Conversational Recommendation: Formulation, Methods, and Evaluation

Wenqiang Lei, Xiangnan He, Maarten de Rijke, Tat-Seng Chua
wenqianglei@gmail.com, hexn@ustc.edu.cn, derijke@uva.nl, dcscts@nus.edu.sg

slides will be available at: https://core-tutorial.github.io
A literature survey based on this tutorial as well as other materials will be available soon.
Information explosion problem?
• Information seeking requirements
  ➢ E-commerce (Amazon and Alibaba)
  ➢ Social networking (Facebook and Wechat)
  ➢ Content sharing platforms (Instagram and Pinterest)

Two major types of information seeking techniques

How to handle?

Search

Recommendation
Recommendation has become prevalent in the IR community. Recommendation became the most popular track in SIGIR of different topics. Submitted papers and accepted papers, along with acceptance rates, are shown in the chart for 2019 SIGIR Hot Topics. SIGIR of different topics were received in 2020.
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)
• Existing Static Recommendation: Collaborative Filtering

- Collaborative filtering
  - matrix factorization and factorization machines

- Deep learning approaches
  - neural factorization machines & deep interest networks

- Graph-based approaches
  - expressiveness and explainability of graphs
• Limitation: Information Asymmetry

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

You may like diaper.

I want beer.
• Existing Online Recommendation: Bandit

Exploration and Exploitation Balance

Online Recommendation:
- Arm → Item/Item Category
- Reward → User feedback
- Environment → User

Bandit Algorithm:
- Exploit-Explore problem
- Cold-Start problem

Multi-Armed Bandit
• Limitation: Lack of Explainability

A model still has no channel to know find the exact reason why a user prefer an item.

Figure credit: Spotx
• Conversation Brings Revolution

Conversational Recommender Systems

- Interactive recommendation
- Using natural languages

The example of a conversational recommender system
• Conversational Recommender Systems In a Broader Perspective
• Tag-based Interaction

The example of tag-based interaction on tiktok

The example of tag-based interaction on kuaishou
Conversational Recommendation Bridges Search and Recommendation

Traditional paradigms for information-seeking: 
Search (pull) or Recommendation (push)

Search: 
User's Intention is clear, explicitly indicated by query

Conversational Recommendation: 
Try to induce user preference through conversation!

Recommendation: 
User's Intention is unclear, implicitly revealed in history
• Conversational Recommender Systems

Four Directions being Explored

1. Question Driven Approaches
2. Multi-turn Conversational Recommendation Strategy
3. Exploitation-Exploration Trade-offs for Cold Users
4. Dialogue Understanding and Generation

- Designing suitable evaluation metrics for each component
- Eliciting user preferences by asking questions and maintaining the multi-turn conversation
- Trading off the exploitation-exploration
- Explaining recommendation results
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)
Multi-turn Conversational Recommendation Strategy

A System needs to choose to ask questions and make recommendations in a multi-turn conversation

**Purpose:** making successful recommendations with less turns of interactions

**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

• Exploitation-Exploration Trade-offs for Cold Users

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

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

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

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

Inflexible, constrained

Fail to understand user intent.

Rule/Templet-based

Neural methods

Casual, more natural.

Extract intent from user utterances.

Express actions in generated responses

Fluent and Consistent.
• Tutorial Outline

❑ A Glimpse of Dialogue System

❑ Four research directions in conversational recommendation system
  ❑ Question Driven Approaches
  ❑ Multi-turn Conversational Recommendation Strategy
  ❑ Dialogue Understanding and Generation
  ❑ Exploitation-Exploration Trade-offs for Cold Users

❑ Summary of Formalizations and Evaluations
• Two Types of Dialogue Systems

• Task-oriented Dialogue System
  - Shopping
  - Booking
  - Setting Memo
  - Playing Media

• Non-task-oriented Dialogue System (Chatbot)
  - Chit chat
  - Chit chat

- Chatbot icon
Typical Structure of Task-oriented Dialogue System

- **Natural Language Understanding**
  - “I want to find a Chinese restaurant.”
  - “Where do you want to eat?”

- **Dialog State Tracking**
  - Inform (cuisine="Chinese")
  - Request (location)

- **Dialog Manager**
  - Dialog State
  - Dialog Policy

- **Dialog Policy**

- **Knowledge Base**

Classical pipeline structure

Zhang et al. Recent advances and challenges in task-oriented dialog system (Science China’ 20)

---

**Recommending music**

**User**

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

**System**

**Booking restaurants**

I want to find a Chinese restaurant.

Where do you want to eat?

Near the center of the town.

What price range do you like?

Moderate is ok.

Hope you enjoy this restaurant:

HaiDiLao Hotpot

**Setting alarms**

Remind me this afternoon.

What time do you want me to remind you this afternoon?

Three O’clock

Okay, I will remind you at 15:00.

Today 15:00
• Natural Language Understanding

• Three Purpose:

1. Domain detection

2. Intent detection

3. Slot value extraction

An example utterance with annotations in IOB format

<table>
<thead>
<tr>
<th>W</th>
<th>find</th>
<th>recent</th>
<th>comedies</th>
<th>by</th>
<th>james</th>
<th>cameron</th>
</tr>
</thead>
<tbody>
<tr>
<td>↓</td>
<td></td>
<td>↓</td>
<td>↓</td>
<td>↓</td>
<td></td>
<td>↓</td>
</tr>
<tr>
<td>S</td>
<td>O</td>
<td>B-date</td>
<td>B-genre</td>
<td>O</td>
<td>B-dir</td>
<td>I-dir</td>
</tr>
<tr>
<td>D</td>
<td>movies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>find_movie</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

where:

- **S**: semantic slots.
- **D**: domain.
- **I**: intent.

In IOB format:

- **O**: a token belongs to no chunk.
- **B-**: the beginning of every chunk.
- **I-**: a token inside a chunk

• Dialogue State Tracking

Aiming to track all the states accumulated across the conversational turns

Recent solutions: latent vector-based methods

1. Classification (picklist-based).
2. Copying (generative)

Zhang et al. Find or Classify? Dual Strategy for Slot-Value Predictions on Multi-Domain Dialog State Tracking (Arxiv’19)
• Jointly Solving Natural Language Understanding and Dialogue State Tracking -- Classification

• Using a classifier as dialogue state tracker

• Jointly Solving Natural Language Understanding and Dialogue State Tracking -- Copying

• Find the text span in original utterances.

