

## RecSys 2021 Tutorial on

# Conversational Recommendation: Formulation, Methods, and Evaluation

Wenqiang Lei

National University of  
Singapore (NUS)

[wenqanglei@gmail.com](mailto:wenqanglei@gmail.com)

Chongming Gao

University of Science and  
Technology of China (USTC)

[chongminggao@mail.ustc.edu.cn](mailto:chongminggao@mail.ustc.edu.cn)

Maarten de Rijke

University of Amsterdam

[m.derijke@uva.nl](mailto:m.derijke@uva.nl)



# Who Are We?



**Wenqiang Lei**  
Postdoc at National  
University of Singapore (NUS)



**Chongming Gao**  
PhD student at University of Science  
and Technology of China (USTC)



**Maarten de Rijke**  
University Professor at  
University of Amsterdam



# Outline

- I. Introduction
- II. Five important challenges
- III. Promising future directions



# Outline

## I. Introduction

- **Background and definition of CRSs**
- **Difference with related topics**
- **The importance of CRS**
- **Introduction of our survey**
- **A glance of the five important challenges**

## II. Five important challenges

## III. Promising future directions

# 1.1 Background: Begin with Information Seeking

## Information explosion problem

- E-commerce (Amazon and Alibaba)
- Social networking (Facebook and Wechat)
- Content sharing platforms (Instagram and Pinterest)



Two major types of information seeking techniques



Search



Recommendation

How to handle?

Information overload

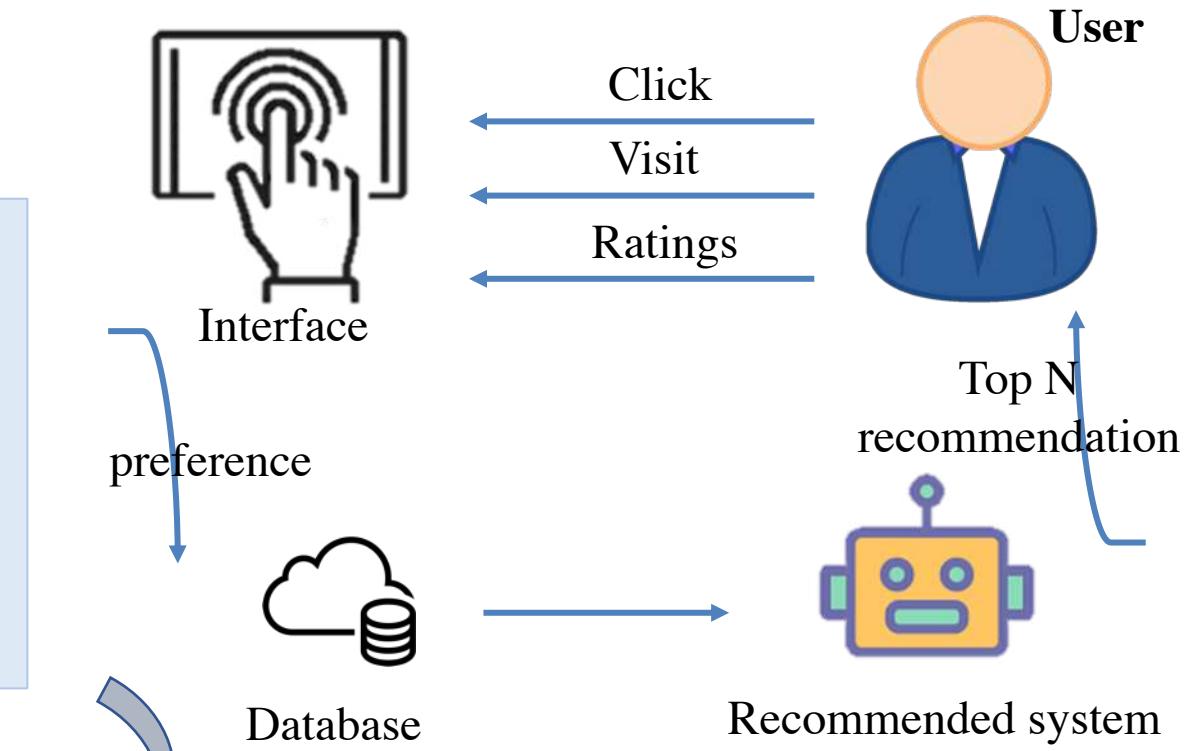


# 1.1 Background: Begin with Information Seeking

## Recommender systems

- predict a user's **preference** towards an item by analyzing their **past behavior**  
(e.g., click history, visit log, ratings on items, etc)

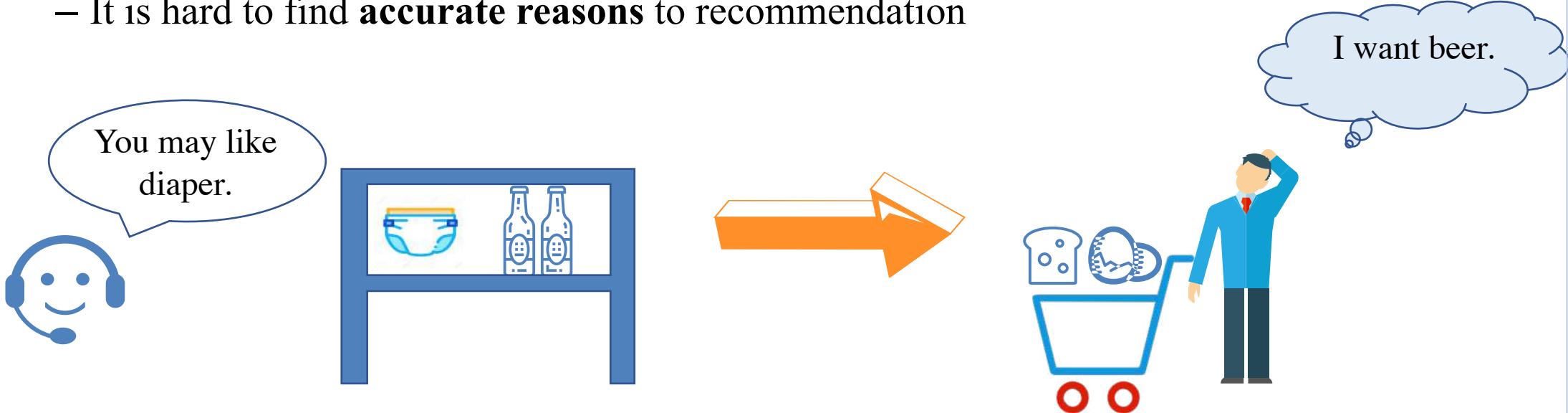
Implicit



# 1.1 Background: Begin with Information Seeking

## Key Problems for Recommendation: Information Asymmetry

- Information asymmetry
  - A system can only **estimate** users' preferences based on their historical data
- Intrinsic limitation
  - Users' preferences often **drift** over times.
  - It is hard to find **accurate reasons** to recommendation



## 1.2 Definition of CRS

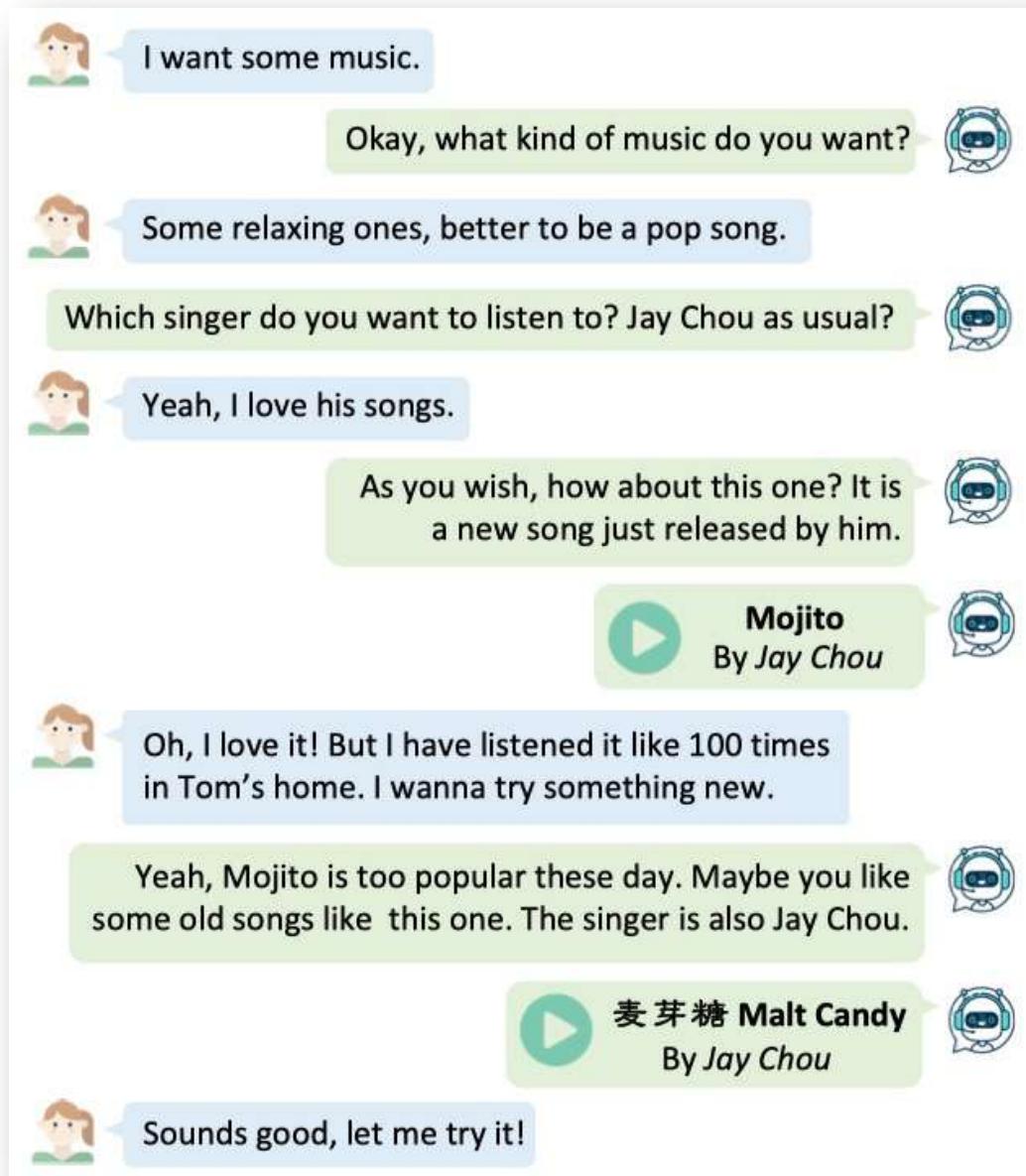
***“A recommendation system that can elicit the dynamic preferences of users and take actions based on their current needs through real-time multturn interactions.”***

Including: written or spoken natural language,  
form fields, buttons, and even gestures.

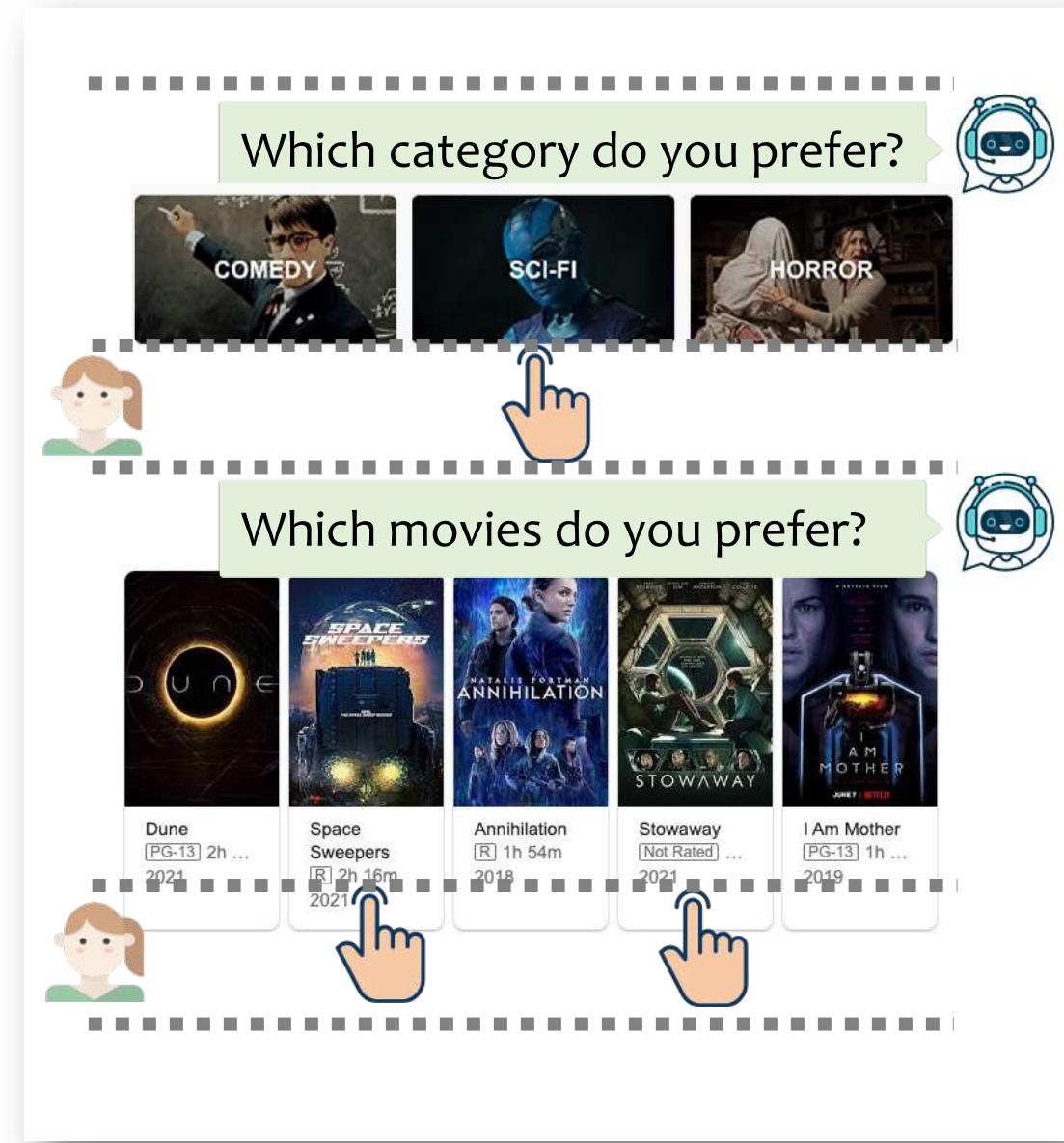
- We don't constrain the form of interaction as long as it can break the information asymmetry

# 1.2 Definition of CRS: toy examples

- CRS based on dialogue interaction



- CRS based on button-clicking interaction

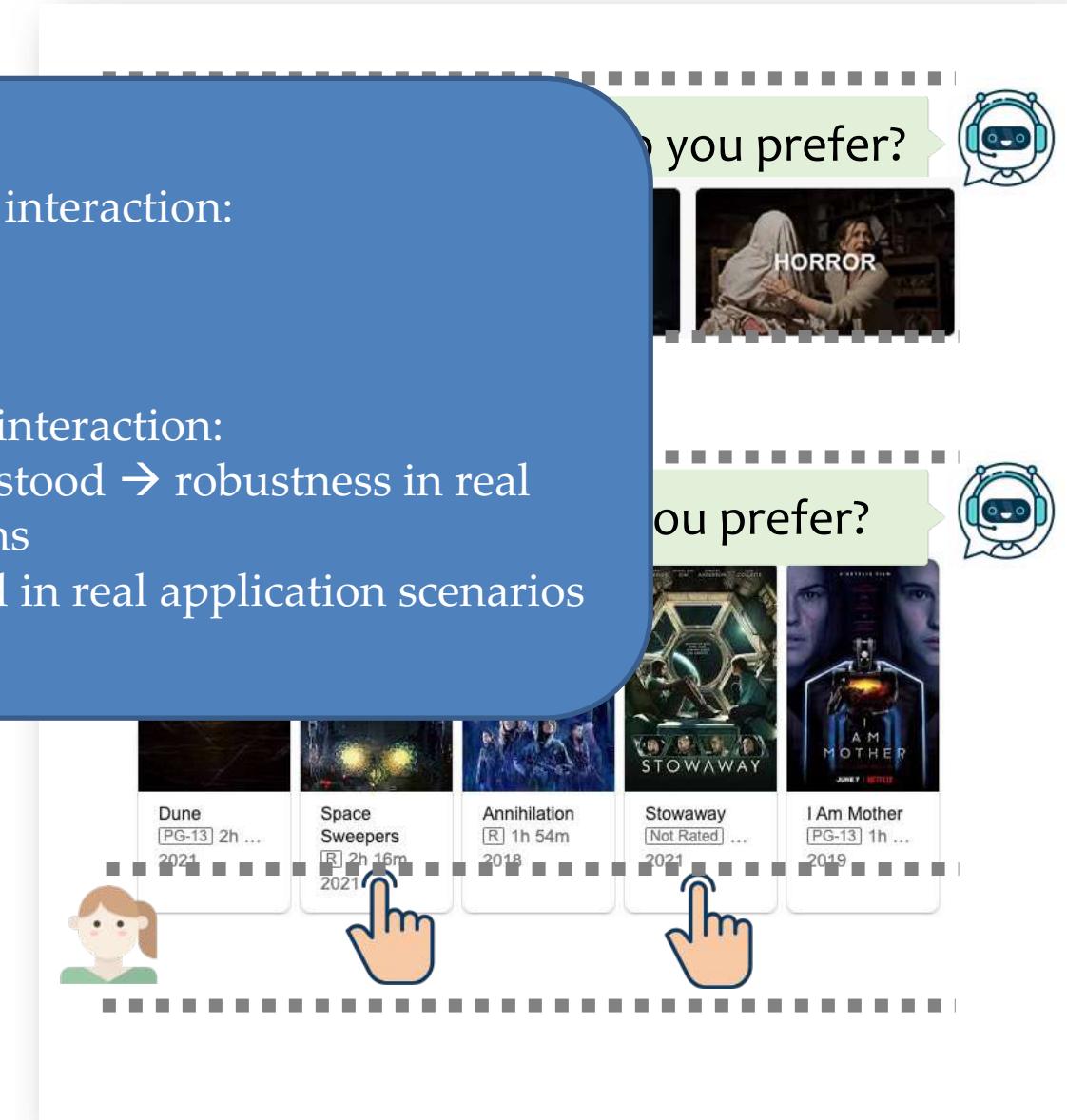


# 1.2 Definition of CRS: toy examples

- CRS based on dialogue interaction



- CRS based on button-clicking interaction



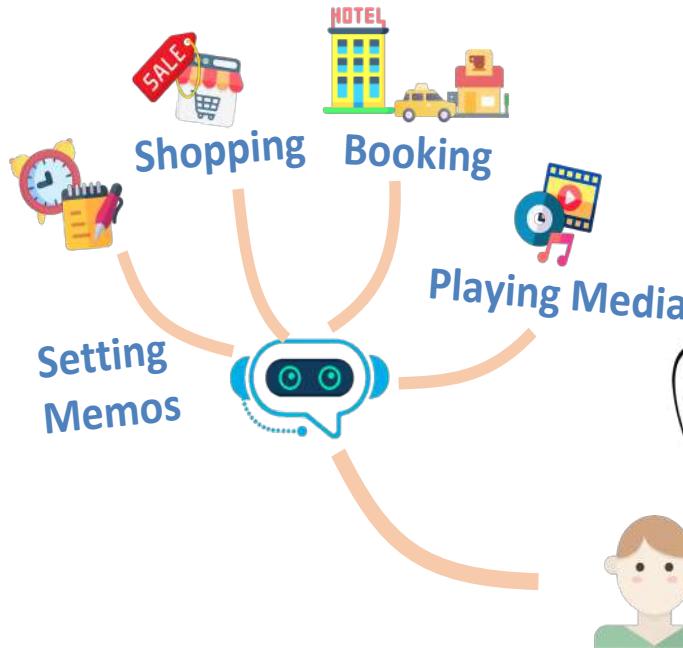
## 1.2 Differences with related topics

### Interactive recommender systems (IRSSs) and CRSs

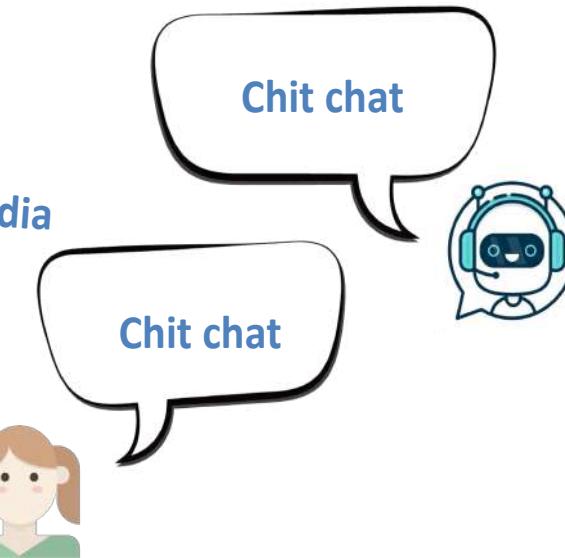
- IRSSs can be seen as an **early form** of CRSs
- IRSSs work by repeating the following two procedure, which is **rigid, inflexible, and inefficient**: 
  1. Making a list of recommendations.
  2. Collecting user feedback, and adjust strategies. Jump to 1.
- CRSs introduce **miscellaneous** types of interaction 
  - They elicit user preferences by asking questions about attributes, which is **more efficient**
  - They only make recommendations when the confidence is high, which improves **user experience**

# 1.2 Differences with related topics

## Task-oriented Dialogue Systems and CRSSs



(a) Task-oriented dialogue systems



(b) Chatbots

**Figure:** Two types of dialogue systems

### □ Problems in traditional dialogue systems:

- Focusing only on **natural language processing**
- Failure to **optimize recommendation strategy**  
Does not **consider click feedback** (Jannach et al.)



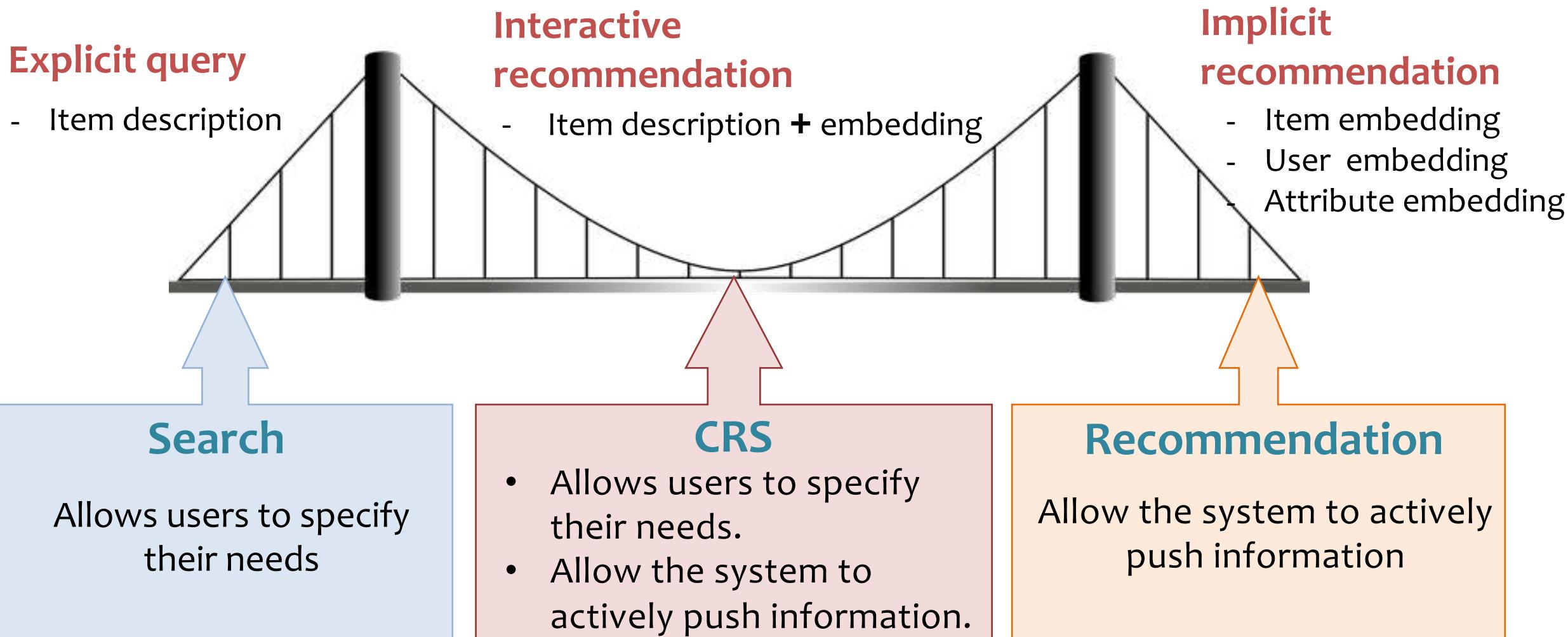
### □ Main focus of CRSSs:



- Aim to elicit **accurate user preferences**, and generate **high-quality recommendations**
- Language understanding not the first priority

## 1.3 Importance of CRSs

- Conversational Recommender Systems (CRSs) can **bridge the gap** between search engines and recommender systems



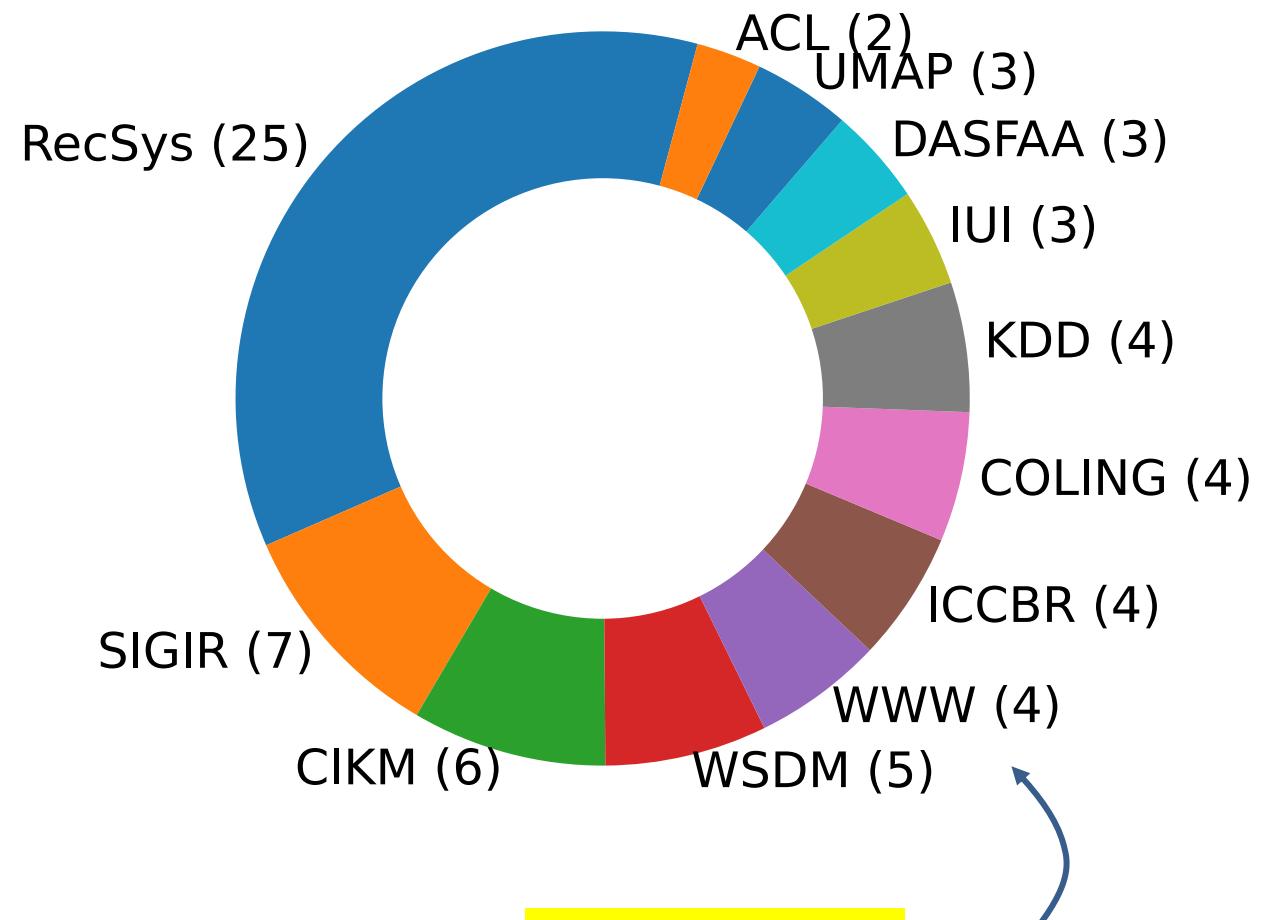
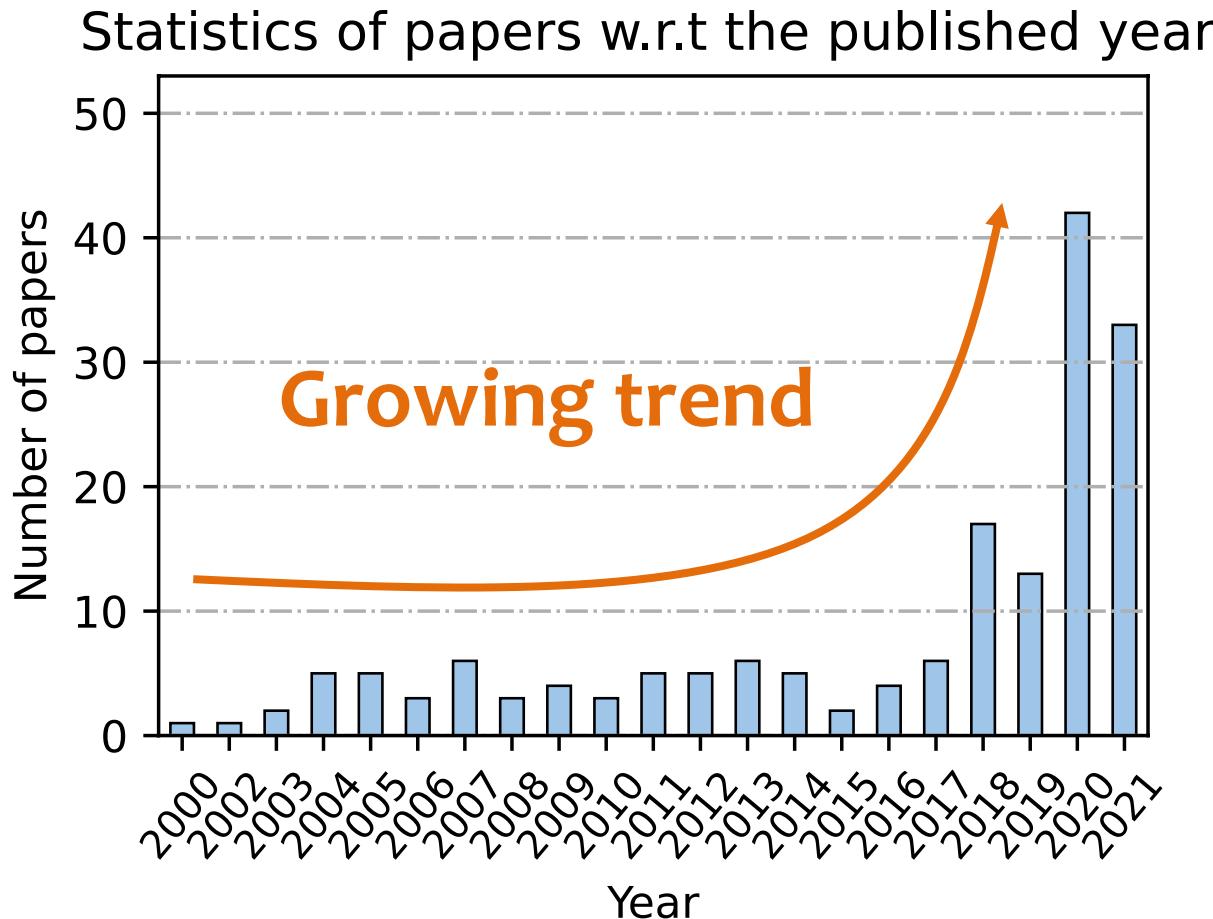
## 1.3 Importance of CRSs

### Conversational Recommender Systems are

- A promising direction for recommendation systems: solving information asymmetry and dynamic preference problem
- An opportunity to converge cutting-edge techniques to push the development of recommendation: reinforcement learning, natural language processing, explainable AI, conversational AI etc.
- An exemplary step towards the big goal of human-machine collaboration

## 1.3 Importance of CRSs

Searching results of “Conversation\* Recommend\*” on DBLP.



