Improving Proactive Dialog Agents Using Socially-Aware Reinforcement Learning

UMAP' 2023

Eunseo Yang

Table of contents

01

Introduction

Introduction

Conversational AI(CA) agents

(Amazon Alexa: 자연어를 사용하여 식료품 쇼핑 관리, Amazon Chatbot: 고객서비스에 사용)

- CA는 널리 이용되고 있으나,
  단순 '보조' 역할로만 사용되고 있음
- 낮은 대화 지능 때문에 복잡한 협업을 필요로 하는 업무에서는 신뢰하지 않는 추세
  (주식 거래 보조, 비즈니스 자문 등)

현재 CA Systems

- 주로 반응적 동작 (이벤트 상기)
- 비교적 낮은 수준의 적극적 행동

복잡한 작업 돕기 위해서, 더 나은 협력을 만들기 위해서 고도의 적극적 대화 전략*이 필요

적극적 대화 전략*: 자발적이고, 예측적인 행동

- 적절한 적극성은 시스템 유용성, 사용자 만족도 증가시켜서 결국 신뢰성을 얻을 수 있음
- 적절하지 않거나 때를 놓치면 매우 부정적

Introduction

"Therefore, the design, modelling, and implementation of effective and trusted proactive dialog is a delicate task."

→ Proactive dialog modelling을 위해 Reinforcement Learning을 활용하자!

- 사용자의 신뢰도는 시스템이 얼마나 사용자의 기대에 부합하는지에 따라 달라진다는 사전 연구 결과를 반영
- RL 기반 적극적 시스템이 얼마나 사용자의 기대에 맞게 적재적소에 원하는 행동을 할 수 있는지, 그에 따라 사용자의 신뢰 수준이 어떻게 변화하는지 확인

보상 함수에 신뢰를 포함하면, 에이전트는 사용자의 신뢰 수준을 높이는 행동을 취하는 것이 더 높은 보상을 받게 되므로, **신뢰를 중시하는 행동 전략을 학습하는 RL 접근법**을 시도

02

Related Work

Related Work

Proactive Conversational AI

- 역할: 사용자의 활동과 목표를 추적하고 자동으로 예측하여 행동
- 범위: “임무 지향적 적극적 대화”에 초점: 의사결정과정을 소통하고 협상하여 시스템 실패의 위험을 최소화하고 효율적으로 작업을 해결
- 주요 도전 과제: 적극적 행동의 타이밍과 수준
- 문제: 작업 도메인의 복잡성으로 인해 규칙 기반 접근법이 사용되어 옴
  - 생성된 적극적 행동 규칙은 다른 시나리오와 작업 도메인으로 전환될 수가 없었음
  - 충분한 행동을 재현하기 위해 필요한 규칙 세트가 너무 많음
  - 한정된 사용자 세트를 대상으로만 설계됨 (모두의 기대 충족 x, 융통성 x)

Related Work

Human-Computer Trust*

Trust: 에이전트가 개인의 목표를 달성하는 데 도움을 줄 것이라는 태도

- **사용자 신뢰의 중요성**
  - 인간-기계 상호작용에서 사용자의 신뢰는 필수적
    사용자가 시스템을 신뢰하지 않으면, 시스템이 제공하는 정보나 제안을 무시하거나 거부
  - 따라서, 시스템이 사용자의 신뢰를 쌓고 유지하는 것은 상호작용의 성공에 중요
- **적극적 행동과 신뢰의 균형**
  - 에이전트가 너무 적극적이거나 자주 개입할 경우, 사용자는 자신의 선택이나 의견이 충분히 고려되지 않는다고 느낄 수 있음
  - 반면, 에이전트가 너무 소극적이면 사용자가 필요로 하는 지원을 제공하지 못할 수 있음
    사용자의 신뢰를 유지하면서 적절한 수준의 적극성을 발휘할 수 있는 전략 채택이 필요

