

Tzu-Hsiang Lin  
Trung Bui  
Doo Soon Kim  
Jean Oh

Carnegie Mellon University,  
Adobe Research

{tzuhsial,hyaejino}@andrew.cmu.edu, {bui,dkim}@adobe.com

NIPS 2018 - 2nd Conversational AI Workshop

## Abstract

In this paper, we present a multimodal dialogue system for Conversational Image Editing. We formulated our multimodal dialogue system as a Partially Observed Markov Decision Process (POMDP) and trained it with Deep Q-Network (DQN) and a user simulator. Our evaluation shows that the DQN policy outperforms a rule-based baseline policy, achieving 90% success rate under high error rates. We also conducted a real user study and analyzed real user behavior.

본 논문에서는 대화식 이미지 편집을 위한 **멀티 모달 대화 시스템**을 제시한다. 우리는 **멀티 모달 대화 시스템을 POMDP (Partially Observed Markov Decision Process)로 공식화하고 이를 DQN (Deep Q-Network) 및 사용자 시뮬레이터로 교육했습니다.** 우리의 **평가에 따르면 DQN 정책은 규칙 기반 기준 정책보다 성능이 뛰어나 오류율이 높은 경우 90 %의 성공률을 달성합니다.** 또한 실제 사용자 연구를 수행하고 실제 사용자 행동을 분석했습니다.

## 1 Introduction

Image editing has been a challenging task due to its steep learning curve. Image editing software such as Adobe Photoshop typically have a complex interface in order to support a variety of operations including selection, crop, or slice. They also require users to learn jargons such as saturation, dodge and feather<sup>1</sup>. In order for a user to achieve a desired effect, some combinations of operations are generally needed. Moreover, image edits are usually localized in a particular region, for instance, users may want to add more color in the trees or remove their eye puffs. In the first case, users need to first select the trees and then adjust the saturation to a certain level. In the second case, users need to select the eye puffs, apply a reconstruction tool that fills the region with nearby pixels, then apply a patching tool to make the reconstruction more realistic. Such complexity makes the editing task challenging even for the experienced users.

학습 곡선이 가파르 기 때문에 이미지 편집은 어려운 작업이었습니다. Adobe Photoshop과 같은 이미지 편집 소프트웨어는 일반적으로 선택, 자르기 또는 슬라이스를 포함한 다양한 작업을 지원하기 위해 복잡한 인터페이스를 가지고 있습니다. 또한 사용자는 saturation, dodge 및 feather과 같은 전문 용어를 배워야합니다. 사용자가 원하는 효과를 달성하기 위해서는, 일반적으로 몇 가지 동작 조합이 필요하다. 또한 이미지 편집은 일반적으로 특정 region에 국한되어 있습니다. 예를 들어, 사용자는 나무에 더 많은 색상을 추가하거나 eye

puffs를 제거 할 수 있습니다. 첫 번째 경우 사용자는 먼저 나무를 선택한 다음 채도를 특정 수준으로 조정해야 합니다. 두 번째 경우, 사용자는 eye puffs를 선택하고 근처 픽셀로 영역을 채우는 재구성 도구를 적용한 다음 패치 도구를 적용하여 재구성을 보다 사실적으로 만들어야 합니다. 이러한 복잡성으로 인해 숙련 된 사용자에게도 편집 작업이 까다로워 집니다.

In this paper, we propose a conversational image editing system which allows users to specify the desired effects in a natural language and interactively accomplish the goal via multimodal dialogue. We formulate the multimodal dialogue system using the POMDP framework and train the dialog policy using Deep Q-Network (DQN)[16]. To train the model, we developed a user simulator which can interact with the system through a simulated multimodal dialogue. We evaluate our approach—i.e., DQN trained with the user simulator—by comparing it against a hand-crafted policy under different semantic error rates. The evaluation result shows that the policy learned through DQN and our user simulator significantly outperforms the hand-crafted policy especially under a high semantic error rate. We also conducted a user study to see how real users would interact with our system. The contributions of the paper are summarized as follows:

본 논문에서는 사용자가 원하는 효과를 자연 언어로 지정하고 대화식으로 멀티 모달 대화를 통해 목표(goal)를 달성 할 수 있는 대화식 이미지 편집 시스템을 제안한다. POMDP 프레임워크를 사용하여 멀티 모달 대화 시스템을 구성하고 DQN (Deep Q-Network)을 사용하여 대화 정책을 학습합니다 [16]. 모델을 훈련시키기 위해 시뮬레이션 된 멀티 모달 대화를 통해 시스템과 상호 작용할 수 있는 사용자 시뮬레이터를 개발했습니다. 우리는 접근 방식, 즉 사용자 시뮬레이터로 훈련 된 DQN을 다른 시맨틱 오류율로 수작업 정책과 비교하여 평가합니다. 평가 결과에 따르면 DQN과 사용자 시뮬레이터를 통해 학습 한 정책은 특히 높은 semantic 오류율에서 수작업 정책보다 훨씬 뛰어납니다. 또한 실제 사용자가 시스템과 어떻게 상호 작용하는지 확인하기 위해 사용자 연구를 수행했습니다. 이 논문의 기여는 다음과 같이 요약됩니다.

- We present a POMDP formulated multimodal dialogue system.
- We developed a multimodal multi-goal user simulator for our dialogue system.
- We present an architecture for Conversational Image Editing, a real-life application of the proposed framework in the domain of image editing
- We present the experiment results of comparing the proposed model against a rule-based baseline.

- POMDP 공식화된 멀티 모달 대화 시스템을 소개합니다.
- 대화 시스템을 위한 멀티 모달 multi-goal 사용자 시뮬레이터를 개발했습니다.
- 이미지 편집 영역에서 제안된 프레임 워크의 실제 응용 프로그램 인 Conversational Image Editing을 위한 아키텍처를 제시합니다.
- 제안 된 모델과 규칙 기반 베이스라인을 비교 한 실험 결과를 제시합니다.

## 2 Related Work

The multimodal system PixelTone [11] shows that an interface combining speech and gestures can help users increase more image operations. Building on its success, we

propose to build a multimodal image editing dialogue system. Previous research on multimodal dialogue systems mostly focus on the architectures [4] for multimodal fusion and did not adopt the POMDP framework [22]. Since the dialogue managers of these systems are based on handcrafted rules, it cannot be directly optimized. Also, real users are essential in evaluating these systems [21], which can be costly and time inefficient.

**멀티 모달 시스템 PixelTone [11]은 음성과 제스처를 결합한 인터페이스가 사용자의 더 많은 이미지 작업을 증가시키는 데 도움이 될 수 있음을 보여줍니다. 우리는 성공을 바탕으로 멀티 모달 이미지 편집 대화 시스템을 구축 할 것을 제안합니다.**

**멀티 모달 대화 시스템에 대한 이전의 연구는 대부분 멀티 모달 융합을 위한 아키텍처에 중점을두고 있으며 [4] POMDP 프레임 워크를 채택하지 않았다 [22]. 이러한 시스템의 대화 관리자는 수작업(hand-crafted) 규칙을 기반으로 하기 때문에 직접 최적화 할 수 없습니다. 또한 실제 사용자는 이러한 시스템을 평가하는 데 필수적입니다 [21]. 비용이 많이 들고 시간이 비효율적 일 수 있습니다.**

Information-seeking dialogue systems such as ticket-booking [7] or restaurant-booking [6, 20] typically focus on achieving one user goal throughout an entire dialogue. Trip-booking [7] is a more complex domain where memory is needed to compare trips and explore different options. The most similar domain is conversational search and browse [9, 8], where the system can utilize gestures and even gazes to help users to locate the desired objects. A recently collected corpus [13] shows that Conversational Image Editing is more challenging, requiring to address not only these aspects but also the composite-task setting [17, 3] where the user may have multiple goals to achieve in a sequential order.

티켓 예약 [7] 또는 식당 예약 [6, 20]과 같은 정보를 구하는 대화 시스템은 일반적으로 전체 대화에서 하나의 사용자 목표(goal)를 달성하는 데 중점을 둡니다. 여행 예약 [7]은 여행을 비교하고 다양한 옵션을 탐색하기 위해 메모리가 필요한 더 복잡한 도메인입니다. 가장 유사한 도메인은 대화식 검색 및 탐색입니다 [9, 8]. 여기서 시스템은 사용자가 원하는 객체를 찾을 수 있도록 제스처와 심지어 시선을 활용할 수 있습니다.

**최근에 수집된 코퍼스[13]는 대화 이미지 편집이 더 어렵다는 것을 보여준다. 이러한 측면뿐만 아니라 사용자가 순차적으로 달성해야 할 여러 목표(goal)를 가질 수 있는 복합 작업 설정[17, 3]을 다루어야 한다.**

Our task is also related to Visual Dialogue [5] which focuses on the vision and the dialogue jointly. The agent in Visual Dialogue needs to recognize objects, infer relationships, understand the scene and dialogue context to answer questions about the image. Our agent also requires a similar level of understanding, but the focus on vision is more of recognizing a localized region in an image. Another closely related area is vision-and-language navigation [1], since both the navigation instructions (e.g., go upstairs and turn right) and image edit requests [15] (e.g., remove the elephant not looking forward) are mostly in imperative form and relates to what the agent sees.

우리의 task는 비전과 대화에 공동으로 초점을 둔 시각적 대화 [5] 와도 관련이 있습니다. 비주얼 대화의 에이전트는 이미지에 대한 질문에 대답하기 위해 개체를 인식하고, 관계를 추측하고, 장면과 대화 컨텍스트를 이해해야 합니다. Google 상담원도 비슷한 수준의 이해가 필요하지만 비전에 중점을 두는 것은 이미지에서 localized 영역을 인식하는 것입니다. 또

다른 밀접하게 관련된 영역은 시각 및 언어 내비게이션 [1]인데, 이는 내비게이션 안내 (예 : 위층으로 이동하여 우회전)와 이미지 편집 요청 (15) (예 : 기대하지 않는 코끼리 제거)이 대부분 명령형(imperative)이기 때문에 에이전트가 보는 것과 관련이 있습니다.

### Markov Property -

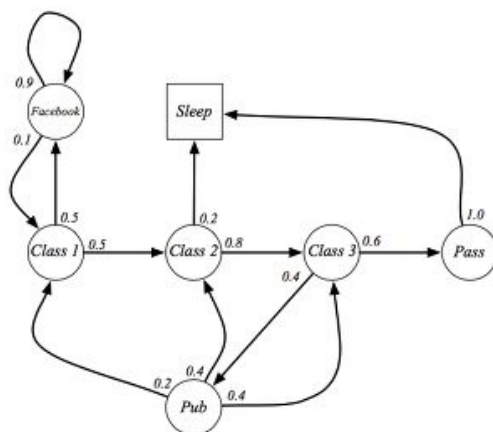
agent가 environment에서 어떠한 action을 하기 위해서는 의사결정이 필요합니다. 그리고 의사결정을 하기 위해서는 environment로 부터 정보들을 받게 되는데 이러한 정보들의 특성을 Markov property라고 합니다.

현재의 상태는 과거의 어떠한 과정들을 거쳐서 발생이 된 것 임으로, 현재 상태에서의 정보들은 과거의 중요한 정보들을 포함하고 있다고 볼 수 있습니다. 모든 과거의 정보들을 포함하고 있지는 않지만 앞으로 다가올 미래를 예측하는데 필요한 정보는 충분히 포함하고 있다는 것입니다.

### Markov Process(Markov Chain) - 마르코프 프로퍼티를 가지는 랜덤 상태의 순서이다.

State 집합과 State Transition probabilities (각 상태로 전이할 확률) Matrix의 튜플로 표현할수 있다.

모든 상태들을 고려하여 앞으로 변경될 확률까지를 고려한 것이 Markov process 혹은 Markov chain 이라고 합니다. 그리고 S와 P matrix로 이를 표현을 할 수 있게 되었습니다.