There are 171 unique publications, and we only visualize the top 12 venues in the circle chart, which contain 70 papers out of all 171 papers at all 102 venues.

# 1.4 Introduction of Our Survey

Accepted by AI Open in June 2021. Link: <https://arxiv.org/abs/2101.09459>

## Advances and Challenges in Conversational Recommender Systems: A Survey

Chongming Gao<sup>a</sup>, Wenqiang Lei<sup>b,\*</sup>, Xiangnan He<sup>a</sup>, Maarten de Rijke<sup>c,d</sup> and Tat-Seng Chua<sup>b</sup>

<sup>a</sup>*University of Science and Technology of China*

<sup>b</sup>*National University of Singapore*

<sup>c</sup>*University of Amsterdam, Amsterdam, The Netherlands*

<sup>d</sup>*Ahold Delhaize, Zaandam, The Netherlands*

---

### ARTICLE INFO

**Keywords:**

conversational recommendation system  
interactive recommendation  
preference elicitation  
multi-turn conversation strategy  
exploration-exploitation

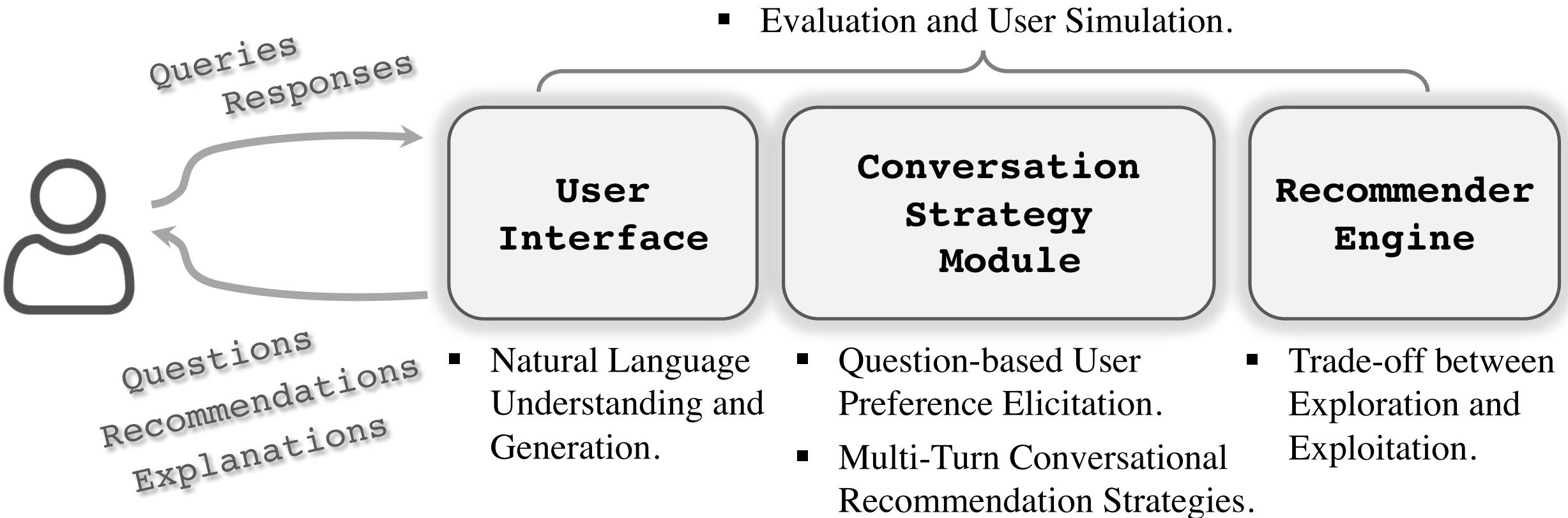
---

### ABSTRACT

Recommender systems exploit interaction history to estimate user preference, having been heavily used in a wide range of industry applications. However, static recommendation models are difficult to answer two important questions well due to inherent shortcomings: (a) What exactly does a user like? (b) Why does a user like an item? The shortcomings are due to the way that static models learn user preference, i.e., without explicit instructions and active feedback from users. The recent rise of conversational recommender systems (CRSs) changes this situation fundamentally. In a CRS, users and the system can dynamically communicate through natural language interactions, which provide unprecedented opportunities to explicitly obtain the exact preference of users.

Considerable efforts, spread across disparate settings and applications, have been put into developing CRSs. Existing models, technologies, and evaluation methods for CRSs are far from mature. In

## 1.4 Introduction of Our Survey



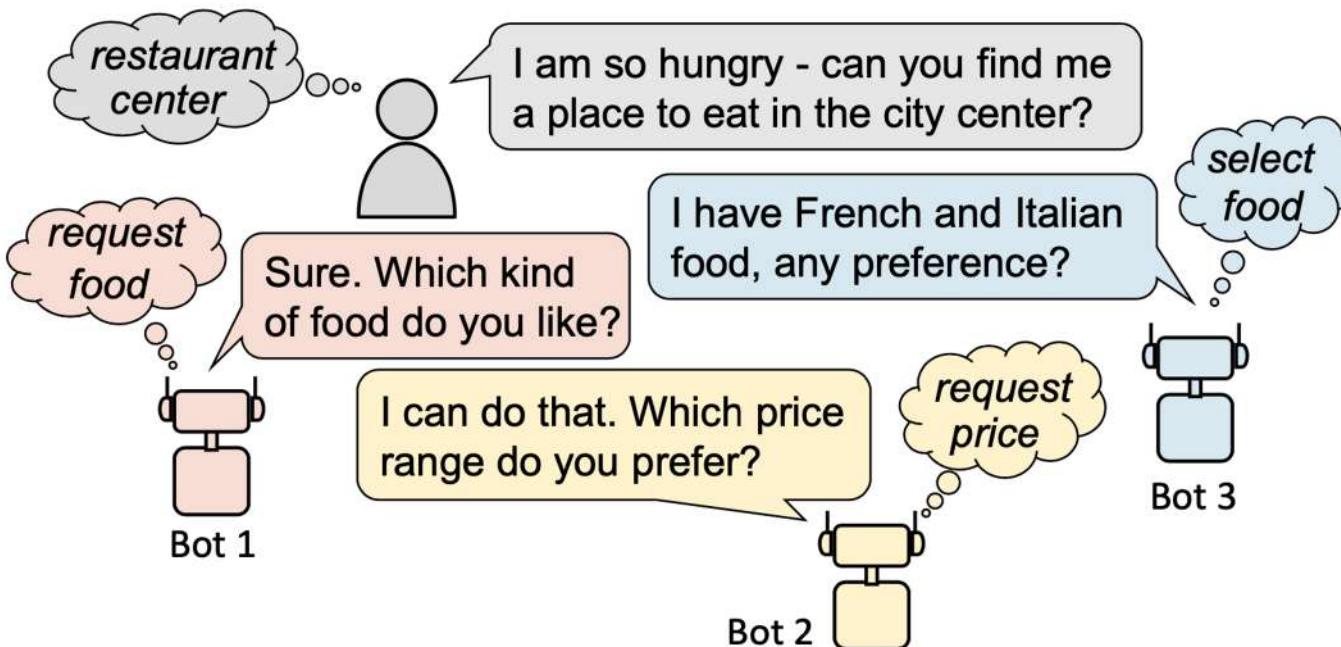
**Figure:** Illustration of the general framework of CRSs and our identified **primary challenges** on the three main components.

# 1.5. Five Important Challenges: A Glance

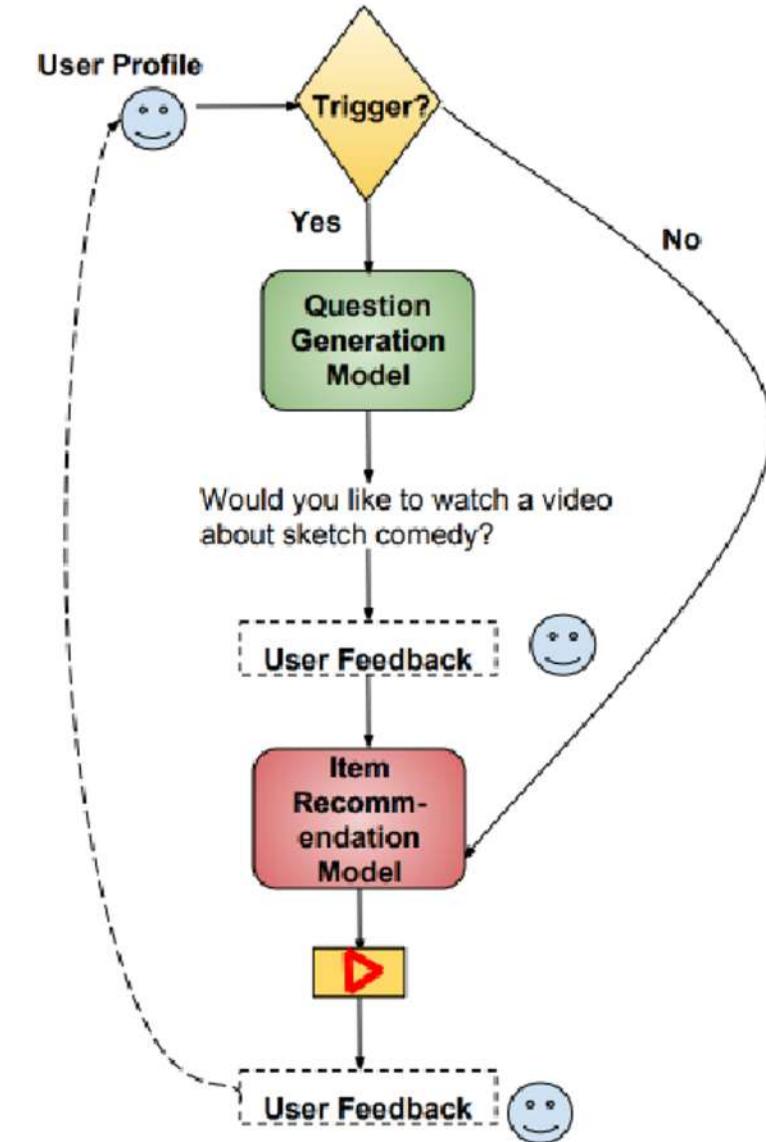
## □ Question-based user preference elicitation

The key advantage of conversational recommendation:  
being able to ask questions

- Ask about **attributes/topics/categories** of items to narrow down the recommended candidates



Zhang et al. Task-Oriented Dialog Systems that Consider Multiple Appropriate Responses under the Same Context (AAAI' 20)



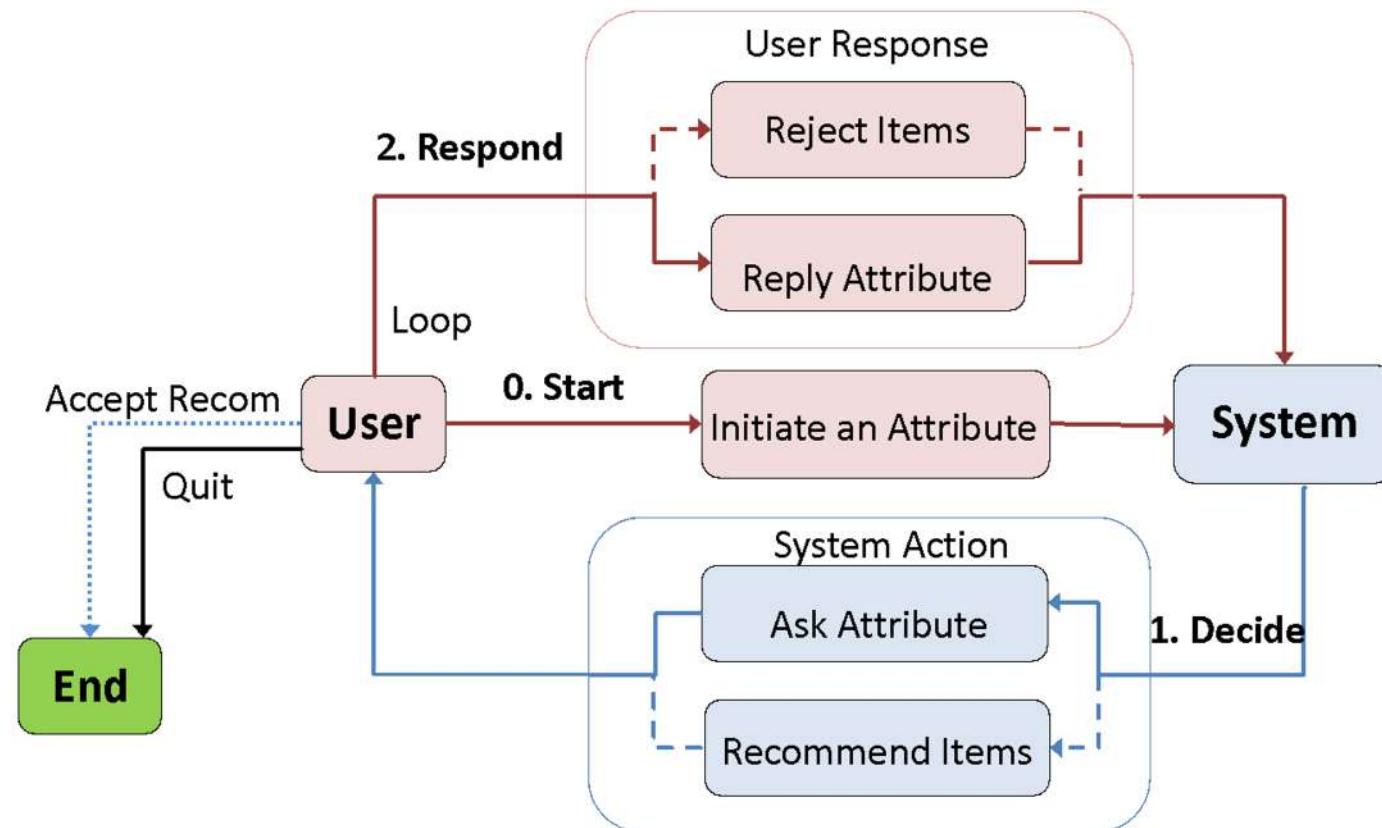
Christakopoulou et al. "Q&R: A Two-Stage Approach toward Interactive Recommendation"(KDD' 18)

# 1.5. Five Important Challenges: A Glance

## □ Multi-turn conversational recommendation strategies

A system can choose to **ask questions** and **make recommendations** in a multi-turn conversation

- **Purpose:** making successful recommendations with less turns of interactions
- **Core challenges to address:**
  1. Which items or attributes to recommend?
  2. When to ask questions and when to make recommendations?
  3. How to adapt user feedback



*Lei et al. "Estimation–Action–Reflection: Towards Deep Interaction Between Conversational and Recommender Systems" (WSDM'20)*

# 1.5. Five Important Challenges: A Glance

## □ Natural language understanding and generation



Inflexible ,  
constrained

Fail to understand  
user intent.

### Rule/Template-based

I want some music.

What category of music do you like?

Pop.

Which Pop singer do you like?

Jay Chou.

Hope you enjoy this song:



七里香 Qi-Li-Xiang

By Jay Chou

Change it.

Hope you enjoy this song:



Change it

By Stevie Ray Vaughan

### Neural methods

I want some music.

Feel tired in work? What do you want?

Yeah, wanna some relaxed music

As you wish, how about this one?  
It is a new song just released by Jay Chou.



Mojito  
By Jay Chou

Oh, I love it! But I have listened it like 100 times. I wanna try something new.

Yeah, Mojito is too popular these day.  
Maybe you like some niche songs like this one. The singer is also Jay Chou.



麦芽糖 Malt Candy  
By Jay Chou



Casual, more natural.

Extract intent from user utterances.

Express actions in generated responses

Fluent and Consistent.

## 1.5. Five Important Challenges: A Glance

### □ Trade-offs between exploration and exploitation (E&E)

Exploration  
(Learning)

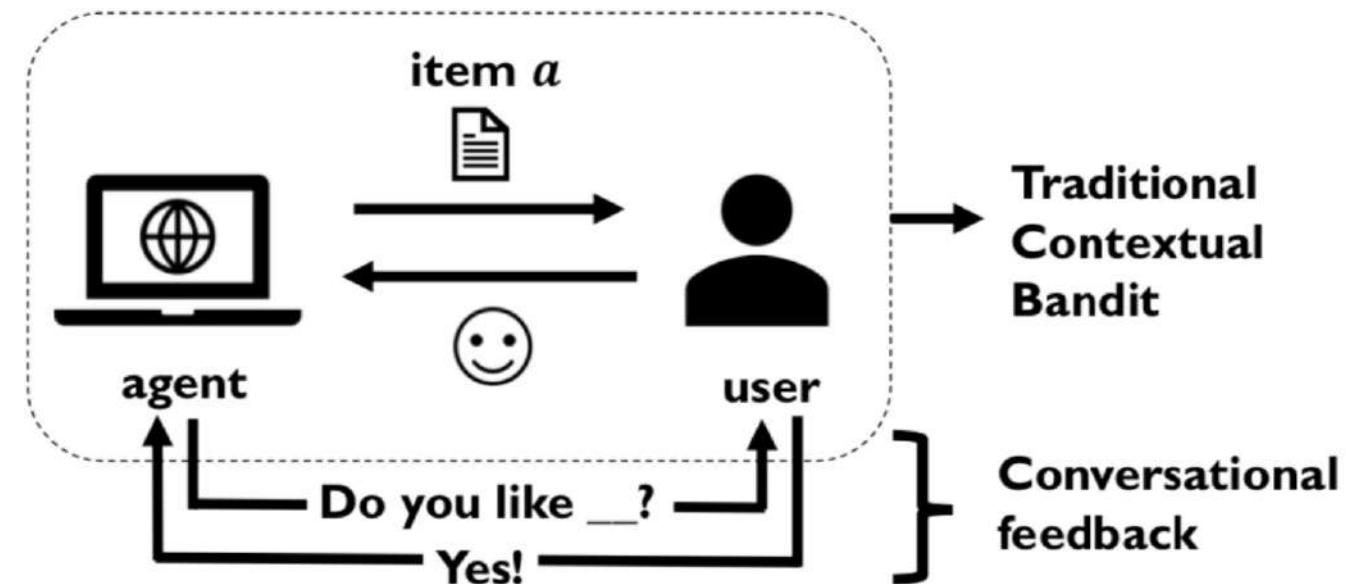
Take some risk to collect information  
about unknown options



Exploitation  
(Earning)

Takes advantage of the best option  
that is known.

Leverage the dynamics of CRS  
to benefit the E&E trade-off for  
cold users/items.



# 1.5. Five Important Challenges: A Glance

## □ Simulation and evaluation

□ How to evaluate CRSs in terms of **turn-level** performance?

- Evaluation of recommendation
- Evaluation of response generation

□ How to evaluate CRSs in terms of **conversation-level** (global) performance?

- Online test (A/B test)  
and Off-policy evaluation
- User simulation

**turn-level evaluation**



**Conversation-level evaluation**



# Outline

## I. Introduction

- **Background and definition of CRSs**
- **Difference with related topics**
- **The importance of CRSs**
- **Introduction of our survey**
- **A glance of the five important challenges**

## II. Five important challenges

## III. Promising future directions



# Outline

## I. Introduction

## II. Five important challenges

### 2.1 Question-based user preference elicitation

### 2.2 Multi-turn conversational recommendation strategies

### 2.3 Natural language understanding and generation

### 2.4 Trade-offs between exploration and exploitation (E&E)

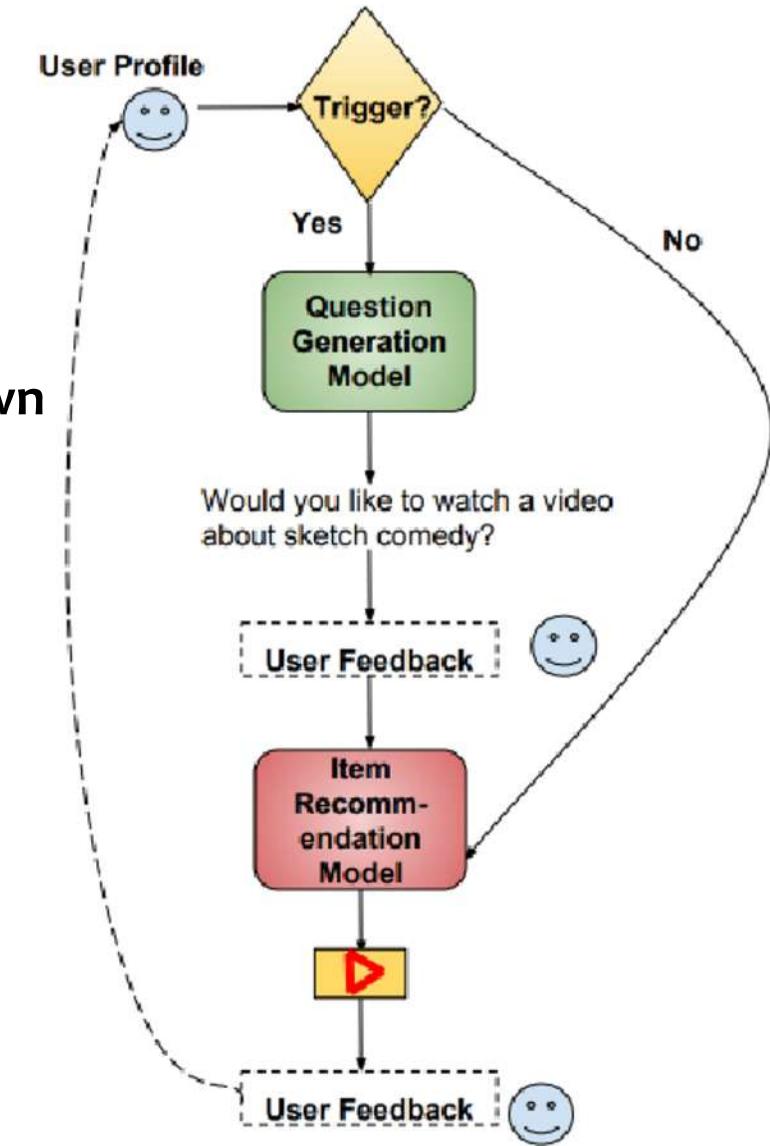
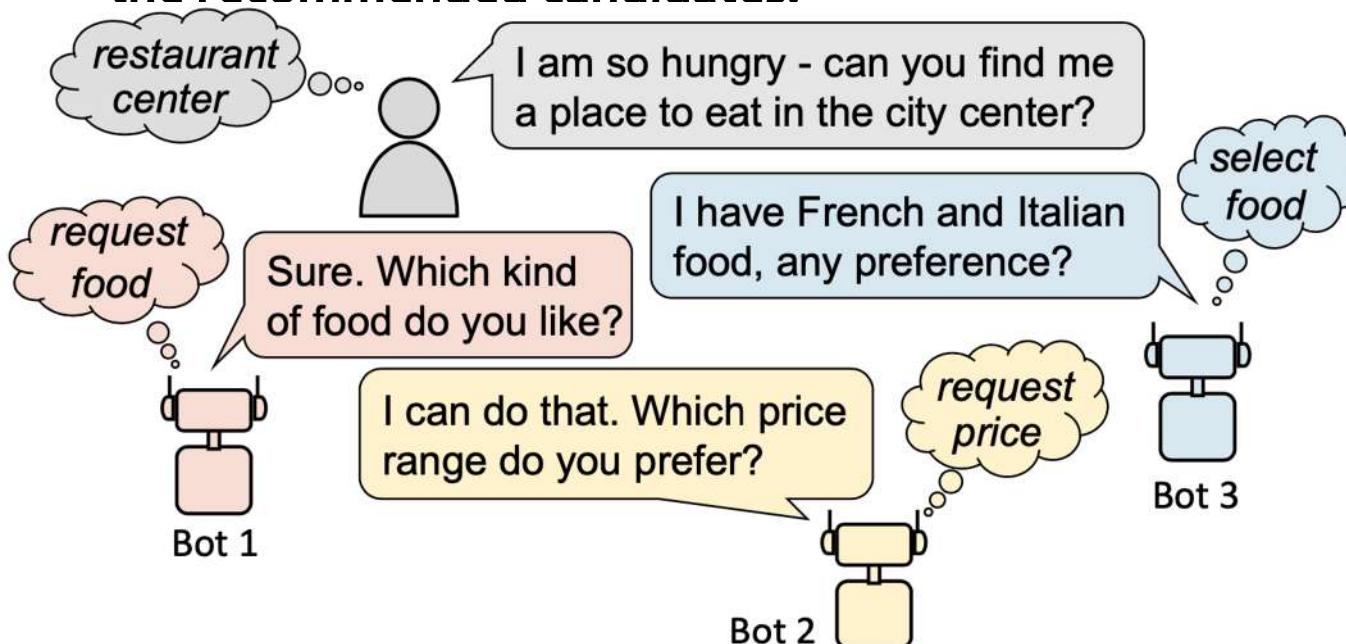
### 2.5 Evaluation and user simulation

## III. Promising future directions

# 2.1 Question-based User Preference Elicitation

The key advantage of conversational recommendation: being able to ask questions

- Ask about **attributes/topics/categories** of items to narrow down the recommended candidates.