Lei et al. Sequicity: Simplifying Task-oriented Dialogue Systems with Single Sequence-to-sequence Architectures (ACL’ 18)
• Dialogue Policy
  • Dialogue act in a session are generated sequentially, so it is formulated as a Markov Decision Process (MDP)

  • Can be address by **Supervised Learning** or **Reinforcement Learning**
- Natural Language Generation

- Strategies:
  - Surface realization
  - Conditioned language generation (RNN-based neural network)

- Challenges:
  - Adequacy: meaning equivalence,
  - Fluency: syntactic correctness,
  - Readability: efficacy in context,
  - Variation: different expression.

Semantically-Conditioned Generative Pre-Training (SC-GPT) Model

Peng et al. Few-shot Natural Language Generation for Task-Oriented Dialog (Arxiv’ 20)

[BOS] Let me confirm that you are searching for Hinton hotel in the center area [EOS]
• **Non-task-oriented Dialogue System**

  • **Chit-chat**: casual and non-goal-oriented.
  • **Open domain and open ended**
  • **Challenges:**
    • Coherence
    • Diversity
    • Engagement
    • ...
  • **Ultimate goal**: to pass Turing Test
• Template-based (Rule-based) Solution

- Unscalable: require human labor
- Inflexible: hard to adopt to unseen topic
• Retrieval-based Solution

Assumption:
• A large candidate response set such that all input utterances can get a proper response.
• **Generation-based Solution -- Classical Sequence to Sequence**

**Challenges:**

• **Blandness**
  - Basic models tend to generate generic responses like “I see” and “OK”.

• **Consistency**
  - Logical self-consistent across multiple turns, e.g., persona, sentiment

• **Lack of Knowledge**
  - Typical sequence-to-sequence models only mimic surface level sequence ordering patterns without understanding world knowledges deeply.

*Wu et al. Deep Chit-Chat: Deep Learning for ChatBots (EMNLP’ 18)*
• **Blandness**: VAE-based solution

• **Problem in chatbot**:
  • The lack of diversity: often generate dull and generic response.

• **Solution**:
  • Using latent variables to learn a distribution over potential conversation actions.
  • Using **Conditional Variational Autoencoders (CVAE)** to infer the latent variable.

---

• Consistency: Persona chat

• Motivation:
  • The lack of a consistent personality
  • A tendency to produce non-specific answers like “I don’t know”

• Solution: endowing machines with a configurable and consistent persona (profile), making chats condition on:
  1. The machine’ own given profile information.
  2. Information about the person the machine is talking to.

Persona of two interlocutors

<table>
<thead>
<tr>
<th>Persona 1</th>
<th>Persona 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>I like to ski</td>
<td>I am an artist</td>
</tr>
<tr>
<td>My wife does not like me anymore</td>
<td>I have four children</td>
</tr>
<tr>
<td>I have went to Mexico 4 times this year</td>
<td>I recently got a cat</td>
</tr>
<tr>
<td>I hate Mexican food</td>
<td>I enjoy walking for exercise</td>
</tr>
<tr>
<td>I like to eat cheetos</td>
<td>I love watching Game of Thrones</td>
</tr>
</tbody>
</table>

Wu et al. “Personalizing Dialogue Agents: I have a dog, do you have pets too?”(EMNLP’ 18)
• Lack of background knowledge: Knowledge grounded dialogue response generation -- Text

• **Solution:** Knowledge retrieval from texts (e.g., Wikipedia) into dialogue responses

*Knowledge retrieval module*

Response generated by integrating knowledge

*Two-Stage*

Transformer Encoder 1

Transformer Decoder

Dialogue Response

Dialogue + knowledge

End-to-End

$L_{\text{NLL}}$

$L_{\text{knowledge}}$

Transformer Encoder 2

Attention

encoded independently

IR System

Knowledge

Dialog Context

Memory

Dinan et al. “Wizard of Wikipedia: Knowledge-Powered Conversational agents” (ICLR’ 19)
• Lack of background knowledge: Knowledge grounded dialogue response generation -- KG

• Solution: **Walking** within a large knowledge graph to
  • track dialogue states.
  • to guide dialogue planning

Blue arrow: walkable paths led to engaging dialogues
Orange arrow: non-ideal paths that never mentioned (Should be pruned)

Moon et al. “OpenDialKG: Explainable Conversational Reasoning with Attention-based Walks over Knowledge Graphs” (ACL’ 19)
• Tutorial Outline

- A Glimpse of Dialogue System
- Four research directions in conversational recommendation system
  - Question Driven Approaches
    - Multi-turn Conversational Recommendation Strategy
    - Dialogue Understanding and Generation
    - Exploitation-Exploration Trade-offs for Cold Users
- Summary of Formalizations and Evaluations
• System Ask – User Respond (SAUR) - Formalization

Research Question -- Given the requests specified in dialogues, the system needs to predict:
1. What questions to ask
2. What items to recommend

Figure 1: Example for conversational search in e-commerce product search or recommendation scenario (best in color).

SAUR – Method -- Representation

- Apply a gated recurrent unit (GRU) on the text description of each item
- The hidden states of each sentence as the representation of \( T_j(s_1, s_2 \ldots s_T) \)

Also a gated recurrent unit (GRU)
- Query sequence \( c_1, c_2 \ldots \) is extracted in conversations
• SAUR - Method

Joint optimize

The Unified Architecture

Question Loss + Search (item) Loss

Figure 3: The Multi-Memory Network (MMN) architecture for conversational search and recommendation, including five components: query and item representations, the memory module, as well as the question and search modules.

• SAUR - Evaluation

Evaluation Criteria:
1. Query prediction
2. Item prediction (e.g., NDCG)

User’s review

Mobile phone
Operating system is Android
Screen size larger than five inches
Storage capacity at least 64G
Brand is Samsung or Huawei
Price within 700 dollars

Top category

Query Representation
I want to find a towel for a bath?

Are you seeking for a cotton related item?

- Yes!

Are you seeking for a beach towel related item?

- No.

Are you seeking for a bathroom towel related item?

- Yes!

The recommendation list:

- Towel A
- Towel B

Question number is determined by pre-set parameters $N_q$.

Making Recommendation only once after $N_q$ questions.

Zou et al. “Towards Question-based Recommender Systems” (SIGIR’ 20)
Qrec - Method -- Offline and Online Optimization

Latent Factor Recommendation

\( Y_{ij} \): the number of attributes of item \( j \) that satisfy the user \( i \) in current conversation. For example, if a user specify [cotton] and [towel], but the item only gets [cotton], \( Y=1 \).

\[
\alpha \log p(U, V | R, Y, \Theta) - \frac{1}{2} \sum_{i,j \in R} (R_{ij} - p^T(u_i \circ v_j))^2 + \frac{Y}{2} \sum_{i,j \in Y} (Y_{ij} - q^T(u_i \circ v_j))^2 + \\
\sum_{i=1}^{M} \frac{\lambda_u}{2} \|u_i\|^2 + \sum_{j=1}^{N} \frac{\lambda_v}{2} \|v_j\|^2 + \frac{\lambda_p}{2} \|p\|^2 + \frac{\lambda_q}{2} \|q\|^2.
\]

Offline Optimization

Online Optimization (feedback from user, (i.e. \( Y \)))

Figure 1: Framework of our proposed question-based recommendation model, Qrec. Cotton is an extracted entity (informative term), \( U, V, p, q \) are model variables, and \( \alpha \) is a hyper-parameter of user belief.

