ABSTRACT

none of previous has considered the issue of popularity bias in a CRS. This paper proposes a human-in-the-loop popularity debiasing framework that integrates real-time semantic understanding of open-ended user utterances as well as historical records, while also effectively managing the dialogue with the user. This allows the CRS to balance the recommendation performance as well as the item popularity so as to avoid the well-known "long-tail" effect.

Introduction:  
Traditional methods for reducing popularity bias in recommendation systems are not easily applicable in a conversational setting. Most of these methods rely on one-shot debiasing, which isn't as effective in a CRS environment where there are multiple rounds of conversation, creating a potential for multi-round popularity bias. This is especially the case in CRSs that use a human-in-the-loop (HitL) learning approach, where the system must actively interact with the user and elicit their preferences.

The authors point out that in a CRS, the system must consider user preference updates during the course of the conversation to balance between recommendation utility and popularity bias. Applying a uniform debiasing strategy across the board can be counterproductive.Furthermore, the system must be able to adjust the weight of different popularity biases based on individual user's history and global popularity statistics.

The authors also discuss the typical framework for CRS, including User Interaction Modeling (UIM), Dialogue State Management (DSM), and Recommendation (REC). UIM involves encoding information from the dialogue and historical user records, while DSM uses this information to select suitable actions or questions in each conversational turn. The REC module then proposes recommendations based on this dialogue state.

However, the existing CRS methods such as collaborative filtering-based ones implicitly encode the user histories, which may mislead the system about user preferences. For instance, a user's history of watching science fiction movies may lead the system to assume they prefer popular science fiction films, while the user might actually have a broader range of interests.

The authors argue that the DSM module should be designed to carefully consider how to ask questions in each round to alleviate the popularity bias that may arise from the UIM module. Unfortunately, existing designs for the DSM module do not account for popularity bias in their objective functions, which is a gap that the authors aim to address.

The authors propose a new method, named Popcorn, for mitigating popularity bias in (CRS). Popcorn includes three modules: User-Interaction Modeling (UIM), Dialogue State Management with Popularity Debiasing (DSM-PD), and Recommendation (REC).

1. \*\*UIM\*\*: This module encodes real-time user-agent dialogue and historical user records to produce a personalized state vector.

2. \*\*DSM-PD\*\*: This module takes the state vector from the UIM and predicts the next action - either asking a question to learn more about the user's preferences or making a recommendation. The DSM-PD module specifically considers the popularity bias of potential recommendations and integrates this as a supervised signal into the learning process. The training procedure for DSM-PD is formalized as a multi-objective Markov decision process, using a new variant of the policy gradient method to learn its parameters.

3. \*\*REC\*\*: This module is triggered by the DSM-PD at an appropriate time to make the final recommendation, using both dialogue context and historical user information.

**PRELIMINARIES**

The paper uses a conversational framework referred to as the System Asks – User Responds (SAUR) paradigm to define the conversational recommendation problem.

In this framework:

- U and V represent the sets of users and items, respectively.

- Each user u has a subset of items 𝑉𝑢 which represents their historical records.

- The CRS agent asks the user a question 𝑞 from a set of candidate questions Q, each associated with a specific attribute.

- The user responds with 𝑝 , which could be a direct answer to the question with an attribute value, or an expression of no preference on the attribute.

The conversation proceeds with the agent selecting the next action based on the current dialogue state s𝑡 =(𝑉𝑢, {𝑝0, . . . , 𝑝𝑡 }, {𝑞1, . . . , 𝑞𝑡 }).

If the agent decides to request further information about the user’s preferences, it will choose the bestquestion 𝑞𝑡+1 from the set of remaining questions Q \{𝑞1, . . . , 𝑞𝑡 }. If it instead decides to provision a recommendation ˆ𝑉𝑢, it will pick top 𝐾 items from the candidate set V \𝑉𝑢 and display them to the user

If the user rejects all recommended items, the system can choose to continue asking new questions or end the conversation. To simulate users quitting due to impatience, a maximum threshold is set on the number of conversational turns in their experiments.