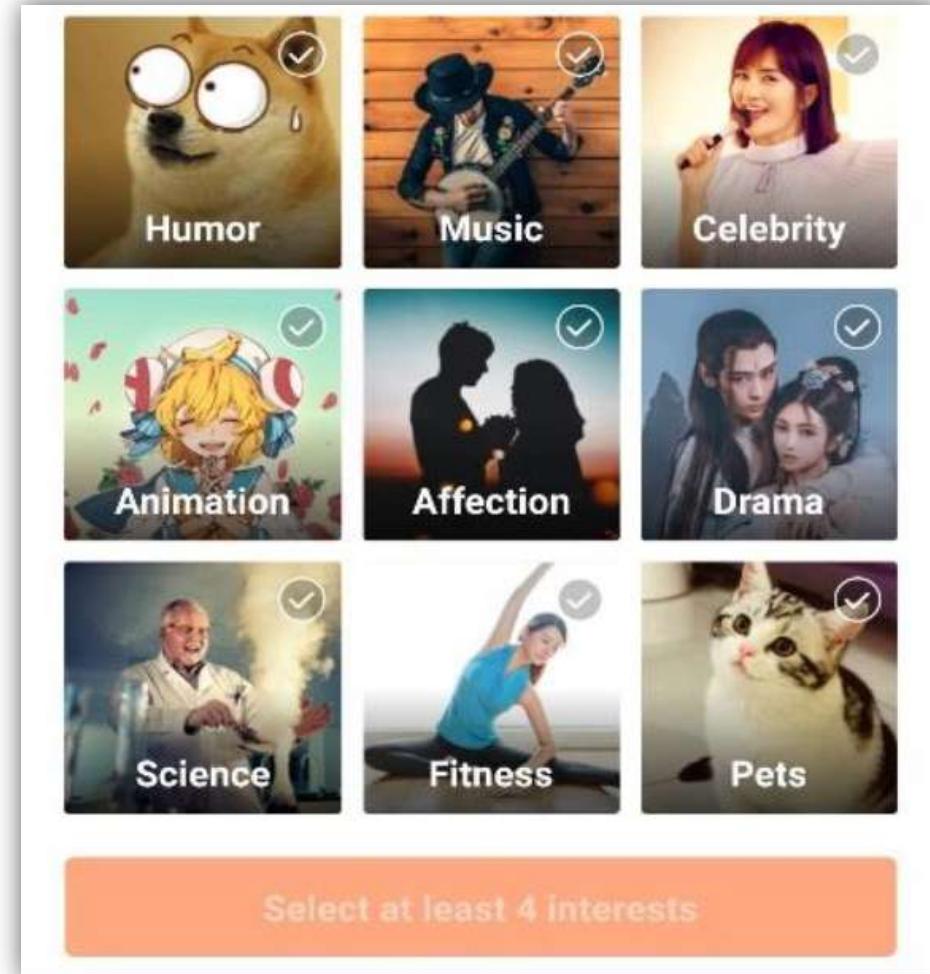


# 2.1 Question-based User Preference Elicitation

## Asking about Items



## Asking about Attributes

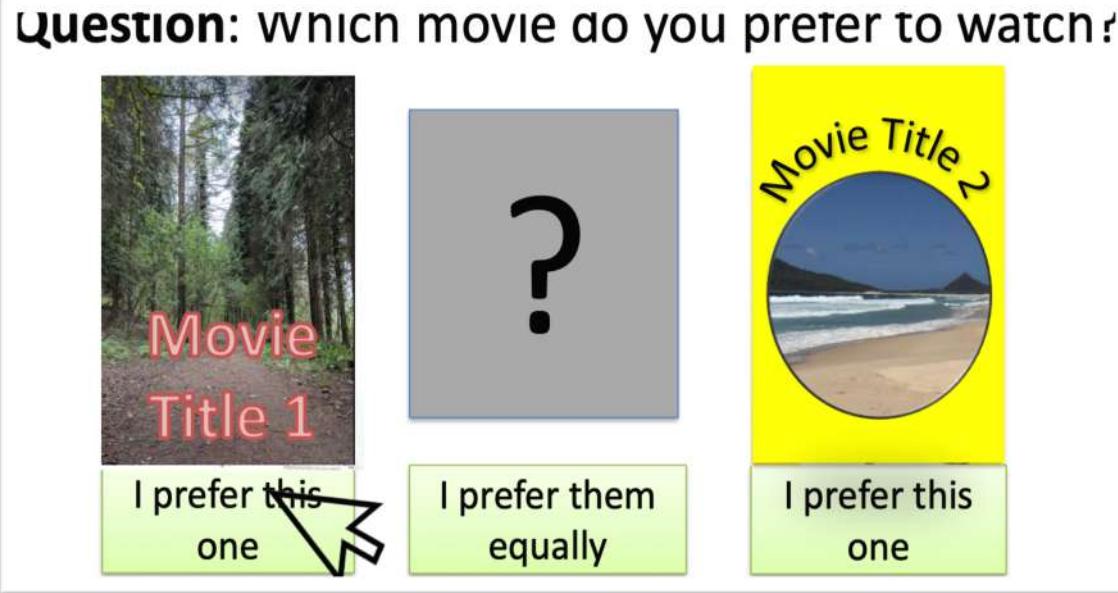


*Figure Credit:* Tong Yu, Yilin Shen, and Hongxia Jin. A Visual Dialog Augmented Interactive Recommender System. KDD' 19

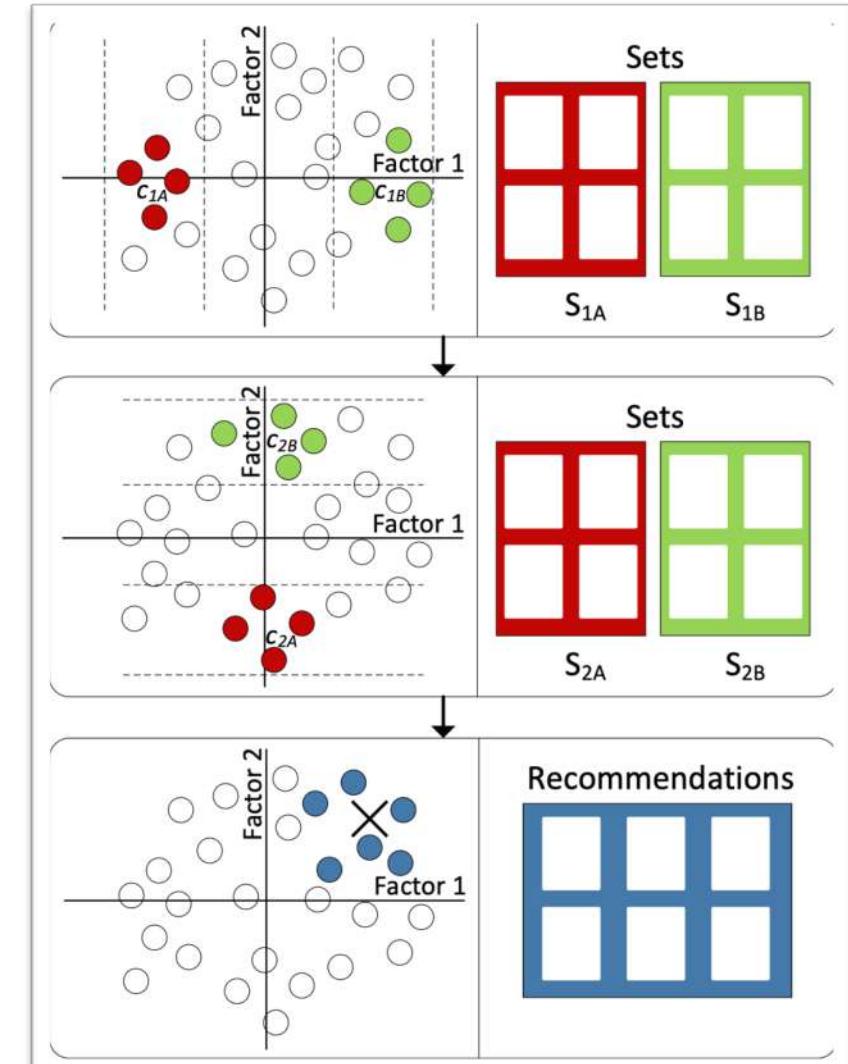
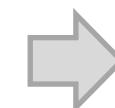
*Figure Credit:* Shijun Li et al. Seamlessly Unifying Attributes and Items Conversational Recommendation for Cold-Start Users. TOIS' 2021.

# 2.1 Question-based User Preference Elicitation

## Asking about Items: (1) Latent factor methods



- Choosing an item from two or more items
- Choosing a set of items from two given lists



## 2.1 Question-based User Preference Elicitation

### Asking about Items: (2) Bayesian preference elicitation

- Preference is represented as a utility function:  $u(x_j, u_i)$

$$u(\mathbf{x}_j, \mathbf{u}_i) = \mathbf{x}_j^T \mathbf{u}_i.$$

- The utility of an item  $j$  for a user  $i$  is computed as the expectation:

$$\mathbb{E}[u(\mathbf{x}_j, \mathbf{u}_i)] = \int_{\mathbf{u}_i \sim \mathcal{U}^{(i)}} P(\mathbf{u}_i) u(\mathbf{x}_j, \mathbf{u}_i) d\mathbf{u}_i.$$

- The item with the maximum expected utility for user  $i$  is considered as the recommendation items:

$$\arg \max_j \mathbb{E}[u(\mathbf{x}_j, \mathbf{u}_i)].$$

## 2.1 Question-based User Preference Elicitation

### Asking about Items: (2) Bayesian preference elicitation

- Based on the utility function, the system can select some items to query.
- The user belief distribution can be updated based on users' feedback. Specifically,

$$P(\mathbf{u}_i | q, r_j) = \frac{P(r_j | q, \mathbf{u}_i) P(\mathbf{u}_i)}{\int_{\mathcal{U}^{(i)}} P(r_j | q, \mathbf{u}_i) P(\mathbf{u}_i) d\mathbf{u}_i}.$$

- There are variations for query strategy, i.e., **selecting which items to ask.**
  - ✓ Single item query.
  - ✓ pairwise comparison query.
  - ✓ Slate query.

(Details can be found in our survey)

## 2.1 Question-based User Preference Elicitation

### Asking about Items: (3) Reinforcement learning

- **Problem:** The Log data is sparse.
- **Solution:** the first attempt to leverage KG for reinforcement learning in interactive recommender systems.

- Use Q-learning to generate items
- Use GCN to represent states

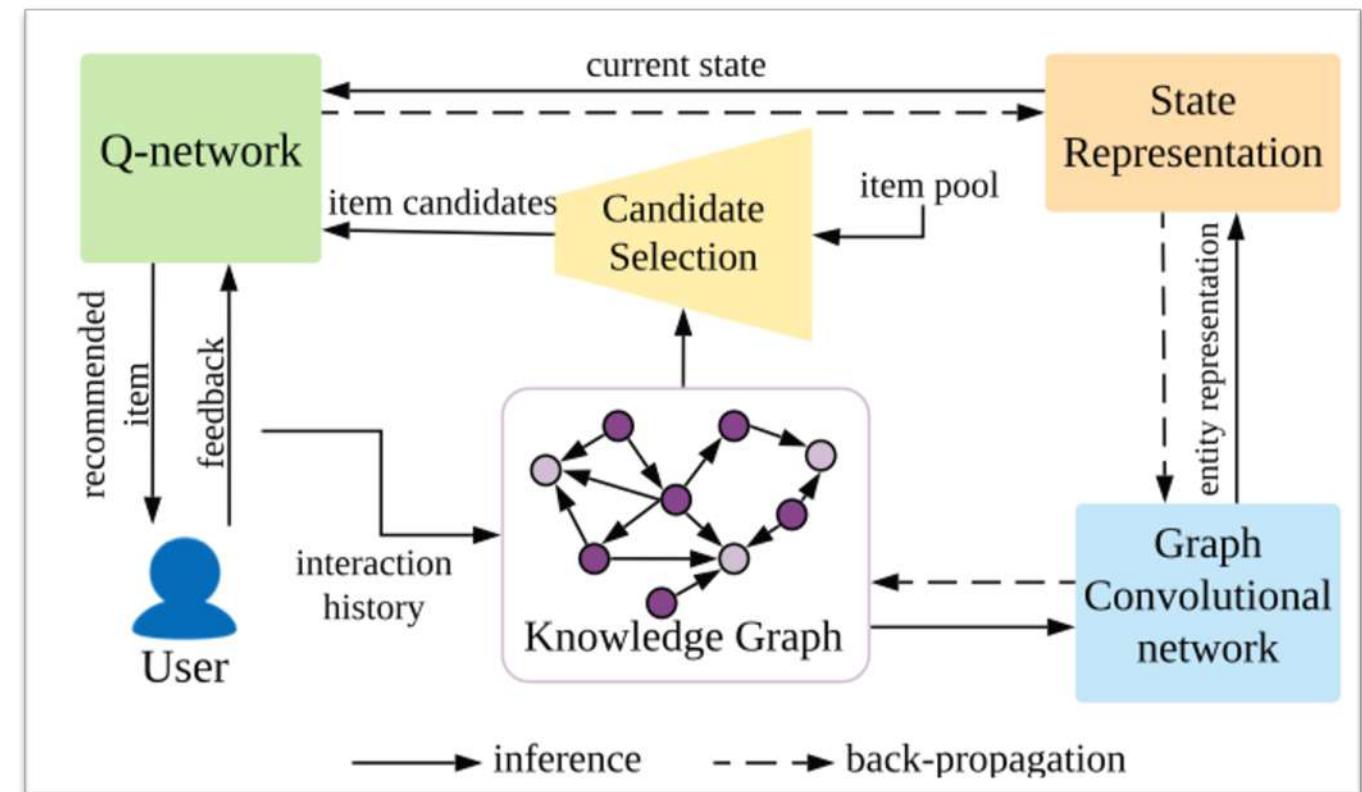


Figure Credit: Sijin Zhou et al. Interactive Recommender System via Knowledge Graph-enhanced Reinforcement Learning. SIGIR' 20

# 2.1 Question-based User Preference Elicitation

## Asking about Attributes: (1) Using sequential model to predict

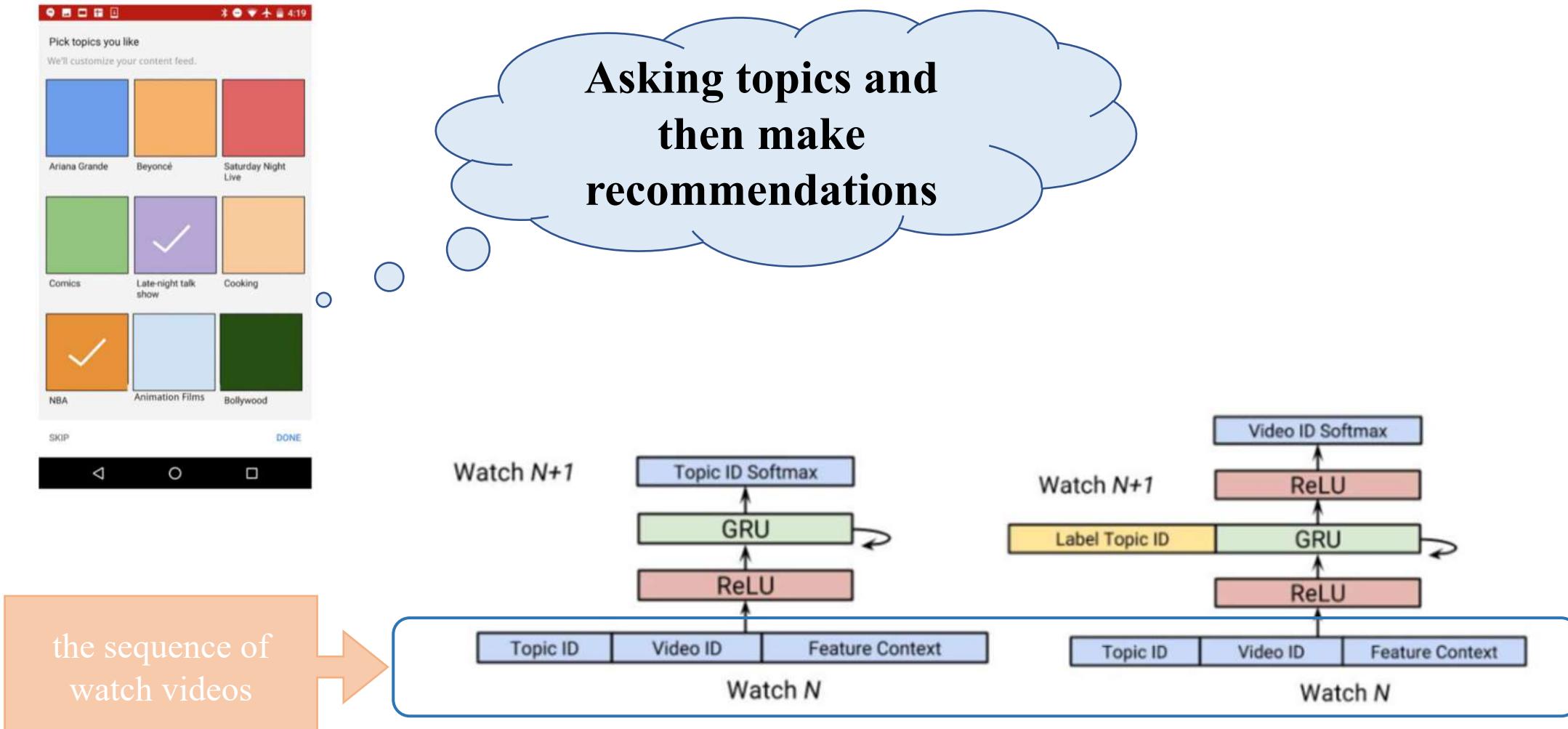


Figure 3: Left: Topic Prediction (Question Ranking) Model. Right: Post-Fusion Approach for Response Model.

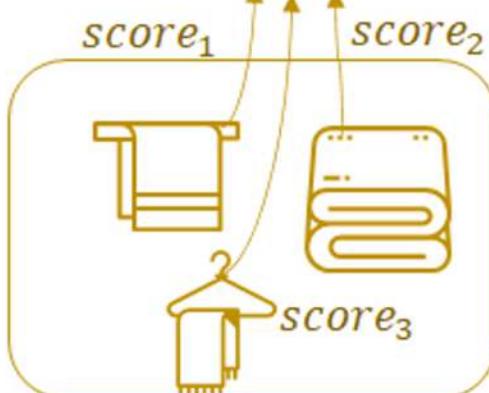
## 2.1 Question-based User Preference Elicitation

### Asking about Attributes: (2) Uncertainty driven

Application scenario: e-commerce

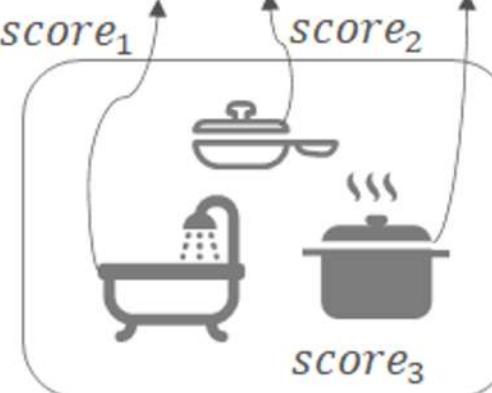
[cotton] preference confidence :  $|S_{\text{like}} - S_{\text{dislike}}|$

like [cotton] score:  $S_{\text{like}}$



[cotton] related  
item set

dislike [cotton] score:  $S_{\text{dislike}}$



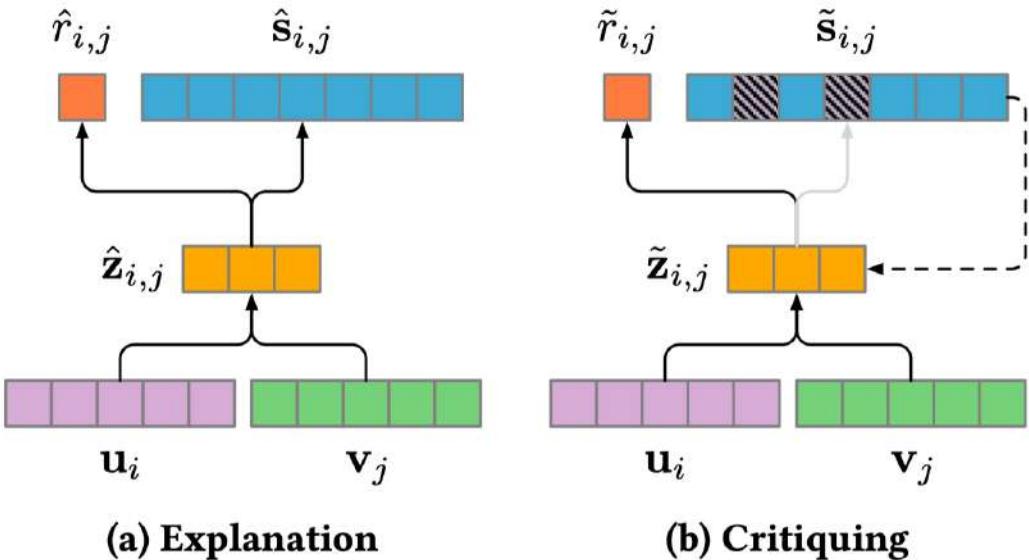
[cotton] unrelated  
item set

The smaller the preference confidence indicate the more uncertain attribute.

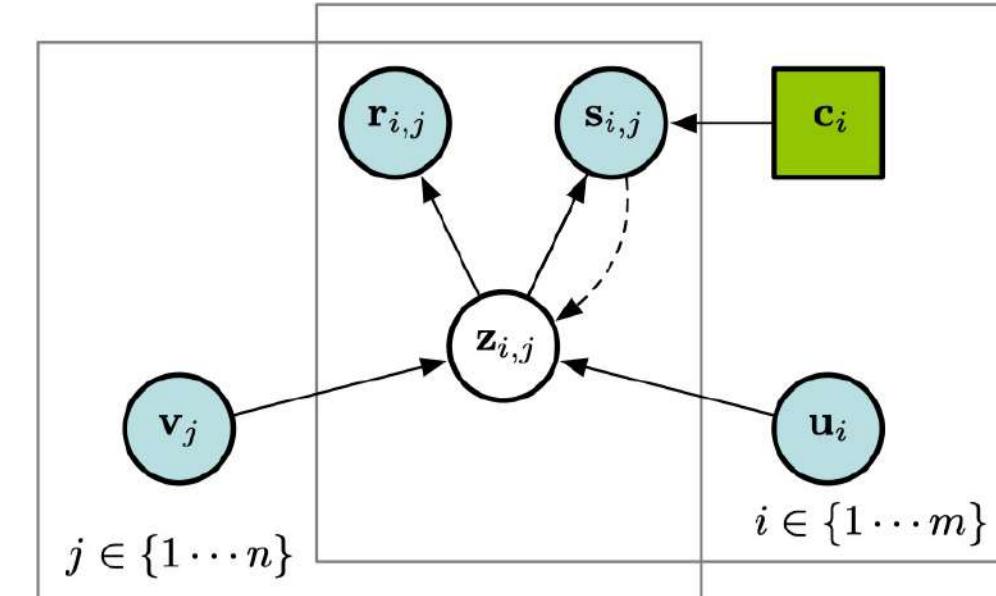
$$score_i \propto (UV^T, Y_i)$$

# 2.1 Question-based User Preference Elicitation

## Asking about Attributes: (3) Explainable recommendation



**Figure 1: Proposed CE-(V)NCF architecture.** (a) Given user  $u_i$  and item  $v_j$  embeddings as input, the network produces a joint embedding  $\hat{z}_{i,j}$  and an initial rating  $\hat{r}_{i,j}$  and explanation  $\hat{s}_{i,j}$  via forward propagation. (b) Shaded squares indicate critiqued keyphrase explanations that modulate the latent space into  $\tilde{z}_{i,j}$  for subsequent recommendations.



**Figure 2: Probabilistic Graphical Model view of the proposed CE-(V)NCF model.** Action node  $c_i$  represents a critiquing action of user  $i$  that modifies the predicted explanation  $s_{i,j}$  into critiqued explanation  $\tilde{s}_{i,j}$ . The dashed arrow denotes posterior inference after critiquing.

# 2.1 Question-based User Preference Elicitation

## Classification w.r.t.

- a) What to ask (item or attribute)
- b) Asking mechanism
- c) Basic model
- d) Type of user feedback
- e) Multi-turn strategy

Asking	Asking Mechanism	Basic Model	Type of User Feedback	Strategy	Publications
Items	Exploitation & Exploration	Multi-Armed bandit	Rating on the given item(s)	No	[217, 32, 220, 184, 205]
	Exploitation & Exploration	Meta learning	Rating on the given item(s)	No	[235, 87]
	Maximal posterior user belief	Bayesian methods	Rating on the given item(s)	No	[171]
	Reducing uncertainty	Choice-based methods	Choosing an item or a set of items	No	[105, 75, 53, 144, 140]
Attributes	Exploitation & Exploration	Multi-Armed bandit	Rating on the given attribute(s)	Yes	[209, 95]
	Reducing uncertainty	Bayesian approach	Providing preferred attribute values	No	[113]
		Critiquing-based methods	Critiquing one/multiple attributes	No	[117, 155, 172, 12, 154]
		Matrix factorization	Answering Yes/No for an attributes	No	[135, 23, 189, 108, 107]
	Fitting historical patterns	Sequential neural network	Providing preferred attribute values	Yes	[31, 210]
			Providing an utterance	No	[94, 25]
			Answering Yes/No for an attributes	Yes	[88, 89]
	Maximal reward	Reinforcement learning	Providing an utterance	Yes	[161, 167, 76]
	Exploring graph-constrained candidates	Graph reasoning	Answering Yes/No for an attributes	Yes	[141]
			Providing an utterance	Yes	[89]
			Providing an utterance	Yes	[25, 104]
			Providing preferred attribute values	Yes	[225, 98]
	Providing preferred attribute values		Providing preferred attribute values	Yes	[193]
			Providing preferred attribute values	No	[123]



# Outline

## I. Introduction

## II. Five important challenges

2.1 Question-based user preference elicitation

2.2 Multi-turn conversational recommendation strategies

2.3 Natural language understanding and generation

2.4 Trade-offs between exploration and exploitation (E&E)

2.5 Evaluation and user simulation

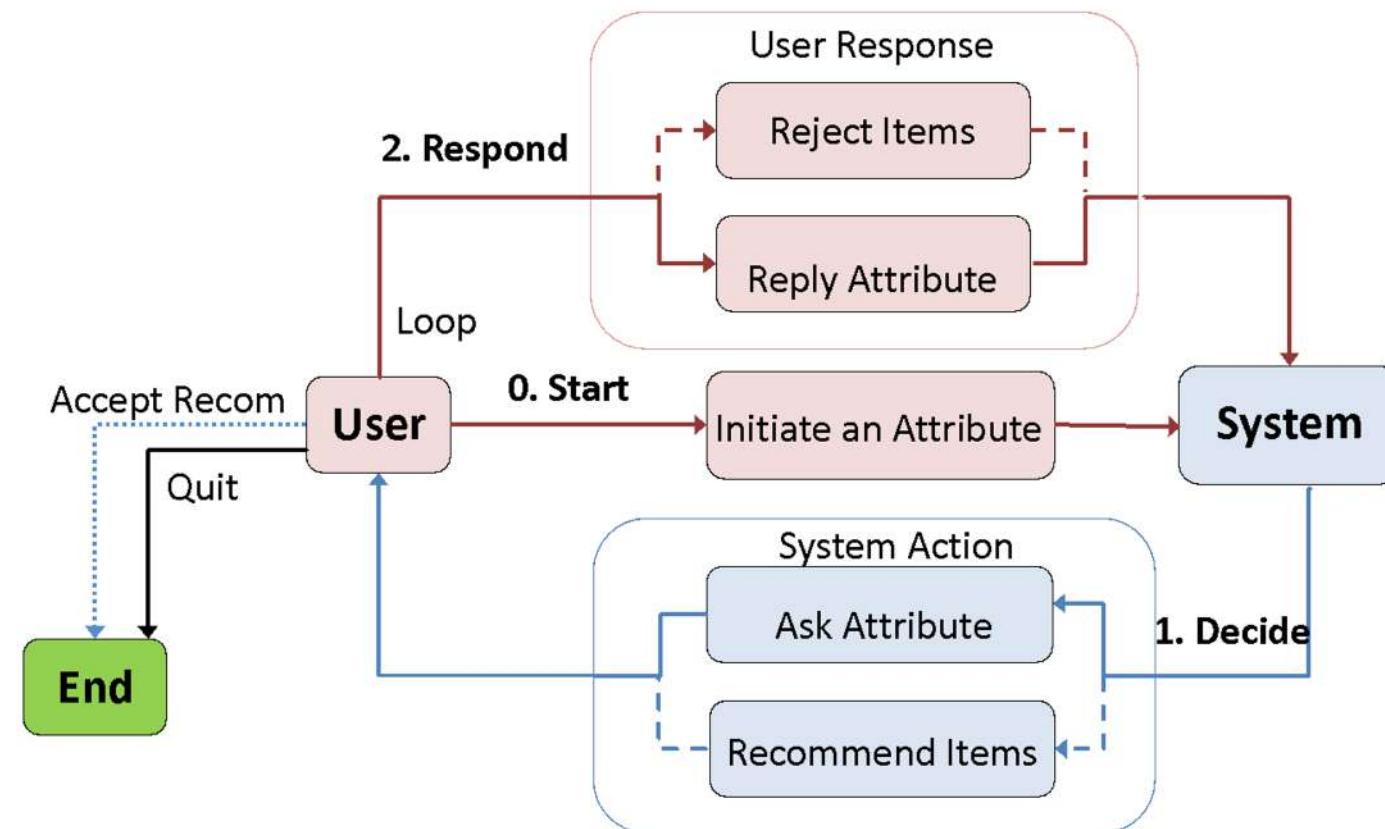
## III. Promising future directions

## 2.2 Multi-turn Conversational Recommendation Strategies

### □ Multi-turn Conversational Recommendation Strategies.

A system can choose to **ask attributes** and **make recommendations** (i.e., ask items) in a multi-turn conversation

- **Purpose:** making successful recommendations with less turns of interactions
- **Core challenges to address:**
  1. Which items to recommend and which attributes to recommend?
  2. When to ask questions and when to make recommendations?
  3. How to adapt user feedback



*Lei et al. "Estimation–Action–Reflection: Towards Deep Interaction Between Conversational and Recommender Systems" (WSDM'20)*

## 2.2 Multi-turn Conversational Recommendation Strategies

- CRM Model

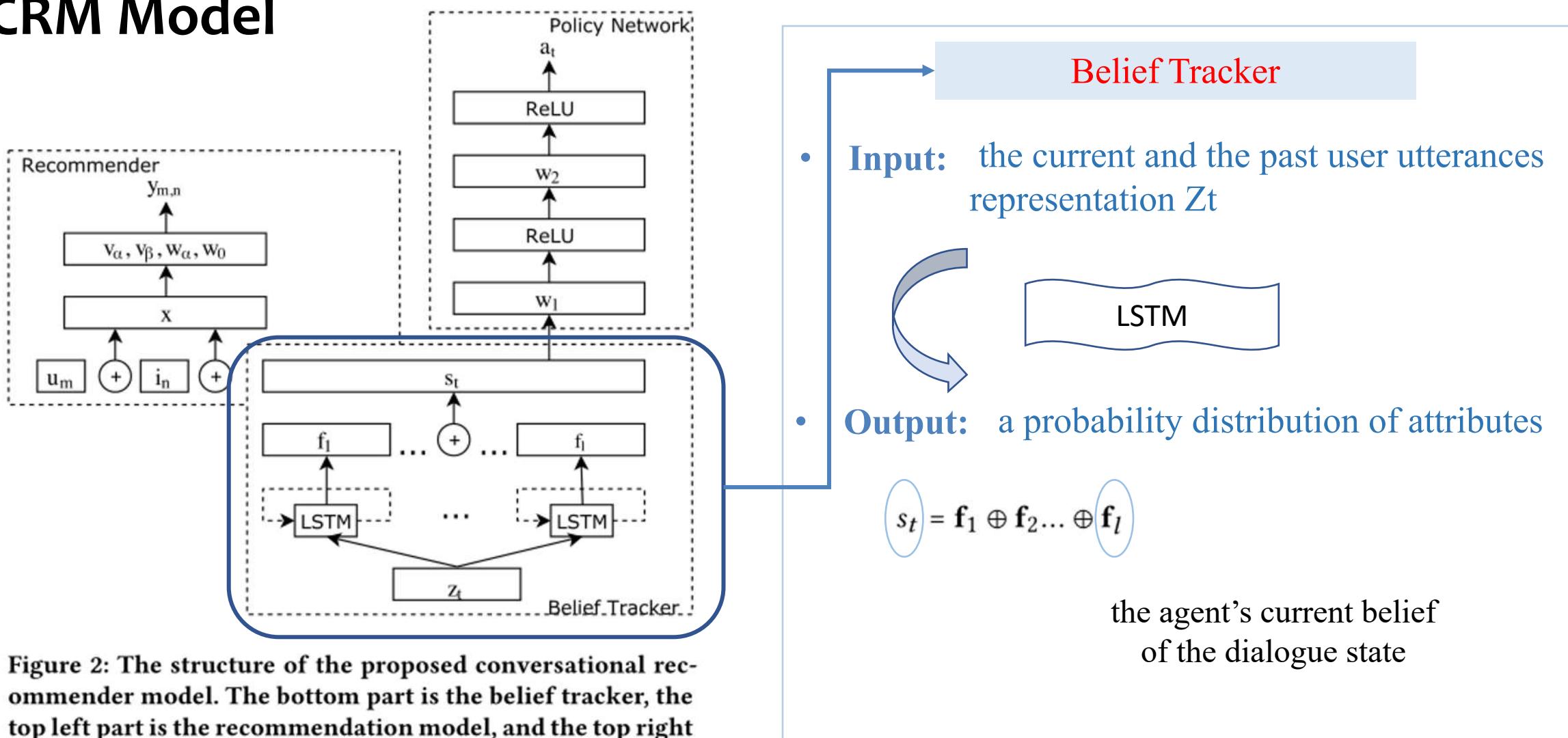


Figure 2: The structure of the proposed conversational recommender model. The bottom part is the belief tracker, the top left part is the recommendation model, and the top right part is the deep policy network.