Zou et al. “Towards Question-based Recommender Systems” (SIGIR’ 20)
Qrec - Method -- Choosing Questions to Ask

Attribute Choosing criteria: Finding the most uncertain [attribute] to ask.

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

- The smaller the preference confidence indicate the more uncertain attribute.

\[ \text{score}_i \propto (UV^T, Y_i) \]

Zou et al. “Towards Question-based Recommender Systems” (SIGIR’ 20)
• Qrec - Evaluation

Evaluation Measures: recall@5, MRR, NDCG only on items! No questions are evaluated, but if question asking strategy is bad, the item recommendation results will not be good.

Dataset: Amazon product dataset

- Using TAGME (an entity linking tool) to find the entities in the product description page as the attributes.

Item Name: “Cotton Hotel spa Bathroom Towel”
Item Attributes: [cotton, bathroom, hand towels]

Zou et al. “Towards Question-based Recommender Systems” (SIGIR’20)
● Question & Recommendation (Q&R) - Formalization

Positive-only type of feedback (click topics)

User is prompted to choose as many topics as they like

Only asking question once and make one recommendation

Incorporates the user feedback to improve video recommendations

Christakopoulou et al. “Q&R: A Two-Stage Approach toward Interactive Recommendation” (KDD’ 18)
**Q&R - Method**

Two Main Tasks

- **What to ask**
- **How to respond**

---

**building better user profiles**

i.e., predicting the sequential future (interested topic)

**given the video (user interests)**

i.e., predicting the video that the user be most interested in

---

**the sequence of watch videos**

---

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

*Christakopoulou et al. “Q&R: A Two-Stage Approach toward Interactive Recommendation” (KDD’ 18)*
• Q&R - Evaluation

**Offline Evaluation**

**Data**

**YouTube user watch sequences**
1. The *watch sequence* of a user up until the previous to last step
2. The *video ID* and *topic ID* of the user’s last watch event

- watched video id (until t)
- watched video topic id (until t)
- feature context (until t)
- video topic id (t+1)
- Target video id (t+1)

**Online Evaluation**

**Figure 2: User Onboarding UI.**

Christakopoulou et al. “Q&R: A Two-Stage Approach toward Interactive Recommendation” (KDD’18)
- Tutorial Outline

- A Glimpse of Dialogue System
- Four research directions in conversational recommendation system
  - Question Driven Approaches
  - Multi-turn Conversational Recommendation Strategy
  - Dialogue Understanding and Generation
  - Exploitation-Exploration Trade-offs for Cold Users
- Summary of Formalizations and Evaluations
• CRM - Formalization

Scenario: single round of a conversation between a user and the system

Figure 1: The conversational recommender system overview

Sun et al. “Conversational Recommender System” (SIGIR’ 18)
• CRM - Method -- Dialogue Component

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.

Belief Tracker

• Input: the current and the past user utterances representation $Z_t$

• Output: a probability distribution of facets

$$s_t = f_1 \oplus f_2 \oplus ... \oplus f_i$$

vector representation for the facet $i$

the agent’s current belief of the dialogue state

Sun et al. “Conversational Recommender System” (SIGIR’ 18)
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.

Sun et al. “Conversational Recommender System” (SIGIR’ 18)
**• CRM - Method**

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.

---

**State:** $s_t = \{f_1 \oplus f_2 \ldots \oplus f_l\}$. Description of the conversation context.

**Action:**
- $\{a_1, a_2, \ldots, a_l\}$, request the value of a facet
- $a_{rec}$, 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.

---

*Sun et al. “Conversational Recommender System” (SIGIR’ 18)*
• CRM - Evaluation

User Simulation

Yelp (the restaurants and food data)

I’m looking for **Italian** food in **San Diego**.

Which **state** are you in?

I’m in **California**. (state="CA")

Which **price** range do you like?

**Low price** (price_range="cheap")

What **rating** range do you want?

**3.5 or higher**. (rating_range>="3.5")

Do you want “**Small Italy Restaurant**”? thank you!

*Item Name: “Small Italy Restaurant”  
*Item Attributes: [Italian, San Diego, California, cheap, rating>=3.5]*

**Evaluation Metrics**

**Evaluation Matrices:**
- **SR @ k** (Success rate at k-th turn)
- **AT** (Average Turns)

\[
SR = \frac{\text{#successful dialogues}}{\text{#dialogues}} \cdot 100\% \\
AT = \frac{\text{dialogue length}}{\text{dialogue length}}
\]

*Sun et al. “Conversational Recommender System” (SIGIR’ 18)*
Objective:
Recommend desired items to user in shortest turns

Workflow of Multi-round Conversational Recommendation (MCR)

Key Research Questions

1. What item/attribute to recommend/ask?
2. Strategy to ask and recommend?
3. How to adapt to user's online feedback?

Method: Attribute-aware FM for Item Prediction and Attribute Preference Prediction

\[
\hat{y}(u, v, P_u) = u^T v + \sum_{p_i \in P_u} v^T p_i
\]

Score function for item prediction

\[
L_{item} = \sum_{(u,v,v') \in D_1} - \ln \sigma (\hat{y}(u, v, P_u) - \hat{y}(u, v', P_u))
+ \sum_{(u,v,v') \in D_2} - \ln \sigma (\hat{y}(u, v, P_u) - \hat{y}(u, v', P_u))
+ \lambda \|\Theta\|^2
\]

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

---

• EAR - Method -- What Item to Recommend and What Attribute to Ask

Method: Attribute-aware FM for Item Prediction and Attribute Preference Prediction

\[
\hat{g}(p|u, P_u) = u^T p + \sum_{p_l \in P_u} p_l^T p_l
\]

Score function for attribute preference prediction

\[
L_{\text{attr}} = \sum_{(u,p,p') \in D_3} - \ln \sigma(\hat{g}(p|u, P_u) - \hat{g}(p'|u, P_u)) + \lambda_\Theta ||\Theta||^2
\]

\[
L = L_{\text{item}} + L_{\text{attr}}
\]

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

---

**• EAR - Method -- Action stage**

**Method:** Strategy to Ask and Recommend? *(Action Stage)*

We use reinforcement learning to find the best strategy.
- policy gradient method
- simple policy network (2-layer feedforward network)

