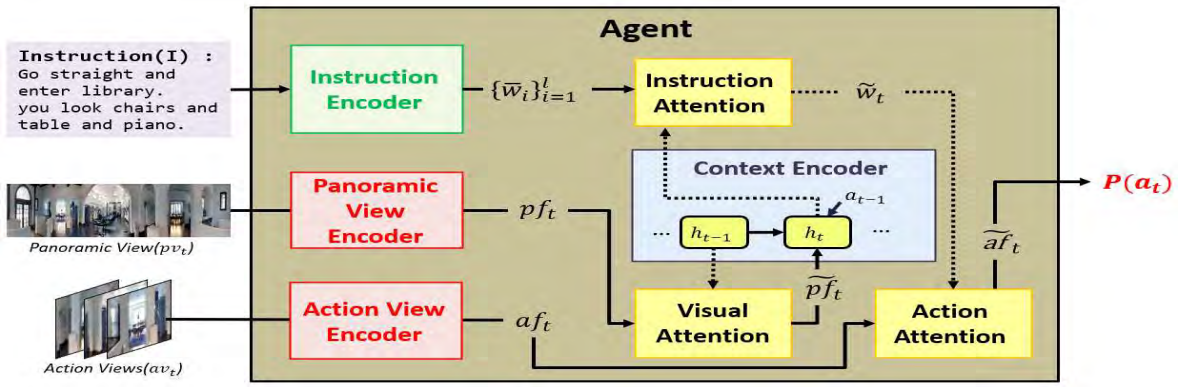
```
1 envs = All environments in R2R data
2 for e in envs:
3     for s1 in states(e):
4         for s2 in states(e):
5             path = dijkstra(s1, s2)
6             if 1 < len(path) and len(path) < 8:
7                 key = cv(path[0], path[1])
8                 value = lp(path)
9                 LPM[key] += value
```

학습과 검증 과정에서는 에이전트의 정책 네트워크가 LPM에 저장된 장소 정보를 고려하여 행동을 결정한다. 에이전트는 환경으로부터 입력받은 이동 가능한 행동 영상을 에이전트 모델의 내부에 있는 행동 영상 인코더(action view encoder)에서 처리한다. (그림 3)는 행동 영상 인코더의 구조를 보인다.



(그림 3) 행동 영상 인코더

우선 행동 영상을 LPM에 키 입력으로 주면 그에 맞는 미리보기 장소 lp 값을 얻게 된다. 이후 (식 1)과 같이 ResNet을 거친 행동 영상과 장소 정보 lp를 연결



(그림 4) VLN 에이전트 모델의 구조

(concatenation)하여 행동 영상에 대한 특징 벡터 af로 활용한다. af를 활용하여 행동을 선택하는 과정은 2.3절에서 소개한다.

$$lp = LPM(cv)$$
$$af = [ResNet(cv); LPM(cv)] \quad \text{(식 1)}$$

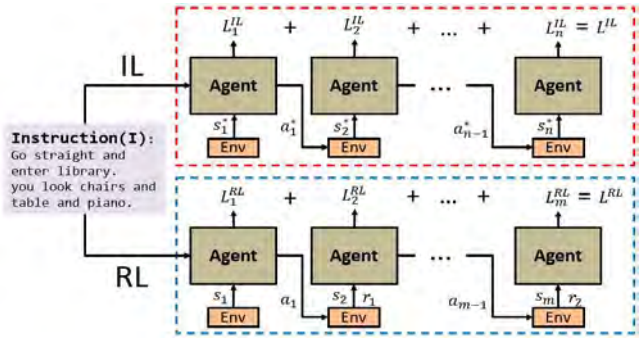
본 논문에서 제안한 장소 미리보기 메모리를 활용하는 방법은 에이전트가 행동을 결정하는 데 있어 과거 경험의 메모리를 통해 현재 위치에서 보이지 않는 장소 정보까지 고려할 수 있도록 만들어준다.