Sample episodes for Student Markov Chain starting from  $S_1 = C1$

$S_1, S_2, \dots, S_T$

- C1 C2 C3 Pass Sleep
- C1 FB FB C1 C2 Sleep
- C1 C2 C3 Pub C2 C3 Pass Sleep
- C1 FB FB C1 C2 C3 Pub C1 FB FB  
FB C1 C2 C3 Pub C2 Sleep

대학생들이 학교에서 수업에 참여하는 혹은 딴짓을 하는 Markov process를 예제로 설명하고 있습니다.

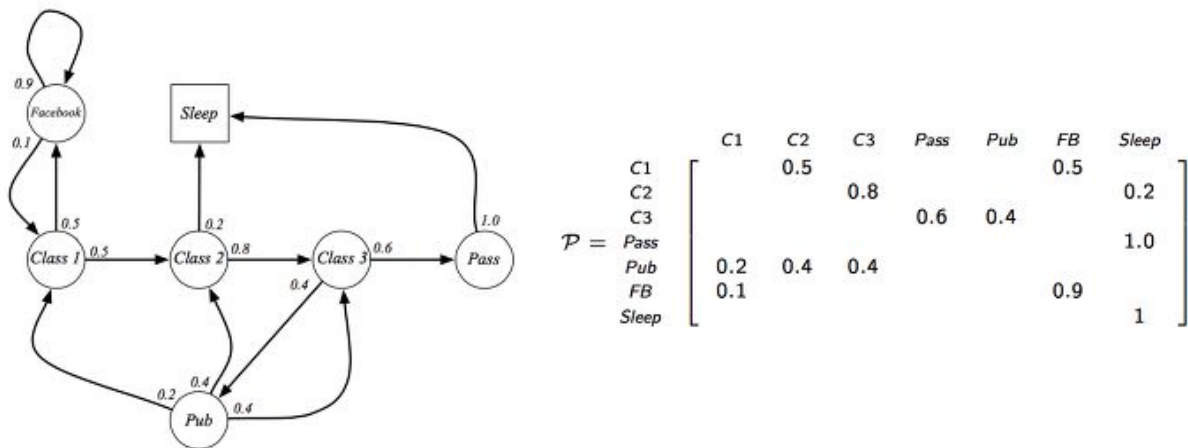
심플하게 3개의 class를 모두 참여하면 해당 과정을 pass 할 수 있는 프로세스를 표현하고 있습니다. 하지만 학생들이 모두 열심히 공부를 하지는 않죠. 그래서 가끔 페이스북을 하거나 수업을 땡땡이 하고 술을 먹으로 가거나 잘 수도 있는 state들도 표현이 되어 있습니다.

class 1에서 시작을 해보죠. class 1을 마치고 학생들이 class 2로 갈 확률이 50%이고 class 2로 가지 않고 facebook으로 갈 확률이 50%입니다. 한번 facebook으로 간

학생들을 90%의 확률로 계속 facebook에 빠져 있게 되고 10%만 다시 수업으로 복귀하게 됩니다.

힘들게 class 2로 갔지만 갑자기 졸음이 몰려와서 자는 경우도 있네요. 열심히 하는 학생들은 class 3까지 완료하고 Pass 를 하게 되는데 모든 학생들이 자러 갑니다. 이때 sleep state를 terminal state라고 합니다. 모든 프로세스가 종료되는 시점을 말합니다.

위의 예제 상황에서 몇가지 에피소드를 만들어보았습니다. 순서대로 상태가 변화하는 것을 볼 수 있습니다. 가장 첫번째의 에피소드는 상당한 모범생 스타일이라는 것을 추론할 수 있습니다.



위 예제에서 이동한 가능한 상태들에 대해서 P matrix에 테이블로서 정리하여 표현을 하면 발생한 케이스와 확률들을 모두 표현할 수 있습니다.

### Markov Reward Process (Markov Process + reward)

A Markov reward process is a Markov chain with values.

#### Definition

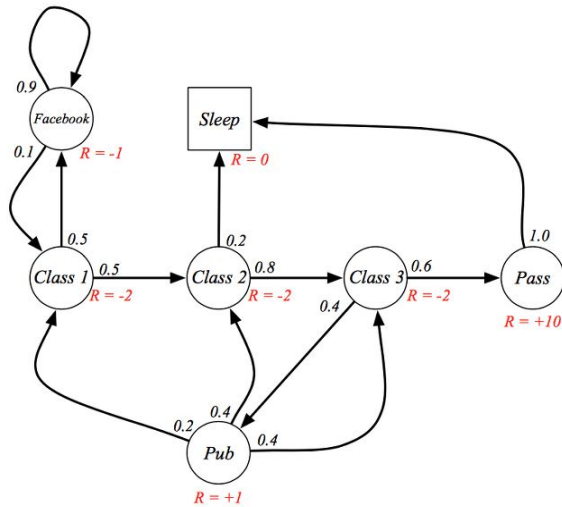
A Markov Reward Process is a tuple  $\langle S, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $S$  is a finite set of states
- $\mathcal{P}$  is a state transition probability matrix,  
 $\mathcal{P}_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s]$
- $\mathcal{R}$  is a reward function,  $\mathcal{R}_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- $\gamma$  is a discount factor,  $\gamma \in [0, 1]$

앞에서 알아본 Markov chain에다가 values (가치)라는 개념을 추가하여 생각해 볼 수 있습니다. 이를 Markov Reward Process 라고 합니다. 이 가치를 판단하기 위해서는 두가지 factor가 추가가 되는데 하나가 reward이고 다른 하나는 discount factor입니다.

reward는 현재 상태를 기준으로 하여 다음 상태에 받게 될 기대되는 Reward를 표현합니다. 그리고 discount factor는 0에서 1 사이의 값으로 할인율 입니다.

$S_t$ 가  $s$ 인 현재 상태일때를 조건으로 하는  $S_{t+1}$ 이  $s'$ 로 다음의 상태가 될 확률을  $P_{ss'}$ 로 표현하기로 합니다. 이를 상태가 발생하는 경우에 수만큼 나열하여 매트릭스의 형태로 표현을 하면 테이블 형태의  $P$ 로도 표현이 됩니다.



학생 예제로 돌아가서 class 1에서 시작하여 학생이 50%의 확률로 class 2를 선택하고 이동하게 되었습니다. 이 때 빨간색의 reward가 추가가 된 것을 볼 수 있습니다. class 2로 이동한 학생들이 받을 Reward이 -2 값이 됩니다. 그리하여 class 3까지 모두 완료한 학생들은 pass가 되면서 +10 이라는 큰 Reward을 받게 되고 모두 자러 갑니다.

모두가 자러가서 이 프로세스가 종료가 되면 reward는 0을 받게 됩니다.

#### Definition

The return  $G_t$  is the total discounted reward from time-step  $t$ .

$$G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

- The discount  $\gamma \in [0, 1]$  is the present value of future rewards
- The value of receiving reward  $R$  after  $k + 1$  time-steps is  $\gamma^k R$ .
- This values immediate reward above delayed reward.
  - $\gamma$  close to 0 leads to "myopic" evaluation
  - $\gamma$  close to 1 leads to "far-sighted" evaluation

우리가 원하는 목표로 하는 것은 이러한 reward에 총합을 구하는 것입니다. 현재 시점에서 앞으로 내가 Reward을 받게 될 모든 것들을 계산해 보는 것이지요.

하지만, 이때 현재 즉시 받는 Reward과 미래에 받게 되는 Reward과는 다른 가치를 가지고 있으므로 할인율을 적용해서 discount를 해줍니다. 이것은 미래 가치를 현재 가치로 환산하는 것이라고 생각하면 될 것 같습니다.



할인율이 0에서 1사이의 실수 값이기 때문에 이를 제공하게 되면 특정한 값으로 수렴하게 됩니다. time step  $k$ 에서 수렴을 하게 될 경우를 보면 바로 다음의 time step  $k+1$ 에서의 Reward이 값이 있더라도 너무 먼 미래의 Reward임으로 고려하지 않게 되는 것과 동일합니다. 이러한 특성이 있기에 무한대로 time step이 진행이 된다고 해도 현재 시점에서 우리가 의사결정시에 고려할 미래 가치는 유한하게 고려되어 집니다.

그리고, 이 할인율이 0이 되면 미래에 받게 될 Reward들이 모두 0이 되어 고려하지 않게 됩니다. 그렇기에 바로 다음의 Reward만을 추구하는 근시안적인 행동을 하게 될 것입니다.

반대로 할인율이 1에 근처에 가까워 지면 질수록 할인율 작아지기 때문에 미래 Reward들을 더 많이 고려하게 되는 원시안적인 행동을 하게 될 것입니다.

The value function  $v(s)$  gives the long-term value of state  $s$

#### Definition

The *state value function*  $v(s)$  of an MRP is the expected return starting from state  $s$

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

현재 상태가  $s$  라는 state일때 앞으로 발생할 것으로 기대되는(E) 모든 rewards의 합을 value 라고 합니다. 이것을 조건부 확률로 수학적으로 표현을 하면 위와 같이 됩니다.

이 value function는 현재 시점에서 미래의 모든 기대하는 Reward들을 표현하고 이를 미래 가치라고 할 수 있습니다.

강화학습에서 핵심적으로 학습을 통해서 찾고자 하는 것이 바로 이 value function을 최대한 정확하게 찾는 것이라고 할 수도 있습니다. 다시 말해서 미래 가치가 가장 큰 의사결정을 하고 행동하는 것이 우리의 최종 목표가 되며 이는 매우 중요한 내용입니다.

Sample **returns** for Student MRP:

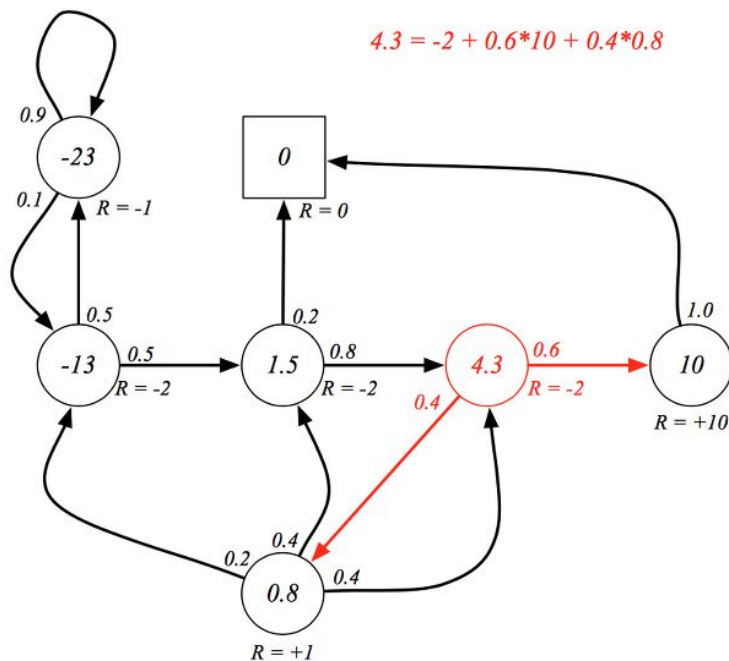
Starting from  $S_1 = C1$  with  $\gamma = \frac{1}{2}$

$$G_1 = R_2 + \gamma R_3 + \dots + \gamma^{T-2} R_T$$

C1 C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 10 * \frac{1}{8}$	=	-2.25
C1 FB FB C1 C2 Sleep	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16}$	=	-3.125
C1 C2 C3 Pub C2 C3 Pass Sleep	$v_1 = -2 - 2 * \frac{1}{2} - 2 * \frac{1}{4} + 1 * \frac{1}{8} - 2 * \frac{1}{16} \dots$	=	-3.41
C1 FB FB C1 C2 C3 Pub C1 ...	$v_1 = -2 - 1 * \frac{1}{2} - 1 * \frac{1}{4} - 2 * \frac{1}{8} - 2 * \frac{1}{16} \dots$	=	-3.20
FB FB FB C1 C2 C3 Pub C2 Sleep			

학생 예제로 다시 돌아서 각각의 에피소드들에 대한 value 를 구해보면 위와 같이 됩니다.