## 2.2 Multi-turn Conversational Recommendation Strategies

- CRM Model

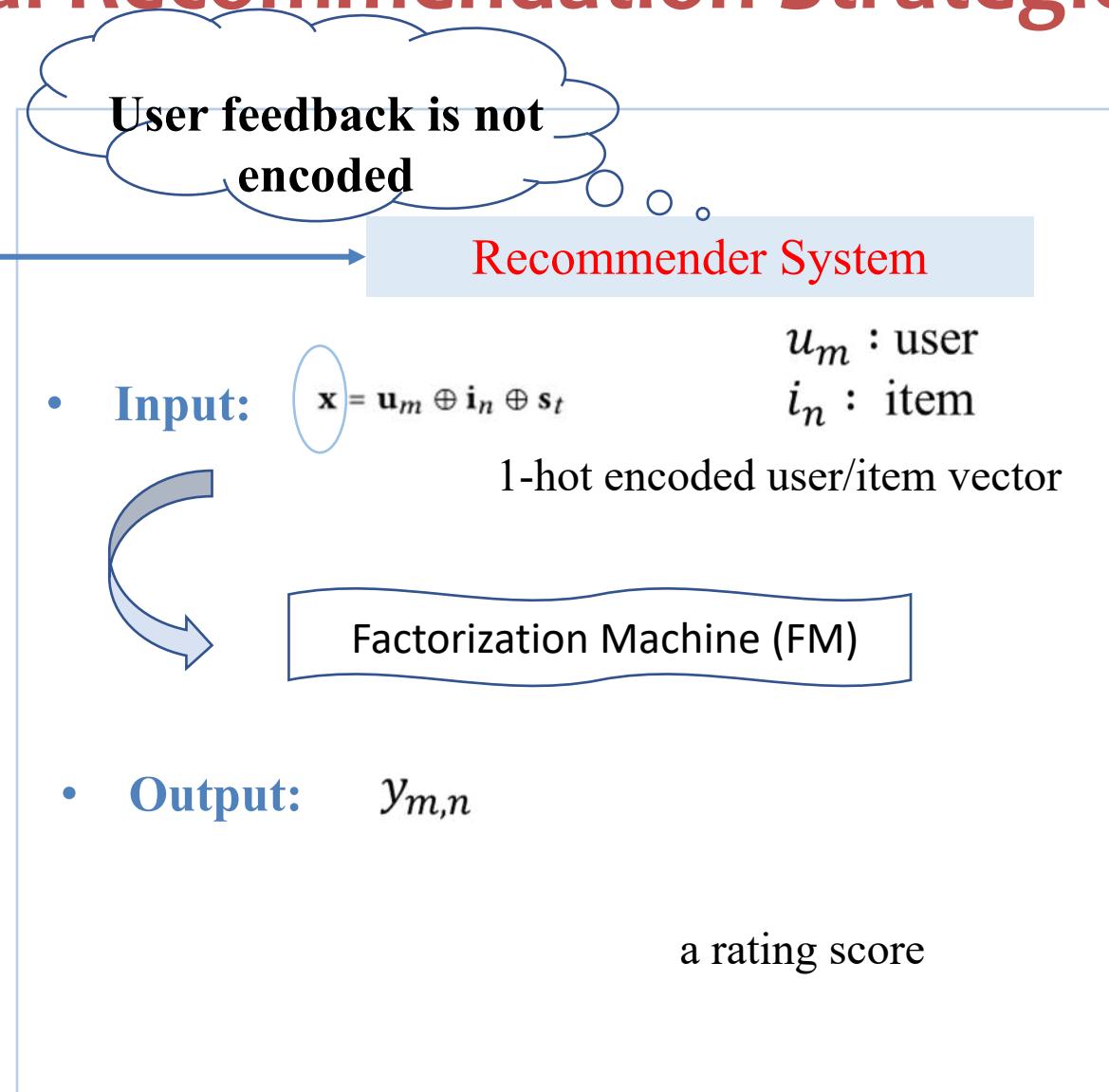
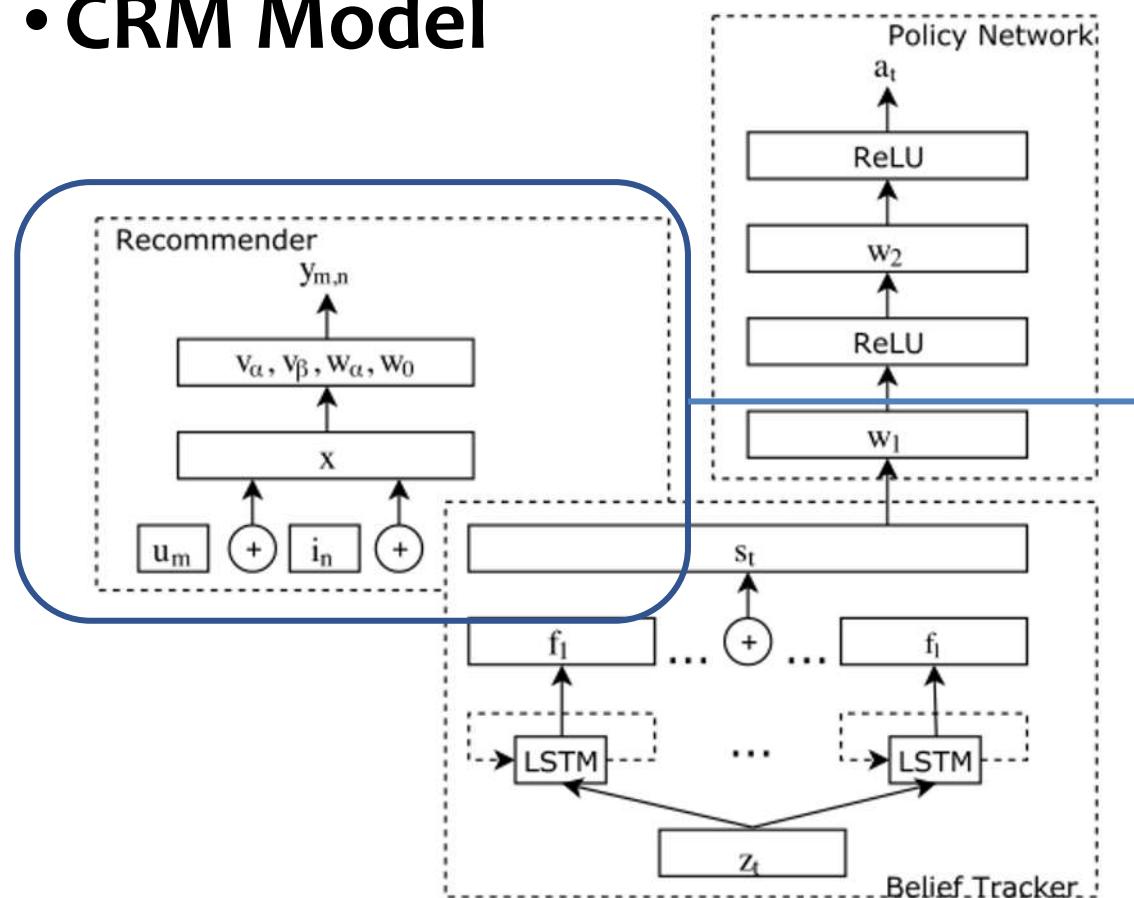
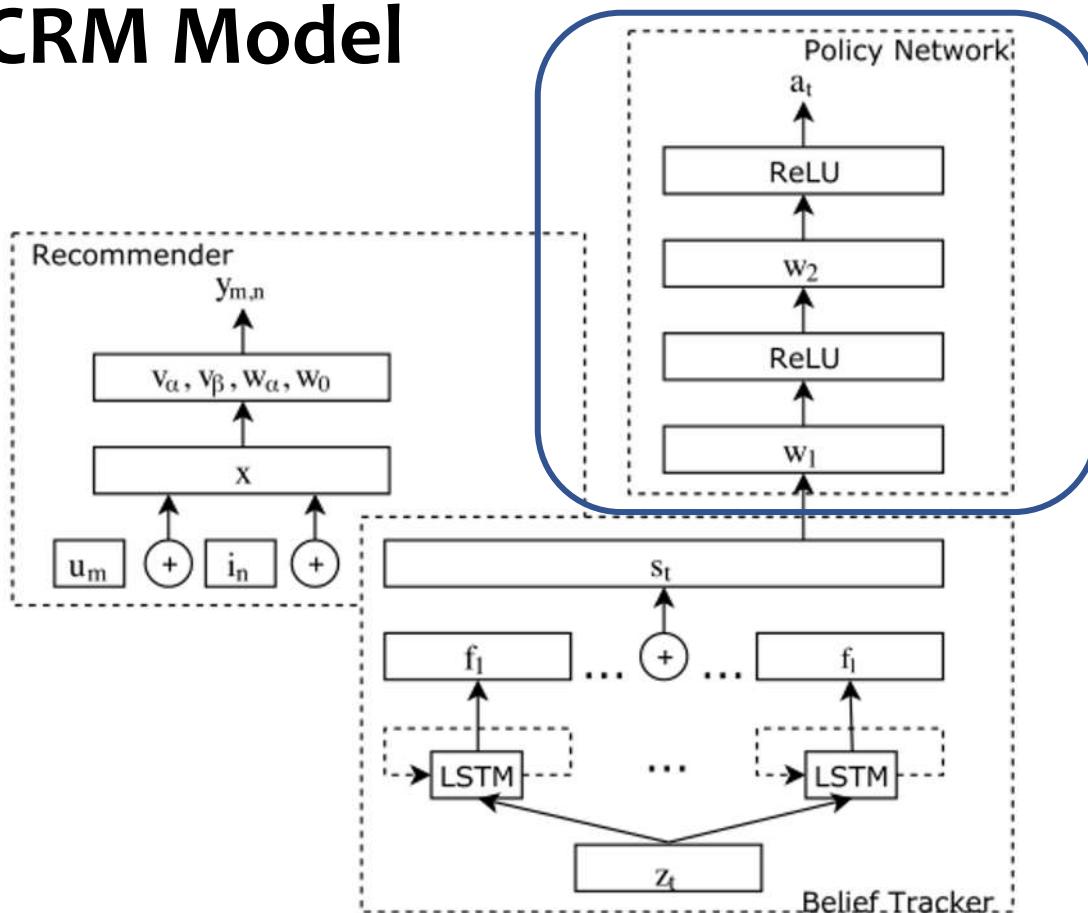


Figure 2: The structure of the proposed conversational recommender model. The bottom part is the belief tracker, the top left part is the recommendation model, and the top right part is the deep policy network.

## 2.2 Multi-turn Conversational Recommendation Strategies

- CRM Model



Decisions based only on the belief tracker

Deep Policy Network

- **State:**  $s_t = \{f_1 \oplus f_2 \dots \oplus f_l\}$ .  
Description of the conversation context
- **Action :**  $\{a_1, a_2, \dots, a_l\}$ , request the value of a facet  
make a personalized recommendation
- **Reward :** benefit/penalty the agent gets from interacting with its environment
- **Policy:**  $\pi(a_t | s_t)$ , two fully connected layers as the policy network

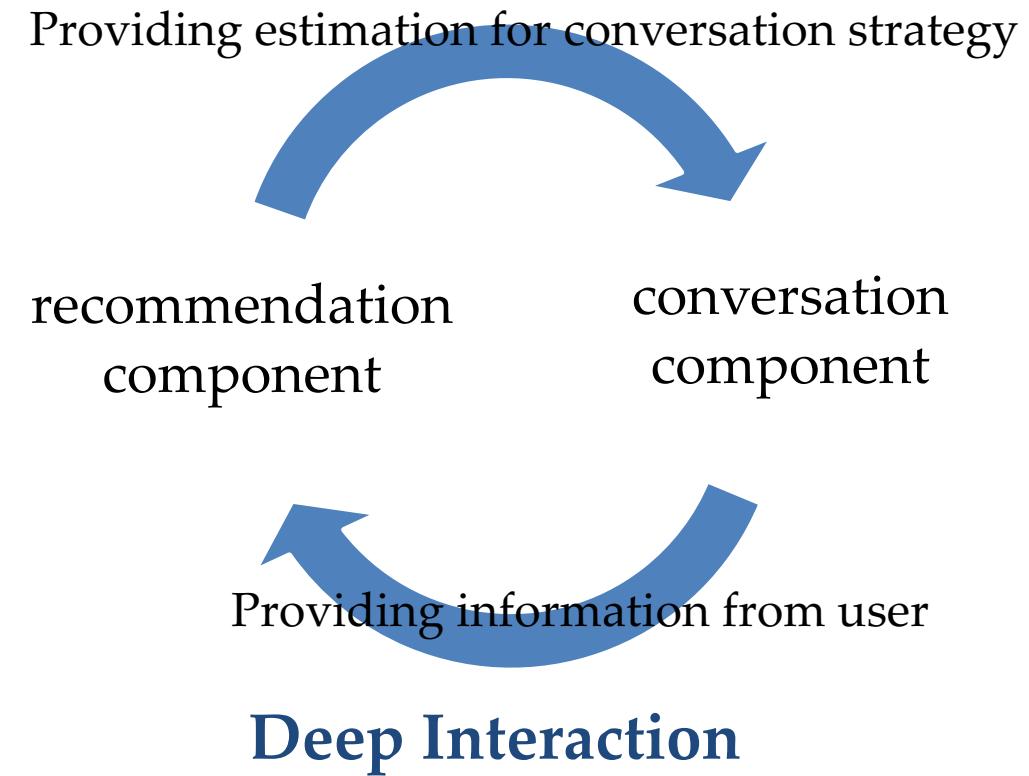
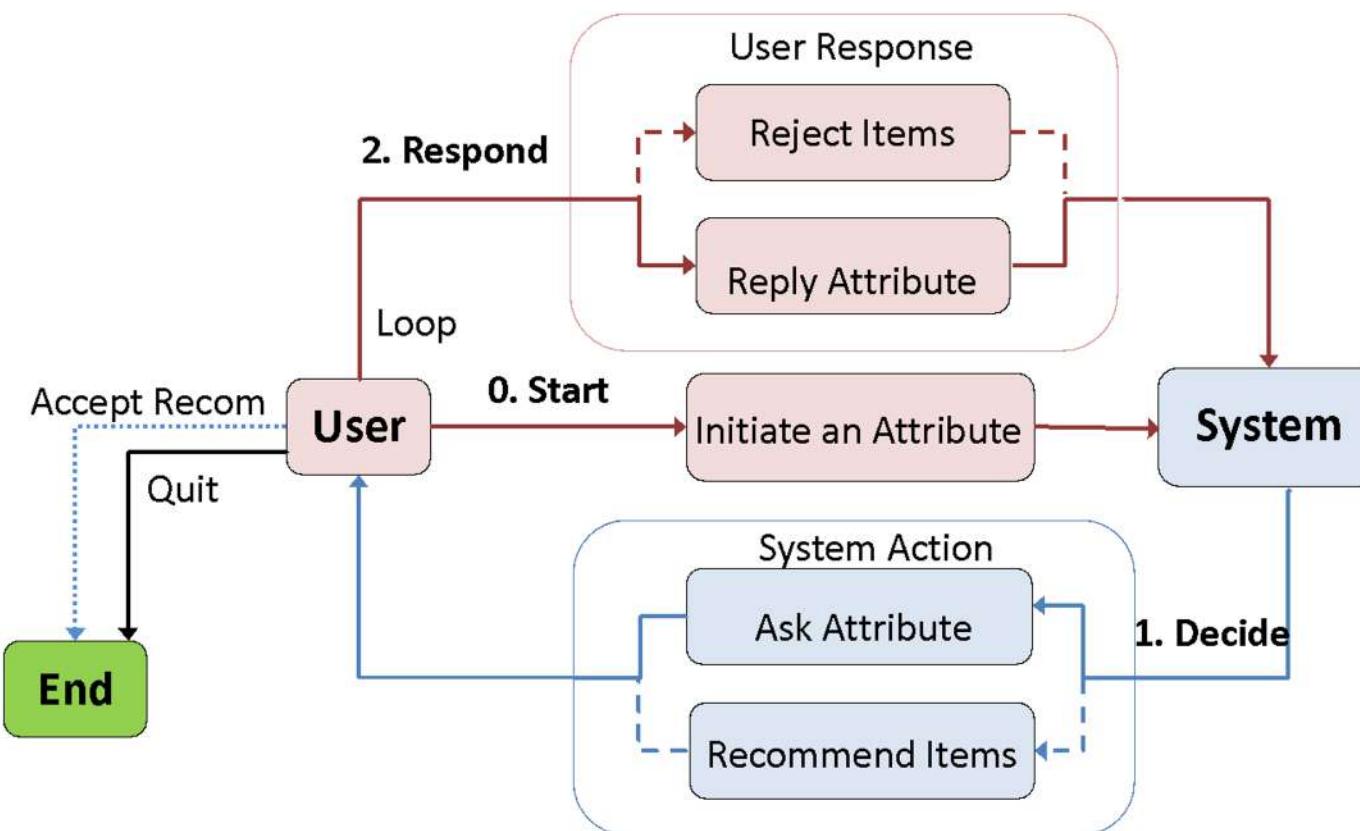
Adopt the policy gradient method of reinforcement learning

Figure 2: The structure of the proposed conversational recommender model. The bottom part is the belief tracker, the top left part is the recommendation model, and the top right part is the deep policy network.

## 2.2 Multi-turn Conversational Recommendation Strategies

- Estimation–Action–Reflection (EAR Model)

A multi-step decision problem under asymmetric information



## 2.2 Multi-turn Conversational Recommendation Strategies

- EAR Model: conversation component supports recommendation component

Notation	Meaning
$p$	A given attribute
$u$	User embedding
$\mathcal{P}_u$	User's known preferred attributes

Notation	Meaning
(Neg. 1) $\mathcal{V}_u^- := \mathcal{V} \setminus \mathcal{V}_u^+$	The ordinary negative sample as in standard BPR.
(Neg. 2) $\widehat{\mathcal{V}_u} := \mathcal{V}_{cand} \setminus \mathcal{V}_u^+$	$\mathcal{V}_{cand}$ is the set of candidate items satisfying user's preferred attributes.
$\mathcal{D}_1 := \{(u, v, v')   v' \in \mathcal{V}_u^-\}$	Paired sample for first kind of negative sample
$\mathcal{D}_2 := \{(u, v, v')   v' \in \widehat{\mathcal{V}_u}\}$	Paired sample for second kind of negative sample

Using attribute to predict item

$$\hat{y}(u, v, \mathcal{P}_u) = \mathbf{u}^T \mathbf{v} + \sum_{p_i \in \mathcal{P}_u} \mathbf{v}^T \mathbf{p}_i$$

Score function for item prediction

$$\begin{aligned} L_{item} &= \sum_{(u, v, v') \in \mathcal{D}_1} -\ln \sigma(\hat{y}(u, v, \mathcal{P}_u) - \hat{y}(u, v', \mathcal{P}_u)) \\ &+ \sum_{(u, v, v') \in \mathcal{D}_2} -\ln \sigma(\hat{y}(u, v, \mathcal{P}_u) - \hat{y}(u, v', \mathcal{P}_u)) \\ &+ \lambda_{\Theta} \|\Theta\|^2 \end{aligned}$$

ordinary negative example

The items satisfying the specified attribute but still are not clicked by the user

## 2.2 Multi-turn Conversational Recommendation Strategies

- EAR Model: recommendation component supports conversation component

Notation	Meaning
$p$	A given attribute
$u$	User embedding
$\mathcal{P}_u$	User's known preferred attributes

Notation	Meaning
(Neg. 1) $\mathcal{V}_u^- := \mathcal{V} \setminus \mathcal{V}_u^+$	The ordinary negative sample as in standard BPR.
(Neg. 2) $\widehat{\mathcal{V}}_u^- := \mathcal{V}_{cand} \setminus \mathcal{V}_u^+$	$\mathcal{V}_{cand}$ is the set of candidate items satisfying user's preferred attributes.
$\mathcal{D}_1 := \{(u, v, v')   v' \in \mathcal{V}_u^-\}$	Paired sample for first kind of negative sample
$\mathcal{D}_2 := \{(u, v, v')   v' \in \widehat{\mathcal{V}}_u^-\}$	Paired sample for second kind of negative sample

Using items to predict attributes

$$\widehat{g}(p|u, \mathcal{P}_u) = u^T p + \sum_{p_i \in \mathcal{P}_u} \mathbf{P}^T \mathbf{P}_i$$

Score function for attribute preference prediction

$$L_{attr} = -\ln \sigma(\widehat{g}(p|u, \mathcal{P}_u) - \widehat{g}(p'|u, \mathcal{P}_u)) + \lambda_\Theta \|\Theta\|^2$$

$$L = L_{item} + L_{attr}$$

Multi-task Learning: Optimize for item ranking and attribute ranking simultaneously.

## 2.2 Multi-turn Conversational Recommendation Strategies

- EAR Model: recommendation component supports conversation component

We use **reinforcement learning** to find the best strategy.

- policy gradient method
- simple policy network (2-layer feedforward network)

- **State Vector**

- $s_{entropy}$ : The entropy of attribute is important.

- $s_{preference}$ : User's preference on each attribute.

- $s_{history}$ : Conversation history is important.

- $s_{length}$ : Candidate item list length.

### Reward

$r_{success}$ : Give the agent a big reward when it successfully recommend!

$r_{ask}$ : Give the agent a small reward when it ask a correct attribute.

$r_{quit}$ : Give the agent a big negative reward when the user quit (the conversation is too long)

Note: 3 of the 4 information come from Recommender Part

Action Space:

$$|\mathcal{P}| + 1$$

$r_{prevent}$ : Give each turn a relatively small reward to prevent the conversation goes too long.

## 2.2 Multi-turn Conversational Recommendation Strategies

- **Conversational Path Reasoning (CPR) model**

**Core idea:** the CRS asks the questions and generates questions based on the generated paths on the graph.

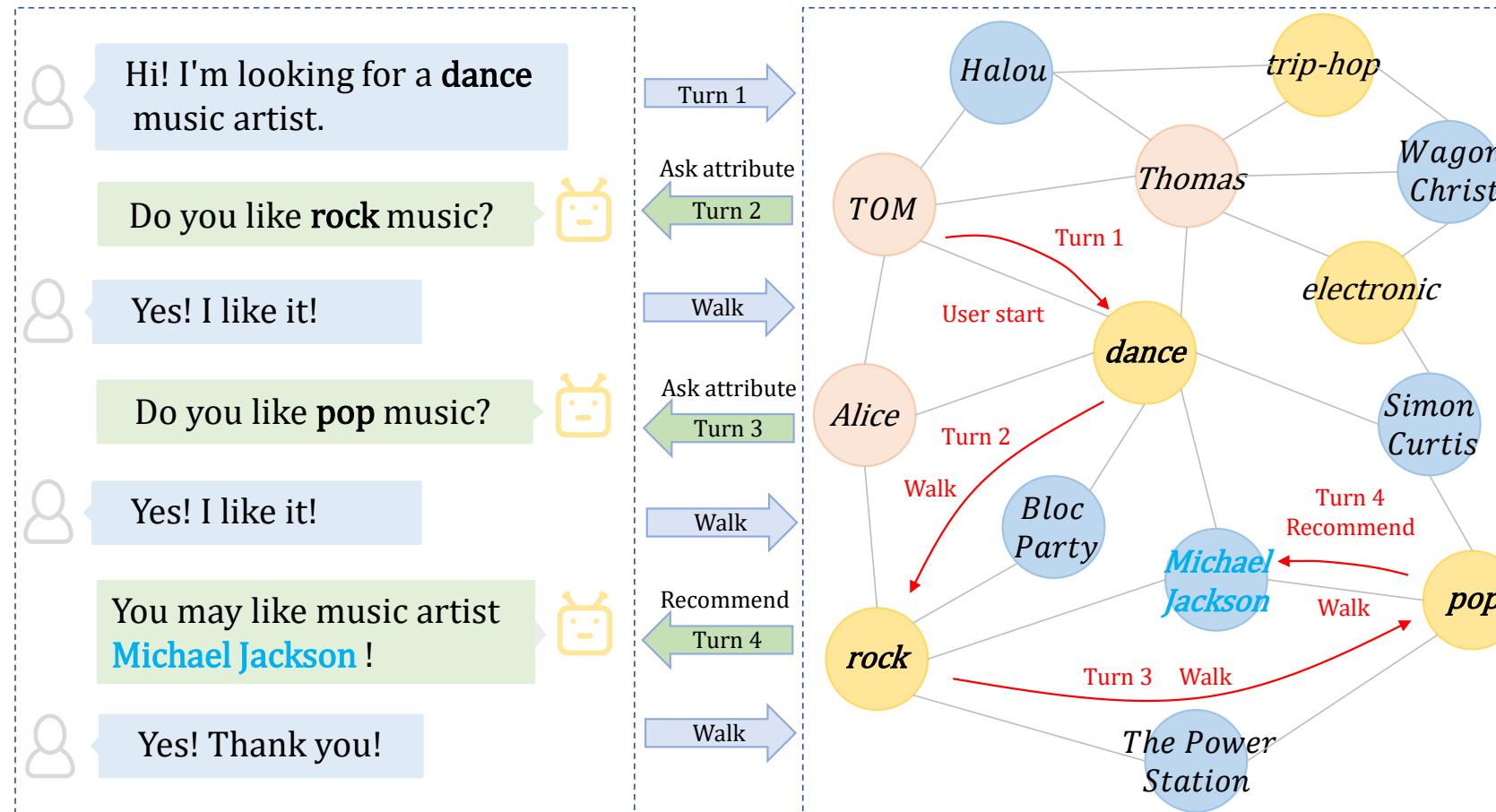


Figure Credit: Wenqiang Lei et al. Interactive Path Reasoning on Graph for Conversational Recommendation. KDD' 20

## 2.2 Multi-turn Conversational Recommendation Strategies

### • CPR - Method

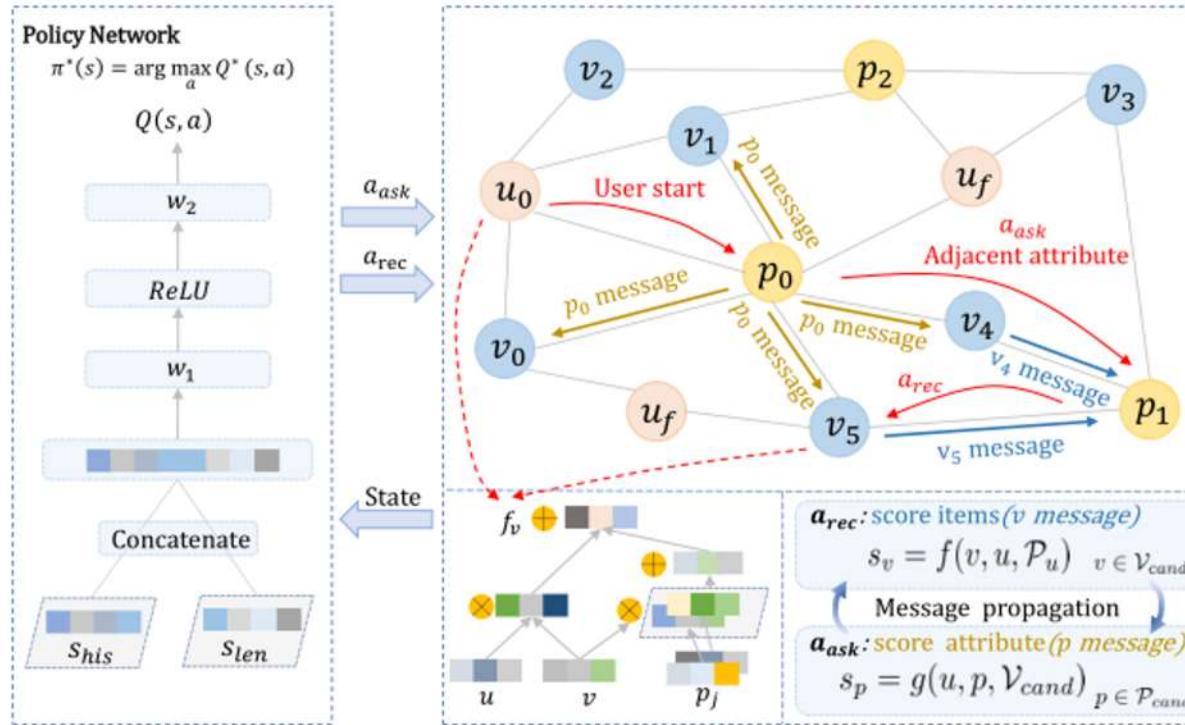


Figure 2: CPR framework overview. It starts from the user  $u_0$  and walks over adjacent attributes, forming a path (the red arrows) and eventually leading to the desired item. The policy network (left side) determines whether to ask an attribute or recommend items in a turn. Two reasoning functions  $f$  and  $g$  score attributes and items, respectively.