4.2 Quantifying Popularity Bias: The authors introduced a method to quantify popularity bias. They defined a set A of m attributes associated with the item set. For a given set of items V subset of V and an attribute a in A, they denoted by V(a) the subset of items whose values on the attribute a is not empty(only includes items for which the specific attribute a is applicable or known.)They then derived two disjoint sets from V(a) - one for popular items and the other for unpopular items, in relation to the attribute a.

In a recommendation scenario, for a specific recommendation set predicted Vu,we define the probability of resulting popularity bias for attribute ai ∈A on the recommendations as follows.



Here, Vu pop(ai) is the popular item set derived from predicted Vu(ai), and Z is the normalization term.Furthermore, motivated by the classical information retrieval evaluation method NDCG, we can also define a utility function to quantify the influence degree of popularity bias for each attribute ai over the recommendations, denoted by p(ai, predicted Vu):

一張含有 文字, 字型, 收據, 白色 的圖片

自動產生的描述

where1[.]∈{0,1}denotes the indicator function and π(v) is the rank of v. Note that this utility function always yields a positive value indicating the degree of popularity bias.

Therefore, to measure the overall effect of popularity bias on the recommendation at the t-th turn, we define the expected popularity bias as follows.

一張含有 字型, 文字, 白色, 印刷術 的圖片

自動產生的描述

Degree \* p

At is the set of attributes associated with the invoked questions in the t-round conversation, while Vu,t is the set of potential CRS recommendations made at turn t that are not actually shown to the user. In the CRS, Eq. 3 may be employed to indicate the expected influence of popularity bias that the unused attributes {a|a∈A,aAt} will bring to the current recommendations.

4.3 Evaluation Metrics

We employ two metrics to measure the attribute-based popularity bias in the recommendations.

Personalized Average Recommendation Popularity (PARP).

We first explore and adapt the personalized popularity distribution of recommended items, which is defined as:  
一張含有 字型, 文字, 白色, 行 的圖片

自動產生的描述

where predicted Vu denotes the recommendation list for user u,A(v) is the set of available attributes of item v, and Apop(v) is the subset of A(v) such that the attribute values are popular. Note that a higher PARP value indicates less exposure of unpopular items. The PARP scores are within the range of [0,1] and the smaller the value, the better the performance of the popularity debiasing.

所以PARP計算的是從Vcand中的某個item它的attributes中popular的那些attribute，佔該item的attribute的多少。然後每一個item都會有一個比例。

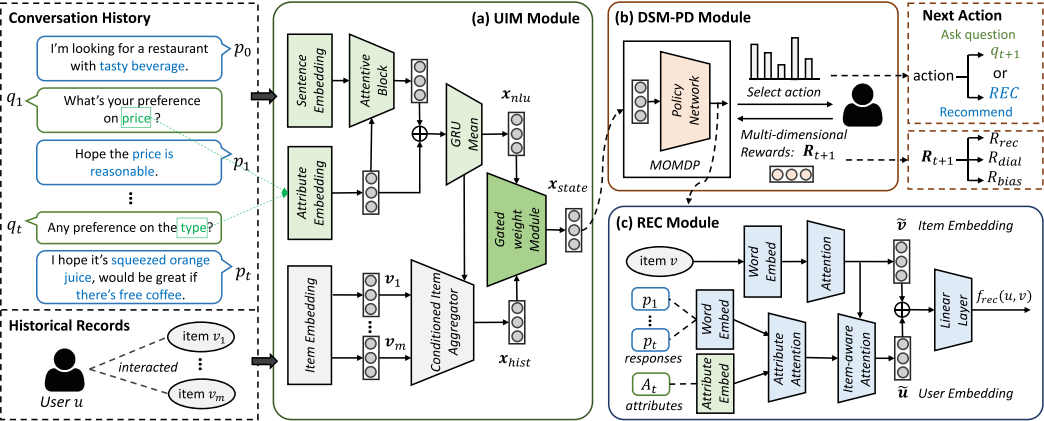
Popularity-rank correlation for users (PRU). PARP neglects the ranking position of the recommended items. Similar to [54], (PRU) definedas follows:



where Ou)is the ground truth set of items, p(·) is Spearman's Rank Correlation, pop(·) calculates the popularity of each item in Ou, and π(,) denotes the rankings of the ground truth items in the recommendations.