<table>
<thead>
<tr>
<th>State Vector</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>s_{entropy}</em>: The entropy of attribute is important.</td>
<td>( r_{success} ): Give the agent a big reward when it successfully recommend!</td>
</tr>
<tr>
<td><em>s_{preference}</em>: User's preference on each attribute.</td>
<td>( r_{ask} ): Give the agent a small reward when it ask a correct attribute.</td>
</tr>
<tr>
<td><em>s_{history}</em>: Conversation history is important.</td>
<td>( r_{quit} ): Give the agent a big negative reward when the user quit (the conversation is too long)</td>
</tr>
<tr>
<td><em>s_{length}</em>: Candidate item list length.</td>
<td>( r_{prevent} ): Give each turn a relatively small reward to prevent the conversation goes too long.</td>
</tr>
</tbody>
</table>

Note: 3 of the 4 information come from Recommender Part

Action Space: \( |\mathcal{P}| + 1 \)

---

Solution: We treat the recently rejected 10 items as negative samples to retrain the recommender, to adjust the estimation of user preference.

\[ L_{\text{ref}} = \sum_{(u, v, v') \in D_4} -\ln \sigma \left( \hat{y}(u, v, P_u) - \hat{y}(u, v', P_u)\right) + \lambda_\theta \|\Theta\|^2 \]

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v^t )</td>
<td>Recently rejected item set.</td>
</tr>
<tr>
<td>( D_4 := {(u, v, v')</td>
<td>v' \in V_u^+ \land v' \in v^t} )</td>
</tr>
</tbody>
</table>

• EAR - Evaluation

Table 1: Dataset statistics.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#users</th>
<th>#items</th>
<th>#interactions</th>
<th>#attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yelp</td>
<td>27,675</td>
<td>70,311</td>
<td>1,368,606</td>
<td>590</td>
</tr>
<tr>
<td>LastFM</td>
<td>1,801</td>
<td>7,432</td>
<td>76,693</td>
<td>33</td>
</tr>
</tbody>
</table>

Item Name: “Small Italy Restaurant”
Item Attributes: [Pizza, Nightlife, Wine, Jazz]

I'd like some **Italian** food.

Got you, do you like some **pizza**?

Yes!

Got you, do you like some **nightlife**?

Yes!

**Do you want “Small Paris”?**

Rejected!

Got you, do you like some **Rock Music**?

No!

**Do you want “Small Italy Restaurant”?**

Accepted!

---

Hi! I'm looking for a dance music artist.

Do you like rock music?

Yes! I like it!

Do you like pop music?

Yes! I like it!

You may like music artist Michael Jackson!

Yes! Thank you!

Lei et al. “Interactive Path Reasoning on Graph for Conversational Recommendation” (KDD ‘20)
CPR Framework

- **Assuming**
  - Current path \( P = p_0, p_1, p_2, \ldots, p_t \)
  - \( u \): user  \( v \): item  \( p \): attribute
  - \( \mathcal{P}_u \): user’s preferred attributes
  - \( \mathcal{V}_{cand} \): candidate items

- **Reasoning**
  - Score items to recommend (v message):
    \[
    s_v = f(v, u, \mathcal{P}_u)
    \]
  - Score attribute to ask (p message):
    \[
    s_p = g(u, p, \mathcal{V}_{cand})
    \]

- **Consultation**
  - Policy network (choose to ask or rec)

- **Transition**
  - Extended path
    \[
    P = p_0, p_1, p_2 \ldots p_t, p_{t+1}
    \]
  - Update candidate item /attribute set \( (\mathcal{V}_{cand}/\mathcal{P}_{cand}) \)

---

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.

Lei et al. “Interactive Path Reasoning on Graph for Conversational Recommendation” (KDD’20)
• CPR - Method

- An instantiation of CPR Framework

- Message propagation from attributes to items
  - Factorization Machine in EAR
  - Item prediction
    \[ f(v, u, P_u) = u^T v + \sum_{p \in P_u} v^T p, \]
  - Optimization:
    Bayesian Personalized Ranking

- Message propagation from items to attributes
  - Information entropy strategy
    Weighted attribute information entropy
    \[ g(u, p, V_{cand}) = -\text{prob}(p) \cdot \log_2(\text{prob}(p)), \]
    \[ \text{prob}(p) = \frac{\sum_{v \in V_{cand} \cap V_u} \sigma(s_v)}{\sum_{v \in V_{cand}} \sigma(s_v)} \]

- \( u \): user
- \( v \): item
- \( p \): attribute
- \( P_u \): user’s preferred attributes
- \( L_{item} \): item prediction loss
- \( L_{atti} \): attribute prediction loss

Lei et al. “Interactive Path Reasoning on Graph for Conversational Recommendation” (KDD ’20)
• CPR - Method

Input

\[ S_{\text{his}}: \text{encodes the conversation history} \]
\[ S_{\text{len}}: \text{encodes the size of candidate items} \]

Output

\[ Q(s, a) \]
\[ Q(s, a): \text{the value of action a in state s} \]
\[ a_{\text{rec}}: \text{the action of recommendation} \]
\[ a_{\text{ask}}: \text{the action of asking attribute} \]

DQN method

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

TD loss: \[ \delta = Q(s, a) - \left( R + \gamma \max_a Q(s', a) \right) \]
CPR can make the reasoning process explainable and easy-to-interpret!

Sample conversations generated by SCPR (left) and EAR (right) and their illustrations on the graph (middle).

Lei et al. “Interactive Path Reasoning on Graph for Conversational Recommendation” (KDD ’20)
• Tutorial Outline

- A Glimpse of Dialogue System
- Four research directions in conversational recommendation system
  - Question Driven Approaches
  - Multi-turn Conversational Recommendation Strategy
  - Dialogue Understanding and Generation
  - Exploitation-Exploration Trade-offs for Cold Users
- Summary of Formalizations and Evaluations
• ReDial - Formalization

Conversational recommendation through natural language (in movie domain)

- **Seeker**: explain what kind of movie he/she likes, and asks for movie suggestions

- **Recommender**: understand the seeker’s movie tastes, and recommends movies

---

**Li et al.** “Towards Deep Conversational Recommendations” (NIPS’18)
• ReDial – Formalization -- Dataset Collection

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># conversations</td>
<td>10006</td>
</tr>
<tr>
<td># utterances</td>
<td>182150</td>
</tr>
<tr>
<td># users</td>
<td>956</td>
</tr>
<tr>
<td># movie mentions</td>
<td>51699</td>
</tr>
</tbody>
</table>

**seeker mentioned**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>seeker mentioned</td>
<td>16278</td>
</tr>
<tr>
<td>recommender suggested</td>
<td>35421</td>
</tr>
</tbody>
</table>

**not seen**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>not seen</td>
<td>16516</td>
</tr>
<tr>
<td>seen</td>
<td>31694</td>
</tr>
<tr>
<td>did not say</td>
<td>3489</td>
</tr>
</tbody>
</table>

**disliked (4.9%)**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>disliked</td>
<td>2556</td>
</tr>
<tr>
<td>liked (81%)</td>
<td>41998</td>
</tr>
<tr>
<td>did not say (14%)</td>
<td>7145</td>
</tr>
</tbody>
</table>

Data annotation on Amazon Mturk Platform
- 2 turkers: **Seeker** and **recommender** converse with each other.