### CPR Framework

#### • Assuming

- Current path  $P = p_0, p_1, p_2, \dots, p_t$
- $u$ : user     $v$ : item     $p$ : attribute
- $\mathcal{P}_u$ : user's preferred attributes
- $\mathcal{V}_{cand}$ : candidate items

#### • Reasoning

- Score items to recommend ( $v$  message):  
$$s_v = f(v, u, \mathcal{P}_u) \quad v \in \mathcal{V}_{cand}$$
- Message propagation
- Score attribute to ask ( $p$  message):  
$$s_p = g(u, p, \mathcal{V}_{cand}) \quad p \in \mathcal{P}_{cand}$$

#### • Consultation

- Policy network (choose to ask or rec)

#### • Transition

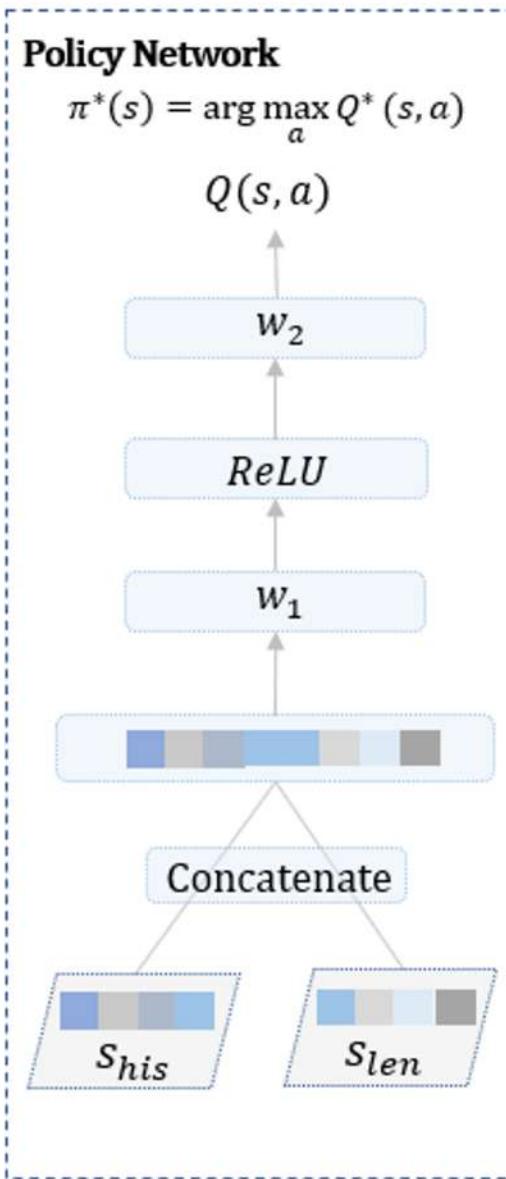
- Extended path

$$P = p_0, p_1, p_2, \dots, p_t, p_{t+1}$$

- Update candidate item /attribute set ( $\mathcal{V}_{cand}/\mathcal{P}_{cand}$ )

## 2.2 Multi-turn Conversational Recommendation Strategies

### • CPR - Method



#### Input

$s_{his}$ : encodes the conversation history

$s_{len}$ : encodes the size of candidate items

#### Output

$$Q(s, a)$$

$Q(s, a)$  : the value of action  $a$  in state  $s$

$a_{rec}$ : the action of recommendation

$a_{ask}$ : the action of asking attribute

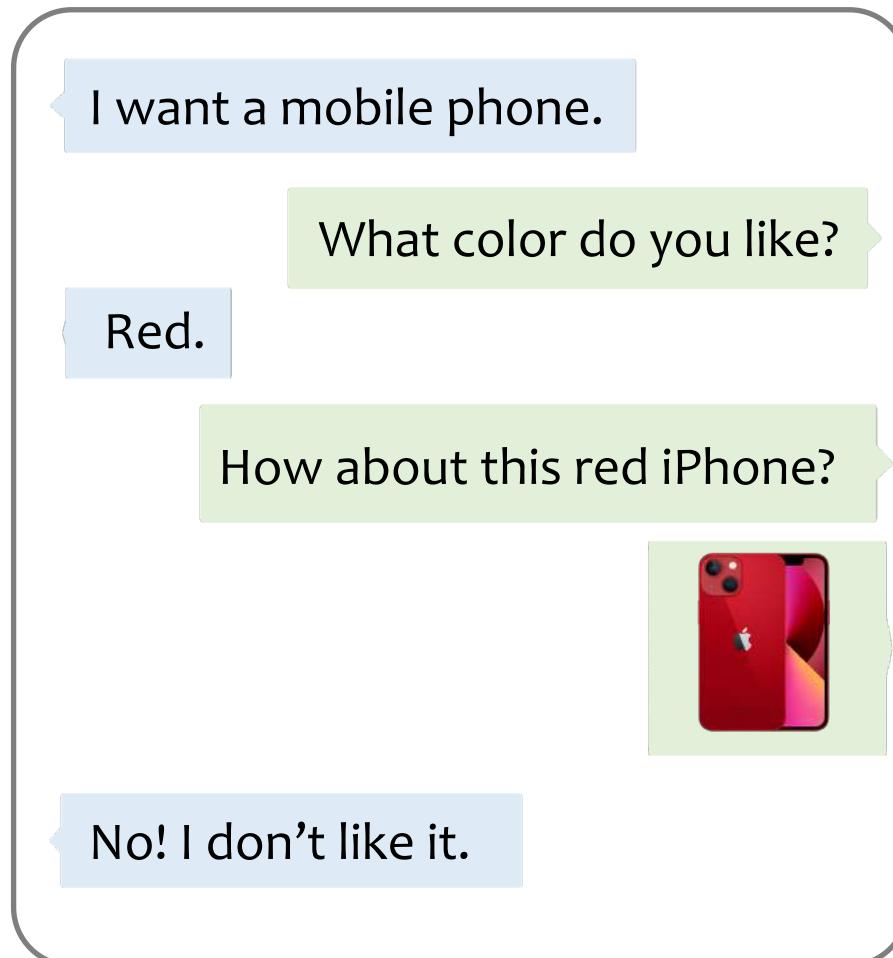
#### DQN method

Policy:  $\pi^*(s) = \arg \max_a Q^*(s, a)$

TD loss:  $\delta = Q(s, a) - \left( R + \gamma \max_a Q(s', a) \right)$

## 2.2 Multi-turn Conversational Recommendation Strategies

- How to handle rejected items/attributes?



**Negative samples in CRM, EAR, CPR models**



The item: red iPhone

**How about attribute-level preference?**



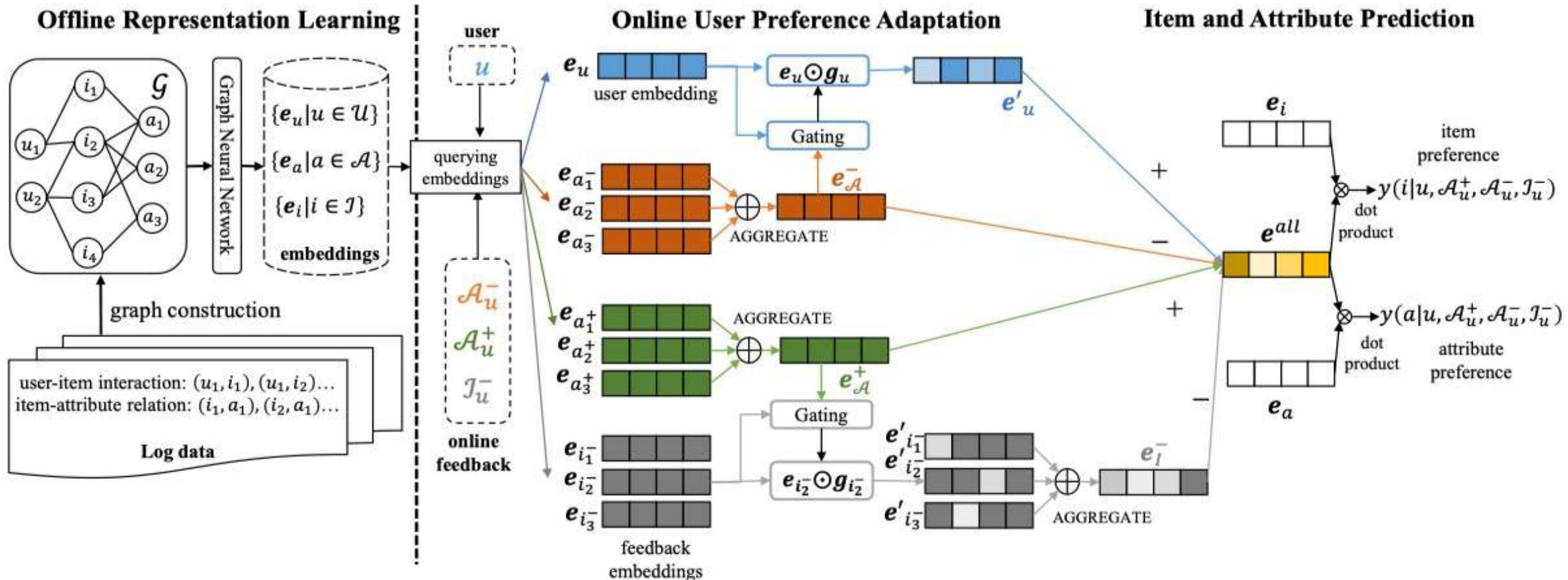
User like explicitly



User might not like

## 2.2 Multi-turn Conversational Recommendation Strategies

- FPAN: disentangle item-level and attribute level feedback



## 2.2 Multi-turn Conversational Recommendation Strategies

- Other efforts

- Problem: too many items making decision making hard

Solution: using actor-critic framework

*Ali Montazer alghaem et al. Large-scale Interactive Conversational Recommendation System using Actor-Critic Framework. RecSys' 21*

- Problem: too sparse reward making policy function hard to converge

Solution: using more fine-grained reward

*Ruiyi Zhang et al. Reward Constrained Interactive Recommendation with Natural Language Feedback. NeurIPS' 19*

*Yaxiong Wu et al. Partially Observable Reinforcement Learning for Dialog-based Interactive Recommendation. RecSys' 21*



# Outline

## I. Introduction

## II. Five Important Challenges

2.1 Question-based User Preference Elicitation.

2.2 Multi-turn Conversational Recommendation Strategies.

2.3 Natural Language Understanding and Generation.

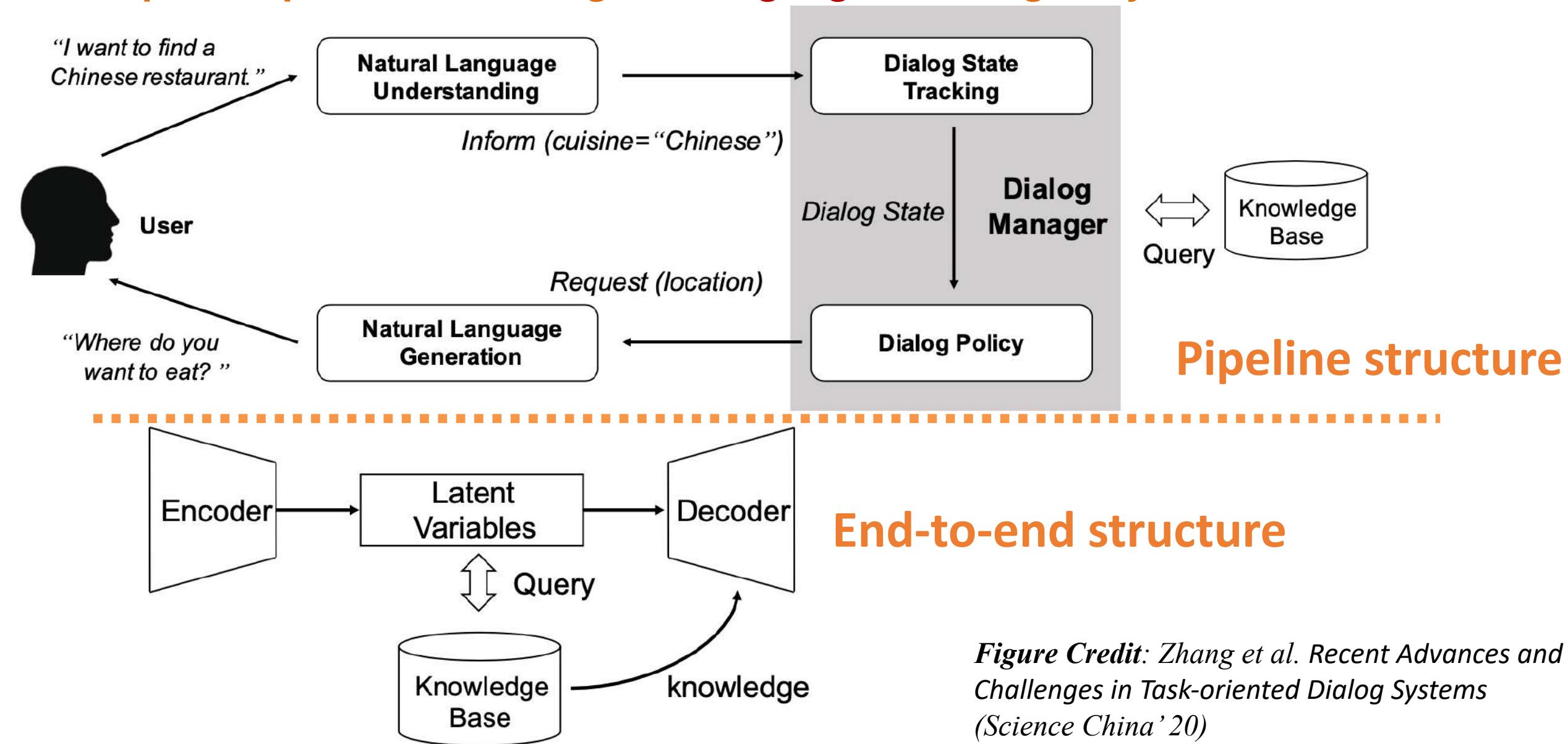
2.4 Trade-offs between Exploration and Exploitation (E&E).

2.5 Evaluation and User Simulation.

## III. Promising Future Directions

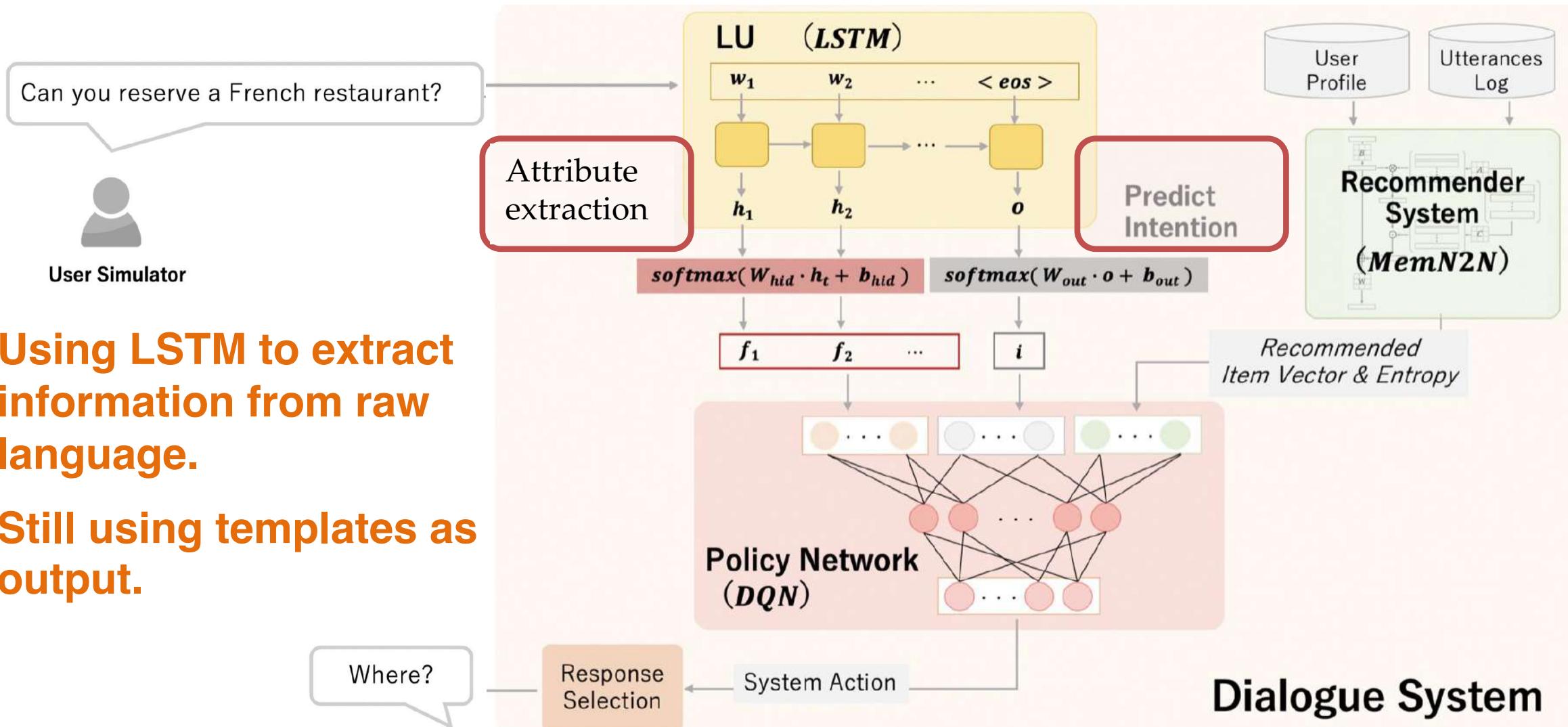
## 2.3 Natural Language Understanding and Generation

- Two philosophies of handling raw language in dialogue systems



## 2.3 Natural Language Understanding and Generation

- An illustration of dialogue system-based CRS



## 2.3 Natural Language Understanding and Generation

### A classic CRS with end-to-end structure. REDIAL Model

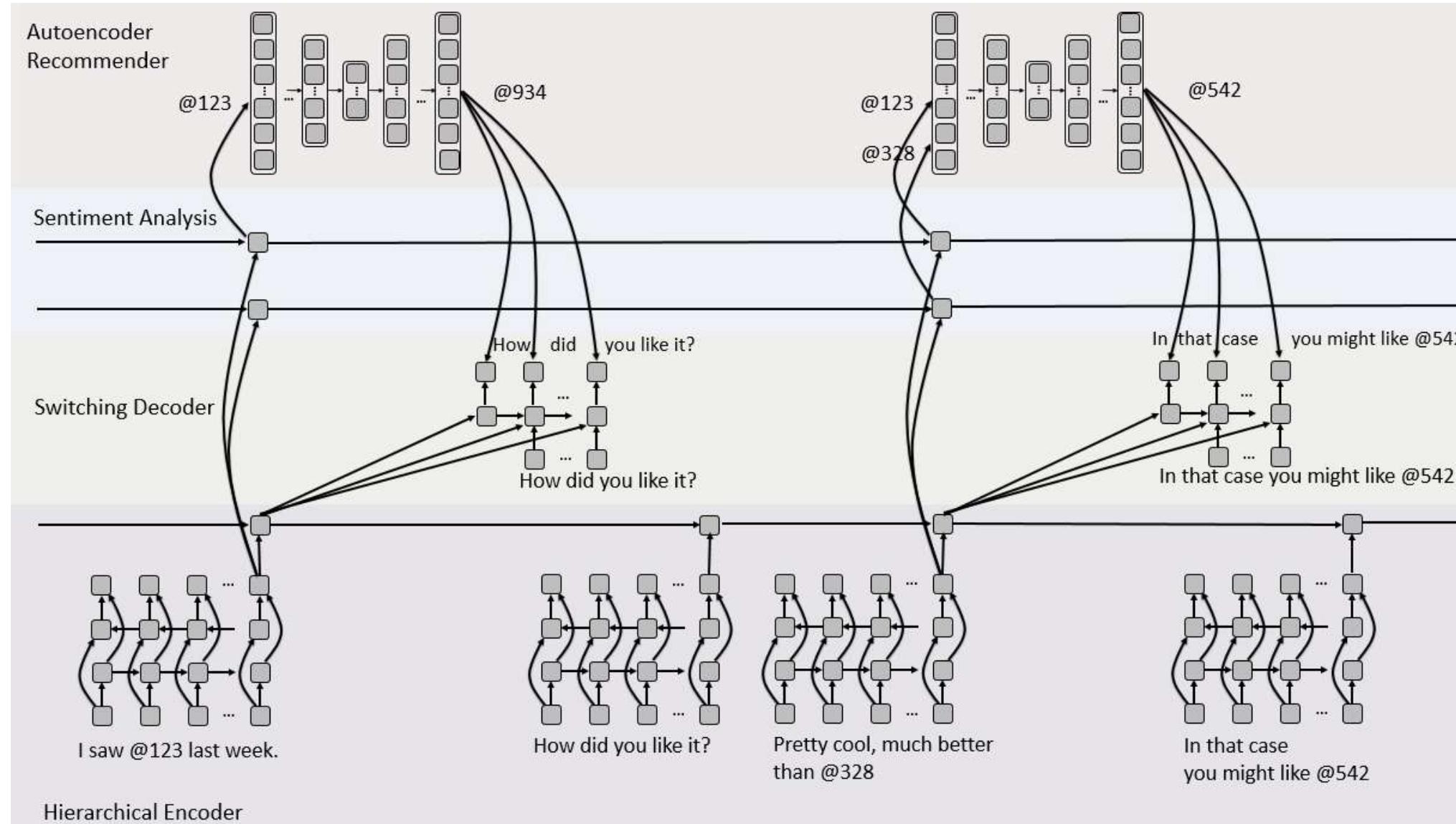
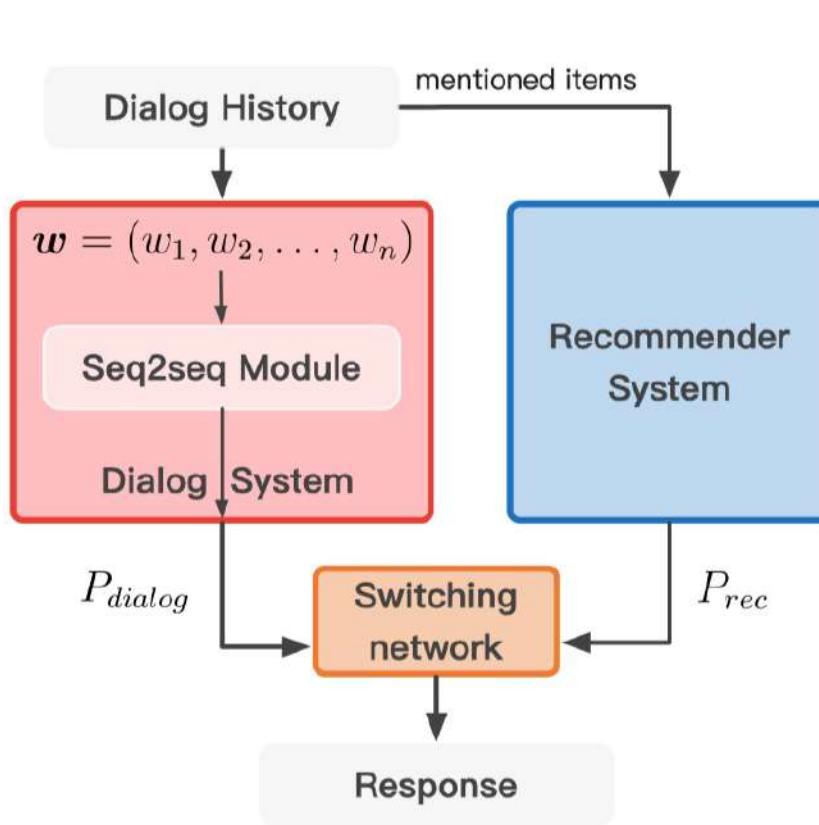


Figure Credit: Raymond Li, et al. Towards Deep Conversational Recommendations. NeurIPS' 18

## 2.3 Natural Language Understanding and Generation

- Introducing Knowledge Graph

**REDIAL model**



**KBRD model**

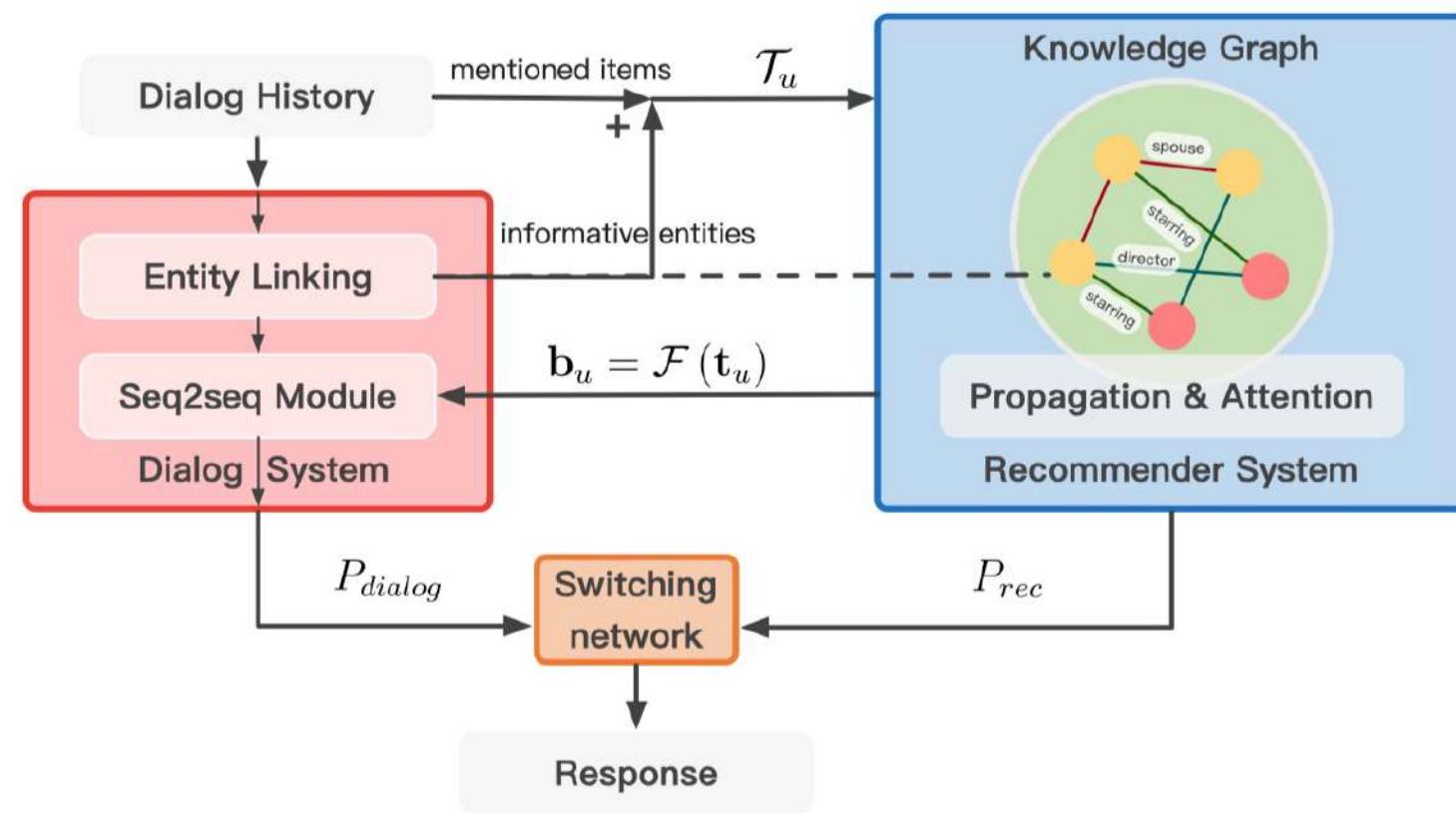
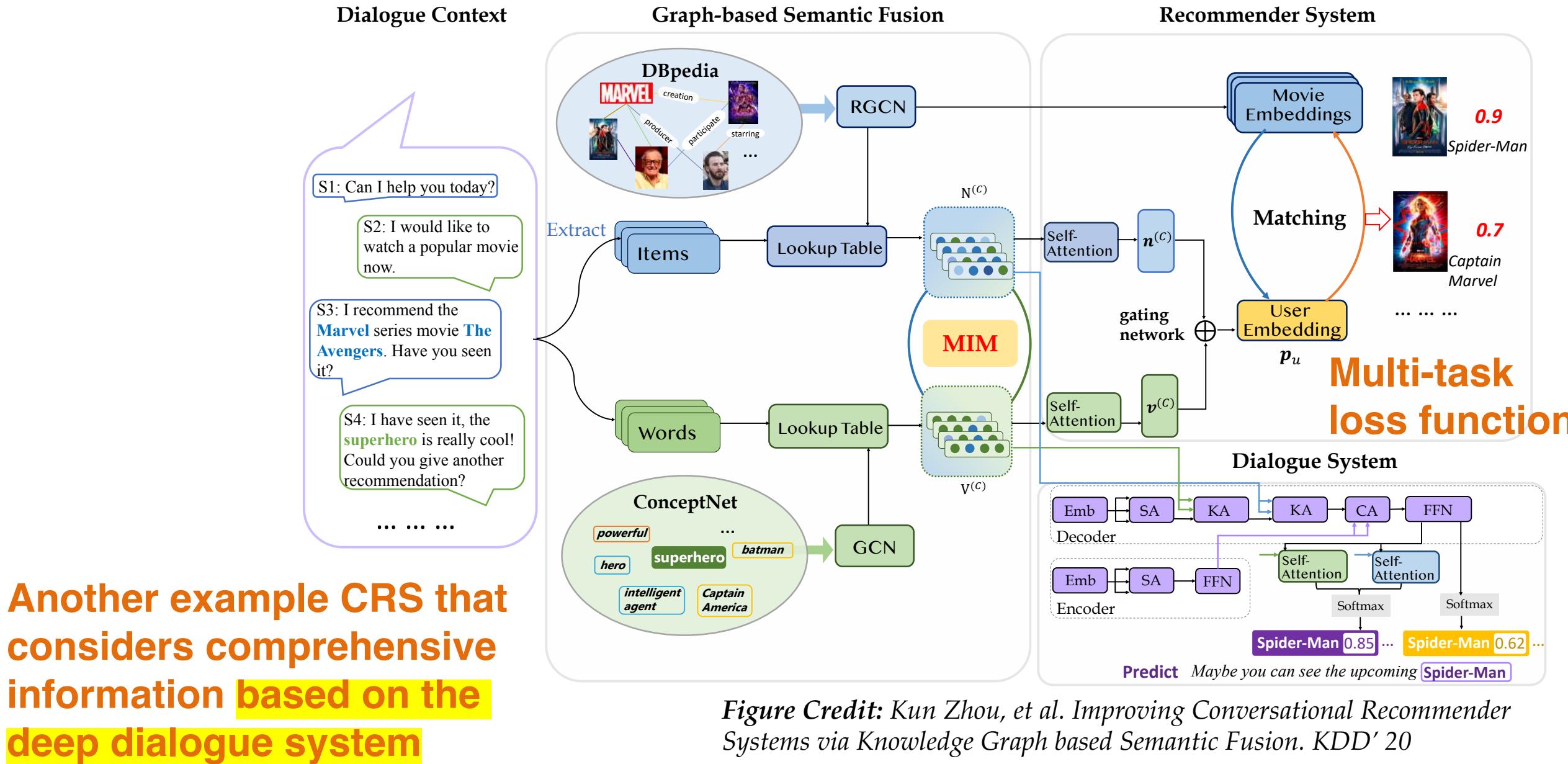