03

Suggestion

Socially-aware RL based dialog management

Simulated Proactive Dialog Environment

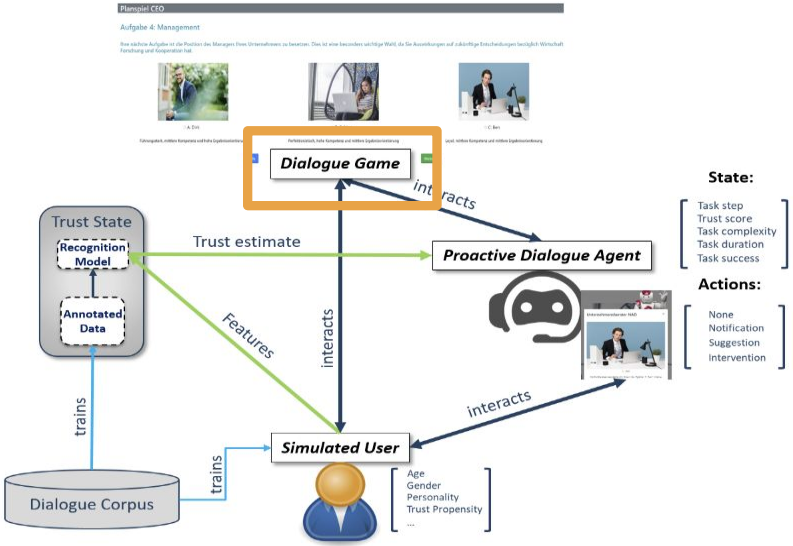


Figure 1: Components and information flows of the simulated RL-based proactive dialog environment.

Dialog Game

Overview

- 사용자가 적극적인 대화 에이전트와 협력하여 각 단계의 작업을 해결하면서 전략적 결정을 내리는 과정을 통해, 신뢰성과 관련된 변수들을 수집하는 연구 목적으로 개발
- 사용자의 성공적인 게임 플레이와 결정은 대화 에이전트의 도움과 피드백에 기반하며, 이를 통해 사용자와 시스템 간의 신뢰 관계를 탐색하고 분석하는 데 필요한 데이터를 마련

Game 구조

- **목표**: 회사의 이익 극대화
- **방법**: 클릭 가능 GUI, Proactive agent
- **과제**: 12단계 작업
- **결정**: 지역 계획, 인사 관리 등에 대한 결정을 수행. 각 결정은 회사 관리의 성공에 영향을 미침

User Interaction

- **선택**: 각 단계마다 다중 선택, 옵션은 3~5개
- **가능한 행동**
  - 옵션 선택
  - 도움 요청 (이전 결정 참고)
  - DA에게 제안 요청
  - 게임 계속하기

신뢰도 측정

- **인지 기반 신뢰:** 사용자는 시스템의 신뢰도 및 인지 된 능력, 예측 가능성, 신뢰성 (reliability)에 대해 5점 리커트 척도로 평가
- **데이터 수집:** 사용자의 인지 기반 신뢰 수준을 측정하는 데 사용

Proactive Dialog Agent

- **매커니즘:** 각 작업 단계마다 최선의 옵션을 선택하기 위해 이전 사용자 결정에 대한 지식을 가지고 게임의 평가 모델을 조회하는 단순한 추론 매커니즘 사용

- **행동 방식 4가지**:
  - None (제안을 요청할 때 까지 기다리기)
  - Notification (알림 메세지 무시할 가능성 주기)
  - Suggestion (예/아니오 답변을 기대)
  - Intervention (사용자의 선택권 X, 자율적으로 에이전트가 옵션을 선택)