2.3 VLN 에이전트 모델

본 논문에서는 VLN 작업을 수행하기 위해 [3]에서 제안한 인코더-디코더(encoder-decoder) 기반의 에이전트 모델을 기초 모델 (baseline)로 사용한다. 에이전트 모델의 구조는 (그림 4)와 같다. 에이전트에게 지시(instruction)가 주어지면 지시 인코더(instruction encoder)에서는 주어진 지시를 단어 임베딩(word embedding) 벡터 $\{w_i\}_{i=1}^{l}$로 변환한다. l은 단어의 수를 의미하며 최대 80으로 제한된다. 이후 순차적인 정보를 고려할 수 있도록 Bi-LSTM(Bidirectional Long Short-Term Memory)을 거친 특징 벡터 $\{\overline{w}_i\}_{i=1}^{l}$를 생성한다.

이후 매 순간 환경으로부터 에이전트 위치에 따른 파노라마 영상 pv_t와 현재 위치에서 이동 가능한 행동 영상들 av_t을 입력받는다. 이러한 영상정보는 에이전트의 시각 인코더(view encoder)에 특징 벡터를 추출하게 된다. 시각 인코더에서는 기본적으로 영상정보가 사전학습된 ResNet을 거치고 에이전트가 바라보는 방향 정보와 연결(concatenation)하는 과정을 거친다. 방향 정보는 가로 방향 ψ_t와 세로 방향 θ_t에 의해 $(\cos\theta_t, \sin\theta_t, \cos\psi_t, \sin\psi_t)$으로 표현한다. 또한, 에이전트는 특징 벡터 af_t을 구하는 과정은 2.2절에서 설명한 것과 같이 LPM에 저장된 미리보기 장소 정보를 av_t에 연결한다.

에이전트는 매 순간 수행된 지시와 수행할 지시를 파악하기 위한 문맥 정보를 담는 LSTM(Long Short-Term Memory) 기반의 문맥 인코더(context encoder)를 갖는다. 에이전트가 행동을 선택하기 위해 우선 생성된 파노라마 영상의 특징 벡터 pf_t에 이전 문맥 정보 h_{t-1}로 주의 집중을 가해 현재까지의 문맥 정보를 반영한 주의 집중된 파노라마 벡터 $\widetilde{pf}_t$를 생성한다. 이후 $\widetilde{pf}_t$을 통해 문맥 인코더에서 이전 문맥 정보 h_{t-1}를 새로운 문맥 정보 h_t로 갱신한다. 다음으로 갱신된 h_t를 사용하여 임베딩된 지시 벡터 $\{\overline{w}_i\}_{i=1}^{l}$에서 현재 따라야 할 지시정보에 주의 집중을 가하여 주의 집중된 지시 벡터 $\widetilde{w}_t$를 생성한다. 마지막으로 이동 가능한 행동 영상의 특징 벡터 af_t에 $\widetilde{w}_t$로 주의 집중을 가해 주의 집중된 영상 벡터 $\widetilde{af}_t$를 생성하는 과정에서 행동 확률 분포 $P(a_t)$를 계산할 수 있게 된다.



(그림 5) 모방 학습과 강화 학습 에피소드

제안 모델을 학습하기 위해서는 (그림 5)와 같이 매 학습 주기마다 모방 학습 손실(imitation loss) IL을 계산하는 에피소드와 강화 학습 손실(reinforcement loss) RL을 계산하는 에피소드를 한 번씩 교대로 진행한다. 이와 같이 행동 정책 학습에 혼합 손실(mixed loss)을 이용하는 것은 낮은 데이터 효율성을 갖는 강화 학습과 학습 데이터에 편향성을 갖는 모방 학습의 문제를 상호 보완하기 위함이다. 각 에피소드에서 에이전트는 정지 행동을 선택하거나 최대 반복횟수에 도달할 때까지 행동을 선택하는 과정을 반복하게 된다. 모방 학습 에피소드에서는 반복 과정에서 매 순간 교차 엔트로피(cross entropy)를 이용한 모방 학습 손실 L^{IL}를 계산하고, 강화 학습 에피소드에서는 A2C(advantage actor-critic) 알고리즘을 기반으로 강화 학습 손실 L^{RL}을 계산한다. 각 손실을 계산하는 식은 (식 2)와 같다.

$$L^{IL} = -\sum_{t=1}^{N} a_t^* \log \pi_{\theta_p}(p_t) \quad \text{(식 2)}$$
$$L^{RL} = -\sum_{t=1}^{M} a_t \log \pi_{\theta_p}(p_t) A_t$$
$$L^{MIX} = \lambda_{IL} L^{IL} + L^{RL}$$

여기서 A_t는 A2C 알고리즘에서의 우세 함수(advantage function)를 의미한다. 이후 두 손실 값을 더한 혼합 손실 L^{MIX}를 사용하여 에이전트를 학습한다. 이때 λ_{IL}는 두 학습 손실 값의 불균형 문제를 해결하기 위한 손실 가중치이다. 이러한 혼합 손실 알고리즘은 [3]에서 제안한 알고리즘을 채용하였다.