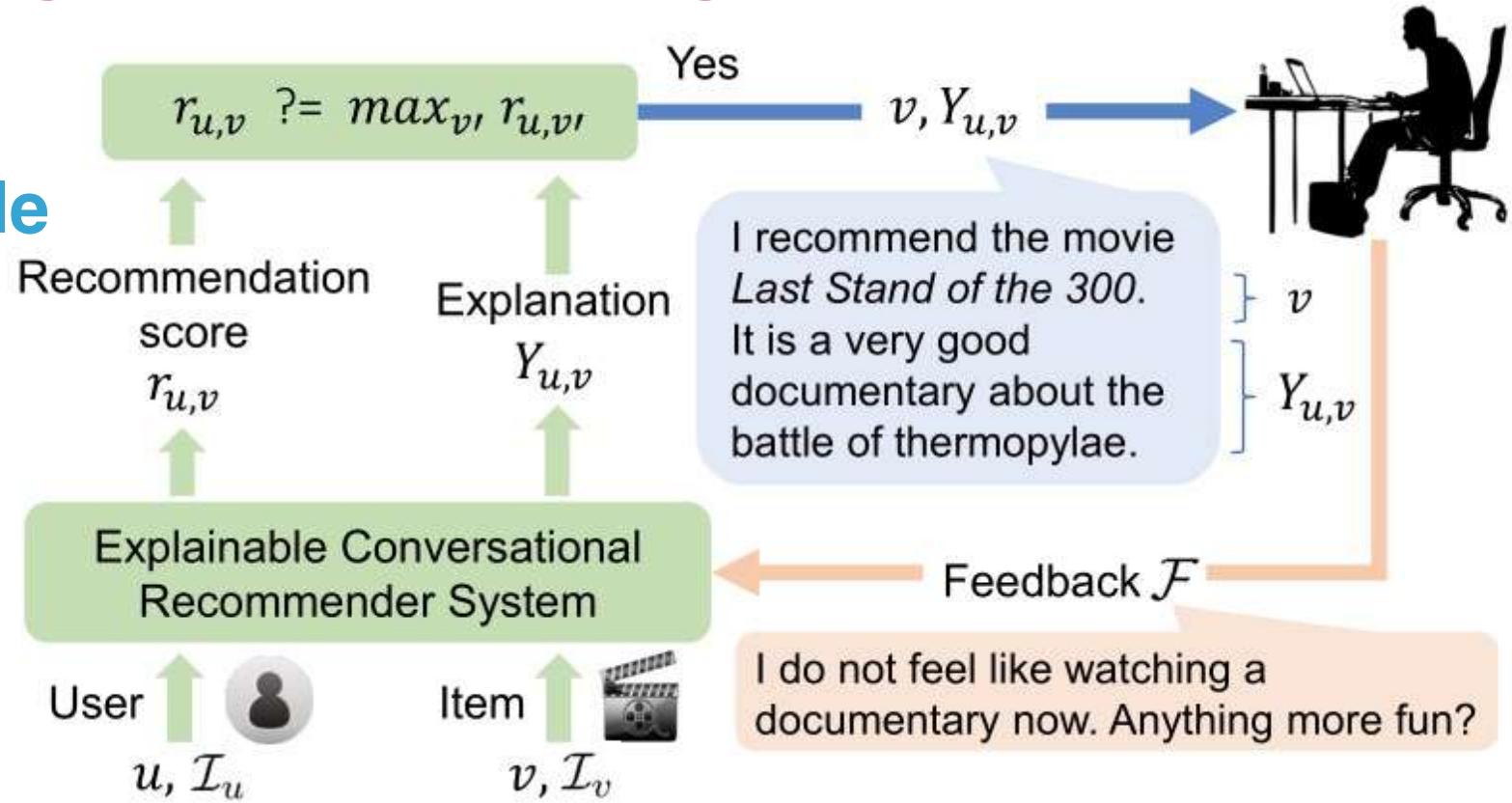
Figure Credit: Qibin Chen, et al. Towards Knowledge-Based Recommender Dialog System.

# 3.3 Natural Language Understanding and Generation



## 2.3 Natural Language Understanding and Generation

- Pipeline of explainable conversational recommendation



**Model:** I recommend Pulp Fiction. This is a dark comedy with a great cast.

**User:** I don't want to watch a comedy right now.

**Model:** How about Ice Age? It is a very good anime with a lot of action adventure.

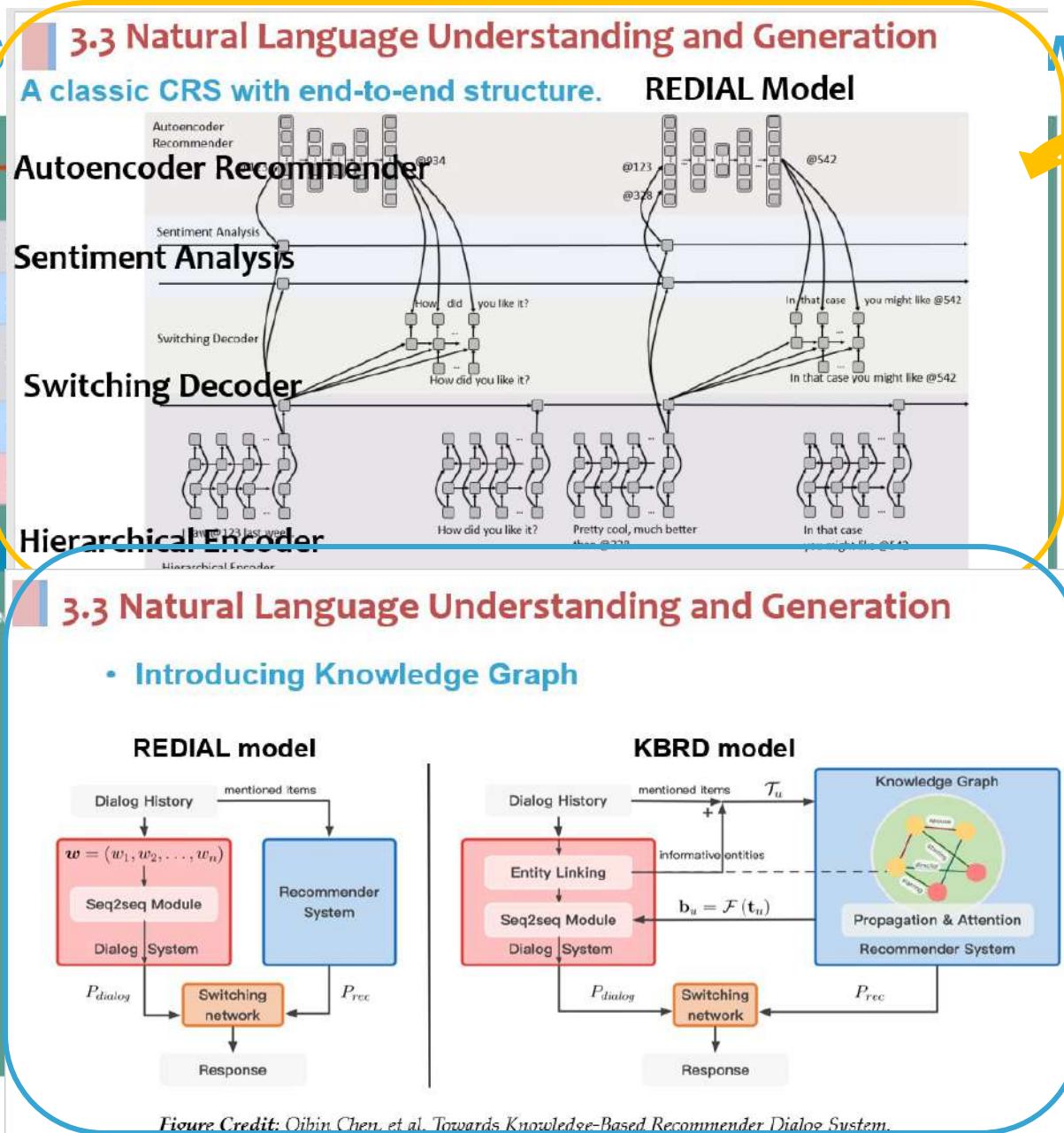
**User:** I don't like anime, but action movie sounds good.

**Model:** I recommend Mission Impossible. This is by far the best of the action series.

**User:** Sounds great. Thanks for the recommendation!

# 2.3 Natural Language Understanding and Generation

- Dis



work is ready for CRSs?

	DeepCRS	KBRD	RB-CRS
Avg. score	3.13	3.46	3.71
Std. deviation	1.49	1.45	1.32

Potential value of relying on retrieval-based components when building a CRS

Figure Credit: Generation-based vs. Retrieval-based Conversational Recommendation: A User-Centric Comparison. RecSys '21

## 2.3 Natural Language Understanding and Generation

- Discussion: Whether generation network is ready for CRSs?
  - Another view: end-to-end learning may have a long way to go?

As a study conducted on the state-of-the-art baselines shows:

1. For each system, about one-third of the system utterances are not meaningful in the given context and would probably lead to a breakdown of the conversation in a human evaluation.
2. Less than two-thirds of the recommendations were considered to be meaningful in a human evaluation.
3. Neither of the two systems "generated" utterances, as almost all system responses were already present in the training data.

## 2.3 Natural Language Understanding and Generation

- **Summarized problems** in existing CRSs based on dialogue systems:
  - Focusing on deep end-to-end NLP models to **fit the patterns** from human conversations.
  - **Failure** to generate new conversation;
  - **Failure** to produce satisfying recommendation (Jannach et al.).

*Source:* Dietmar Jannach and Ahtsham Manzoor. 2020. *End-to-End Learning for Conversational Recommendation: A Long Way to Go?* (RecSys Workshop 2020)

- However, **it is worthy of trying**, since natural language have the advantages:
  - **Flexible.**
  - **Natural for users.**



# Outline

## I. Introduction

## II. Five important challenges

2.1 Question-based user preference elicitation

2.2 Multi-turn conversational recommendation strategies

2.3 Natural language understanding and generation

2.4 Trade-offs between exploration and exploitation (E&E)

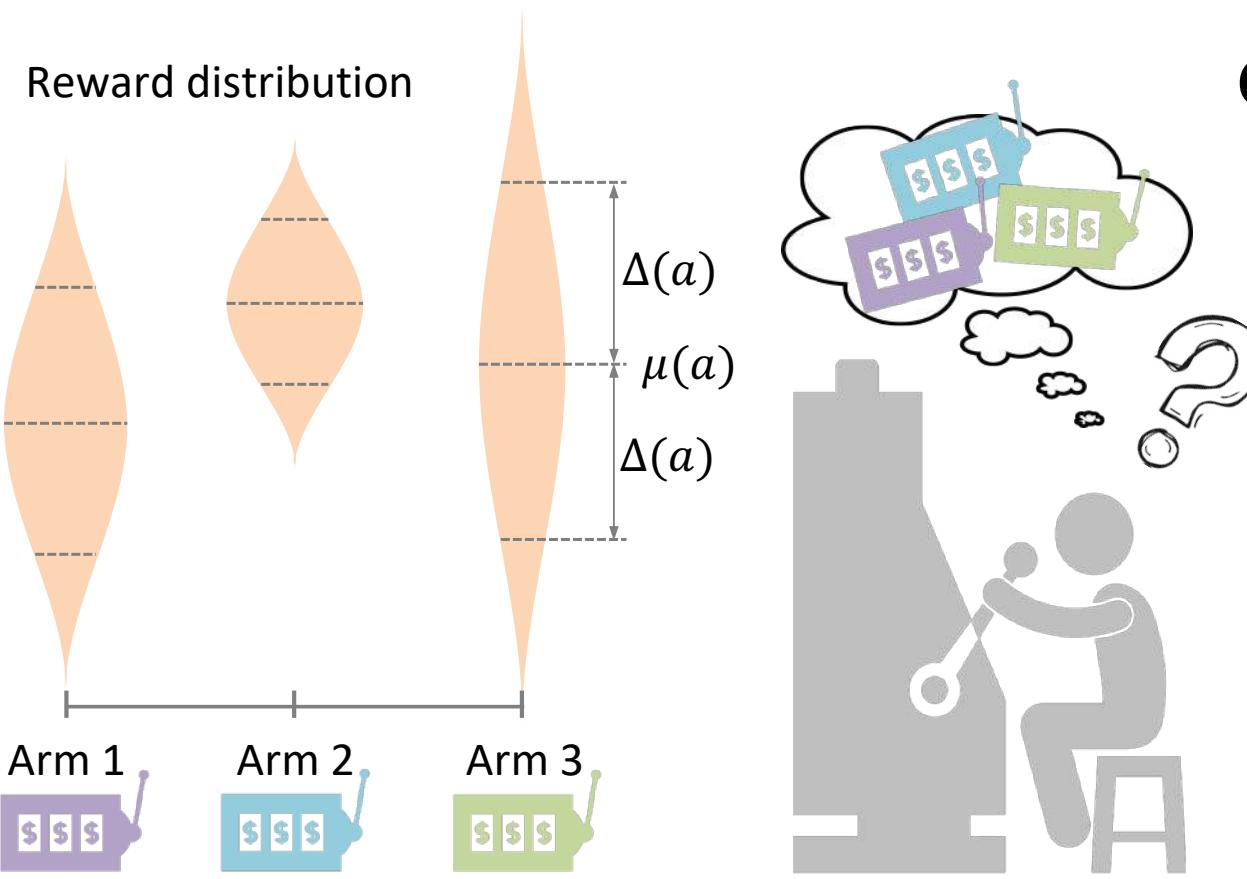
2.5 Evaluation and user simulation

## III. Promising future directions

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

**Multi-armed Bandit problem:** A gambler needs to decide which arm to pull to get the maximal reward.

He can only estimate the statistics, e.g., the mean  $\mu(a)$  and uncertainty  $\Delta(a)$  of each arm by doing experiments.



**Goal:** To maximize the cumulative reward, which can be formulated as minimizing the **regret function** (the difference between the theoretically optimal expected cumulative reward and the estimated expected cumulative reward):

$$\mathbf{E} \left[ \sum_{t=1}^T r_{t,a^*} \right] - \mathbf{E} \left[ \sum_{t=1}^T r_{t,a} \right]$$

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Exploration (Learning)

- ✓ Take some risk to collect information about unknown options

### Exploitation (Earning)

- ✓ Takes advantage of the best option that is known.



Multi-armed bandit example: which arm to select next? 

#(Successes)  
—  
#(Trials)

Arm 1  
2/5

Arm 2  
0/1

Arm 3  
3/8

Arm 4  
1/3

...

Common intuitive ideas:

- **Greedy:** trivial exploit-only strategy
- **Random:** trivial explore-only strategy
- **Epsilon-Greedy:** combining Greedy and Random.
- **Max-Variance:** only exploring w.r.t. uncertainty.

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Upper Confidence Bounds (UCB)

Arm selection strategy:

$$\hat{a} = \arg \max_a \hat{Q}(a) + \Delta(a)$$

Exploitation      Exploration

$$\hat{Q}(a) = \frac{1}{N} \sum_{t=1}^{N_a} r_{t,a}$$

$Q(a)$  : The true mean of reward of arm  $a$ .

$\hat{Q}(a)$  : The mean of estimated reward of arm  $a$ .

$\Delta(a)$  : The uncertainty of  $\hat{Q}(a)$ .

According to  
Hoeffding's Inequality

$$P[Q(a) > \hat{Q}(a) + \Delta(a)] \leq e^{-2N_a \Delta(a)}$$

By setting:  $p = e^{-2N_a \Delta(a)}$ , we have:  $\Delta(a) = \sqrt{\frac{-\log p}{2N_a}}$

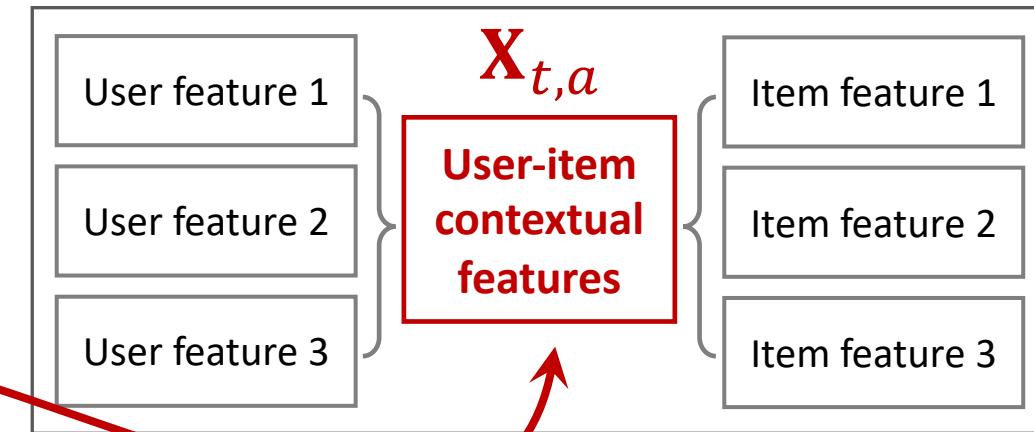
## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### A Contextual-Bandit Approach with Linear Reward (LinUCB)

#### Solution to personalized recommendation:

- Modelling **contextual information** into the bandit reward function by assuming expected payoff of a arm  $a$  is **linear** in its  $d$ -dimensional feature  $\mathbf{X}_{t,a}$

$$E[r_{t,a} | \mathbf{X}_{t,a}] = \mathbf{X}_{t,a}^T \boldsymbol{\theta}_a$$



- Let  $\mathbf{D}_a$  be a matrix of dimension  $m \times d$  at trial  $t$  (i.e.,  $m$  contexts  $\mathbf{X}_{t,a}^T$  that are observed previously for arm  $a$ ), the close-form solution of  $\boldsymbol{\theta}_a$  is

$$\boldsymbol{\theta}_a = (\mathbf{D}_a^T \mathbf{D}_a + \mathbf{I}_d)^{-1} \mathbf{D}_a^T c_a$$

- The arm selection strategy is:

**Exploitation**    **Exploration**

$$a_t \stackrel{\text{def}}{=} \arg \max_a \left( \boxed{\mathbf{X}_{t,a}^T \boldsymbol{\theta}_a} + \alpha \sqrt{\boxed{\mathbf{X}_{t,a}^T \mathbf{A}_a^{-1} \mathbf{X}_a}} \right)$$

$$\text{where } \mathbf{A}_a \stackrel{\text{def}}{=} \mathbf{D}_a^T \mathbf{D}_a + \mathbf{I}_d$$
$$\alpha = 1 + \sqrt{\ln(2/\delta)/2}$$

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### E&E-based methods adopted in IRSs (interactive RSs) and CRSs

	Mechanism	Publications
MAB in IRSs	Linear UCB considering item features	[92]
	Considering diversity of recommendation	[137, 103, 40]
	Cascading bandits providing reliable negative samples	[84, 231]
	Leveraging social information	[205]
	Combining offline data and online bandit signals	[145]
	Considering pseudo-rewards for arms without feedback	[30]
	Considering dependency among arms	[180]
	Considering exploration overheads	[198]
MAB in CRSs	Traditional bandit methods in CRSs	[32]
	Conversational upper confidence bound	[209]
	Conversational thompson sampling	[95]
	Cascading bandits augmented by visual dialogues	[205]
Meta learning for CRSs	Learning to learn the recommendation model	[87, 235, 188]

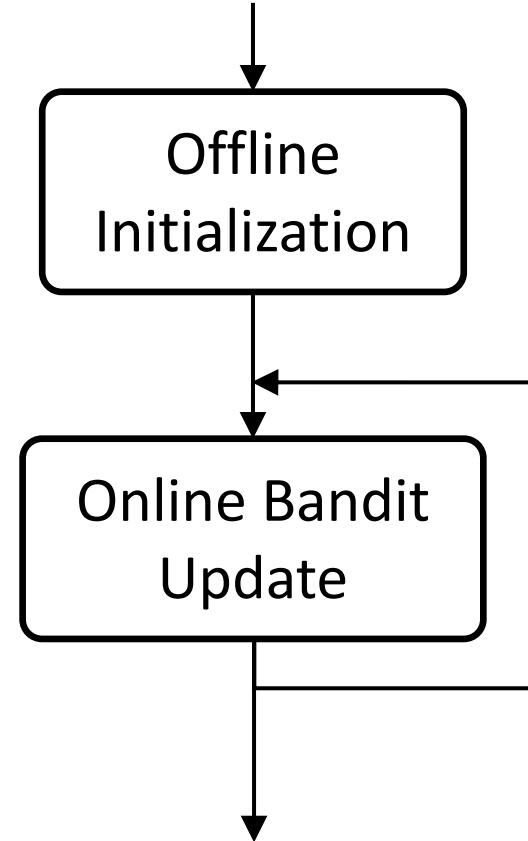
## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Setting:

- Applying multi-armed bandit algorithms in interactive recommendation applications.
- The model is initialized from offline data, and updated in the dynamic interactions.

### Advantages:

- The model can modify its parameters on the fly.
- Diversity of the model is explored, and users have chances to see new item they never interacted before.



## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Bandit algorithm in Conversational Recommendation System

#### Traditional recommendation model

**Absolute Model.** First, let us assume that we have observed tuples of the form (user  $i$ , item  $j$ , 1/0).<sup>4</sup> The model estimates the *affinity* of user  $i$  to item  $j$  based on the biases and traits. The generative procedure is:

1. User  $i$  has traits  $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I})$ , bias  $\alpha_i \sim \mathcal{N}(0, \sigma_2^2)$ .
2. Item  $j$  has traits  $\mathbf{v}_j \sim \mathcal{N}(\mathbf{0}, \sigma_1^2 \mathbf{I})$ , bias  $\beta_j \sim \mathcal{N}(0, \sigma_2^2)$ .
3. (a) The (unobserved) affinity is

$$y_{ij} = \alpha_i + \beta_j + \mathbf{u}_i^T \mathbf{v}_j. \quad (1)$$

Observations are modeled as the noisy estimate  $\hat{y}_{ij} \sim \mathcal{N}(y_{ij}, \epsilon_{ij})$ , where  $\epsilon_{ij}$  models the affinity variance, accounting for noise in user preferences. This yields an observation of whether the user likes an item ( $\hat{r}_{ij}$ ):

$$\hat{r}_{ij} = \mathbf{1}[\hat{y}_{ij} > 0]. \quad (2)$$

#### Traditional MF-based recommendation model



#### bandit model

**Greedy:**  $j^* = \arg \max_j y_{ij}$

A trivial *exploit*-only strategy: Select the item with highest estimated affinity mean.

**Random:**  $j^* = \text{random}(1, N)$

A trivial *explore*-only strategy.

**Maximum Variance (MV):**  $j^* = \arg \max_j \epsilon_{ij}$

A *explore*-only strategy, variance reduction strategy: Select the item with the highest noisy affinity variance.

**Maximum Item Trait (MaxT):**  $j^* = \arg \max_j \|\mathbf{v}_j\|_2$

Select the item whose trait vector  $\mathbf{v}_j$  contains the most information, namely has highest L2 norm  $\|\mathbf{v}_j\|_2 = \sqrt{v_{j1}^2 + v_{j2}^2 + \dots + v_{jd}^2}$ .

**Minimum Item Trait (MinT):**  $j^* = \arg \min_j \|\mathbf{v}_j\|_2$

Select the item with trait vector with least information.

**Upper Confidence (UCB):**  $j^* = \arg \max_j y_{ij} + \epsilon_{ij}$

Based on UCB1 [3]: Pick the item with the highest upper confidence bound, namely mean plus variance (95% CI)

**Thompson Sampling (TS) [5]:**  $j^* = \arg \max_j \hat{y}_{ij}$

For each item, sample the noisy affinity from the posterior. Select item with the maximum sampled value.

#### Common bandit strategies

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Setting: Offline initialization + Online updating

- Offline stage: M users interact with N items. For each user, we sample 10 dislikes
- Online stage: Ask 15 questions. Each question is followed by a recommendation.
- Metric: Average precision  $AP@10$ , which is a widely used recommendation metric.

#### Synthetic data:

- Offline learning on generated **N=200 restaurant and M=200 users**. The types of restaurants and users are listed in the table.
- For each offline user, we sample 10 items from their liked category as likes and 10 items from the rest of the categories as dislikes
- Online learning for **60 cold-start users for each type**.

Restaurant types	%
expensive	15%
cheap & spicy	5%
cheap & not-spicy	10%
only cheap	35%
only not-spicy	15%
only spicy	20%

User types	%
Like expensive	20%
Like spicy	15%
Like not-spicy	25%
Like cheap	30%
Like only not-spicy	5%
Like only spicy	5%

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Setting: Offline initialization + Online updating

- Offline stage: M users interact with N items. For each user, we sample 10 dislikes.
- Online stage: Ask 15 questions. Each question is followed by a recommendation.
- Metric: Average precision  $AP@10$ , which is a widely used recommendation metric.

### Real data: collected from restaurant searching logs

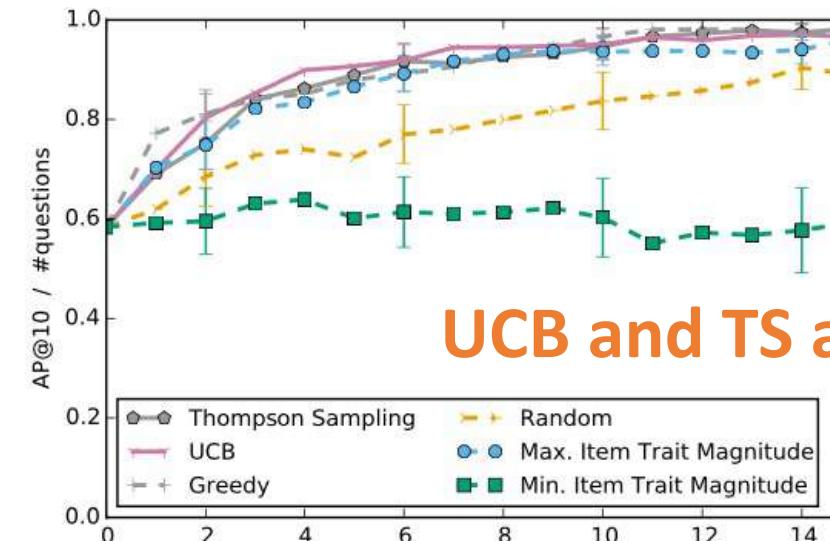
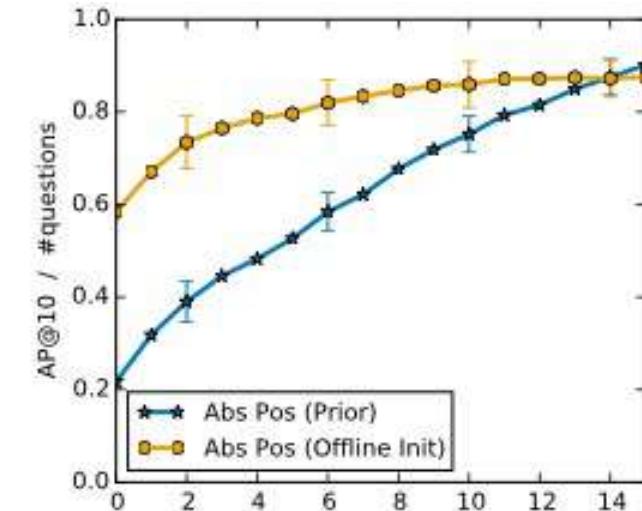
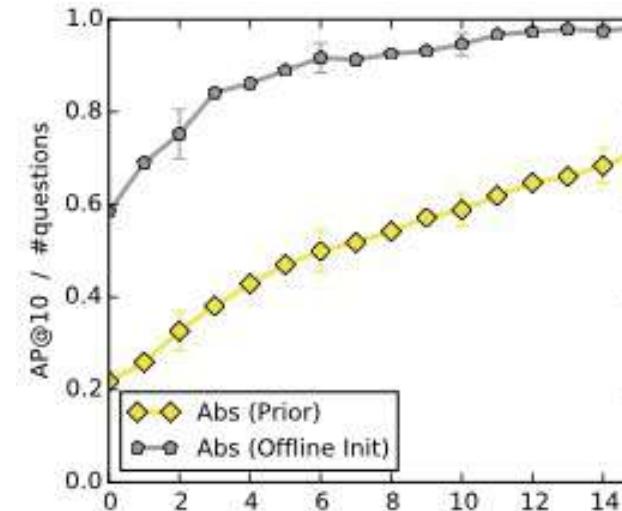
- Offline learning on collected **M = 3549 users, N = 289 restaurants, and 9330 positive observations.**
- Recruit **28 users** to rate on the **selected 10 restaurants**.
- Online cold-start user preference learning: **Sample 50 user based on the 28 ground truth:**
  1. Randomly sample one of the 28 participants.
  2. Observe the sampled user's labels on the pool of 10 restaurants asked in the user study.
  3. Infer user's preference vector  $u_i$
  4. Sample  $\hat{u}_i \sim u_i$ . Set  $\hat{u}_i$  to be the new prior of  $u_i$ .
  5. With this prior, infer the ratings  $r_i$  distribution.
  6. Sample ratings from their distribution  $\hat{r}_i \sim r_i$

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

The offline initialization improve performance

### Conclusion:

- The bandit can help improve model performance.
- Offline initialization brings significant improvement.
- E&E (UCB and Thompson Sampling) methods outperform the trivial Exploit-only and Explore-only methods.