Simulated Proactive Dialog Environment

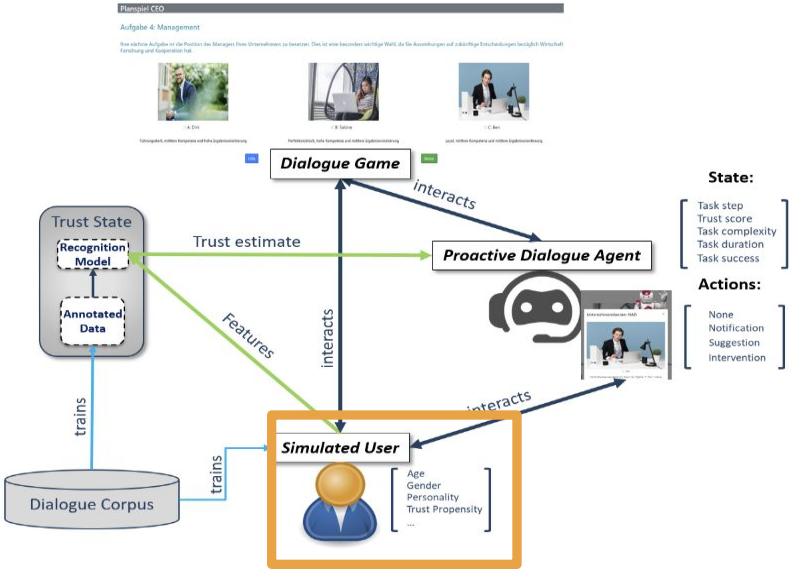


Figure 1: Components and information flows of the simulated RL-based proactive dialog environment.

Simulated User

User가 어떤 사람인가?

- 실제 사용자 대화 데이터를 기반으로 구축
- 사용자들은 순차적 의사결정 문제로 모델링된 대화 게임에서 적극적인 DA와 협력

- **인지 기반 신뢰 관련 변수:** 각 대화 교환마다, 사용자는 시스템에 대한 신뢰성, 사용자가 인지한 시스템의 능력, 예측 가능성, 신뢰성에 대해 자가 보고 방식으로 평가 (1-5)
- **과제 관련 속성 & 시스템 행동 변수:** 복잡성, 대화 교환 기간, 사용자 행동과 같은 과제 관련 속성과 시스템의 행동 (특정 교환에서 발생한 상황과 시스템의 반응에 대한 세부 정보를 제공)
- **사용자 정보:** 나이, 성별, 성격, 도메인 전문 지식 정보 수집 (설문지 통해 얻음)

Simulated User

사용자 대화 관리자: User가 어떤 상호작용을 할까?

- 다양한 사용자 유형을 모델링하여 시뮬레이터가 다양한 사용자의 특성과 반응 패턴을 반영할 수 있도록 했음. **사용자의 성별, 연령, 기술 친화성, 성격** 등의 특성을 포함하는 사용자 프로필을 생성
- JSON, CSV 형태로 저장
- 정의된 프로파일을 기반으로 통계적 분석 or 머신러닝 알고리즘을 사용하여 대화 동안의 패턴, 선호, 반응 유형을 기준으로 구분
- 각 사용자 유형에 대해 예상되는 반응 및 행동 패턴을 시뮬레이션 로직으로 구현
- **확률적 결정:** 각각의 행동은 사용자 모델에 기반해 확률로 결정. 같은 상황에서도 사용자 모델(예: 사용자의 기술 친화도, 신뢰 성향 등)에 따라 다른 반응을 확률적으로 계산