가장 모범생의 행동 스타일이 가장 좋은 value 값을 가지고 있는 걸 보니 가장 잘 한 학생인거 같아 보입니다.^^



다시 학생 예제로 돌아가서 class 3를 현재 상태라고 하고 value function을 계산하면 위와 같이 됩니다. 현재 시점에서의 Reward -2를 더하고 확률적으로 선택될 다음 상태들의 value를 계산하면 4.3의 값이 나오게 됩니다. 이 값이 class 3에서의 value 입니다.

### Markov Decision Process (Markov Process + reward + action)

MRP에 의사결정에 대한 개념을 더 추가하면 MDP됩니다.(역시 모든 state가 Markov 인 환경에서 이루어집니다.)



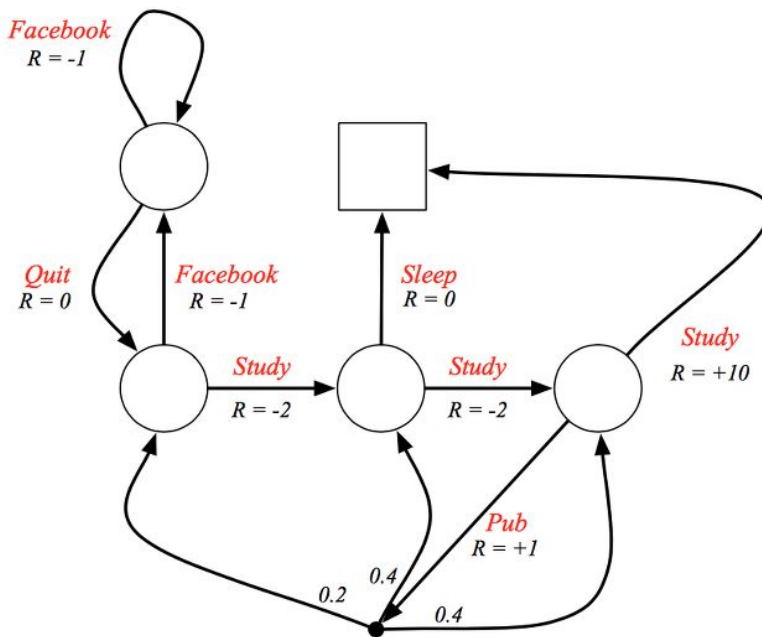
A 라고 하는 action 이 가능한 집합을 표현하는 notation 이 하나가 더 추가가 되었습니다. 이를 통해서 현재 상태  $s$  에서  $a$  라고 하는 action 을 할 때 다음 상태  $s'$  로 가게 될 확률을  $P$  에 대한 내용으로 표현을 하게 됩니다.  
 $R$  도 reward 에 대한 함수인데 마찬가지로 현재 상태  $s$  에서  $a$  라는 action 을 할 때 기대되어지는 Reward 을 표현하게 됩니다.

A Markov decision process (MDP) is a Markov reward process with decisions. It is an *environment* in which all states are Markov.

#### Definition

A Markov Decision Process is a tuple  $\langle S, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $S$  is a finite set of states
- $\mathcal{A}$  is a finite set of actions
- $\mathcal{P}$  is a state transition probability matrix,  
 $\mathcal{P}_{ss'}^a = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$
- $\mathcal{R}$  is a reward function,  $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
- $\gamma$  is a discount factor  $\gamma \in [0, 1]$ .



다시 이전에 사용했던 학생 예제를 살펴보면, class 1 의 상태에서 할 수 있는 action 은 study, facebook 이 될 겁니다. 만약 study 라는 action 을 취한다면 reward 를 -2 얻고 다음 state 인 class 2로 이동하게 되겠습니다.  
facebook이라는 행동을 취했다면 reward를 -1 얻고 다음 상태인 facebook으로 이동하게 되겠지요. 여기서는 다시 facebook 과 quit 이라는 action 을 취하여 state 를 transition 하게 될 것입니다.

## Definition

A policy  $\pi$  is a distribution over actions given states,

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

- A policy fully defines the behaviour of an agent
- MDP policies depend on the current state (not the history)
- i.e. Policies are *stationary* (time-independent),  
 $A_t \sim \pi(\cdot|S_t), \forall t > 0$

이번에는 policy 에 대해서 살펴보겠습니다. policy는 agent가 행동을 하는데 중요한 역할을 하게 됩니다. **현재 state에 대해서 어떤 action을 취할 확률**을 나타냅니다. 또는 **현재 state와 action을 mapping 하는 것이라고도 표현**합니다. 그리하여 policy, 정책은 파이에 대한 함수로 표현이 됩니다.

MDP policy에서는 현재 state만 고려하고 과거의 것들을 고려하지 않고 action을 하는데, 이때에 stochastic 하게, 확률적으로 action을 결정하게 됩니다.

그리고 policy는 time step의 변화에 무관하게 독립적으로 진행이 되므로 stationary 하다고 할 수 있습니다.

- Given an MDP  $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$  and a policy  $\pi$
- The state sequence  $S_1, S_2, \dots$  is a Markov process  $\langle \mathcal{S}, \mathcal{P}^\pi \rangle$
- The state and reward sequence  $S_1, R_1, S_2, \dots$  is a Markov reward process  $\langle \mathcal{S}, \mathcal{P}^\pi, \mathcal{R}^\pi, \gamma \rangle$
- where

$$\mathcal{P}_{s,s'}^\pi = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{P}_{ss'}^a$$
$$\mathcal{R}_s^\pi = \sum_{a \in \mathcal{A}} \pi(a|s) \mathcal{R}_s^a$$

하나의 MDP가 주어졌을 때 하나의 policy가 존재합니다. 이 policy 개념을 Markov process에 적용을 하여 표현을 하면 **P** **윗첨자에 파이가 표시**가 되고, 이를 **policy 파이를 따르는 P**라고 읽습니다. 왜냐하면 policy 를 벗어난 state transition은 발생할 수 없기 때문입니다.

마찬가지로 Markov reward process 에서의 주요 항목들도 policy 하에서의 P 와 R로 표현이 추가 됩니다.

예를들어서, 우리가 공원에 산책을 나갔습니다. 현재 공원 벤치에 있다고 생각해보겠습니다. 벤치에서 할 수 있는 action은 '누워서 잔다' 혹은 '좀더 산책을 한다'를 선택할 수 있을 것입니다. 하지만 이 공원에 정책상 벤치에서 누워서 자는 것은 허락되지 않습니다. 그러면 우리는 산책을 하는 행동을 취하게 되는 것과 비슷합니다. 물론 또 다른 선택지가 있다면 그에 대한 선택할 확률이 적용될 것입니다.

#### Definition

The *state-value function*  $v_{\pi}(s)$  of an MDP is the expected return starting from state  $s$ , and then following policy  $\pi$

$$v_{\pi}(s) = \mathbb{E}_{\pi} [G_t \mid S_t = s]$$

#### Definition

The *action-value function*  $q_{\pi}(s, a)$  is the expected return starting from state  $s$ , taking action  $a$ , and then following policy  $\pi$

$$q_{\pi}(s, a) = \mathbb{E}_{\pi} [G_t \mid S_t = s, A_t = a]$$

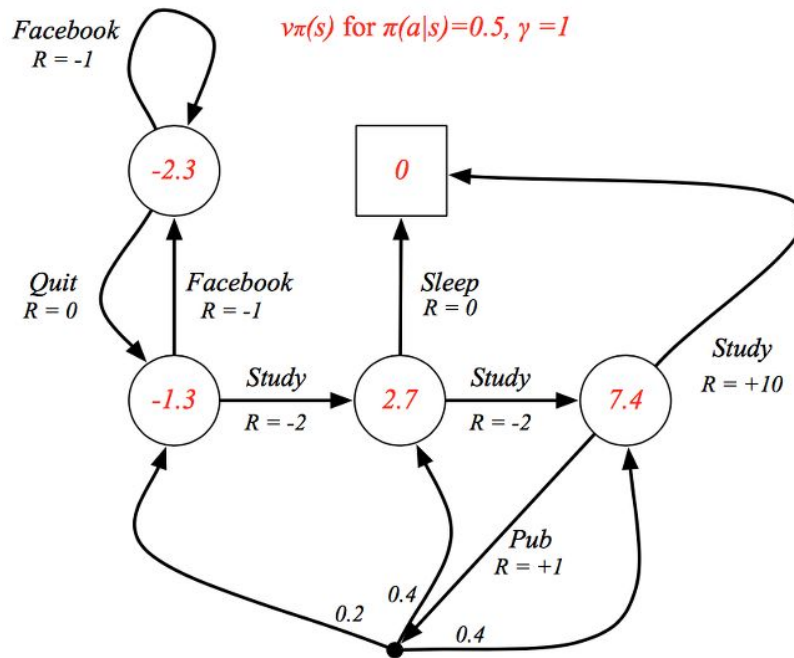
이번에는 value function에 policy를 적용해보겠습니다.

위의 공식을 말로 풀어보면, **state  $s$  에서 policy를 따르는  $v$  가 되며 이것은 동일하게 state  $s$  인 조건에서 policy를 따르는 기대되는 미래 Reward들을 모든 합의 값이 됩니다.** 이를 state-value function,  $v$  이라고 합니다.

이를 시맨틱적으로 다시 말하면, 현재 상태에서 이 policy를 따랐을 때 얼마나 좋은지, 얼마나 가치있는지를 나타내는 값이 되겠습니다.

여기다가 **action  $a$  개념을 추가하면 action-value function,  $q$  가 됩니다. 이것은 state  $s$  이면서 action  $a$  인 조건에서 policy를 따르는 기대되는 미래 Reward들을 모두 합의 값이 됩니다.**

이를 시맨틱적으로 다시 말하면, policy를 따라서 이 action을 행동했을때 얼마나 좋은지, 가치를 나타내는 값이 됩니다.



학생 예제를 통해 표현을 해보면, policy 의 값이 0.5 인 상황이므로 현재 state에서 두가지 action을 취하게 될 확률이 반반이 될 겁니다. class 1의 위치에서 본다면 study를 할 확률이 반이고 facebook을 할 확률이 반이 되는 것입니다. 이러한 policy 를 따르는 value 값을 표현하면 빨간색과 같이 될 수 있습니다.

The state-value function can again be decomposed into immediate reward plus discounted value of successor state,

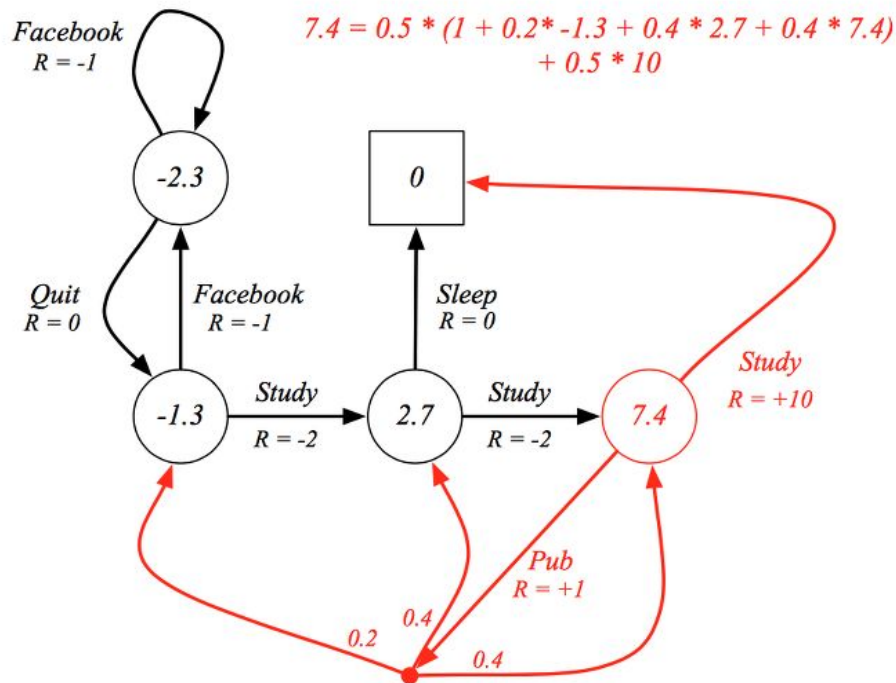
$$v_{\pi}(s) = \mathbb{E}_{\pi} [R_{t+1} + \gamma v_{\pi}(S_{t+1}) \mid S_t = s]$$

The action-value function can similarly be decomposed,

$$q_{\pi}(s, a) = \mathbb{E}_{\pi} [R_{t+1} + \gamma q_{\pi}(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

지금까지 살펴본 v function과 q function을 Bellman equation을 사용해서 분리 하면 위와 같이 됩니다. 현재 상태에서 policy를 따르는 value는 즉시 받게 되는 reward, R 과 할인율을 적용한 다음 상태에서의 value 값에 합으로 나타낼 수 있게 되었습니다.

q도 동일하게 Bellman equation을 사용해서 표현하면 위와 같이 되는 것을 알 수 있으실 겁니다. 현재 상태가 s 일때 취할 수 있는 액션들중에서 a 라는 액션을 했을때의 가치를 나타냅니다.