3. 구현 및 실험

3.1 데이터 집합과 모델 학습

실험 환경은 Python 3.7, Pytorch 1.2 라이브러리를 이

용하여 구현하였다. 에이전트를 학습하고 평가하기 위해 Matterport3D[1]에서 제공하는 R2R 데이터 집합을 사용하였다. R2R 데이터는 시작 위치에서 목적 위치로 이동하는 4~6걸음의 경로와 그 경로를 설명하는 세 개의 자연어 지시로 구성되어 있다. R2R 데이터 집합에서 학습 데이터(training data)는 14,025개, 경험 환경 검증 데이터(seen validation data)는 1,020개, 미-경험 환경 검증 데이터(unseen validation data)는 2,349개, 비-경험 환경 테스트 데이터(unseen test data)는 2,349개의 지시로 각각 구성된다. 시뮬레이터 상의 상태마다 입력되는 RGB 영상의 특징 추출을 위해서는 사전 학습된 ResNet-152 모델을 이용하였다. 배치 크기(batch size)는 64, 손실 가중치 λ_{IL}는 0.05, 학습률(learning rate)은 0.0001로 각각 설정하였다.

3.2 성능 분석 실험

본 논문에서는 LPM을 적용 후의 효과를 분석하고, 기존 모델들과의 비교를 통해 제안 모델의 우수성을 입증하기 위한 실험을 수행하였다. 실험에 사용된 성능 평가 척도는 성공률 SR(Success Rate)과 경로 길이를 고려한 성공률 SPL(Success rate weighted by Path Length)이다. VLN 작업은 마지막 위치가 목적지와의 거리가 3m 이내일 때 성공으로 간주한다. SPL은 정답 경로의 길이를 에이전트가 실제 이동한 경로의 길이로 나눈 0에서 1 사이 값으로 성공의 정도를 평가하게 된다.

첫 번째 실험은 본 논문에서 제안한 미리보기 장소 메모리(LPM)의 긍정적 효과를 입증하기 위한 실험이다. 또한 이 실험에서는 'n 걸음 앞 미리보기 장소 저장'에서 n의 크기에 따른 성능 변화도 함께 분석하고자 한다. 이 실험을 위해 경험한 환경(seen)뿐만 아니라 미-경험 환경(unseen)에서도 에이전트의 사전 탐사를 통해, 미리보기 장소 lp에 대한 정보를 LPM에 저장하도록 하였다.

<표 3> LPM 적용 전과 후의 성능 비교

Pre-Explore Setting	Seen		Unseen	
	SR	SPL	SR	SPL
none	0.634	0.602	0.489	0.449
LPM (n = 4)	0.665	0.638	0.572	0.536
n = 5	0.662	0.637	0.597	0.563
n = 6	0.677	0.654	0.613	0.581
n = 7	**0.691**	**0.662**	**0.634**	**0.605**
n = 8	0.680	0.654	0.631	0.601
n = 9	0.677	0.653	0.626	0.598
n = 10	0.685	0.659	0.626	0.599

이 실험의 결과는 <표 3>과 같다. 표에서 none은 미리보기 장소 메모리(LPM)를 사용하지 않는 모델을 나타낸다. 실험 결과를 보면 lp를 사용하는 것이 기본 에이전트보다 성능 향상이 일어남을 확인할 수 있다. 이를 통해 미리보기 장소 정보를 활용하는 것이 에이전트의 시각-언어 이동 작업에 도움을 준다는 것을 확인된다. 특히 미-경험 환경(unseen env)에서 기본 에이전트보다 큰 폭의 성능 향상이 일어난다. 이는 각 환경의 장소 정보에 대한 학습이 작업 성능에 큰 영향을 미친다는 것을 의미한다. 또한, n이 6-10일 때 비슷한 수준에서 변동이 일어나고 7걸음 앞 미리보기 장소를 활용하는 것이 최고의 성능을 보인다. 이는 R2R 데이터의 경로가 최대 6걸음으로 이루어진 특성에 기인한 것으로 판단된다.