Simulated Proactive Dialog Environment

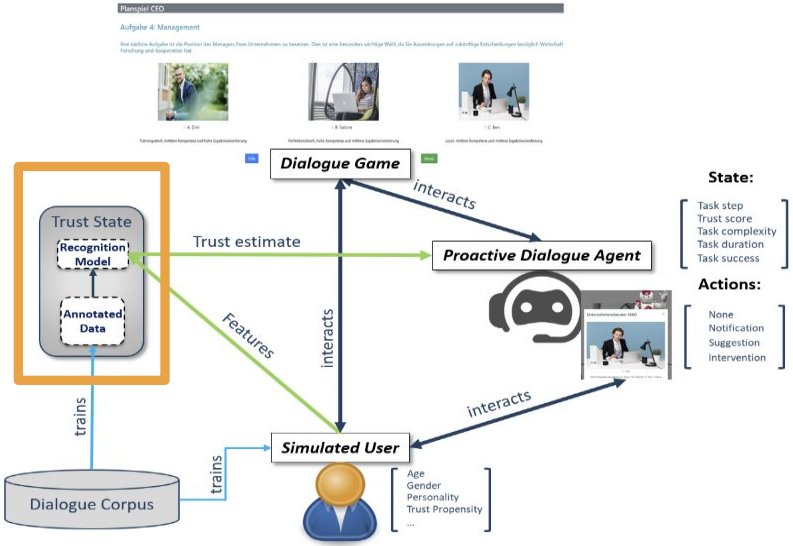


Figure 1: Components and information flows of the simulated RL-based proactive dialog environment.

Trust State Model

실제 사용자 데이터와 시뮬레이션 결과와 유사도를 측정해서 검증

적극적인 대화 에이전트와 상호작용하는 동안 사용자의 신뢰 수준을 예측(1~5)하기 위해 **SVM**을 사용.

- **개인 사용자 매개변수:** 사용자의 **Big5, 나이, 신뢰 성향, 기술 친화도, 도메인 전문성**
- **상호작용 매개변수:** 사용자와 대화 에이전트 간의 상호 작용에서 **발생하는 정보**, 대화 **에이전트의 행동 유형(알림, 제안, 개입, 없음)**, **작업 단계 (작업의 단계, 복잡성)**
- **시간적 상호작용 매개변수:** 대화가 진행되면서 변경될 수 있는 정보, **과거의 대화 내용** (이전 대화에서 에이전트가 얼마나 유용했는가의 기록, 과거의 긍/부정 경험)이나 **사용자의 행동 패턴**(사용자가 일관되게 제안이나 정보에 반응하는지, 어떤 패턴으로 반응하는지)

사용자가 에이전트를 얼마나 신뢰하는지, 그리고 어떤 조건에서 신뢰 수준이 변할 수 있는지 예측할 때 사용

Simulated Proactive Dialog Environment

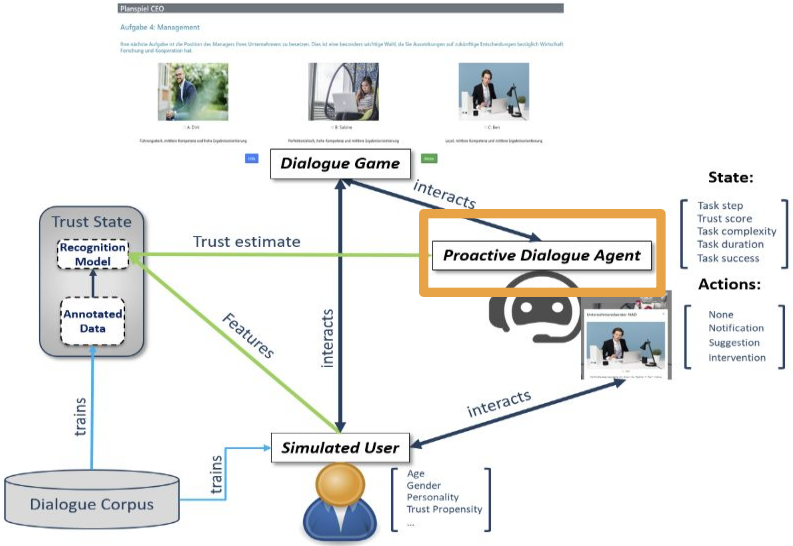
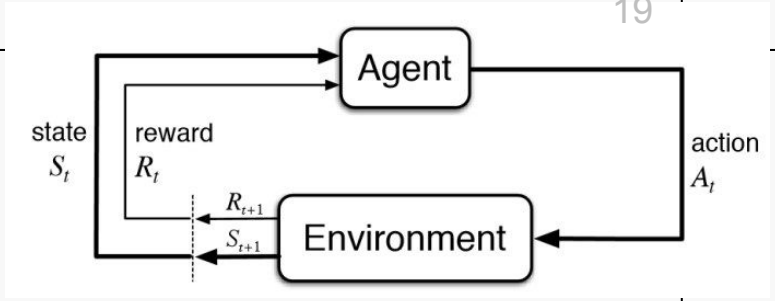


Figure 1: Components and information flows of the simulated RL-based proactive dialog environment.