다시 학생 예제를 꺼내서 한번 살펴보겠습니다. 빨간색 원으로 표시된 class 3의 state가 현재 상태라고 할때,  $v$  값을 구하면 위와 같이 됩니다. 이 예제에서의 policy는 0.5 임으로 반반의 확률로 study를 선택하거나 pub을 선택하게 될 겁니다. study를 action으로 취한 경우에는  $0.5 * 10$  이 되어 간단하게 계산이 됩니다.

반대로 50%의 확률로 선택이 될 pub을 action으로 취한 경우에는 Reward +1을 받고 20%의 확률로 class 1로 가게 되어 value 가 -1.3 이되는 것이 하나입니다. ( $0.2 * -1.3$ ) 그런데 여기서 하나가 아니죠. 40% 확률로 class 2로 가고 ( $0.4 * 2.7$ ) 또 40% 확률로 class 3으로 다시 돌아오는 것 ( $0.4 * 7.4$ ) 모두를 고려해서 계산을 하면 되겠습니다.

#### Definition

The *optimal state-value function*  $v_*(s)$  is the maximum value function over all policies

$$v_*(s) = \max_{\pi} v_{\pi}(s)$$

The *optimal action-value function*  $q_*(s, a)$  is the maximum action-value function over all policies

$$q_*(s, a) = \max_{\pi} q_{\pi}(s, a)$$

- The optimal value function specifies the best possible performance in the MDP.
- An MDP is "solved" when we know the optimal value fn.

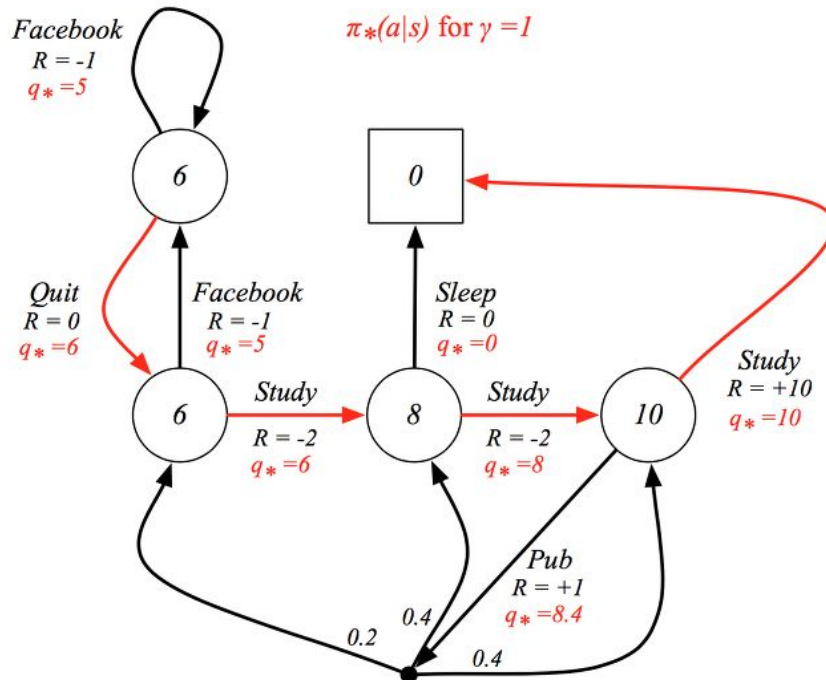
지금까지 배운 내용을 토대로 하여 최적화하는 방법에 대하여 알아보도록 하겠습니다. optimal value function에 대하여 생각해 봅시다.



state-value function이 갖는 값이 최대값이 되도록 하는 max 를 구한다면 이것이 가장 최적의  $v$  가 될 것입니다. 이때의  $v$  를  $v^*$ 로 표현을 하며 이것이 곧 optimal state-value function이 됩니다.

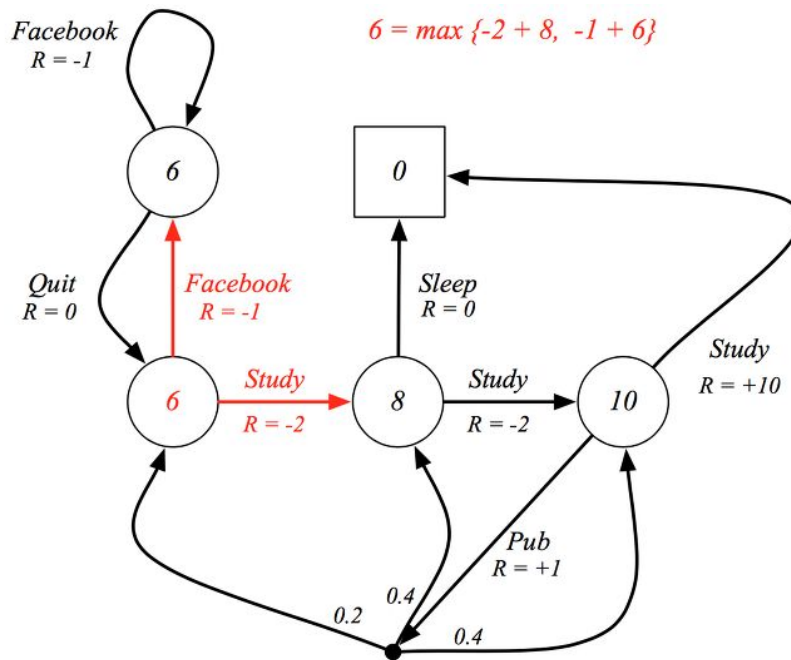
마찬가지로 action-value function이 갖는 값이 최대값이 되도록 하는 max를 구한다면 이때의  $q$  를  $q^*$ 로 표현하고 이는 곧 optimal action-value function이 됩니다.

이렇게 최적화된 value function을 찾는 것이 우리의 목표가 되며 MDP를 푼다고 표현을 하기도 합니다.



$q^*$ 를 활용해서 optimal policy를 예제에 표현을 해보면 각 state에서 최적의 행동을 취하도록 할 것입니다. 그리하여 그 행동을 따르게 되면 빨간색 화살표와 같이 움직이게 되겠습니다. 공부를 열심히 하도록 만들어져 있군요~





학생의 예제에서 첫번째 state (빨간색)에 있다고 할 때 최적이 되는 value 값은 6이 됩니다. 두가지의 action 이 있고 이에 대한 Reward가 미래가치를 구해서 max가 되는 것을 찾음...

- Infinite and continuous MDPs
- Partially observable MDPs
- Undiscounted, average reward MDPs

MDPs 환경들도 다양한 환경이 있습니다. 무한하고 지속되는 환경도 있고 부분적으로만 볼 수 있고 전체적인 것을 알 수 없는 환경도 있습니다. 또 미래 Reward를 할인하지 않거나 다르게 처리해야 하는 환경도 있을 것입니다.

출처: <https://daeson.tistory.com/318?category=710652>

## POMDP

MDP에 기반한 대화 관리 시스템에서는, 모든 환경이 MDP를 만족하는 상태로 표현하기 위해 매우 많은 상태들이 요구된다. 즉, MDP에 기반한 대화 관리 시스템은, 상기와 같이 많은 상태가 요구됨에 따라, 강화 학습이 어려울 뿐만 아니라 훈련 데이터베이스의 수집이 대규모로 이뤄져야 하는 단점이 있다.

MDP에 기반한 대화 관리 시스템은, 실제로 텍스트 기반의 자연어를 이해하거나 음성을 인식하는 경우에 오류가 발생하는 경우(즉, 모든 환경 상태를 완전히 믿을 수 없는 상태)에 대화 관리 성능이 크게 떨어질 수 있다.

즉, 실제 세계에서 환경 상태에 대한 정확하고 완전한 정보를 가진다는 것은 거의 불가능하기 때문에, MDP에 기반한 대화 관리 시스템은 실제 세계를 고려하여 부분적인 불확실성이 포함된

상태에서 행동을 선택하고, 종종 환경 상태에 대한 정보를 늘려서 좀더 효과적으로 행동을 선택할 필요성도 있다.

이에, MDP의 관측에서 부분적으로 믿을 수 없는 문제를 해결하기 위해, 부분 관측 마르코프 의사 결정 과정(Partially Observable Markov Decision Process, 이하 "POMDP"라 함)에 기반한 대화 관리 시스템이 제안되었다. 이때, POMDP에 기반한 대화 관리 시스템은 MDP에 기반한 대화 관리 시스템과 마찬가지로, 사용자의 발화 및 그에 대응되는 시스템 동작으로 구성된 대화 코퍼스로부터 정책을 훈련하여 대화 관리 시스템의 동작과 응답을 결정하는 방식을 이용한다.

POMDP에서는 MDP에서 환경 상태를 정확히 관찰하지 못하는 경우를 모델링하여 최적 행위를 계산할 수 있도록 한다.

즉, POMDP에서는 에이전트가 부분적인 불확실성이 있는 상황에서도 의사결정을 할 수 있도록 MDP를 일반화한다. 이를 위해, POMDP에서는 MDP와 수학적인 정의가 흡사하나, 관찰값의 오류를 확률적으로 모델링하는 부분이 추가된다.

POMDP는 하기 6개의 파라미터에 의해 수학적으로 정의된다.

- (1) 에이전트가 행동을 수행하는 환경상태(world state)를 정의하는 상태 집합:  $S$
- (2) 에이전트가 수행할 수 있는 행동(action)을 정의하는 행동 집합:  $A$
- (3) 에이전트가 환경상태  $s$ 에서 행동  $a$ 를 수행했을 때, 받는 보상(reward)을 산출하는 Reward 함수:  $R(s, a)$
- (4) 에이전트가 환경상태  $s$ 에서 행동  $a$ 를 수행했을 때, 다음 환경상태가  $s'$ 이 될 상태 천이 확률:  $T(s, a, s')$
- (5) 에이전트가 관찰할 수 있는 값들을 정의하는 관찰 집합:  $Z$
- (6) 환경 상태가  $s$ 일 때, 개체의 관찰값이  $z$ 일 관찰 확률:  $O(s, z)$

Partially Observed Markov Decision Process  
is a tuple  $\langle S, A, P, R, \Omega, \mathcal{O} \rangle$

- ①  $S, A, P, R$  are the same as in MDP
- ②  $\Omega$  – finite set of observations
- ③  $\mathcal{O} : S \times A \mapsto \Delta(\Omega)$  – observation function, which gives  $\forall (s, a) \in S, A$ , a probability distribution over  $\Omega$ , i.e.  
 $p(o | s_{t+1}, a_t) \quad \forall o \in \Omega$

POMDP에서는 에이전트가 환경상태를 직접 볼 수 없다고 가정하기 때문에, 에이전트는 관찰값의 시퀀스로부터 실제 환경상태를 유추(infer)해야만 한다. 따라서 POMDP에서의 정책은 관찰값의 시퀀스로부터 행동으로 매핑하는  $\pi: Z^* \rightarrow A$ 로 정의된다. 이때, POMDP에서는 관찰값의 시퀀스가 상태추정 확률분포

$$b = [p(s_1), \dots, p(s_n)]$$

로 요약된다고 알려져 있다.