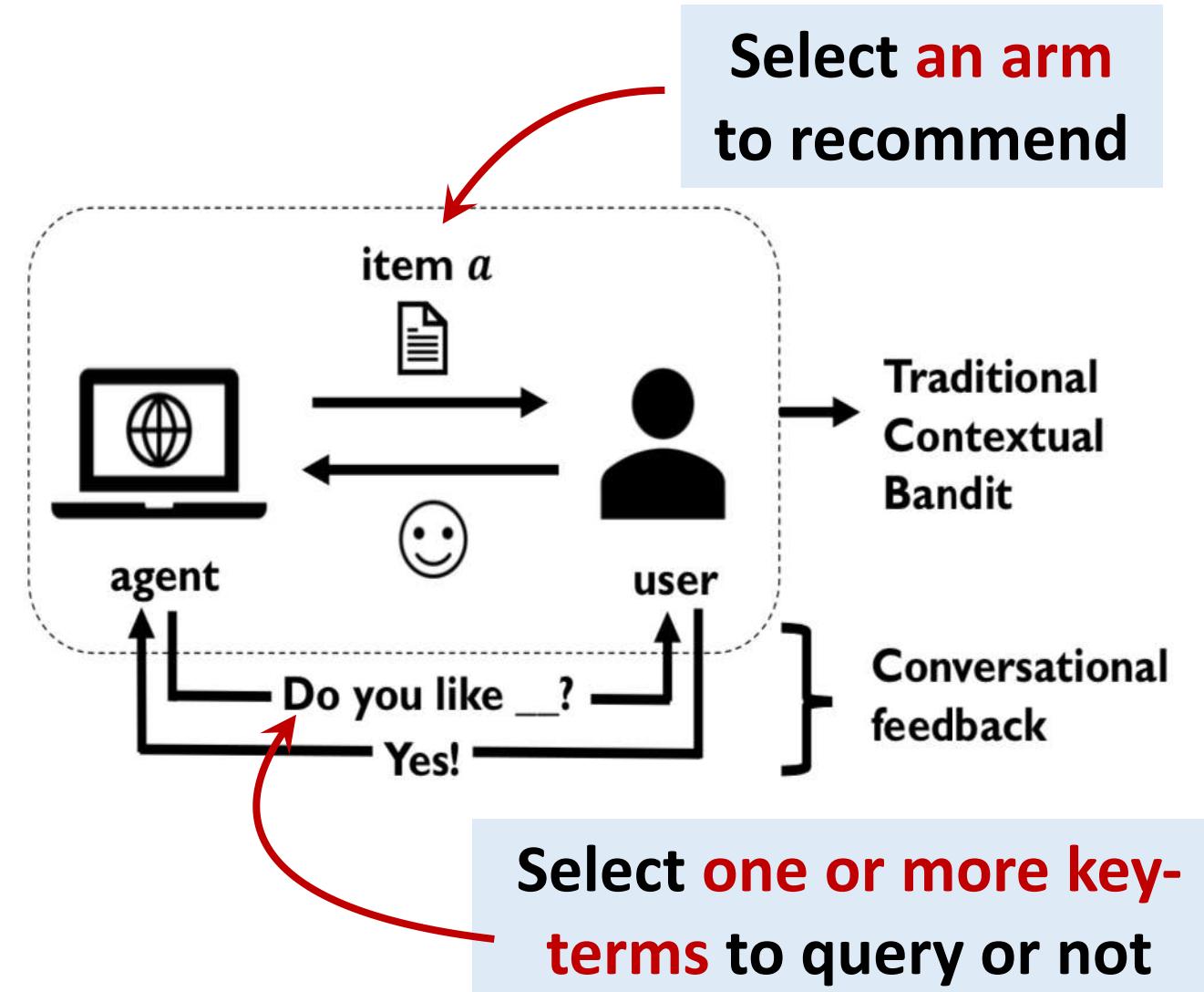
UCB and TS are the best

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### ConUCB Model:

#### Setting:

- Asking questions about not only the **bandit arms (items)**, but also the **key-terms (categories, topics)**.
- One key-term is related to a subset of arms. Users' preference on key-terms can propagate to arms.
- Each arm has its own features.



## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### ConUCB Model:

Select one or more key-terms to query

Select an arm to recommend

---

#### Algorithm 1: General algorithm of ConUCB

---

**Input:** arms  $\mathcal{A}$ , key-terms  $\mathcal{K}$ , graph  $(\mathcal{A}, \mathcal{K}, W)$ ,  $b(t)$ .

- 1 **for**  $t = 1, \dots, T$  **do**
- 2     observe contextual vector  $\mathbf{x}_{a,t}$  of each arm  $a \in \mathcal{A}_t$ ;
- 3     If conversation is allowed at round  $t$ , i.e.,  $q(t) = 1$ , select key-terms to conduct conversations and receive conversational feedbacks  $\{\tilde{r}_{k,t}\}$ ;
- 4     select an arm  $a_t = \arg \max_{a \in \mathcal{A}_t} \tilde{R}_{a,t} + C_{a,t}$ ;
- 5     receive a reward  $r_{a_t,t}$ ;
- 6     update model ;

Exploitation      Exploration

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### When to query the key-terms:

- Define a function  $b(t)$ , which determines:
  - (1) whether to converse at round  $t$ .
  - (2) the number of conversations until round  $t$ .

- Consider the function  $q(t)$ :

$$q(t) = \begin{cases} 1, & b(t) - b(t-1) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

- If  $q(t) = 1$ , query about key-term for  $b(t) - b(t-1)$  times;
- If  $q(t) = 0$ , does not query about a key-term;
- For users' experience, key-term-level conversations should be less frequent than arm-level interactions, i.e.,  $b(t) \leq t, \forall t$ .

### Examples:

- 1) The agent makes  $k$  conversations in every  $m$  rounds.

$$b(t) = k \left\lfloor \frac{t}{m} \right\rfloor, m \geq 1, k \geq 1,$$

- 2) The agent makes a conversation with a frequency represented by the logarithmic function of  $t$ .

$$b(t) = \lfloor \log(t) \rfloor$$

- 3) There is no conversation between the agent and the user.

$$b(t) \equiv 0$$

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

The core strategy to select arms and key-terms:

- Selecting the arm with the largest upper confidence bound derived from both arm-level and key-term-level feedbacks, and receives a reward.

User preference computed on key-term-level rewards

$$\tilde{\theta}_t = \arg \min_{\tilde{\theta}} \sum_{\tau=1}^t \sum_{k \in \mathcal{K}_\tau} \left( \frac{\sum_{a \in \mathcal{A}} w_{a,k} \tilde{\theta}^T \mathbf{x}_{a,\tau}}{\sum_{a \in \mathcal{A}} w_{a,k}} - \tilde{r}_{k,\tau} \right)^2 + \lambda \|\tilde{\theta}\|_2^2,$$

User preference computed on arm-level rewards

$$\theta_t = \arg \min_{\theta} \lambda \sum_{\tau=1}^{t-1} (\theta^T \mathbf{x}_{a_\tau, \tau} - r_{a_\tau, \tau})^2 + (1-\lambda) \|\theta - \tilde{\theta}_t\|_2^2.$$

Constrain  $\theta$  to be close to  $\tilde{\theta}$

The strategy of arm selection is

$$a_t = \arg \max_{a \in \mathcal{A}_t} \underbrace{\mathbf{x}_{a,t}^T \theta_t}_{\tilde{R}_{a,t}} + \underbrace{\lambda \alpha_t \|\mathbf{x}_{a,t}\|_{M_t^{-1}} + (1-\lambda) \tilde{\alpha}_t \|\mathbf{x}_{a,t}^T M_t^{-1}\|_{\tilde{M}_t^{-1}}}_{C_{a,t}}$$

Exploitation

Exploration

where  $M_t$  is the function of  $\theta$  and  $\tilde{\theta}$

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

The core strategy to select arms and key-terms:

- **Selecting the key-terms** that reduce the learning error most, and enquires the user's preference over the key-terms. The natural idea is to select the key-term  $k$  that minimizes the expectation error  $E[||X_t \theta_t - X_t \theta_*||_2^2]$ , where  $\theta_*$  is the unknown ground-truth user preference vector.

$$k = \arg \max_{k'} \left\| \mathbf{X}_t \mathbf{M}_t^{-1} \tilde{\mathbf{M}}_{t-1}^{-1} \tilde{\mathbf{x}}_{k',t} \right\|_2^2 / \left( 1 + \tilde{\mathbf{x}}_{k',t}^T \tilde{\mathbf{M}}_{t-1}^{-1} \tilde{\mathbf{x}}_{k',t} \right)$$

where  $\tilde{\mathbf{x}}_{k,t} = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \mathbf{x}_{a,t}$ .

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Evaluation setting:

- Metric: regret function:

$$E \left[ \sum_{t=1}^T r_{t,a^*} \right] - E \left[ \sum_{t=1}^T r_{t,a} \right],$$

where:

$a$  is the selected arm.

$a^*$  is the true optimal arm.

- Synthetic data generation:

Synthesizing  
features of arms  
and key-terms

- 1) We generate a pseudo feature vector  $\dot{x}_k$  for each key-term  $k$ , where each dimension is drawn independently from a uniform distribution  $U(-1, 1)$
- 2) For each arm  $a$ , we sample  $n_a$  key-terms uniformly at random from  $K$  without replacement as its related key-terms set  $y_a$ .
- 3) Each dimension  $i$  of the feature  $x_a$  is independently drawn from  $N(\sum_{k \in y_a} \frac{\dot{x}_k(i)}{n_a}, \sigma_g^2)$

Synthesizing  
user preferences

- 4) We generate  $N_u$  users, each of whom is associated with a  $d$ -dimensional vector  $\theta_u$ , i.e., the ground-truth of user  $u$ 's preference. Each dimension of  $\theta_u$  is drawn from a uniform distribution  $U(-1, 1)$ .

Synthesizing  
true reward

- 5) The true arm-level reward  $r_{a,t}$  is  $r_{a,t} = x_{a,t}^T \theta + \epsilon_t$ ;  $\epsilon_t$  is the noise sampled from  $N(0, \sigma_g^2)$ .
- 6) The true key-term-reward  $\tilde{r}_{k,t}$  is:  $\mathbb{E}[\tilde{r}_{k,t}] = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \mathbb{E}[r_{a,t}], \quad k \in \mathcal{K}$ .

## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Evaluation setting for real data:

- How to simulate users' ground-truth rewards on unobserved arms?

1. Use interactions of test set as known rewards  $r_{a,t}$
2. Given users' feature  $x_{a,t}$  on an arm  $a$ .
3. Estimate users preferences  $\theta$  using ridge regression:

$$\theta = \arg \min_{\theta} \sum_{t=1}^{|T_a|} (x_{a,t}^T \theta - r_{a,t})^2 + \|\theta\|^2$$

4. Simulate the ground-true arm-level reward  $r_{a,t}$  on unobserved arms by:

$$r_{a,t} = x_{a,t}^T \theta + \epsilon_t$$

5. Simulate the ground-true key-term-level reward  $\tilde{r}_{k,t}$  by:

$$\mathbb{E}[\tilde{r}_{k,t}] = \sum_{a \in \mathcal{A}} \frac{w_{a,k}}{\sum_{a' \in \mathcal{A}} w_{a',k}} \mathbb{E}[r_{a,t}], \quad k \in \mathcal{K}.$$

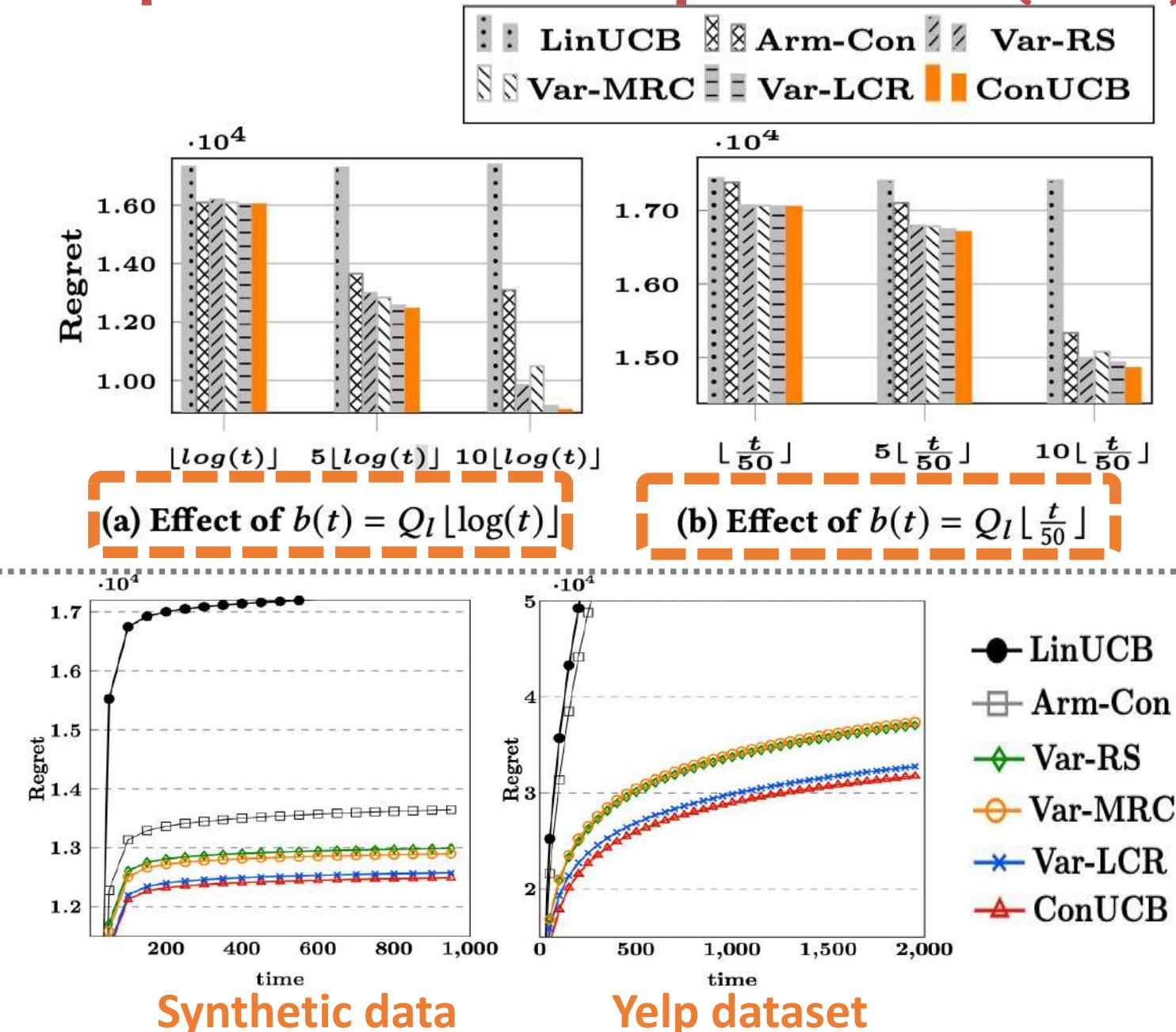
## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### Evaluation result:

- How  $b(t)$  (times of query key-terms) affect the bandit regret:

**The more times of queries,  
the better the performance**

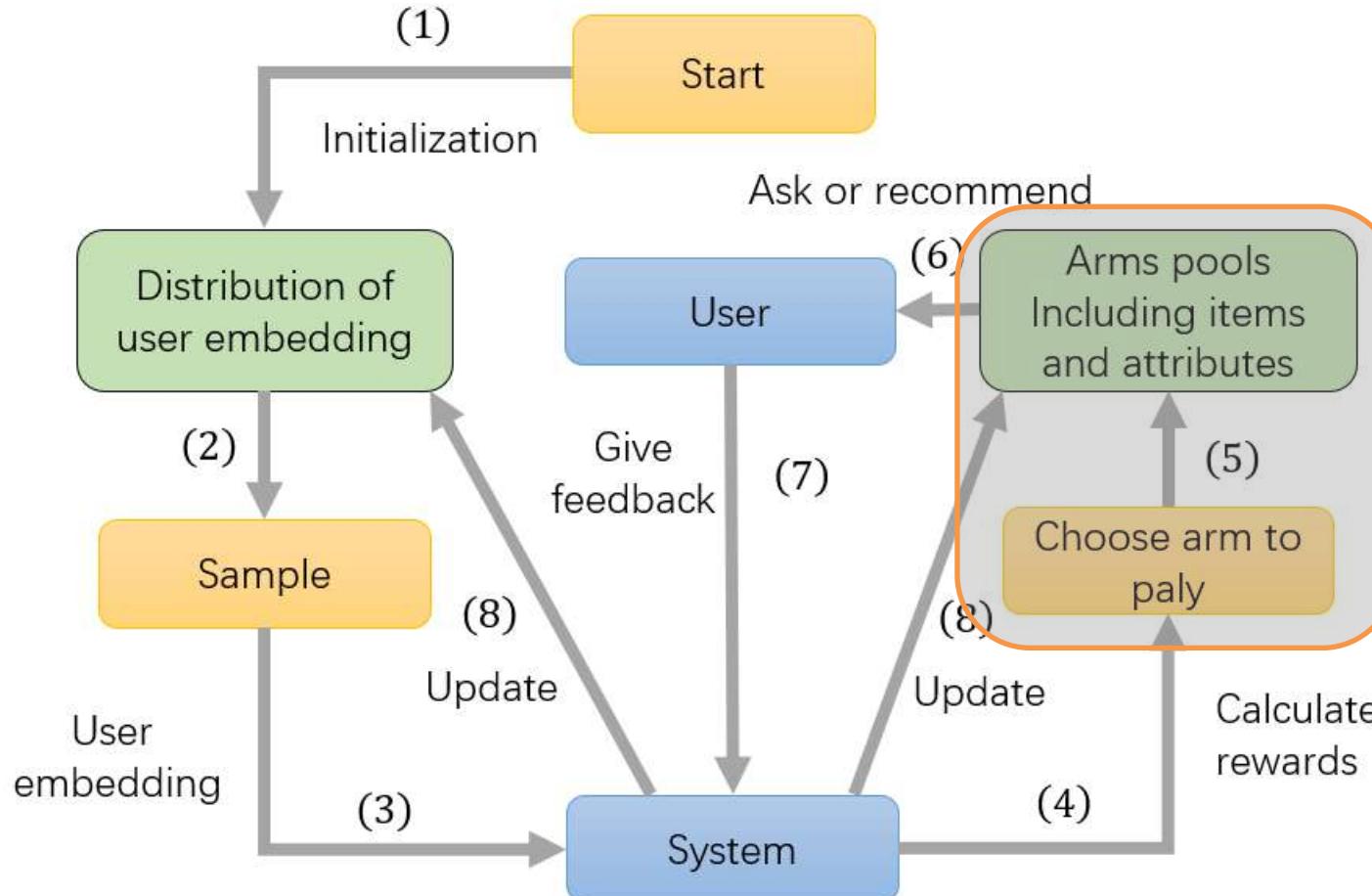
- The performance of different algorithms: **The proposed ConUCB outperforms others in terms of Regret.**



## 2.4 Trade-offs between Exploration and Exploitation (E&E)

ConTS model:

1. Automatically alternate asking questions and making recommendations.
2. Addressed the cold-start user problem.



## 2.4 Trade-offs between Exploration and Exploitation (E&E)

### ConTS model: Core idea

- The expected reward of arm  $a$  (either an item or an attribute) for user  $u$  as:

$$\mathbb{E}[r(a, u, \mathcal{P}_u)] = \mathbf{u}^T \mathbf{x}_a + \sum_{p_i \in \mathcal{P}_u} \mathbf{x}_a^T \mathbf{p}_i,$$

Preference for the item      Preference for the attributes of the item

**Arm Choosing:** selecting the arm with highest reward.

**Indiscriminate arms for items and attributes:**

- If the arm with highest reward is attribute: system asks.
- If the arm with highest reward is item: system recommends top K items.



# Outline

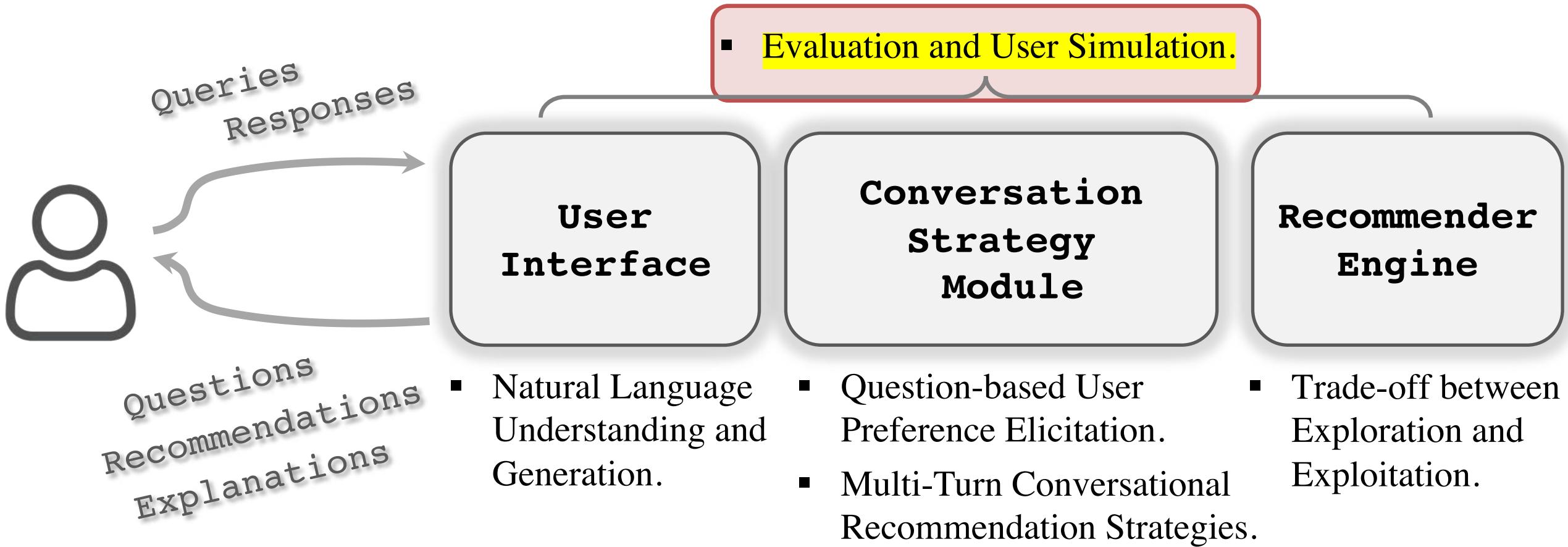
## I. Background

## II. Five Important Challenges

- 2.1 Question-based User Preference Elicitation.
- 2.2 Multi-turn Conversational Recommendation Strategies.
- 2.3 Natural Language Understanding and Generation.
- 2.4 Trade-offs between Exploration and Exploitation (E&E).
- 2.5 Evaluation and User Simulation.

## III. Promising Future Directions

# Outline



## 2.5 Evaluation and User Simulation

- How to evaluate CRSs in terms of **turn-level** performance?
  - How good is the recommendation?
  - How good is the response generation?
- How to evaluate CRSs in terms of **conversation-level** (global) performance?
  - Online test (A/B test) and off-policy evaluation
  - User simulation

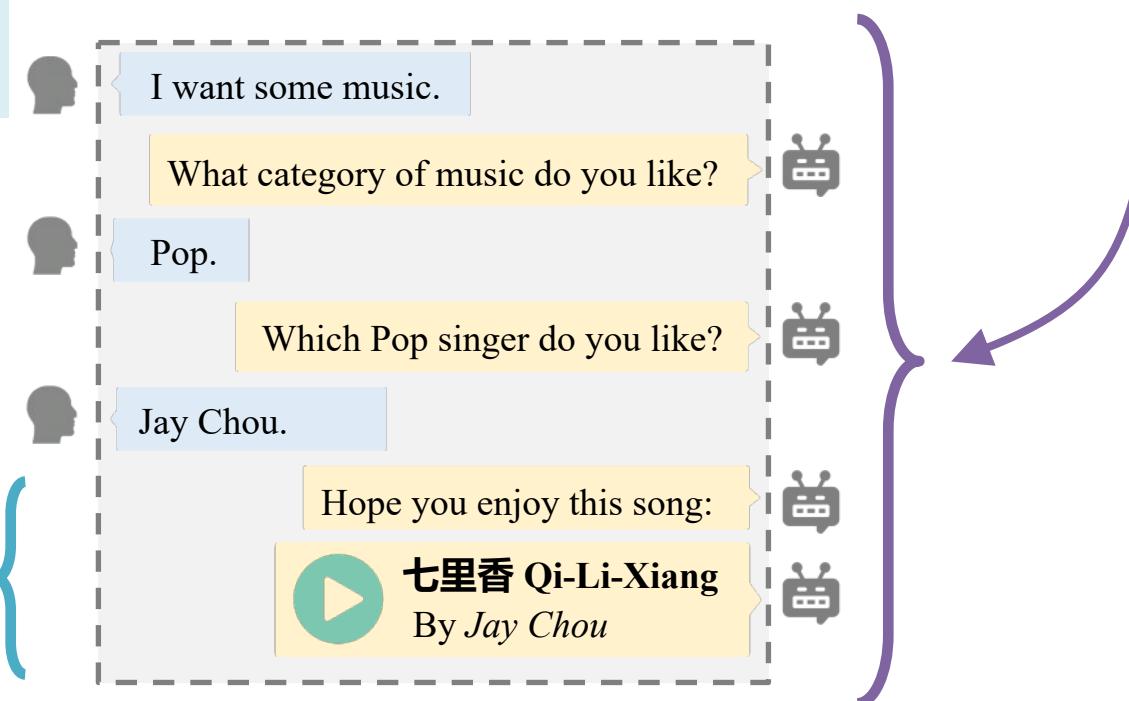
# 2.5 Evaluation and User Simulation

## Two kinds of metrics:

- Turn-level Evaluation
- Evaluation of Recommendation:  
RMSE, MSE, recall, precision,  
F1-score, Hit, NDCG, MAP, MRR
- Evaluation of Dialogue  
Generation: BLEU, Rouge

### □ Conversation-level Evaluation:

- AT (average turn), the lower the better as the system should achieve the goal as soon as possible.
- SR@ $k$  (success rate at  $k$ -th turn), the higher the better.



## 2.5 Evaluation and User Simulation

**Turn-level evaluation: assume we know the ground-truth answer**

- Metric for evaluate recommendation performance
  - **Rating-based metrics:** Mean Squared Error (MSE), Root Mean Squared Error (RMSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}}$$

**Measuring the difference between the actual and predicted answer**

- **Ranking-based metrics:** Hits, Precision, Recall, F1-score, Mean Reciprocal Rank (MRR), Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG)

$$Precision = \frac{|\{True\} \cap \{Predicted\}|}{|\{Predicted\}|}$$

$$Recall = \frac{|\{True\} \cap \{Predicted\}|}{|\{True\}|}$$

**Measuring whether the algorithm ranks items proportional to their relevance**

## 2.5 Evaluation and User Simulation

**Turn-level evaluation: assume we know the ground-truth answer**

□ **Potential problem:**

- Some studies only sample a small set of irrelevant items and calculate the ranking metrics on this small set.
- (*Krichene and Rendle, KDD' 20*) show that measuring the results on the sampling set could be inconsistent with the true ranking results.

□ **Suggestion:** avoiding sampling when measuring.

## 2.5 Evaluation and User Simulation

**Turn-level evaluation: assume we know the ground-truth answer**

- Metric for evaluate the performance of response generation
  - **Traditional metrics:** BLEU, ROUGE, etc.

$$BLEU = \frac{|\{Reference\ words\} \cap \{Generated\ words\}|}{|\{Reference\ words\}|}$$

(Similar to **Precision** in recommendation)

$$ROUGE = \frac{|\{Reference\ words\} \cap \{Generated\ words\}|}{|\{Generated\ words\}|}$$

(Similar to **Recall** in recommendation)

## 2.5 Evaluation and User Simulation

**Turn-level evaluation:** assume we know the ground-truth answer

- **Problem:** sensitive to lexical variation, e.g., “good” and “great”
- **Our goal:** not to predict the response with the highest probability, but rather the long-term success of the dialogue.
- **Specialized Metrics:** fluency, consistency, readability, informativeness, diversity, and empathy.

## 2.5 Evaluation and User Simulation

### Conversation-level evaluation: for long-term gain

#### □ Problems in turn-level evaluation:

- CRS is not a supervised learning task. The answer is not known in advance.
- The interaction process is not i.i.d., but rely on historical actions and user feedback.

#### □ Solution: Using conversation-level evaluation to measure the long-term gain.

## 2.5 Evaluation and User Simulation

### Conversation-level evaluation: for long-term gain

#### ☐ Metrics:

- Average turn (AT) : The smaller the better.
- Success rate at the  $t$ -th turn (SR@ $t$ ): The larger the better.
  - Definition of success: click, watching time.
- Cumulative rewards:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi(\tau)} \left[ \sum_{t=0}^T \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

$\tau$  : A sequence of historical interactions  $((s_1, a_1) \dots (s_t, a_t))$

$\pi(\tau)$  : The probability distribution of trajectory  $\tau$  under policy  $\pi$

$\gamma$  : Discount factor.

$r(s_t, a_t)$ : The mean of estimated reward of arm  $a$ .

## 2.5 Evaluation and User Simulation

### Conversation-level evaluation: for long-term gain

#### Online User Test (A/B test)

Interact with true users and compute the cumulative reward:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi(\tau)} \left[ \sum_{t=0}^T \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

- **Problems: Not practical in reality!**
  1. Too slow and inefficient.
  2. Hurt user experience.