RL-based Proactive Agent



MDP(Markov Decision Process)

1. **상태(State):** 상태는 에이전트가 인식할 수 있는 환경의 모든 가능한 상황.
2. **행동(Action):** 행동은 에이전트가 선택할 수 있는 모든 가능한 조치
3. **보상(Reward):** 보상은 에이전트가 특정 행동을 취했을 때 환경으로부터 받는 피드백
4. **상태 전이 확률(Transition Probabilities):** 에이전트가 특정 행동을 취했을 때 한 상태에서 다른 상태로 이동할 확률을 나타냅니다. 대화 에이전트의 경우, 이것은 특정 대화 행동이 어떻게 대화의 흐름을 변화시킬 수 있는지를 모델링합니다.

RL-based Proactive Agent

강화학습 요약

1. **문제 정의:** 강화학습 모델을 돌리기 전에 에이전트가 해결해야 할 문제를 MDP*로 정의
   학습 과정에서 에이전트와 환경 간의 상호작용을 어떻게 모델링할지 결정
2. **학습 과정:** 강화학습 알고리즘은 에이전트가 환경과 상호작용하며 학습을 진행
   이 과정에서 에이전트가 보상을 최대화하기 위한 최적의 행동 전략을 채택
3. **정책 평가:** 에이전트는 학습 과정을 통해 다양한 행동을 시도하고, 이러한 행동이
   가져오는 결과(보상)를 평가. 이 데이터를 사용하여 에이전트의 행동 정책을 지속적으로
   개선
4. **정책 최적화:** 학습이 진행됨에 따라, 에이전트는 점차 최적의 행동 정책을 발전시키며,
   이는 결국 MDP의 해결책.

RL-based Proactive Agent

상태(State)	**대화의 현재 단계(작업 수행 단계), 그 단계의 복잡성, 사용자의 신뢰 수준, 이전 작업의 성공 여부, 마지막 작업의 지속 시간**
행동(Action)	**‘None’, ‘Notification’, ‘Suggestion', ‘Intervention’**
보상(Reward)	**사용자의 신뢰 수준, 작업 성공, 작업의 지속 시간**

보상 함수 상세 내용

- **신뢰 보상:** 사용자 신뢰 수준이 5(매우 높음)이면 20점, 1(매우 낮음)이면 -20점의 보상
- **작업 성공 보상:** 작업 성공 점수가 평균보다 높으면 15점, 낮으면 5점의 보상
- **작업 지속 시간 보상:** 작업을 평균 시간 이내에 완료하면 10점의 보상

RL-based Proactive Agent

- **문제 상황:** 대화가 진행될 수 있는 방식이 매우 다양하기 때문에(약 90,000가지 상태), 모든 가능성을 고려하는 전통적인 방식으로는 효과적인 학습이 어려움
- **딥-Q-네트워크(DQN) 사용:** 이 문제를 해결하기 위해, 연구팀은 딥러닝 기반의 강화학습 기법인 딥-Q-네트워크(DQN) 사용. DQN은 크고 복잡한 상태 공간에서도 효과적으로 최적의 행동을 학습 가능
- **네트워크 구성:** DQN은 여러 층(layer)을 가진 인공신경망으로 구성. 이 연구에서는 두 개의 주요 층인 다층 퍼셉트론(MLP) 사용.
  (신경망은 대화 상태(현재 대화 단계, 작업의 복잡성, 사용자의 신뢰 수준1-5, 이전 작업의 성공0-40, 마지막 작업 단계의 지속 시간(s))를 입력으로 받고, 각 가능한 행동에 대한 보상 예측값(Q-값)을 출력)
- **훈련 데이터:** DQN은 시뮬레이션된 사용자와의 25,000회의 대화 게임(총 300,000개의 작업 단계)을 통해 훈련. 이 과정에서 상태 공간은 최소-최대 스케일링을 통해 정규화 해서, 학습 속도를 빠르게 유지