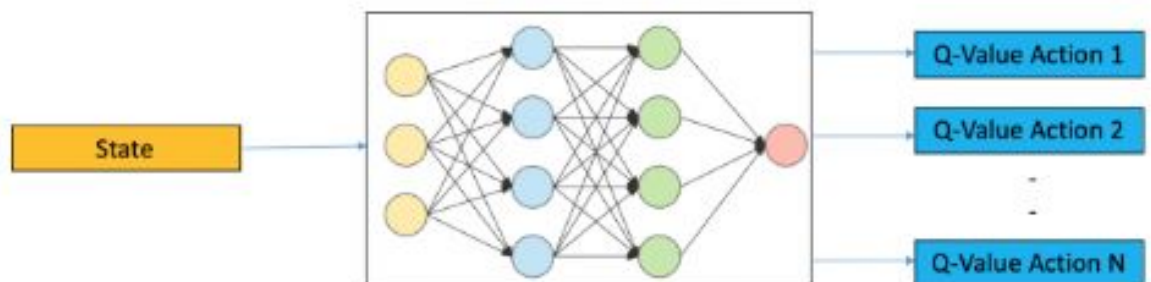
결론적으로, POMDP에서 정책이라 함은 '가능한 모든 상태추정 확률분포'에 대해 어떤 행동을 수행할지를 결정하는 것을 시맨틱한다. 즉, '가능한 모든 상태추정 확률분포의 집합을 'B'라 하면, POMDP에서 정책은 ' $\pi: B \rightarrow A$ '로 정의된다.

## Q-Learning

$$Q(s, a) = r(s, a) + \gamma \max_a Q(s', a)$$

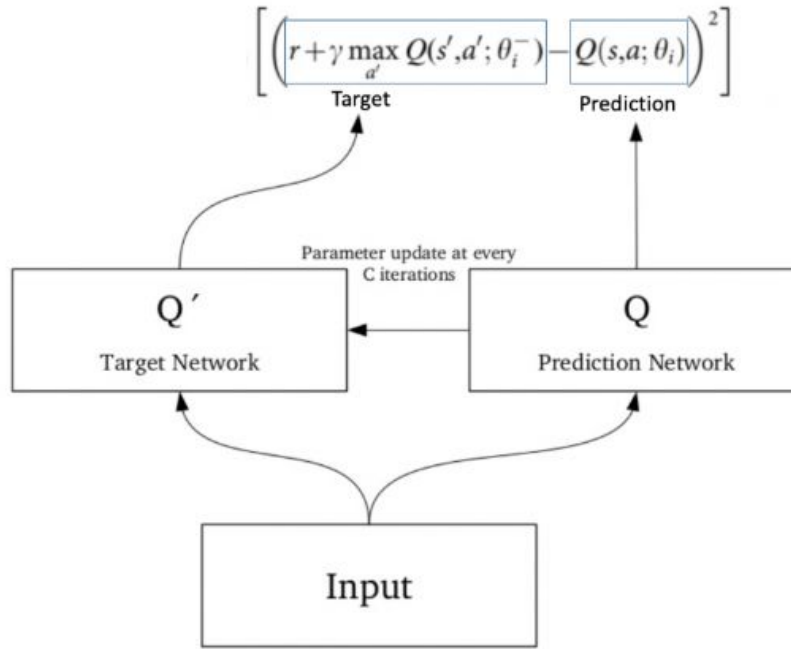


Q Learning



Deep Q Learning

$$Loss = (r + \gamma \max_{a'} Q(s', a'; \theta') - Q(s, a; \theta))^2$$



### 3 Partially Observed Markov Decision Process Formulation

In this section, we formulate our image editing dialogue system as a Partially Observable Markov Decision Process (POMDP) [22]. POMDP dialogue framework combines Belief State Tracking and Reinforcement Learning. Belief State Tracking represents uncertainty in dialogue that may come from speech recognition errors and possible dialogue paths, whereas Reinforcement Learning helps the dialogue manager discover an optimal policy.

이 섹션에서는 이미지 편집 대화 시스템을 Partially Observable Markov Decision Process (POMDP) [22]로 공식화합니다. **POMDP 대화 프레임 워크는 Belief State Tracking 및 Reinforcement Learning을 결합합니다.** **Belief State Tracking**은 음성 인식 오류 및 가능한 대화 경로에서 비롯될 수 있는 대화의 불확실성을 나타내며, **강화 학습**은 대화 관리자가 최적의 정책을 찾으도록 도와줍니다.

**POMDP 대화 프레임 워크 = Belief State Tracking + Reinforcement Learning**

POMDP is composed of belief states  $B$ , actions  $A$ , rewards  $R$ , and transitions  $T$ . The goal is to optimize a parametrised policy  $\pi : B \rightarrow A$  to maximize the expected sum of rewards .

$$\pi : B \rightarrow A$$

$$R = \sum_t \gamma^t r_t.$$

POMDP는 belief 상태  $B$ , action  $A$ , rewards  $R$  및 transitions  $T$ 로 구성됩니다.

목표(goal)는 예상되는 rewards 합계  $R$  를 최대화하기 위해 매개 변수화 된 정책  $\pi : B \rightarrow A$ 를 최적화하는 것입니다.

## State

Our state space  $B = B_u \oplus B_e$  includes the user state  $B_u$  and the image edit engine state  $B_e$ .  $B_u$ , the estimation of the user goal at every step of the dialogue, is modeled as the probability distribution over possible slot values. For gesture related slots (e.g., gesture\_click, object\_mask\_str), we assume these values hold 100% confidence and assigns probability score 1 if a value is present and 0 otherwise.  $B_e$  is the information provided by the engine that is related to the task at hand.

상태 공간  $B = B_u \oplus B_e$ 에는 사용자 상태  $B_u$ 와 이미지 편집 엔진 상태  $B_e$ 가 포함됩니다. 대화의 모든 단계에서 사용자 목표(goal)를 추정하는  $B_u$ 는 가능한 슬롯 값에 대한 확률 분포로 모델링됩니다.

제스처 관련 슬롯 (예 : gesture\_click, object\_mask\_str)의 경우 이 값이 100 % 신뢰를 유지하고 값이 있으면 확률 점수 1을 지정하고 그렇지 않으면 0을 할당한다고 가정합니다.  $B_e$ 는 현재 작업과 관련된 엔진에서 제공하는 정보입니다.

The main difference from convention dialogue systems is that we include information from the engine. Since the image edit engine displays the image, executes edits and stores edit history, we hypothesize that including this information can help our system achieve a better policy. One example is, if the edit history is empty, users will unlikely to request an Undo. Examples of our state features is presented in Table 1.

컨벤션 대화 시스템과의 주요 차이점은 엔진의 정보를 포함한다는 것입니다. 이미지 편집 엔진은 이미지를 표시하고 편집을 실행하며 편집 기록을 저장하므로 이 정보를 포함하면 시스템이 더 나은 정책을 달성하는 데 도움이 될 수 있다는 가설을 세웁니다. 편집 기록이 비어있는 경우 사용자가 실행 취소를 요청하지 않을 수 있습니다. 상태 feature의 예는 표 1에 나와 있습니다.

Type	Feature Type	Examples
Speech	Distribution over possible values	<i>intent, attribute</i>
Gestures	Binary	<i>image_path</i>
Image edit engine	Binary	<i>has_next_history</i>

Table 1: Example state features used for dialogue policy learning

다이얼로그 정책 학습에 사용되는 state 피쳐 예

## Action

We designed 4 actions for our dialogue system: (i) Request, (ii) Confirm, (iii) Query, and (iv) Execute. Request and Confirm actions are each paired with a slot. Request asks users for the value of a slot and Confirm asks users whether the current value stored by the system is correct. Query takes the current value in slot object and queries the vision engine to predict segmentation masks of object. Execute is paired with an intent. Execute passes its paired intent and the intent's children slots (Figure 1) to the image edit engine for execution. If any of the arguments are missing, the execution will fail, and the image will remain unchanged.

우리는 대화 시스템을 위해 (i) 요청(Request), (ii) 확인(Confirm), (iii) 쿼리(Query) 및 (iv) 실행(Execute)의 4 가지 action을 설계했습니다. 요청(Request) 및 확인(Confirm) action은 각각 슬롯과 쌍을 이룹니다. 요청(Request)은 사용자에게 슬롯 값을 요구하고 확인(Confirm)은 사용자가 시스템에 저장된 현재 값이 올바른지 묻습니다. 쿼리(Query)는 슬롯 개체의 현재 값을 가져 와서 비전

엔진을 쿼리하여 개체의 분할 마스크를 예측합니다. 실행(Execute)은 인텐트와 쌍을 이룹니다. 실행(Execute)는 페어링 된 인텐트(intent)와 인텐트(intent)의 하위 슬롯 (그림 1)을 이미지 편집 엔진에 전달하여 실행합니다. 인수가 누락되면 실행이 실패하고 이미지는 변경되지 않습니다.

Unlike information-seeking dialogue systems [6, 20] which attempt to query a database at every turn, we make Query an independent action because of two reasons: (i) modern vision engines are mostly based on Convolutional Neural Networks (CNNs) and frequent queries may introduce performance latency; (ii) segmentation results should be stored, and consecutive queries will override previous results.

매번 데이터베이스를 쿼리하려고 시도하는 정보를 찾는 대화 시스템 [6, 20]과는 달리, 우리는 두 가지 이유로 쿼리(Query)를 독립적인 action으로 만듭니다. (i) 현대 비전 엔진은 주로 CNN (Convolutional Neural Networks)을 기반으로 하며 빈번한 쿼리는 성능 지연을 초래할 수 있습니다. (ii) 분할(segmentation) 결과가 저장되어야 하며 연속 쿼리(Query)는 이전 결과를 덮어씁니다.

## Reward

We define two reward functions. The first reward function is defined based on PyDial [19]. The reward is given at the end of a dialogue and defined as  $20 * 1(D) - T$ , where 20 is the success reward, 1(D) is the success indicator and T is the length of the dialogue. The second reward function gives a positive reward  $r_p$  when an user goal is completed, and a negative reward  $r_n$  if an incorrect edit is executed. The main idea for the second reward function is that since image editing dialogues have multiple image edit requests, additional supervision reward will better help train the dialogue system. However, we did not observe a huge difference between the two reward functions in our initial experiments. Therefore, we only present the results on the first reward function.

우리는 두 가지 reward 함수를 정의합니다. 첫 번째 Reward 함수는 PyDial [19]에 따라 정의됩니다.

Reward은 대화의 끝에 주어지며  $20 * \mathbb{1}(D) - T$ 로 정의됩니다. 여기서 20은 성공 Reward, 1(D)는 성공 표시기(indicator), T는 대화 길이입니다. 두 번째 Reward 함수는 사용자 목표가 완료되면

긍정적인 Reward  $r^p$  을 제공하고, 부정확 한 편집이 실행되면 부정적인 Reward  $r^n$  을 제공합니다. 두 번째 Reward 함수의 주요 아이디어는 이미지 편집 대화에는 여러 개의 이미지 편집 요청이 있기 때문에 추가 감독(supervision) Reward이 대화 시스템 교육에 더 도움이 된다는 것입니다. 그러나 초기 실험에서 두 Reward 함수간에 큰 차이는 관찰되지 않았습니다. 따라서 첫 Reward 함수에 대한 결과만 제시합니다.