## 2.5 Evaluation and User Simulation

### Conversation-level evaluation: for long-term gain

#### □ Off-policy Evaluation (Counterfactual Evaluation)

what would have happened if instead of  $\pi_\beta$  we would have used  $\pi_\theta$ ?

$\pi_\theta$  is our current target policy,  $\pi_\beta$  is the behavior policy (logging policy) under which we collect historical data.

- Key idea: Using importance sampling or inverse propensity score:

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\beta(\tau)} \left[ \frac{\pi_\theta(\tau)}{\pi_\beta(\tau)} \sum_{t=0}^T \gamma^t r(s_t, a_t) \right]$$

$w(\tau) = \frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}$  is the weight to address the distribution mismatch between  $\pi_\beta$  and  $\pi_\theta$

## 2.5 Evaluation and User Simulation

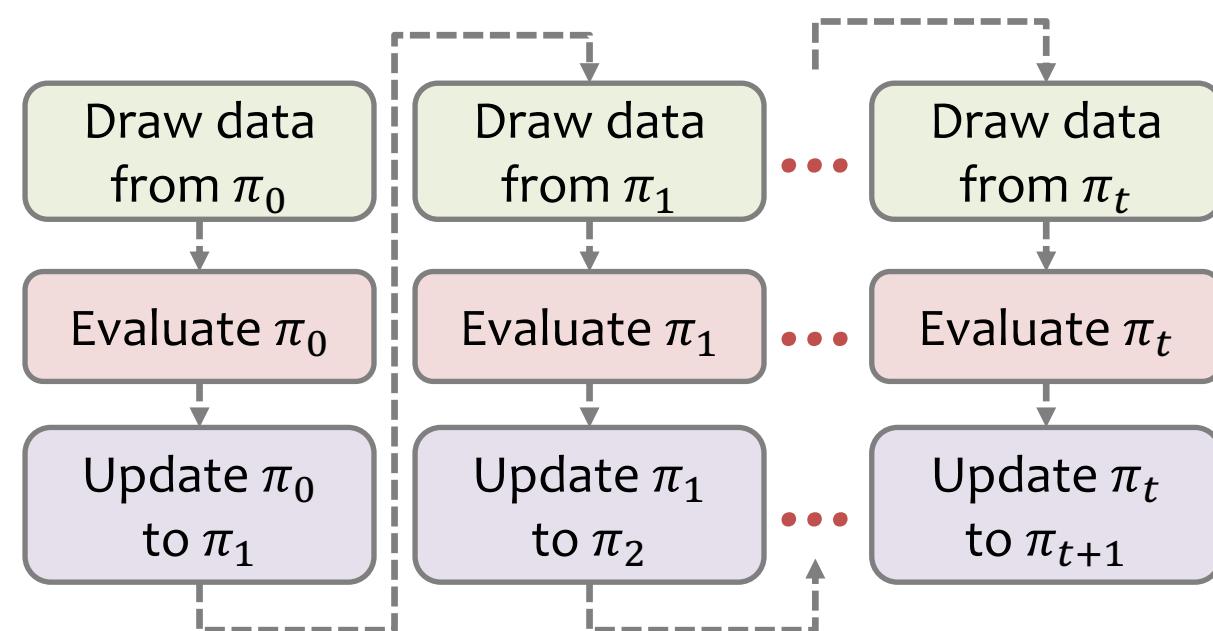
### Conversation-level evaluation: for long-term gain

#### □ Online User Test

$$J(\pi) = \mathbb{E}_{\tau \sim \pi(\tau)} \left[ \sum_{t=0}^T \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

#### □ Off-policy Evaluation

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\beta(\tau)} \left[ \frac{\pi_\theta(\tau)}{\pi_\beta(\tau)} \sum_{t=0}^T \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$



## 2.5 Evaluation and User Simulation

### Conversation-level evaluation: for long-term gain

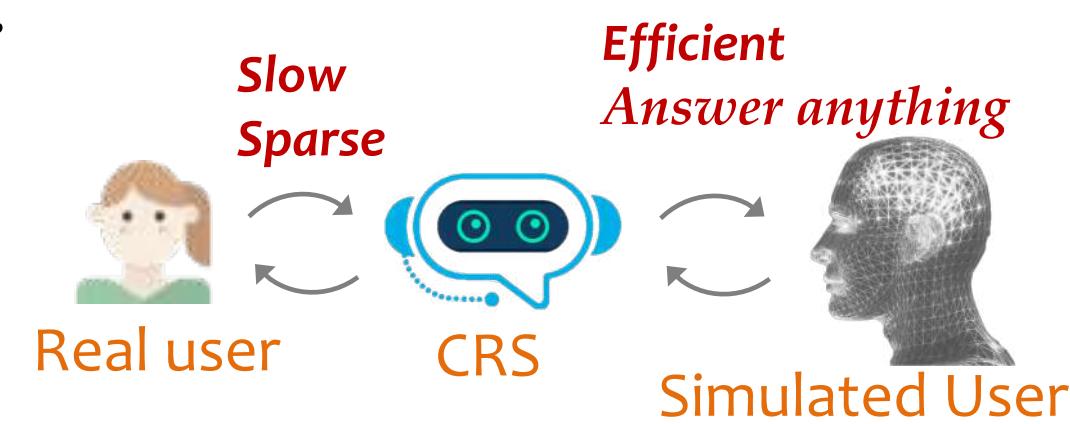
#### □ Off-policy Evaluation:

- **Advantages:**
  - Efficient: using historical data to evaluate current policy
  - Unbiased: using importance sampling
- **Problems:**
  - High variance of the estimator  $w(\tau) = \frac{\pi_\theta(\tau)}{\pi_\beta(\tau)}$
- **Remedy:**
  1. Weight clipping to limit  $w(\tau)$  by an upper bound.
  2. Trusted region policy optimization (TRPO) to bound policy update.

## 2.5 Evaluation and User Simulation

### User Simulation: an intuitive way to evaluate CRS

- Motivation: problems in online evaluation and off-policy evaluation
  - Online evaluation: very slow and expensive.
  - Off-policy evaluation: action space is too large, and historical data is too sparse!
- A natural solution: using simulate users.
  - Efficient.
  - Can answer any question or query.



## 2.5 Evaluation and User Simulation

### 4 kinds of user simulation:

1. Using direct interaction history of users
  - Similar to traditional recommendation.
  - Disadvantage: Very sparse.

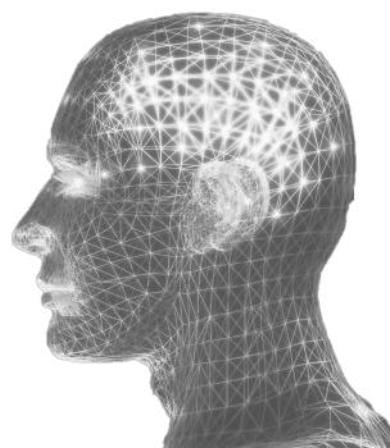
2. Estimating user preferences on all items in advance
  - Solved the missing data problem
  - Disadvantage: May introduce estimating error

3. Extracting from user reviews

- Explicitly mentions attributes, which can reflect the personalized opinions of the user on this item.
- Disadvantage: Hard to distinguish user sentiment

4. Imitating human conversational corpora

- Used in the dialogue system-driven CRSs
- Disadvantage: non-transparent and hard to interpret



## 2.5 Evaluation and User Simulation

### Using direct User Click History:

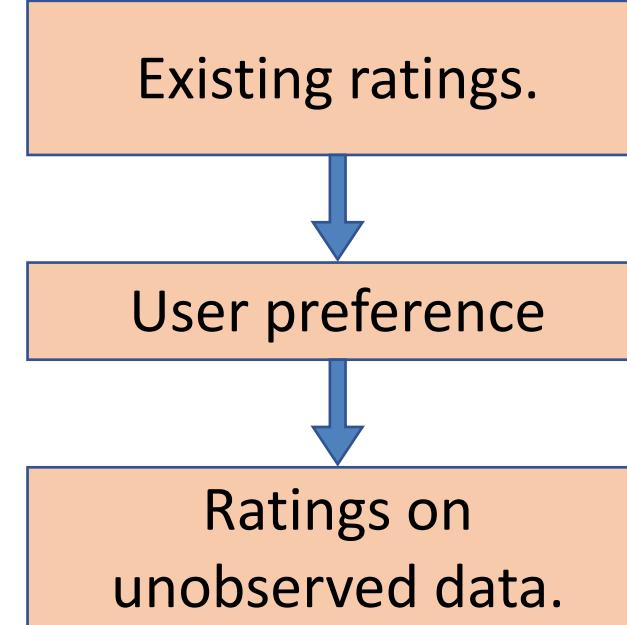
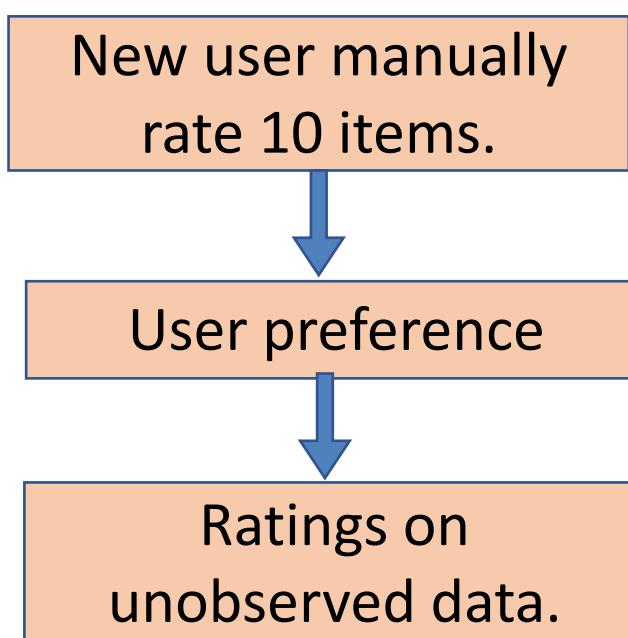
- Observed (user – item) pairs are used as positive samples, unobserved once as negative samples.
- During one conversation session, we sample one (user – item) pair.
  - During this session, the user will only like this item.
  - During this session, the user will only like the attributes of this item.



## 2.5 Evaluation and User Simulation

### Generalize to the Whole Candidate Testing Set

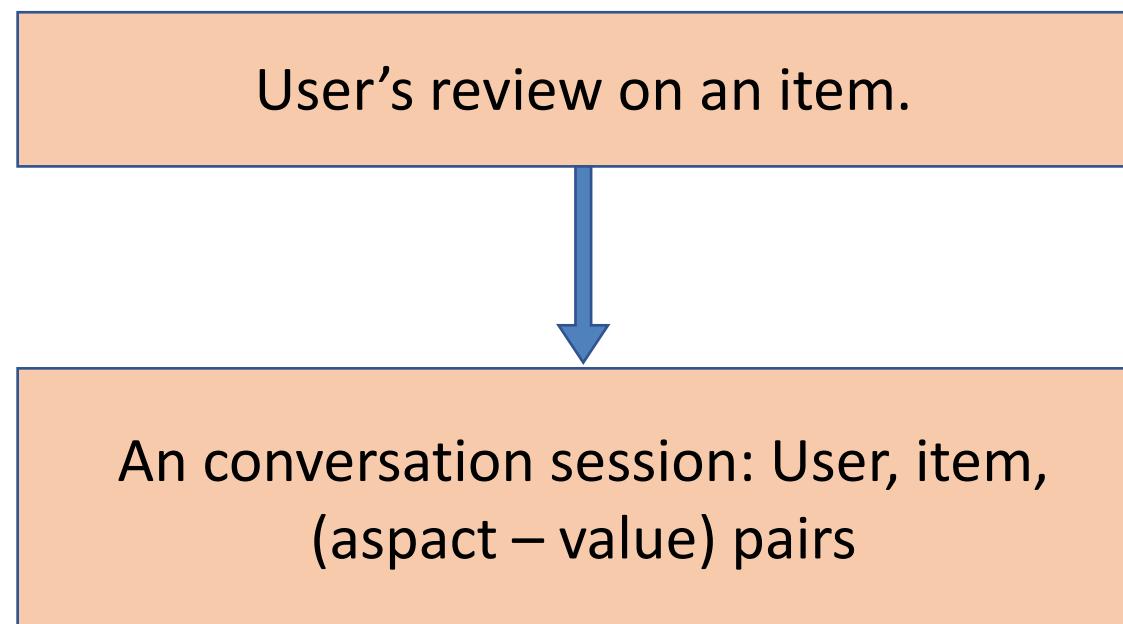
- Get user's ground-truth preference score on a small amount of data.
- Infer user's preference for the full dataset.



## 2.5 Evaluation and User Simulation

### Extract from user review:

- Each review will be used to generate a conversation session.
- “Aspect – Value” pairs would be extracted from the review (e.g. “price” = “high”, ‘OS’ = “Android”).



## 2.5 Evaluation and User Simulation

### Imitating human conversational corpora

**HUMAN:** hello, can i help you find a movie ?

**HRED:** hi what kind of movies do you like

**OURS:** hi, how can i help you ?

**SEEKER:** hello ! i would like to find a suspenseful, but clean family friendly movie .

...

**SEEKER:** ...  
2001 : a space odyssey might be a great option. anything else that you would suggest ?

**HUMAN:** you can try planet of the apes the older one is quite suspenseful and family friendly .

**i.e. corpus**

User actually likes “Star Wars” and dislikes “the planet of the apes”.

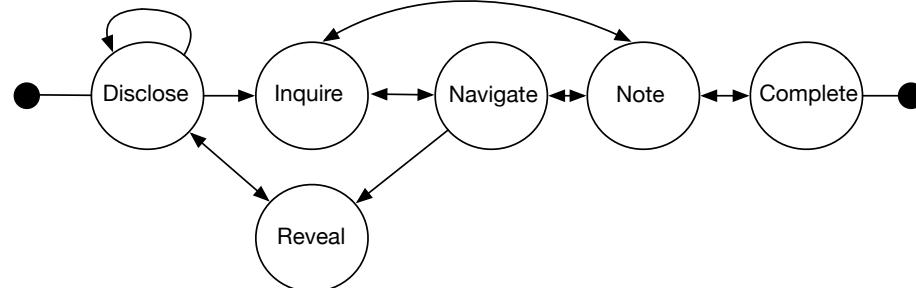
- User’s preference is recorded “as is” in the corpus. The evaluation is actually biased on responses in the corpus (which is often generated on AMTurker).

# 2.5 Evaluation and User Simulation

## Simulated user

CRS

Stack-like simulation strategy



Defined state transition rule

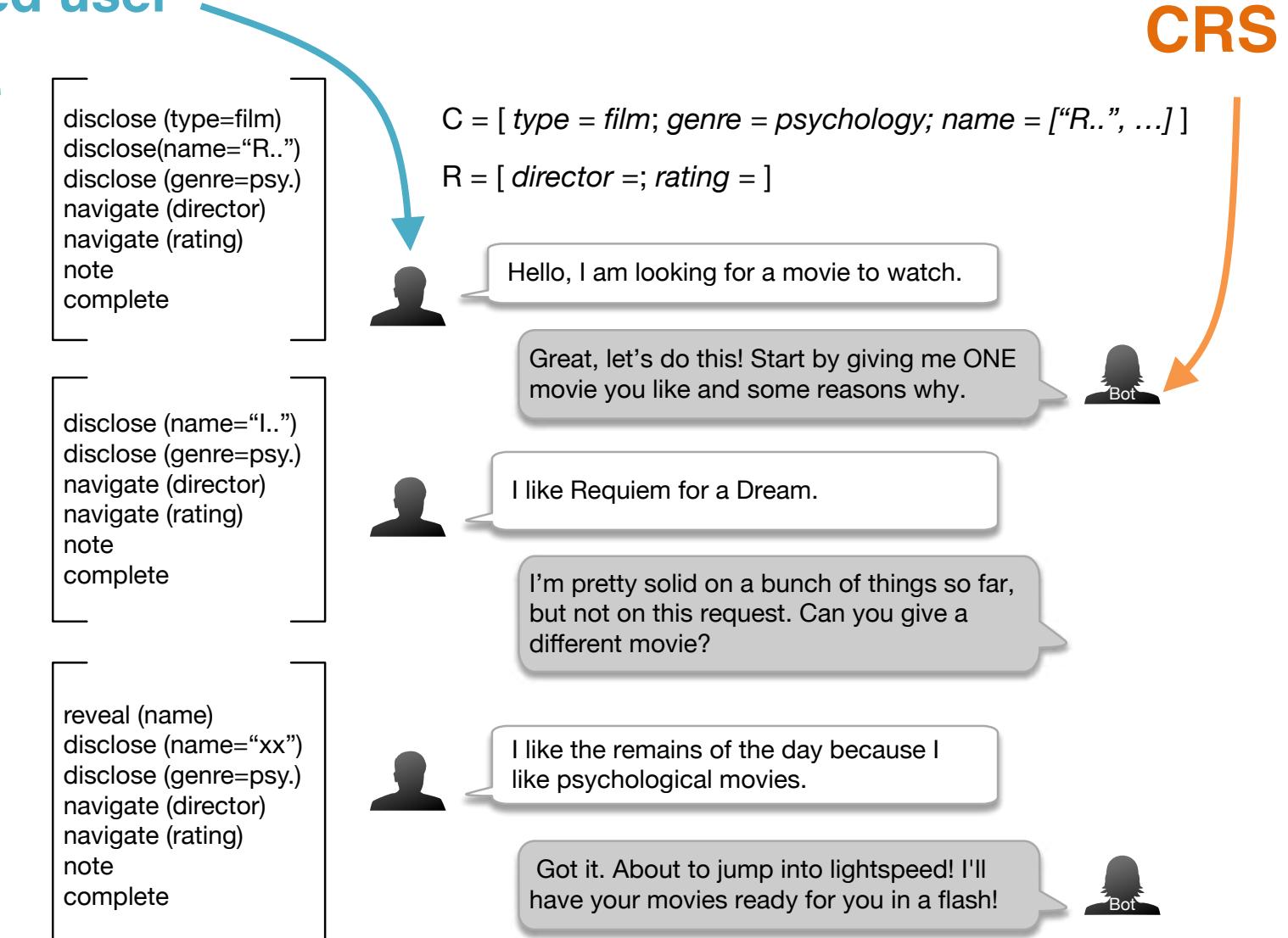


Figure Credit: Shuo Zhang and Krisztian Balog. Evaluating Conversational Recommender Systems via User Simulation. KDD' 20

## 2.5 Evaluation and User Simulation

### Datasets:

Simulated from traditional RS data (without dialogues)

Dataset	#Dialogs	#Turns	Dialogue Type	Domains	Dialogue Resource	Related Work
MovieLens [7]				Movie	From item ratings	[217, 104]
LastFM [7]		Depended on the dialogue simulation process		Music	From item ratings	[87, 69, 104]
Yelp				Restaurant	From item ratings	[88, 89, 104]
Amazon [116]				E-commerce	From item ratings	[161, 88, 210, 47, 104]
TG-ReDial [227]	10,000	129,392	Rec., chitchat	Movie, Multi topics	From item rating, and enhanced by multi topics	[227]
DuRecDial [104]	10,190	155,477	Rec., QA, etc.	Movie, restaurant, etc.	Generated by workers	[104]
Facebook_Rec [41]	1M	6M	Rec.	Movie	From item ratings	[41]
OpenDialKG [123]	15,673	91,209	Rec., chitchat	Movie, Book, Sport, etc.	Generated by workers	[123]
ReDial [94]	10,006	182,150	Rec., chitchat	Movie	Generated by workers	[94, 25, 104]
COOKIE [47]	No given	11,638,418	Rec.	E-commerce	From user activities and item meta data	[47]
MGConvRex [193]	7.6K+	73K	Rec.	Restaurant	Generated by workers	[193]
GoRecDial [76, 111]	9,125	170,904	Rec.	Movie	Generated by workers	[76]
INSPIRED [56]	1,001	35,811	Rec.	Movie	Generated by workers	[56]



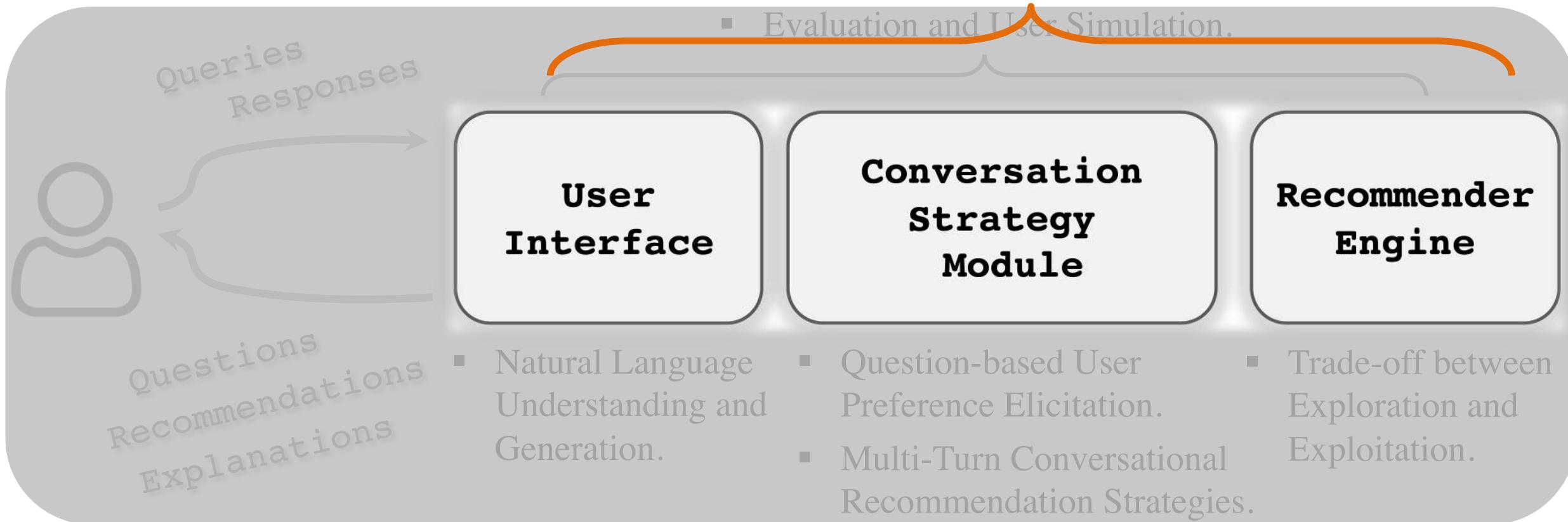
# Outline

I. Introduction

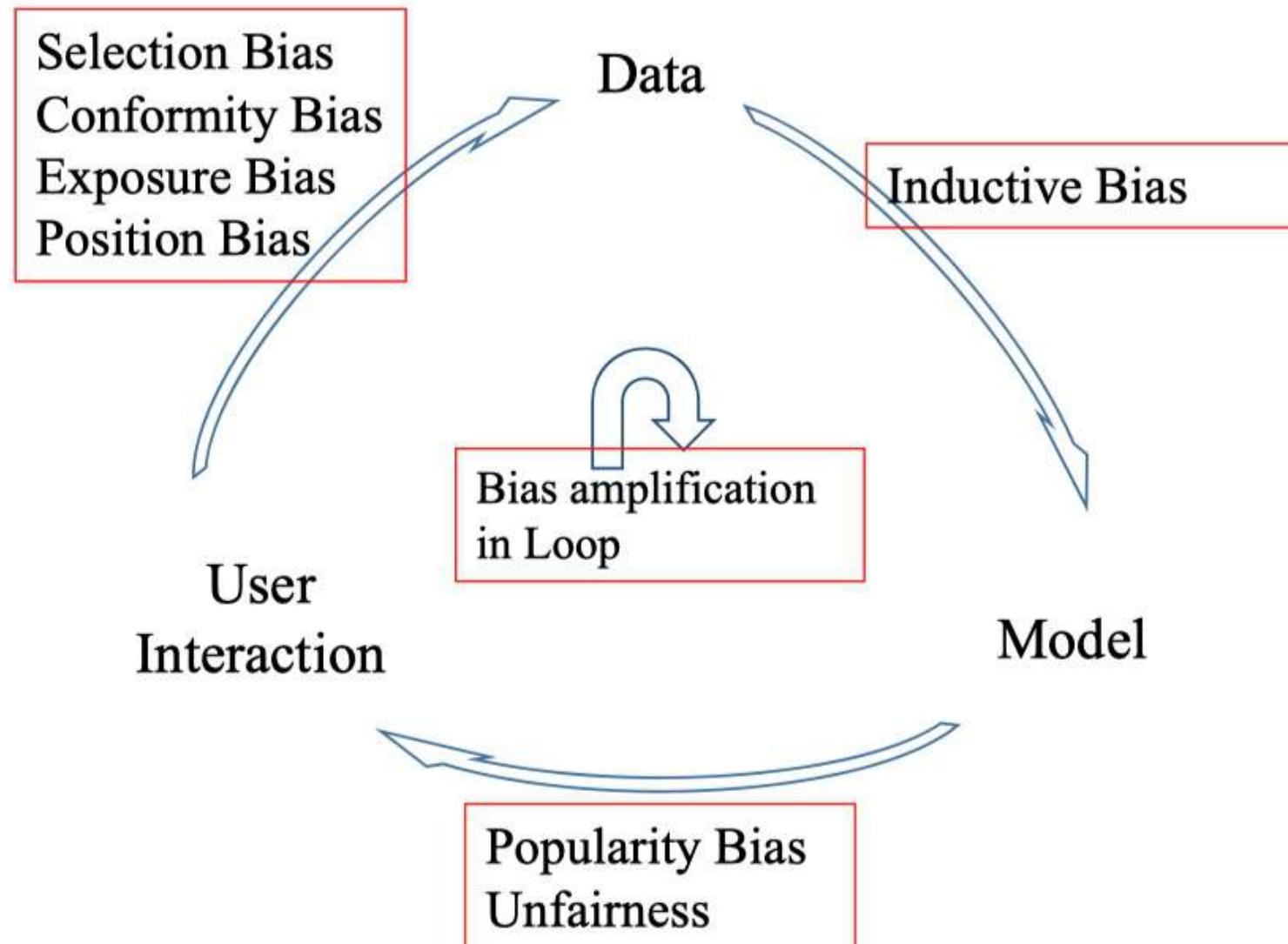
II. Five important challenges

**III. Promising future directions**

# 3.1 Future Directions: Jointly Optimizing Three Tasks



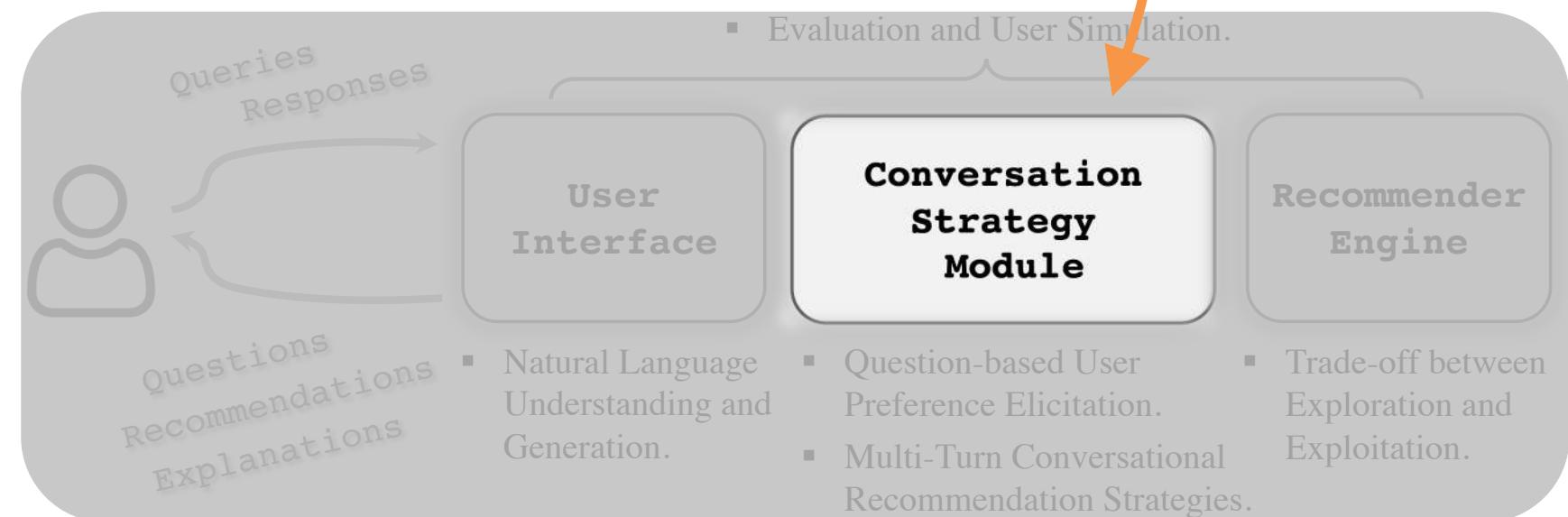
## 3.2 Future direction: Bias and Debiasing in CRSs



**Figure Credit:** Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. *Bias and Debias in Recommender System: A Survey and Future Directions*. arXiv preprint

## 3.3 Future direction: Sophisticated Strategies

- How to handle negative feedback?
- How to handle delayed feedback?
- How to design the reward function based on the feedback?
- How to handle sparse rewards?



## 3.4 Future direction: Knowledge Enrichment

- To import common sense knowledge?
- To import visual, sound modality?

## 3.5 Future direction: Better Evaluation and User Simulation

- How to simulate **reliable** users?

# CRS itself has a promising future!

## Conversational Recommender Systems are

- A promising direction for recommendation systems: solving information asymmetry and dynamic preference problem
- An opportunity to converge cutting-edge techniques to push the development of recommendation: reinforcement learning, natural language processing, explainable AI, conversational AI etc.
- An exemplary step towards the big goal of human-machine collaboration

# Thanks!

Wenqiang Lei

National University of  
Singapore (NUS)

[wenqianglei@gmail.com](mailto:wenqianglei@gmail.com)

Chongming Gao

University of Science and  
Technology of China (USTC)

[chongminggao@mail.ustc.edu.cn](mailto:chongminggao@mail.ustc.edu.cn)

Maarten de Rijke

University of Amsterdam

[m.derijke@uva.nl](mailto:m.derijke@uva.nl)

A literature survey related to this tutorial has been published at <https://arxiv.org/abs/2101.09459>