RL-based Proactive Agent

Q-learning

- 가치 기반 강화학습 알고리즘 중 하나
- 에이전트가 어떤 상태에서 어떤 행동을 취했을 때, 얻을 수 있는 '가치'를 학습하는 방법
- '가치': 미래에 받을 수 있는 보상의 총합을 예측한 값
- Q-table을 사용하여 모든 상태와 행동 조합에 대한 가치를 저장하고, 업데이트하며 진행
- 에이전트가 Q-table을 참조하여 어떤 행동이 최선인지 결정해서 커지면 관리 여려움

환경 불변 상황 & 비교적 간단한 문제에

Deep-Q-Network

- Q-learning의 발전된 형태
- Q-table 사용 대신 신경망을 사용
- 신경망은 input으로 상태를, output으로 행동에 대한 가치를 제공

복잡한 문제나 환경에 적용 가능

대화형 시스템에서의 RL의 경우 적합

04

Results

Evaluation Method

- 평가 지표: 평균 전반적인 신뢰 평가, 전반적인 과제 성공 점수, 전반적인 과제 소요 시간을 사용
- 전략 간 차이에 대한 유의성 검정은 t-검정을 사용, 본페로니 교정을 적용하여 다중 검정을 고려

Results 1. Trust

- RL-based DA는 3번째로 높은 평균 신뢰 값
- None agent, Rule-based agent와 유의미한 차이 없음

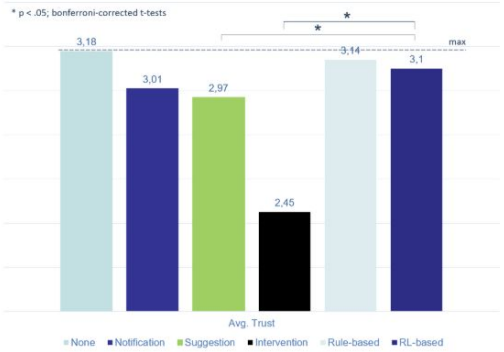
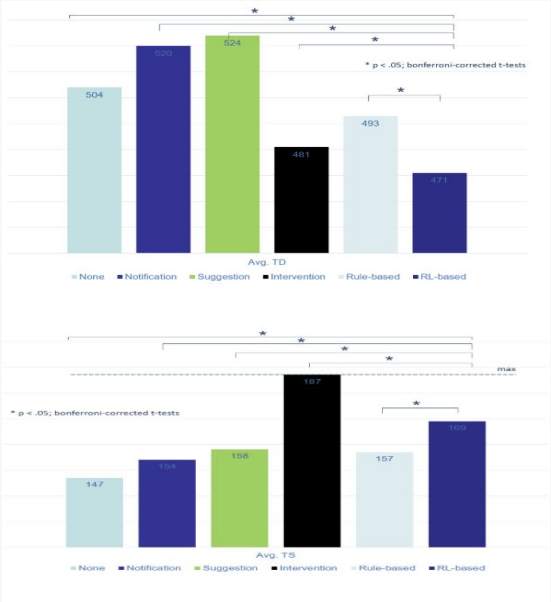


Figure 2: Average trust ratings on a 5-point Likert scale per DA (*max* indicates maximum value among DA).