## Transition

Our transitions are based on the user simulator and the image edit engine. Every time step  $t$ , the system observes belief state  $b_t$  and outputs system action  $a_t$ . The image edit engine observes system action  $a_t$  and update its state to  $b_{t+1}^e$ . The user simulator then observes both  $b_t$  and  $a_t$  then updates its state to  $b_{t+1}^u$ . Next state  $b_t = b_t^u \oplus b_t^e$  will pass to the system for the next turn. Both the user simulator and the image edit engine are updated according to predefined rules.

Transitions은 사용자 시뮬레이터와 이미지 편집 엔진을 기반으로 합니다. 단계 (t)마다, 시스템은 belief 상태 (bt)를 관찰하고 시스템 동작을 출력한다. 이미지 편집 엔진은 시스템 동작을 관찰하고 상태를  $b_{t+1}^e$ 로 업데이트합니다. 그런 다음 사용자 시뮬레이터는  $b_t$ 를 모두 관찰 한 다음 상태를  $b_{t+1}^u$ 로 업데이트합니다. 다음 상태  $bt = b_{t+1}^u \oplus b_{t+1}^e$ 는 다음 차례를 위해 시스템으로 전달됩니다. 사용자 시뮬레이터와 이미지 편집 엔진 모두 사전 정의 된 규칙에 따라 업데이트됩니다.



## Dialogue Policy

We present two policies for dialogue management.  
우리는 대화 관리를 위한 두 가지 정책을 제시합니다.

**Rule-based:** We hand-crafted a rule-based policy to serve as our baseline. The rule-based policy first requests the intent from the user. After knowing the intent, it then requests all the slots that correspond to that particular intent and then executes the intent. To obtain the localized region (object\_mask\_str), the rule-based policy first queries the vision engine and then requests object\_mask\_str if the vision engine result is incorrect.

**Rule-based :** 우리는 baseline 역할을 하는 규칙 기반 정책을 수작업(hand-crafted)으로 만들었습니다. 규칙 기반 정책은 먼저 사용자에게 인텐트를 요청합니다. 인텐트를 알면 해당 인텐트에 해당하는 모든 슬롯을 요청한 다음 인텐트를 실행합니다. localized 영역 (object\_mask\_str)을 얻기 위해 규칙 기반 정책은 먼저 비전 엔진을 쿼리 한 다음 비전 엔진 결과가 잘못된 경우 object\_mask\_str을 요청합니다.

**Deep Q-Network:** Deep Q-Network [16] combines artificial neural networks and reinforcement learning and takes state  $b$  and  $t$  as inputs to approximate action values  $Q(s_t, a_t)$  for all action  $a$ . Deep Q-Networks are shown to succeed in spoken dialogue systems learning [19] due to its capability to model uncertainty in spoken language understanding and large domain space.

**Deep Q-Network :** Deep Q-Network [16]은 인공 신경 네트워크와 강화 학습을 결합하고 상태  $b$  and  $t$ 를 모든 action에 대한 action 값  $Q(s_t, a_t)$ 에 대한 입력으로 취합니다. Deep Q-Networks는 음성 언어 이해와 넓은 도메인 공간에서 불확실성을 모델링하는 능력으로 인해 음성 대화 시스템 학습에서 성공한 것으로 나타났습니다 [19].

## 4 Conversational Image Editing System

In this section, we first present the ontology used in our system, then describe the role of each system component in further detail.

이 섹션에서는 먼저 시스템에 사용된 온톨로지를 제시한 다음 각 시스템 구성 요소의 역할에 대해 자세히 설명합니다.

### 4.1 Domain Ontology

Conversational Image Editing is a multimodal dialogue domain that consists of multiple user intents and a natural hierarchy. Intents are high level image edit actions that may come with arguments or entities [13, 14]. Also, most edit targets are localized regions in the image. This introduces a hierarchy where the system has to first request the name of the object and then query vision engine to obtain the object's segmentation mask.

대화식 이미지 편집은 여러 사용자 인텐트와 natural 계층으로 구성된 멀티 모달 대화 도메인입니다. 인텐트는 인수 또는 엔티티와 함께 제공되는 고급 이미지 편집 action입니다 [13, 14]. 또한 대부분의 편집 대상은 이미지에서 localized 영역입니다. 이것은 시스템이 먼저 객체의 이름을 요청한 다음 비전 엔진을 쿼리하여 객체의 분할 마스크를 가져와야 하는 계층 구조를 소개합니다.

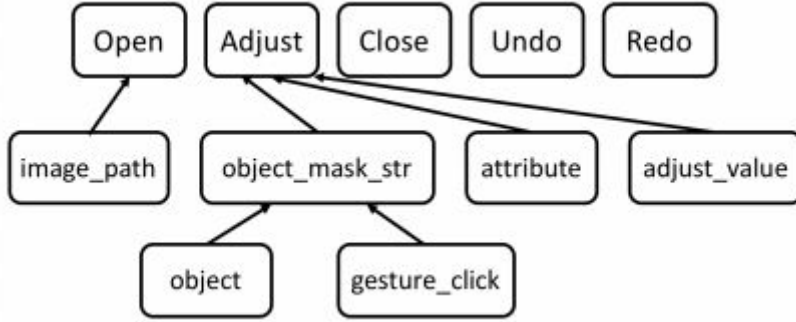


Figure 1: This figure depicts the domain ontology of our conversational image editing system. Top level nodes represent intents; Mid level and low level nodes represent slots. Arrows indicate dependencies. Mid level nodes that directly point to top level nodes are arguments directly associated with that intent. Right three intents do not require additional arguments.

그림 1 :이 그림은 대화형 이미지 편집 시스템의 도메인 온톨로지를 보여줍니다. 최상위 노드는 인텐트를 나타냅니다. 중간 레벨 및 낮은 레벨 노드는 슬롯을 나타냅니다. 화살표는 종속성을 나타냅니다. 최상위 노드를 직접 가리키는 중간 레벨 노드는 해당 인텐트와 직접 연관된 인수입니다. 오른쪽 세 가지 인텐트에는 추가 인수가 필요하지 않습니다.

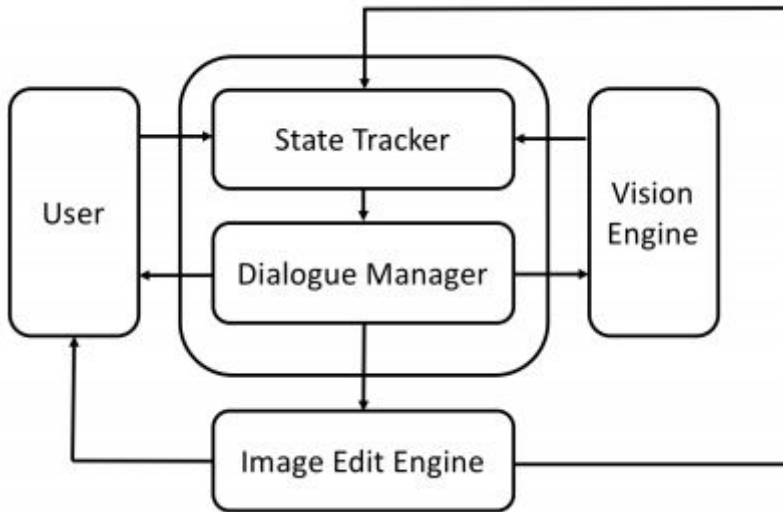


Figure 2: This figure illustrates the components in our system and interactions with the user. State Tracker observes information from User and Image Edit Engine then passes the state to Dialogue Manager. Dialogue Manager then selects an action. If the action is Query, Dialogue Manager will query Vision Engine and store the results in the state. The action will then be observed by both the User and Image Edit Engine.

그림 2 :이 그림은 시스템의 구성 요소와 사용자와의 상호 작용을 보여줍니다. 상태 추적기는 사용자 및 이미지 편집 엔진의 정보를 관찰 한 후 상태를 대화 관리자로 전달합니다. 그런 다음 대화 관리자가 action을 선택합니다. action이 쿼리 인 경우 대화 관리자는 Vision Engine을 쿼리하고 결과를 상태에 저장합니다. 이 action은 사용자 및 이미지 편집 엔진 모두에서 관찰됩니다.

We handpicked a subset of user intents from DialEdit [13] and Editme [15] corpus. The user intents are Open, Adjust, Close, Undo, Redo. Open and Close are inspired by

Manuvinakurike et al. [14]; Adjust modifies the attributes of a region; Undo reverts incorrect edits and Redo can redo edits if Undo is accidentally executed. Open requires slot image\_path; Adjust requires slots object\_mask\_str, adjust\_value, and attribute. Slot object\_mask\_str further depends on slots object, and gesture\_click. Close, Undo, and Redo do not require slots. Our ontology is depicted in Figure 1.

우리는 **DialEdit [13] 및 Editme [15] 코퍼스에서 사용자 인텐트의 하위 집합을 직접** 선택했습니다. **사용자 인텐트는 Open, Adjust, Close, Undo, Redo**입니다. Open and Close는 Manuvinakurike et al.에서 영감을 얻었습니다. [14]; **Adjust는 영역의 속성을** 수정합니다. Undo는 잘못된 편집을 되돌리고 실수로 실행 취소가 실행 된 경우 다시 실행은 편집을 다시 실행할 수 있습니다. **열기(Open)에는 슬롯 image\_path가 필요합니다.** **Adjust에는 슬롯 object\_mask\_str, adjust\_value 및 속성이 필요합니다. Slot** **object\_mask\_str은 슬롯 오브젝트 및 gesture\_click에 더 의존합니다.** 닫기(Close), 실행 취소(Undo) 및 다시 실행(Redo)에는 슬롯이 필요하지 않습니다. 우리의 온톨로지는 그림 1에 묘사되어있다.

## 4.2 Components

Our system architecture consists of four components: (i) Multimodal State Tracker (ii) Dialogue Manager (iii) Vision Engine (iv) Image Edit Engine.

시스템 아키텍처는 다음 **네 가지 구성 요소로 구성됩니다. (i) 멀티 모달 상태 추적기 (ii) 대화 관리자 (iii) 비전 엔진 (iv) 이미지 편집 엔진.**

### Multimodal State Tracker

Input to our state tracker consists of three modalities (i) user utterance, (ii) gestures (iii) image edit engine state. For (i), we trained a two-layer bi-LSTM on the Editme [15] corpus for joint intent prediction and entity detection. We select the Editme corpus because it is currently the only available dataset which contains image editing utterances. Editme has about 9k utterances and 18 different intents. Since Editme is collected by crowd-sourcing image edit descriptions, the utterances are often at a high level, and the annotation can span up to a dozen words. For example, in Editme, "lower the glare in the back room wall that is extending into the doorway" has the following labels: intent is labeled as "adjust"; attribute is labeled as "the glare"; region is labeled as "in the back room wall that is extending into the doorway". Our tagger achieved 0.80% intent accuracy and 58.4% F1 score on the validation set. Since there exists a discrepancy between our ontology and Editme, an additional string matching module is added to detect numerical values and intents not present in Editme. For (ii), we directly take the output as state values. That is, if gestures are present, then the gestures slot values will 1. For (iii), we designed several features including ones mentioned in Table Table 1. The final output is the concatenation of the results of (i), (ii) and (iii), which is the belief state B in the previous section.