---

Li et al. “Towards Deep Conversational Recommendations” (NIPS’18)
• ReDial – Methods – Overall

Li et al. “Towards Deep Conversational Recommendations” (NIPS’18)
Notations:
- We have $|M|$ users and $|V'|$ movies.
- User-movie Rating Matrix:
- A user can be represented by

$$\mathbf{R} \in \mathbb{R}^{M \times |V'|}$$

$$\mathbf{r}^{(u)} = (R_{u,1}, \ldots, R_{u,|V'|})$$

**AutoRec: Autoencoders Meet Collaborative Filtering (WWW15)**

- Then Loss function:

$$L_R(\theta) = \sum_{u=1}^{M} \| \mathbf{r}^{(u)} - \hat{h}(\mathbf{r}^{(u)}; \theta) \|^2 + \lambda \| \theta \|^2$$

*Li et al. “Towards Deep Conversational Recommendations” (NIPS’ 18)*
---

### ReDial – Methods – Decoder with a Movie Recommendation Switching Mechanism

**Responsibility:**
- When decoding the next token, decide to mention a movie name, or an ordinary word.

**Purpose:**
- Such a switching mechanism allows to include an explicit recommendation system in the dialogue agent.

---

*Li et al. “Towards Deep Conversational Recommendations” (NIPS’ 18)*
• ReDial – Evaluation – Formalization

Evaluation settings:
Corpus-based evaluation. (Similar to the evaluation in dialogue system)

![Diagram showing history dialogues, output utterance, and ground truth in corpus]

Evaluation Metrics in this work:
- Kappa score: Sentiment analysis subtask
- RMSE score: Recommendation subtask
- Human evaluation: Dialogue generation

Li et al. “Towards Deep Conversational Recommendations” (NIPS’ 18)
The ReDial (NIPS18) paper has two shortage:

- Only mentioned items are used for recommender system.

- Recommender cannot help generate better dialogue.

*Lord of the Rings* is really my all-time-favorite! In fact, I love all J. R. R. Tolkien’s work!

Chen et al. “Towards Knowledge-Based Recommender Dialog System” (EMNLP ’19)
Chen et al. “Towards Knowledge-Based Recommender Dialog System” (EMNLP’ 19)
KBRD – Experiments – Does Recommendation Help Dialog?

<table>
<thead>
<tr>
<th>Movie</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Star Wars</td>
<td>space</td>
<td>alien</td>
<td>sci-fi</td>
<td>star</td>
<td>sci</td>
<td>robot</td>
<td>smith</td>
<td>harry</td>
</tr>
<tr>
<td>The Shining</td>
<td>creepy</td>
<td>stephen</td>
<td>gory</td>
<td>horror</td>
<td>scary</td>
<td>psychological</td>
<td>haunted</td>
<td>thriller</td>
</tr>
<tr>
<td>The Avengers (2012)</td>
<td>marvel</td>
<td>superhero</td>
<td>super</td>
<td>dc</td>
<td>wait</td>
<td>batman</td>
<td>thor</td>
<td>take</td>
</tr>
<tr>
<td>Beauty and the Beast</td>
<td>cute</td>
<td>disney</td>
<td>animated</td>
<td>live</td>
<td>music</td>
<td>child</td>
<td>robin</td>
<td>kids</td>
</tr>
</tbody>
</table>

Recommendation-Aware Dialog

\[ P_{\text{dialog}} = \text{softmax} (Wo + b) \]

\[ b_u = \mathcal{F}(t_u) \] \hspace{1cm} Vocabulary Bias

\[ P_{\text{dialog}} = \text{softmax} (Wo + b + b_u) \]

- We select words with Top 8 vocabulary bias. We can see that these words have strong connection with the movie.

Chen et al. “Towards Knowledge-Based Recommender Dialog System” (EMNLP’19)
Recap the settings in NIPS 18:
- **Seeker**: explain what kind of movie he/she likes, and asks for movie suggestions
- **Recommender**: understand the seeker’s movie tastes, and recommends movies

The dialogue types are very limited!

In this work, 4 types of dialogues:
- Recommendation
- Chitchat
- QA
- Task

Liu et al. “Towards Conversational Recommendation over Multi-Type Dialogues” (ACL ’20)
Very similar to the dataset collection process as in NIPS 18: Two workers, one for seeker, one for recommender.

It is further supported by following elements:

Explicit Seeker Profile
- For the consistency

<table>
<thead>
<tr>
<th>Goals</th>
<th>Goal description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal1: QA (dialog type) about the movie &lt;Stolen life&gt; (dialog topic)</td>
<td>The seeker takes the initiative, and asks for the information about the movie &lt;Stolen life&gt;; the recommender replies accordingly to the given knowledge graph; finally the seeker provides feedback.</td>
</tr>
<tr>
<td>Goal2: chitchat about the movie star Xun Zhou</td>
<td>The recommender proactively changes the topic to movie star Xun Zhou as a short-term goal, and conducts an in-depth conversation;</td>
</tr>
<tr>
<td>Goal3: Recommendation of the movie &lt;The message&gt;</td>
<td>The recommender proactively changes the topic from movie star to related movie &lt;The message&gt;, and recommend it with movie comments, and the seeker changes the topic to Rene Liu’s movies;</td>
</tr>
<tr>
<td>Goal4: Recommendation of the movie &lt;Don’t cry, Nanking!&gt;</td>
<td>The recommender proactively recommends Rene Liu’s movie &lt;Don’t cry, Nanking!&gt; with movie comments. The seeker tries to ask questions about this movie, and the recommender should reply with related knowledge. Finally the user accepts the recommended movie.</td>
</tr>
</tbody>
</table>

Task Template
- Constrain the complicated task

Knowledge Graph:
- Further assist the workers

Liu et al. “Towards Conversational Recommendation over Multi-Type Dialogues” (ACL’ 20)
Liu et al. “Towards Conversational Recommendation over Multi-Type Dialogues” (ACL’ 20)
• MGCG – Evaluation – Setting

Evaluation Metrics:

Dialogue generation:
- BLEU – Relevance
- Perplexity – Fluency
- DIST – Diversity
- Hits@1/3 -- Retrieval model (1 ground truth, 9 randomly sampled.)