Results

Results 2. Task Duration, and Task Success

- **Duration**: Intervention agent 다음으로 가장 빠름 (하지만, 유의미한 차이는 없었음. 다만, 평균 작업 시간이 다른 것들에 비해 유의미하게 낮았음)
- **Success**: Intervention agent 다음으로 가장 성공적



행동 방식 4가지:
- None (제안을 요청할 때 까지 기다리기)
- Notification (알림 메세지 무시할 가능성 주기)
- Suggestion (예/아니오 답변을 기대)
- Intervention (사용자의 선택권 X, 자율적으로 에이전트가 옵션을 선택)

Results

Results 3. RL-based 전략에 따른 Agent 행동

신뢰 값과 행동 선택

- 신뢰==1: Notification
- 신뢰==2: Notification 64%
- 신뢰==3: Notification 41%
- 신뢰==4: Intervention 37%

적극

- 신뢰==5: None 49% (적극성 최소화)

게임 점수와 행동 선택

- 점수==0: Notification 85%
- 점수==10: Notification 57%, Intervention 27%

적극

- 점수 ==20: Intervention 44%
- 점수 ==30: None 89%
- 점수 ==40: None (적극성 최소화)

- *낮은 신뢰 값, 낮은 점수는 더 많은 알림 유발*
- *신뢰 값이 높아지면 적극적 행동 감소*
- *점수가 높아지면 시스템은 개입을 줄임*

05

Discussion

Discussion

Trust and Efficiency Balancing

- Agent가 Reactive할 수록 Trust 증가
- Agent가 Proactive할 수록 Task efficiency가 증가

→ Agent는 사용자의 과제 성공률을 향상시키는 선을 찾아야 하지만, 사용자의 신뢰를 잃어서 시스템의 사용을 중단시킬 수도 있음

→ RL-based가 효과적이라는 것을 알 수 있음

Table 2: Task efficiency, trust, and cooperation scores per dialog policy.

Policy	*Task Efficiency*	*Trust*	*Cooperation*
RL-based	0.359	3.10	**1.14**
None	0.292	**3.18**	0.96
Notification	0.296	3.01	0.91
Suggestion	0.302	2.97	0.92
Intervention	**0.389**	2.46	0.97
Rule-based	0.318	3.14	1.05

$$\text{Task efficiency} = \frac{Success}{Duration}$$

$$\text{Cooperation} = \text{Task efficiency} \times Trust$$

행동 방식 4가지:
- None (제안을 요청할 때 까지 기다리기)
- Notification (알림 메세지 무시할 가능성 주기)
- Suggestion (예/아니오 답변을 기대)
- Intervention (사용자의 선택권 X, 자율적으로 에이전트가 옵션을 선택)

Some recommendations

신뢰도에 기반한 행동 전략

- **신뢰도가 낮거나 중간일 때는** ‘Notification(알림 보내고 무시 가능하게 하기)’를 고려하기
- **신뢰도가 적당히 높을 때에는** ‘Suggestion(예/아니오로 답하게 하기)’를 고려하기
- **신뢰도가 너무 높을 때는**, 단순히 Reactive하게 남아있기

점수에 기반한 행동 전략

- **User의 실패 경험에서는** 중간 수준의 개입을 고려하기
- **User의 점수가 낮고, 신뢰도가 높을 때는** 매우 적극적 행동 고려하기
- **User의 성공 행동을 감지하면**, 단순히 Reactive하게 남아있기

06

Limitation

Limitation

시뮬레이션 데이터의 사용

실제 데이터가 아닌, 시뮬레이션 데이터의 사용이라 추가 검증이 필요

협력 가능한 AI를 위한 추가적인 고려 요소 존재

신뢰도 뿐만 아니라, 사용자 참여도 측정, 대화 성실성, 커뮤니케이션 능력 측정 등을 포함해야 함

다양한 접근 방식 탐구

자연어를 기반으로 한 대화 시스템에서의 협력 효율성을 위한 RL에 대한 연구 필요

Conclusions

사회적 & 작업 효과적
상호작용을 위한 RL 대화
Agent 개발

작업 효율성에 기여하는
최상의 절충안을 달성, 신뢰도
유지에 적합함을 증명

Thanks!

Do you have any questions?

Paper DOI: https://dl.acm.org/doi/pdf/10.1145/3565472.3595611