**상태 추적기에 대한 입력은 세 가지 양식 (i) 사용자 발화, (ii) 제스처 (iii) 이미지 편집 엔진 상태로 구성됩니다. (i)의 경우, 조인트 인텐트 예측(prediction) 및 엔티티 감지(detection)를 위해 Editme [15] 모음에서 2 계층 bi-LSTM을 학습했습니다. Editme 코퍼스는 현재 이미지 편집 발언이 포함 된 유일한 사용 가능한 데이터 셋이므로 Editme 코퍼스를 선택합니다. Editme은 약 9k 발화와 18 개의 다른 인텐트를 가지고 있습니다. Editme은 클라우드 소싱 이미지 편집 기술(descriptions)으로 수집되기 때문에 발화의 수준이 높으며 주석은 최대 12 개의 단어로 확장 될 수 있습니다. 예를 들어, Editme에서 "lower the glare in the back room wall that is extending into the doorway"에는 다음 레이블이 있습니다. 인텐트는 "adjust"으로 표시됩니다. 속성은 "the glare"로 표시됩니다. 영역은 "문간으로 뻗어가는 뒷방벽에(in the back room wall that is extending into the doorway)"로 표시됩니다. 우리의 태거는 검증 세트에서 0.80 % 인텐트 정확도와 58.4 % F1 점수를 달성했습니다. 온톨로지와 Editme간에 불일치가 있기 때문에 Editme에 없는 숫자 값과 인텐트를 감지하기 위해 추가**

문자열 일치 모듈이 추가됩니다. (ii)의 경우 출력을 상태 값으로 직접 가져옵니다. 즉, 제스처가 있는 경우 제스처 슬롯 값은 1이됩니다. (iii)의 경우 표 표 1에 언급된 기능을 포함한 여러 기능을 설계했습니다. 최종 결과는 (i), (ii)의 결과를 연결한 것입니다. 그리고 (iii) 이전 섹션의 belief 상태 B입니다.

## Dialogue Manager

Dialogue manager is a rule-based policy or a parametrised policy that observes the dialogue state, and performs actions to complete user's image edit requests. Detailed of actions are presented in the previous section.

대화 관리자는 대화 상태를 관찰하고 사용자의 이미지 편집 요청을 완료하기 위한 action을 수행하는 규칙 기반 정책 또는 매개 변수화 된 정책입니다. 자세한 작업은 이전 섹션에 나와 있습니다.

## Vision Engine

Vision engine performs semantic segmentation and is called when system selects action Query.

Vision engine takes the image and slot object's value as query and outputs a list of candidate masks to be shown to the user. If present, gesture\_click will be used filter the candidate masks by checking whether it overlaps with any of the candidate masks. We leave extracting state features from vision engine as future work.

비전 엔진은 시맨틱 세그먼트 화를 수행하며 시스템이 action Query를 선택할 때 호출됩니다. 비전 엔진은 이미지 및 슬롯 객체의 값을 쿼리로 가져와서 사용자에게 표시 할 후보 마스크 목록을 출력합니다. 존재하는 경우, gesture\_click은 후보 마스크가 임의의 후보 마스크와 겹치는지를 검사함으로써 후보 마스크를 필터링하는데 사용될 것이다. 비전 엔진에서 추출한 상태 features을 향후 작업으로 남겨 둡니다.

## Image Edit Engine

Image edit engine is an image edit application that acts as an interface to our user and as an API to our system. At every turn, our system loads the candidate masks stored in slot object\_mask\_str to the engine for display. When system performs an Execute action, the executed intent and associated arguments will be passed to the engine for execution. If the intent and slots are valid, then an edit can be performed. Else, the execution will result in failure, and image will remain unchanged. We developed a basic image edit engine using the open-source OpenCV [2] library. The main features of our engine include image edit actions, region selectors, and history recording.

이미지 편집 엔진은 사용자의 인터페이스 및 시스템의 API 역할을 하는 이미지 편집 응용 프로그램입니다. 매번 시스템은 슬롯 object\_mask\_str에 저장된 후보 마스크를 엔진에 로드하여 표시합니다. 시스템이 실행 action를 수행하면 실행된 인텐트 및 관련 인수가 엔진에 전달되어 실행됩니다. 인텐트와 슬롯이 유효하면 편집을 수행 할 수 있습니다. 그렇지 않으면 실행이 실패하고 이미지는 변경되지 않습니다. 우리는 오픈 소스 OpenCV [2] 라이브러리를 사용하여 기본 이미지 편집 엔진을 개발했습니다. 엔진의 주요 기능에는 이미지 편집 작업, 영역 선택기 및 기록 기록이 있습니다.

## 4.3 Multimodal Multi-goal User Simulator

We developed an agenda based [18] multimodal multi-goal user simulator to train our dialogue system. User agenda is a sequence of goals which needs to be executed in order. A goal is defined as an image edit request that is composed of an intent and it depending slots in the tree Figure 1. A successfully executed goal will be removed from the agenda, and the dialogue is considered as success when all the user goals are executed. If an incorrect intent is executed, the simulator will add an Undo goal to the agenda to undo the incorrect edit. On the other hand, if the system incorrectly executes Undo, the simulator will

add a Redo goal to the agenda. Our simulator is programmed not to inform object\_mask\_str unless asked to.

우리는 대화 시스템을 훈련시키기 위해 의제에 기초한 [18] **멀티모달 다중 목표(goal) 사용자 시뮬레이터**를 개발했다. **사용자 아젠다는 순서대로 실행해야 하는 일련의 목표(goal)**입니다. **목표(goal)는 인텐트와 트리의 슬롯에 따라 구성된 이미지 편집 요청으로 정의됩니다.** 그림 1. 성공적으로 실행된 **목표(goal)**는 agenda에서 제거되고 모든 사용자 **목표(goal)**가 실행될 때 대화가 성공으로 간주됩니다. **잘못된 인텐트가 실행되면 시뮬레이터는 Undo 목표(goal)를 아젠다에 추가하여 잘못된 편집을 취소합니다.** 반면에 시스템이 **Undo를 잘못 실행하면 시뮬레이터가 Redo 목표(goal)를 agenda에 추가**합니다. 시뮬레이터는 요청하지 않는 한 object\_mask\_str에 알리지 않도록 프로그래밍되어 있습니다.

## 5 Simulated User Evaluation

### 5.1 Experimental setting

User Goal We sampled 130 images from MSCOCO [12] and randomly split them into 100 and 30 for training and testing, respectively. Goals are randomly generated using our ontology and the dataset.

**사용자 목표(goal)** 우리는 MSCOCO [12]에서 130 개의 이미지를 샘플링하여 각각 훈련과 테스트를 위해 100과 30으로 무작위로 나누었습니다. **목표(goal)**는 온톨로지와 데이터 세트를 사용하여 무작위로 생성됩니다.

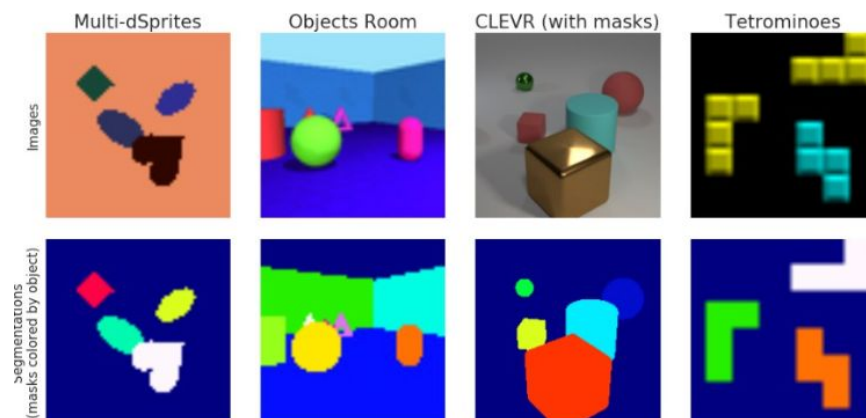
For simulated users, we set the number of goals to 3 which starts with an Open goal, followed by an Adjust goal, then ends with a Close goal. Maximum dialogue length is set to 20. To simulate novice behavior, a  $\theta$  parameter is defined as the probability a slot will be dropped in the first pass. We set  $\theta$  to 0.5.

**시뮬레이션된 사용자의 경우 목표(goal) 수를 3으로 설정하여 Open 목표(goal)를 시작으로 설정하고 조정(Adjust) 목표(goal)를 시작한 다음 Close 목표(goal)로 끝냅니다. 최대 대화 길이는 20으로 설정됩니다.** 초보자 행동을 시뮬레이션하기 위해  $\theta$  매개 변수는 첫 번째 패스에서 슬롯이 떨어질 확률로 정의됩니다.  $\theta$ 를 0.5로 설정했습니다.

### Vision Engine

We directly take ground truth segmentation masks from the dataset as query results.

**우리는 데이터 세트에서 ground truth segmentation masks를 쿼리 결과로 직접 가져옵니다.**



Training Details For DQN, we set the hidden size to 40. We set batch size to 32 and freeze rate to 100. We used 0.99 for reward decay factor  $\gamma$ . The size of experience replay pool is 2000. We used Adam [10] and set the learning rate to  $1e-3$ .

Training 세부 정보 DQN의 경우 hidden size를 40으로 설정했습니다. 배치 크기를 32로 설정하고 freeze rate를 100으로 설정했습니다. Reward 감쇠 계수  $\gamma$ 로 0.99를 사용했습니다. experience replay 풀의 크기는 2000입니다. 우리는 Adam [10]을 사용하고 learning rate를  $1e-3$ 으로 설정했습니다.

## 5.2 Results

We evaluate four metrics under different semantic error rates (SER): (i) Turn (ii) Reward (iii) Goal (iv) Success. (i), (ii), and (iv) follow standard task-oriented dialogue system evaluation. Since image editing may contain multiple requests, having more goals executed should indicate a more successful dialogue, which success rate cannot capture. Therefore, we also include (iii).

우리는 서로 다른 시맨틱 오류율 (SER)로 4 가지 메트릭을 평가합니다. (i) Turn (ii) Reward (iii) Goal (iv) Success. (i), (ii) 및 (iv) 표준 태스크 지향 대화 시스템 평가를 따릅니다. 이미지 편집에는 여러 개의 요청이 포함될 수 있으므로 더 많은 목표를 실행하면 더 성공적인 대화를 나타내야 합니다. 따라서 우리는 (iii)도 포함합니다.

Table 2 shows the results of our simulated user evaluation. At low SER ( $<0.2$ ), we can see that rule-based and DQN can successfully complete the dialogue. As SER increases ( $>0.2$ ), the success rate of the rule-based policy decreases. On the other hand, our DQN policy manages to learn a robust policy even under high SERs and achieved 90% success rate.

표 2는 시뮬레이션된 사용자 평가 결과를 보여줍니다. 낮은 SER ( $<0.2$ )에서는 규칙 기반 및 DQN이 대화를 성공적으로 완료 할 수 있음을 알 수 있습니다. SER이 증가하면 ( $> 0.2$ ) 규칙 기반 정책의 성공률이 감소합니다. 반면, DQN 정책은 높은 SER 환경에서도 강력한 정책을 배우고 90 %의 성공률을 달성했습니다.

SER	Rule-based				DQN			
	Turn	Reward	Goal	Success	Turn	Reward	Goal	Success
0.0	7.5	13.50	3.0	1.0	7.43	13.56	3.0	1.0
0.1	7.2	13.80	3.0	1.0	7.27	13.73	3.0	1.0
0.2	9.16	11.13	2.9	0.97	8.33	12.66	3.0	1.0
0.3	15.0	4.6	2.87	0.93	9.23	11.76	3.0	1.0
0.4	17.3	-6.1	2.30	0.53	12.1	8.2	2.93	0.97
0.5	18.2	-16.8	1.50	0.07	13.6	5.3	2.8	0.90

Table 2: Results of our simulated user evaluation. SER denotes semantic error rate. Our DQN policy still retains high success rate even under high semantic errors compared to rule-based baseline.

표 2 : 시뮬레이션 된 사용자 평가 결과. SER은 시맨틱 오류율을 나타냅니다. 우리의 DQN 정책은 규칙 기반 기준과 비교할 때 높은 시맨틱 오류로도 높은 성공률을 유지합니다.

## 6 Real User Study



While the experiment in the previous section shows effectiveness under a simulated setting, real users may exhibit a completely different behavior in an multimodal dialogue. Therefore, we built a web interface(Appendix A for our system. Our interface allows text input and gestures. Users can input gesture\_click by putting a marker on the image, and input object\_mask\_str by putting a bounding box on the image as the localized region.