두 번째 실험은 기존의 VLN 모델들과의 성능을 비교를 통해, 제안 모델의 우수성을 입증하기 위한 실험이다. 이 실험에서는 발화자 모델을 이용해 에이전트가 경로를 잘 따랐는지 판별한 RCM[2], 학습 데이터 증강(data argumentation)을 위한 환경 드롭아웃(dropout) 기능과 혼합 손실 함수(mixed loss function)를 채용한 Env-Dropout[3], 보조 작업(auxiliary task)을 통해 에이전트의 추론 능력을 높인 AuxRN[4], 그리고 장소 미리보기 메모리를 활용한 본 논문의 제안 모델 LPM의 VLN 작업 성능을 비교하였다. 공정한 비교를 위해 각 모델이 미-경험 환경을 사전 탐사하는 것을 허용하였다. 각 모델의 학습을 위해 공통적으로 사전 탐사를 통해 학습 데이터를 증강(data augmentation)하였고, 본래의 R2R 데이터와 증강된 데이터를 모두 학습에 이용하였다.

<표 4> 기존 모델들과의 성능 비교

Model	Seen		Unseen	
	SR	SPL	SR	SPL
RCM[2]	0.67	-	0.61	-
AuxRN[4]	0.70	0.67	-	-
Env_Dropout[3]	0.62	0.59	0.65	0.61
LPM	**0.73**	**0.70**	**0.73**	**0.71**

이 실험의 결과는 <표 4>와 같다. 비교 모델 중에서 본 논문의 제안 모델이 모든 척도에서 가장 높은 성능을 보이는 것을 확인할 수 있다. 특히 미-경험 환경(unseen)에서도 비교 모델들보다 작업 성능이 향상된 것으로 보아 사전 탐사를 통한 환경에 대한 경험을 효과적으로 활용하였음을 입증한다.

4. 결 론

본 논문에서는 새로운 환경에서의 효과적인 시각-언어 이동 작업을 위해 장소 미리보기 메모리(LPM)를 채용한 새로운 VLN 에이전트 모델을 제안하였다. LPM은 에이전트의 이동 방향에 따라 마주할 가능성이 있는 장소들을 사전 탐사 과정을 통해 미리 저장하는 메모리이다. 본 논문에서는 이러한 LPM을 활용하여 미-경험 환경에서도 효과적으로 행동 정책을 학습할 수 있는 학습 전략도 제시하였다. 그리고 Matterport3D 시뮬레이션 환경과 R2R 데이터를 활용한 다양한 실험을 통해, 제안 모델의 우수성을 입증하였다. 향후에는 환경 내 주요 장소뿐만 아니라 가구나 물체 정보도 활용할 수 있도록 현재의 모델을 확장해나갈 계획이다.

참고 문헌

[1] P. Anderson, Q. Wu, and D. Teney, et al, "Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments," *Proc. of CVPR-2018*, pp. 3674-3683, 2018.

[2] X. Wang, Q. Huang, and A. Celikyilmaz, et al, "Reinforced Cross-Modal Matching and Self-Supervised Imitation Learning for Vision-Language Navigation," *Proc. of CVPR-2019*, pp. 6629-6638, 2019.

[3] H. Tan, L. Yu, and M. Bansal, "Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout," *Proc. of NAACL-2019*, pp. 2610-2621, 2019.

[4] F. zhu, Y. zhu, and X. Chang, et al, "Vision-Language Navigation with Self-Supervised Auxiliary Reasoning Task," *Proc. of CVPR-2019*, 2019.

[5] D. Fried, V. Cirik, and A. Rohrbach, et al, "Speaker-Follower Models for Vision-and-Language Navigation," *Proc. of NeurIPS-2018*, pp. 3314-3325, 2018.

[6] A. Chang, A. Dai, and T. Funkhouser, et al, "Matterport3D: Learning from RGB-D Data in Indoor Environments," *Proc. of 3DV*, 2017.