Human Evaluation:
- Turn level: fluency, appropriateness, informativeness, and proactivity.
- Dialogue level: Goal success rate and Coherence

Liu et al. “Towards Conversational Recommendation over Multi-Type Dialogues” (ACL’ 20)
Motivation:
Existing dialogue systems only utilize textual information, which is not enough for full understanding of the dialogue.

- What is “these”?
- What is “it”?

Background: Fashion Match!

User utterance \( u_1, u_2, \cdots, u_t \)
Agent utterance \( \hat{u}_1, \hat{u}_2, \cdots, \hat{u}_{t-1} \)

\( u \) be both Text and Image modality

• KMD – Method – Overview

Instead of CNN to capture image feature, they used taxonomy-based feature. They argued that CNN only captures generic features, but they want to capture the rich domain knowledge in specific domain.

• KMD – Method – EI Tree

- **EI Loss**: Compare the predicted leaf node against ground truth, and optimize the cross entropy loss.
- **Pairwise ranking loss** is used to regularize the model to match text and image feature.

\[
p(c_n | c_0 \rightarrow c_n, f, W_{EI}) = \prod_{i=1}^{n} p(c_i | c_{i-1}, f, W_{EI})
\]

**Optimization**

**Fashion Tips**: if the user asks for advice about matching tips of **NUS hoodie**, the matching candidates such as the **Livi’s jeans** might not co-occur with it in the whole training corpus or conversation history.
They incorporated knowledge into HRED model (hierarchical recurrent encoder-decoder)

\[ h'_t = h_t + s. \]

Each EI tree leaf gets a memory vector: the averaging of the image representation corresponds to the leaf node.

Towards Building Large Scale Multimodal Domain-Aware Conversation Systems (AAAI 18) MMD Dataset

**Evaluation Metrics:**

**Text generation:**
- BLEU Score
- Diversity (unigram)

**Image response generation:**
- Recall @ K

• Tutorial Outline

- A Glimpse of Dialogue System
- Four research directions in conversational recommendation system
  - Question Driven Approaches
  - Multi-turn Conversational Recommendation Strategy
  - Dialogue Understanding and Generation
  - Exploitation-Exploration Trade-offs for Cold Users
- Summary of Formalizations and Evaluations
• Bandit algorithms for Exploitation-Exploration trade-off

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

<table>
<thead>
<tr>
<th>Arm 1</th>
<th>Arm 2</th>
<th>Arm 3</th>
<th>Arm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/5</td>
<td>0/1</td>
<td>3/8</td>
<td>1/3</td>
</tr>
</tbody>
</table>

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.
• Upper Confidence Bounds (UCB) - Method

Arm selection strategy:

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

\[ \hat{Q}(a) = \frac{1}{N} \sum_{t=1}^{N_a} r_{t,a} \] : The estimated mean of reward of arm \( a \).

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

Estimating rewards by averaging the observed rewards:

<table>
<thead>
<tr>
<th>Arm 1</th>
<th>Arm 2</th>
<th>Arm 3</th>
<th>Arm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{Q}(a) = \frac{2}{5} )</td>
<td>( \hat{Q}(a) = 0/1 )</td>
<td>( \hat{Q}(a) = \frac{3}{8} )</td>
<td>( \hat{Q}(a) = \frac{1}{3} )</td>
</tr>
</tbody>
</table>

\[ \hat{a} = \arg \max_a \hat{Q}(a) + \alpha \frac{\log T}{t_a} \]
A Contextual-Bandit Approach with Linear Reward (LinUCB) - Method

Estimating reward by introducing the feature vector $\mathbf{X}_{t,a}^T : \hat{Q} = \mathbf{X}_{t,a}^T \mathbf{\theta}_a$

List of arm selection tables:

<table>
<thead>
<tr>
<th>Arm 1</th>
<th>Arm 2</th>
<th>Arm 3</th>
<th>Arm 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{X}_{t,a}^T \mathbf{\theta}_a = 2/5$</td>
<td>$\mathbf{X}_{t,a}^T \mathbf{\theta}_a = 0/1$</td>
<td>$\mathbf{X}_{t,a}^T \mathbf{\theta}_a = 3/8$</td>
<td>$\mathbf{X}_{t,a}^T \mathbf{\theta}_a = 1/3$</td>
</tr>
</tbody>
</table>

The arm selection strategy is:

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

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

Li et al. “A Contextual-Bandit Approach to Personalized News Article Recommendation” (WWW 10)
• Bandit algorithm in Conversational Recommendation System - Formalization

**Setting:**

• For cold start users, the user embedding is initialized as the average embedding of existing users.
• Asking only whether a user likes items (no attributes questions).
• The model updates its parameters at each turn.

*Christakopoulou et al. “Towards Conversational Recommender Systems” (KDD’ 16)*
Method:

**Traditional recommendation model** + **bandit model**

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

1. User 1 has traits \(u_i \sim N(0, \sigma_1^2 I)\), bias \(\alpha_i \sim N(0, \sigma_2^2)\).
2. Item 1 has traits \(v_j \sim N(0, \sigma_3^2 I)\), bias \(\beta_j \sim N(0, \sigma_4^2)\).
3. (a) The (unobserved) affinity is

\[
y_{ij} = \alpha_i + \beta_j + u_i^T v_j. \tag{1}
\]

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

\[
\hat{r}_{ij} = 1[\hat{y}_{ij} > 0]. \tag{2}
\]

**Terminology**

- **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):** \(j^* = \arg \max_j \epsilon_{ij}\)
  - An explore-only strategy, variance reduction strategy: Select the item with the highest noisy affinity variance.
- **Maximum Item Trait (MaxT):** \(j^* = \arg \max_j \|v_j\|_2\)
  - Select the item whose trait vector \(v_j\) contains the most information, namely has highest L2 norm \(\|v_j\|_2 = \sqrt{v_{j1}^2 + v_{j2}^2 + \ldots + v_{jd}^2}\).
- **Minimum Item Trait (MinT):** \(j^* = \arg \min_j \|v_j\|_2\)
  - 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):** \(j^* = \arg \max_j \hat{y}_{ij}\)
  - For each item, sample the noisy affinity from the posterior. Select item with the maximum sampled value.

**Common bandit strategies**

Christakopoulou et al. “Towards Conversational Recommender Systems” (KDD’16)
• Bandit algorithm in Conversational Recommendation System - Evaluation

**Setting: Offline initialization + Online updating**

- Online stage: Ask 15 questions of 10 items. 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 3549 users 289 restaurants, and 9330 positive observations.
- Recruit 28 users to rate on the selected 10 restaurants.
- Online cold-start user study: each one of the 28 users rates 10 carefully selected restaurants, based on which his/her preference $u_i$ is inferred. Then, run 50 times:
  1. Sample a user $i$.
  2. Sample $\hat{u}_i \sim u_i$.
  3. Use $\hat{u}_i$ to simulate reward of each restaurant.