이전 섹션의 실험은 시뮬레이션 된 설정에서 효과를 보여 주지만 실제 사용자는 멀티 모달 대화에서 완전히 다른 동작을 보일 수 있습니다. 따라서 웹 인터페이스 (시스템의 부록 A. 인터페이스는 텍스트 입력과 제스처를 허용 함)를 만들었습니다. 사용자는 이미지에 마커를 넣어서 gesture\_click을 입력하고, 이미지에 경계 상자를 localized region으로 넣어 object\_mask\_str을 입력 할 수 있습니다.

We recruited 10 subjects from the author's affiliation and conducted a study testing them against our Rule-based Policy and the DQN policy. One person among the 10 is familiar with image editing. 10 goals were sampled from the test set, and each person is asked to complete two dialogues of the same policy. In every dialogue, there is only one Adjust goal and it is presented to the user in semantic form. Since number of goals is reduced, we set maximum dialogue length to 10.

저자의 소속에서 10 명의 피험자를 모집하고 규칙 기반 정책 및 DQN 정책에 대해 테스트하는 연구를 수행했습니다. 10 명 중 한 명이 이미지 편집에 익숙합니다. 테스트 세트에서 10 개의 목표가 샘플링되었으며 각 개인은 동일한 정책에 대한 두 개의 대화를 완료해야 합니다. 모든 대화에는 하나의 Adjust 목표(goal) 만 있으며 사용자에게 시맨틱 형태로 표시됩니다. 목표의 수가 줄어들었으므로 최대 대화 길이를 10으로 설정했습니다.

Table 3 presents the turns and success rate in our real user study. Both policies achieved 0.8 success rate, while the DQN policy has fewer number of turns. Due to the small amount of goals, we focus on the insights gained from manually inspecting the dialogue data.

표 3은 실제 사용자 연구의 turns과 success을 보여줍니다. 두 정책 모두 0.8 성공률을 달성했으며 DQN 정책은 turn 횟수가 적습니다. 적은 양의 목표로 인해 대화 데이터를 수동으로 검사하여 얻은 통찰력에 중점을 둡니다.

	Rule-based		DQN	
	Turn	Success	Turn	Success
Real User	5.7	0.8	3.9	0.8

Table 3: Result metrics of our real user study on the 10 sampled goals from the test set. Rule-based and DQN policies have the same 0.8 success rate, and the DQN policy has fewer number of turns.

표 3 : 테스트 세트에서 10 개의 샘플링 된 목표(goal)에 대한 실제 사용자 연구의 결과 메트릭. 규칙 기반 및 DQN 정책의 성공률은 0.8이며 DQN 정책의 회전 횟수는 더 적습니다.

Turn Number: We discovered the turn number in our study mostly depend on the user behavior. For users who prefer to express more information, the turn number is smaller. For users who prefer to interact mostly with text and leave gestures as a last resort, the turn number is larger.

turn 수 : 연구에서 turn 수는 주로 사용자 행동에 달려 있음을 발견했습니다. 더 많은 정보를 표현하려는 사용자의 경우 turn 수가 적습니다. 텍스트와 주로 상호 작용하고 제스처를 최후의 수단으로 사용하려는 사용자의 경우 turn 수가 더 큼니다.

Success Rate: The semantic form of the user goal makes it easy for the users to express and inform slot values (Appendix A)), so that most dialogues are successful. All the failure cases are incorrect

understanding of the slot adjust\_value, which is a result of the string matching module in our state tracker. Table 4 presents a success dialogue and Table 5 presents a failure dialogue.

성공률 : 사용자 목표(goal)의 시맨틱 form는 사용자가 슬롯 값(부록 A)을 쉽게 표현하고 알릴 수 있도록 하여 대부분의 대화가 성공한다.

모든 실패 사례는 슬롯 adjust\_value에 대한 잘못된 이해로, 이는 우리의 상태 추적기에 있는 문자열 매칭 모듈의 결과물이다.

표 4는 성공 대화를 나타내고 표 5는 실패 대화를 나타냅니다.

Turn	Speaker	Utterance	Gesture
1	User	increase the man's saturation by 10	object_mask_str
1	System	What object_mask_str do you want?	
2	User	man	
2	System	Execute: intent=adjust, adjust_value=10 attribute=saturation, object_mask_str=iVBOR	
Success			

Table 4: Success dialogue example with DQN policy. The user first interacts only with text input in the first turn, then inputs both the "man" (object) and gesture (object\_mask\_str) in the second turn.

표 4 : DQN 정책을 사용한 성공 대화 예

사용자는 먼저 첫 번째 턴에서 텍스트 입력과 상호 작용 한 다음 두 번째 턴에서 "man"(개체) 및 제스처(object\_mask\_str)를 모두 입력합니다.

Turn	Speaker	Utterance	Gesture
1	User	make the man 30% less bright	object_mask_str
1	System	Execute: intent=adjust, adjust_value=30 attribute=brightness, object_mask_str=iVBOR	
Failure			

Table 5: Failure dialogue example with DQN policy. In the first turn, our issues a image edit request and also the localized region (object\_mask\_str). However, our state tracker failed to interpret "30% less bright" into the value "-30", and executes with the incorrect value 30 (adjust\_value).

표 5 : DQN 정책이있는 실패 대화 예

첫 번째 턴에서는 이미지 편집 요청과 localized 영역(object\_mask\_str)을 발행합니다. 그러나 상태 추적기는 "30 % 덜 밝음"을 "-30"값으로 해석하지 못했으며 잘못된 값 30 (adjust\_value)으로 실행됩니다.

## 7 Conclusions and Future Work

We present a multimodal dialogue system for Conversational Image Editing. We derived the POMDP formulation for Conversational Image Editing, and our simulated evaluation results show that the DQN policy significantly outperforms a sophisticated rule-based baseline under high semantic error rates.

Our real user study shows that the language understanding component is crucial to success and real users may exhibit more complex behavior. Future work includes frame-based state tracking, expanding the ontology to incorporate more intents and modeling multimodal user behavior.

대화 이미지 편집을 위한 멀티 모달 대화 시스템을 소개합니다. 우리는 대화 이미지 편집을 위한 POMDP 공식을 도출했으며, 시뮬레이션 된 평가 결과는 DQN 정책이 높은 시맨틱 오류율에서 정교한 규칙 기반 기준을 크게 능가 함을 보여줍니다. 실제 사용자 연구에 따르면 언어 이해 구성 요소가 성공에 중요하며 실제 사용자는 더 복잡한 행동을 보일 수 있습니다. 향후 작업에는 프레임 기반 상태 추적, 온톨로지를 확장하여 더 많은 인텐트를 통합하고 다중 사용자 행동을 모델링하는 작업이 포함됩니다.

## Acknowledgement

This work is in part supported through collaborative participation in the Robotics Consortium sponsored by the U.S Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement W911NF-10-2-0016. This work should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. We would like to thank the anonymous reviewers for their insightful comments.

이 작업은 협업 기술 제휴 프로그램, 협력 계약 W911NF-10-2-0016에 따라 **미 육군 연구소가 후원하는 로봇 공학 컨소시엄에 협력적으로 참여함으로써 부분적으로 지원됩니다.** 이 작업이 미국 정부의 육군 연구소의 공식 정책을 나타내거나 암시하는 것으로 해석해서는 안 됩니다. 미국 정부는 저작권 표시에도 불구하고 정부 목적을 위해 재인쇄물을 복제하고 배포할 수 있다. 익명의 검토 자에게 통찰력있는 의견을 보내 주셔서 감사합니다.

## References

- [1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, 2018.
- [2] G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.
- [3] Pawel Budzianowski, Stefan Ultes, Pei hao Su, Nikola Mrksic, Tsung-Hsien Wen, Iñigo Casanueva, Lina Maria Rojas-Barahona, and Milica Gasic. Sub-domain modelling for dialogue management with hierarchical reinforcement learning. In SIGDIAL Conference, 2017.
- [4] Trung Bui. Multimodal dialogue management-state of the art. 2006.
- [5] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José M.F. Moura, Devi Parikh, and Dhruv Batra. Visual Dialog. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
- [6] Bhuwan Dhingra, Lihong Li, Xiujuan Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. Towards end-to-end reinforcement learning of dialogue agents for information access. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, pages 484–495, 2017.
- [7] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. Frames: a corpus for adding memory to goal-oriented dialogue systems. In Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, pages 207–219, 2017.
- [8] Dilek Hakkani-Tür, Malcolm Slaney, Asli Celikyilmaz, and Larry Heck. Eye gaze for spoken language understanding in multi-modal conversational interactions. In Proceedings of the 16th International Conference on Multimodal Interaction, pages 263–266. ACM, 2014.

- [9] Larry Heck, Dilek Hakkani-Tür, Madhu Chinthakunta, Gokhan Tur, Rukmini Iyer, Partha Parthasarathy, Lisa Stifelman, Elizabeth Shriberg, and Ashley Fidler. Multi-modal conversational search and browse. In First Workshop on Speech, Language and Audio in Multimedia, 2013.
- [10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
- [11] Gierad P Laput, Mira Dontcheva, Gregg Wilensky, Walter Chang, Aseem Agarwala, Jason Linder, and Eytan Adar. Pixeltone: A multimodal interface for image editing. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pages 2185–2194. ACM, 2013.
- [12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014.
- [13] Ramesh Manuvinakurike, Jacqueline Brixey, Trung Bui, Walter Chang, Ron Artstein, and Kallirroi Georgila. DialEdit: Annotations for Spoken Conversational Image Editing. In Proceedings of the 14th Joint ACL - ISO Workshop on Interoperable Semantic Annotation, Santa Fe, New Mexico, August 2018. Association for Computational Linguistics. URL <https://aclanthology.info/papers/W18-4701/w18-4701>.
- [14] Ramesh Manuvinakurike, Trung Bui, Walter Chang, and Kallirroi Georgila. Conversational image editing: Incremental intent identification in a new dialogue task. In Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, pages 284–295, 2018.
- [15] Ramesh R. Manuvinakurike, Jacqueline Brixey, Trung Bui, Walter Chang, Doo Soon Kim, Ron Artstein, and Kallirroi Georgila. Edit me: A corpus and a framework for understanding natural language image editing. In LREC. European Language Resources Association (ELRA), 2018.
- [16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.
- [17] Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao, Asli Çelikyilmaz, Sungjin Lee, and Kam-Fai Wong. Composite task-completion dialogue policy learning via hierarchical deep reinforcement learning. In EMNLP, 2017.
- [18] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers, pages 149–152. Association for Computational Linguistics, 2007.

[19] Stefan Ultes, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, and Steve Young. PyDial: A Multi-domain Statistical Dialogue System Toolkit. In Proceedings of ACL 2017, System Demonstrations, pages 73–78, Vancouver, Canada, July 2017. Association for Computational Linguistics. URL <http://aclweb.org/anthology/P17-4013>.

[20] Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, volume 1, pages 438–449, 2017.

[21] Steve Whittaker and Marilyn Walker. Evaluating dialogue strategies in multimodal dialogue systems. In Spoken Multimodal Human-Computer Dialogue in Mobile Environments, pages 247–268. Springer, 2005.

[22] Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. Proceedings of the IEEE, 101(5):1160–1179, 2013.

## A System Interface

### Prototype Image Editing Dialogue System


Instructions

Select Goal

10

Intent=adjust, object=zebra, attribute=lightness, adjust\_value=5  
Show/Hide object location

Image



Gestures ☐ gesture\_click(marks a point) ☐ object\_mask\_str(selects a region)

Turn Count: 1

User Input

System:  
Hi, this is a prototype image editing dialogue system. How may I help you?

© Adobe Research 2018

Figure 3: Interface of our system prototype. A goal is presented to the user in semantic form. Users can input a click (`gesture_click`) or select a bounding box (`object_mask_str`)

그림 3 : 시스템 프로토타입의 인터페이스. 목표(goal)는 시맨틱 형태로 사용자에게 제시된다. 사용자는 클릭 (`gesture_click`)을 입력하거나 경계 상자 (`object_mask_str`)를 선택할 수 있습니다(select a region)니다.