*Christakopoulou et al. “Towards Conversational Recommender Systems” (KDD’ 16)*
• Conversational UCB algorithm (ConUCB) - Formalization

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.

Zhang et al. “Conversational Contextual Bandit: Algorithm and Application” (WWW’ 20)
Select attributes (key-terms) to query

Select an item (arm) to recommend

Zhang et al. "Conversational Contextual Bandit: Algorithm and Application" (WWW’ 20)
• ConUCB - Method

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 $b(t) - b(t - 1)$ key-terms;
• 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.

$$b(t) = k \left\lceil \frac{t}{m} \right\rceil, m \geq 1, k \geq 1.$$ 

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

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

1) There is no conversation between the agent and the user.

$$b(t) \equiv 0$$

Zhang et al. “Conversational Contextual Bandit: Algorithm and Application” (WWW’ 20)
• **ConUCB - Method**

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 feedback, and receives a reward.

User preference computed on key-term-level rewards

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

User preference computed on arm-level rewards

\[
\theta_t = \arg \min_{\theta} \lambda \sum_{\tau=1}^{t-1} (\theta^T x_{a,\tau} - r_{a,\tau})^2 + (1 - \lambda) \| \theta - \tilde{\theta}_t \|^2
\]

**Constrain \( \theta \) to be close to \( \tilde{\theta} \)**
The core strategy to select arms and key-terms:

- Selecting the key-terms that maximize the reward of the corresponding items.

\[ k = \arg \max_{k'} \left\| X_t M_t^{-1} \tilde{M}_{t-1}^{-1} \tilde{x}_{k',t} \right\|_2^2 / \left( 1 + \tilde{x}_{k',t}^T \tilde{M}_{t-1}^{-1} \tilde{x}_{k',t} \right) \]

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

Zhang et al. “Conversational Contextual Bandit: Algorithm and Application” (WWW’ 20)
• **Thompson Sampling**
  
  • **Bayesian bandit problem**: instead of modeling the probability of reward as a scalar, Thompson Sampling assumes the user preference comes from a distribution.
• Contextual Thompson Sampling
  
  - Assume that user preference comes from a **multidimensional Gaussian distribution**.

**Arm selection strategy:**

\[ \hat{a} = \arg \max_a x_a^T \theta_u \]

\( \theta_u \) denotes user preference. In each turn, it is sampled from a Gaussian distribution:

\[ \mathcal{N}(\mu_u, I^2 B_u^{-1}) \]
This time, we focus on cold-start users

Objective:
Recommend desired items to user in shortest turns

Treat items and attributes as indiscriminate arms.

Make theoretical customization for contextual TS to adapt to cold-start users in conversational recommendation.

**ConTS (Conversational Thompson Sampling) -- Workflow**

---

**Li et al. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users (arxiv' 20)**
Arm Choosing: It is very simple, 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.

\[
\mathbb{E}[r(a, u, \mathcal{P}_u)] = u^T x_a + \sum_{p_i \in \mathcal{P}_u} x_a^T p_i,
\]

We addresses the strategy for recommendation issue by our indiscriminate designs of arms.

*Li et al. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users (arxiv’ 20)*
• ConTS -- Method -- Update

\[ \mathbb{E}[r(a, u, P_u)] = u^T x_a + \sum_{p_i \in P_u} x_{a}^T p_i, \]

Update of Arm Pool: \( P_u \)

• If user rejects an item / attribute: remove them from arm pool.
• If user likes an attribute: append it to the known attribute set for better estimation and narrow down the candidate item pool accordingly.

Update parameters of: \( \mathcal{N}(\mu_u, l^2B_u^{-1}) \)

\[
\begin{align*}
B_u &= B_u + x_{a(t)} x_{a(t)}^T \\
r'_a &= r_a - x_{a(t)}^T(u_{\text{init}} + \sum_{p_i \in P_u} p_i) \\
f_u &= f_u + r'_a * x_{a(t)} \\
\mu_u &= B_u^{-1} f_u
\end{align*}
\]

The known preferred attributes are used to estimate reward of arms as well as narrow down the candidate item pool.

Li et al. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users (arxiv’20)
User ID: 333, Item ID: 666

Item Name: “Small Italy Restaurant”
Item Attributes: [Pizza, Nightlife, Wine, Jazz]

I’d like some **Italian** food.

Got you, do you like some **pizza**?
Yes!

Got you, do you like some **nightlife**?
Yes!

**Do you want “Small Paris”?**
Rejected!

Got you, do you like some **Rock Music**?
No!

**Do you want “Small Italy Restaurant”?**
Accepted!

Check, I don’t want “Small Paris”

Check, I don’t want “Rock Music”

Template-based utterances
ConTS unifies items and attributes and keeps EE balance.

Li et al. Seamlessly Unifying Attributes and Items: Conversational Recommendation for Cold-Start Users (arxiv’ 20)
• VDA IRS -- Formalization

A Visual Dialog Augmented Interactive Recommender System

Yu et al. (KDD’19)

Round 1

I prefer boots.

I prefer blue color.

Round 2

I prefer high boots.

I prefer high heel.

Yu et al. A Visual Dialog Augmented Interactive Recommender System (KDD’19)
Yu et al. A Visual Dialog Augmented Interactive Recommender System (KDD ’19)
The comments and images are encoded to help elicit the user preferences and narrow down the candidate set.

Yu et al. A Visual Dialog Augmented Interactive Recommender System (KDD ’19)
forall $t = 1, \cdots, n$ do

Sample the parameter $\theta$ from its posterior

$\theta_t \sim N(\bar{\theta}_t, S_t)$

forall $k = 1, \cdots, K$ do

$\quad a_k^t \leftarrow \arg \max_{e \in \mathcal{L} - \{a_1^t, \cdots, a_{k-1}^t\}} x_e^T \theta_t$

end

$A_t \leftarrow (a_1^t, \cdots, a_k^t)$

Observe click $C_t \in \{1, \cdots, K, \infty\}$

Each turn, the model will recommend a list of the Top $K$ items.
• VDA IRS -- Evaluation

Dataset:
- A footwear dataset where 10,000 images for offline training the visual dialog encoder and 4,658 images for evaluating different interactive recommenders.

User simulator:

```
For each run of experiment, choose a category of items as desired items

For each turn of dialog

Probabilistic rules to decide whether the user will comment

Comment generated by relative captioner (a pretrained NLP model)

Recommend and user feedback

E.g., “sneakers”, “boots” and “flats”
```

Yu et al. A Visual Dialog Augmented Interactive Recommender System (KDD’19)
• Strategies in the conversational recommendation bandit (ConUCB)

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-truth reward on unobserved arm and key terms of by this estimated $\theta$.

Zhang et al. “Conversational Contextual Bandit: Algorithm and Application” (WWW’ 20)
• Tutorial Outline

- A Glimpse of Dialogue System
- Four research directions in conversational recommendation system
  - Question Driven Approaches
  - Multi-turn Conversational Recommendation Strategy
  - Dialogue Understanding and Generation
  - Exploitation-Exploration Trade-offs for Cold Users
- Summary of Formalizations and Evaluations
Mainstream settings for CRS:

- Only consult on items.
- Ask 1 turn, recommend 1 turn.
- Ask X turn, recommend 1 turn (X is predefined).
- Ask X turn, recommend 1 turn (The system need to decide X).
- Ask X turn, recommend X turn.
- Natural Language Understanding and Generation.

• Summary – Formalization
The system only consult users on their preference on items.
- Cannot leverage on the advantage of explicitly consulting on attributes.

**Summary – Formalization – Ask 1 Turn, Recommend 1 Turn**

- The session will end regardless the recommendation successes or not.
- The session will continue till the recommendation successes.

![Diagram](image)

Yu et al. A Visual Dialog Augmented Interactive Recommender System (KDD’19)

Christakopoulou et al. “Q&R: A Two-Stage Approach toward Interactive Recommendation” (KDD’18)
Are you seeking for a cotton related item?  

Yes!

Are you seeking for a beach towel related item?  

No.

Are you seeking for a bath-room towel related item?  

Yes!

The recommendation list:  

- Towel A
- Towel B

The number of questions $X = 3$ is a hyper-parameter which is specified.

- Ask $K$ question and then recommend one batch of items. ($X$ is pre-defined)
- Do not take long-term strategy into account.

Making Recommendation only once $N_q$ questions

Zou et al. “Towards Question-based Recommender Systems” (SIGIR’ 20)
• Summary – Formalization – Ask X Turn, Recommend 1 turn

- Ask X question and then recommend one batch of items. (X is decided by model)
- The session will end regardless the recommendation succeeds or not.
- Only consider strategy in a shallow way (e.g. after asking 3, 4 or 5 question, should I recommend?)
• Summary – Formalization – Ask X turn, Recommend X turn

- Ask X question and then recommend one batch of items.
- The session will go on even if the recommendation is not successful!
• **Summary – Formalization – Natural Language Understanding and Generation**

<table>
<thead>
<tr>
<th></th>
<th>HUMAN:</th>
<th>SEEKER:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>hello, can i help you find a movie ?</td>
<td>hello ! i would like to find a suspenseful, but clean family friendly movie .</td>
</tr>
<tr>
<td></td>
<td>hi what kind of movies do you like</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>hi, how can i help you ?</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>2001 : a space odyssey might be a great option. anything else that you would suggest ?</td>
<td>2001 : a space odyssey might be a great option. anything else that you would suggest ?</td>
</tr>
<tr>
<td></td>
<td>you can try planet of the apes the older one is quite suspenseful and family friendly .</td>
<td>you can try planet of the apes the older one is quite suspenseful and family friendly .</td>
</tr>
<tr>
<td></td>
<td>i haven’t seen that one yet but i heard it was good. i haven’t seen that one. have you seen the last house on the left ?</td>
<td>i haven’t seen that one yet but i heard it was good. i haven’t seen that one. have you seen the last house on the left ?</td>
</tr>
<tr>
<td></td>
<td>star wars : the force awakens is also a good one return of the jedi all good movies</td>
<td>star wars : the force awakens is also a good one return of the jedi all good movies</td>
</tr>
<tr>
<td></td>
<td>those sound good ! i ’m going to look into those movies.</td>
<td>those sound good ! i ’m going to look into those movies.</td>
</tr>
<tr>
<td></td>
<td>i hope you enjoy, have a nice one</td>
<td>i hope you enjoy, have a nice one</td>
</tr>
<tr>
<td></td>
<td>have you seen foxcatcher ? it ’s about a man who has a rich guy.</td>
<td>have you seen foxcatcher ? it ’s about a man who has a rich guy.</td>
</tr>
<tr>
<td></td>
<td>i hope i was able to help you find a good movie to watch</td>
<td>i hope i was able to help you find a good movie to watch</td>
</tr>
<tr>
<td></td>
<td>thank you for your help ! have a great night ! good bye</td>
<td>thank you for your help ! have a great night ! good bye</td>
</tr>
</tbody>
</table>

- This is more likely to be a special type of dialogue system. More popular in NLP community.

*Li et.al. “Towards Deep Conversational Recommendations” (NIPS’18)*
The session will go on even if the recommendation is successful.

- Maximize Profit
- Increase the time users stay
Summary – User Preference Simulation

Mainstream approaches to simulate user preference:

- **User click history**: EAR (WSDM20), CPR (KDD20), CRM(SIGIR18)

- **Generalize to the full datasets**: (KDD16) ConUCB (WWW20)

- **Extract from user review**: SAUR (CIKM18)

- **Corpus based**: the line of NLU/NLG works
• Summary – User preference simulation – User Click History

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.

Item Name: “Small Italy Restaurant”
Item Attributes: [Pizza, Nightlife, Wine, Jazz]

Template-based utterances:
- I'd like some Italian food.
  - Got you, do you like some pizza?
    - Yes!
    - Got you, do you like some nightlife?
      - Yes!
      - Do you want “Small Paris”?
        - Rejected!
        - Got you, do you like some Rock Music?
          - No!
          - Do you want “Small Italy Restaurant”?
            - Accepted!
• Summary – User preference 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.

New user manually rate 10 items.

User preference

Ratings on unobserved data.

Existing ratings.

User preference

Ratings on unobserved data.

Christakopoulou et al. “Towards Conversational Recommender Systems” (KDD’ 16)

Zhang et al. “Conversational Contextual Bandit: Algorithm and Application” (WWW’ 20)
• Summary – User preference simulation – Extract from User Review

**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”).

Zou et al. “Towards Question-based Recommender Systems”(SIGIR’ 20)
Conversational recommendation through natural language.

- 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).

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

Li et.al. “Towards Deep Conversational Recommendations” (NIPS’18)
• Discussion on Future Researches

Formalization (problem setting):
- If a user accepts the recommendation, is it possible to recommend more?
- Can we optimize other goals other than clicking? For example, maximizing profits in E-commerce; maximizing total time spending in video sharing platform ...

Evaluation (simulating user preferences):
- How to reliably simulate user preferences and action in conversational recommendation scenarios!