計算popular item 與 ranking item的平均相似度。The negative sign indicates that the PRU is being calculated as a negative correlation, meaning a higher PRU indicates a lower popularity bias。

5 OUR METHOD



In this section, we introduce the proposed POPCORN CRS, which consists of three components, as illustrated in Fig.3,including a user interaction modeling module (UIM), a dialogue state management module with popularity debiasing(DSM-PD),and a bias-aware recommendation module (REC). The UIM module encodes the open-ended descriptive language from user responses and questions to extract pertinent semantic cues. It also integrates such information with a user's historical records by automatically determining to what extent they can contribute towards estimating the user preferences. Subsequently, the UIM's outputs are relayed to the DSMPD to update the dialogue state and select the next action in each conversational turn. In particular, the latter must decide whether to propose a question regarding potential attributes or to issue a particular recommendation considering popularity bias factors. Finally,the REC module makes recommendations and shows the results to the user with a re ranking mechanism to balance between popular and unpopular items.

5.1 User Interaction Modeling

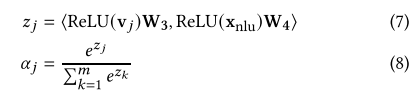
we propose the user-interaction modeling module that captures implicit semantic cues in the user responses as well as historical records.

Given a dialogue up to the t-turn {po,qi,p1,.…,qt,pt},we first encode the semantics of a user's real-time preferences as a latent vector. Sentence embedding：We represent each question or response as a sequence of l words and concatenate the question and the response at each turn with a special separator"[SEP]", resulting in t sentences denoted by {si} i∈[t]. We use a d-dimensional pre-trained word em-bedding model to vectorize each word in the input and compute their average to obtain the sentence representation,denoted by{si} i∈[t] with si E R^d.Let S ∈ R^txd be a matrix encoding t input utterances.Attention embedding: At the same time, each question is associated with an attribute, which may provide structured information on potential items. Thus, we also incorporate an attribute representation into the dialogue. Let {a1,...,at} be the sequence of attributes corresponding to t questions. We embed each attribute in a d-dimensional space and combine all the attribute vectors into a matrix A =[a1,.,at] ∈ R^txd.In order to detect the importance and usefulness of the utterances in the past t turns, we use a variant of the self-attention block to learn attentive information over the t-turn dialogue:



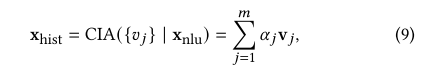
where both dialogue matrix S and attribute matrix A are encoded into the latent semantic vector x\_nlu ∈ R^d.

Item embedding:Next, we connect real-time user information with historical records to obtain a more informed estimate of the user's preferences.The goal is to select the most salient features that capture current user preferences. Suppose that user u has interacted with m historical items,i.e., Vu ={v1,...,vm}.We encode each item vj∈Vu into a vector denoted by vj, j∈[m].In order to know which items are useful to model the user's current preferences conditioned on the real-time dialogue information, we propose a component called Conditioned Item Aggregator (CIA), which learns an attention-based importance factor aj of each item vj automatically.



Here<,>is the dot product operation, W3,W₄ ∈ R^dxd are learnable parameters that map two vectors vj and x\_nlu into the same space.

Therefore, the historical user preferences can be represented by the weighted sum over all item embeddings:



where xhist is the output vector that captures historical records.To aggregate both real-time and historical information, we introduce the Gated Weight Module (GWM) to encode the state vector by automatically balancing between the real-time information and historical user preferences. Specifically, it learns a weighting factor



where W5 ∈ R^dxd, W6 ∈ R^dx1 are learnable parameters. Then, the final state vector can be computed as follows:



This latent state vector encodes rich information regarding user preferences from both the ongoing real-time conversation and from the long-term historical behavior, and the balance between them is automatically adjusted by the weighting factor β.

5.2 Dialogue State Management withPopularity Debiasing

existing work neglects the issue of popularity bias during action selection,To address this, we formulate the DSM-PD module as a multi-objective Markov decision process(MOMDP) and seek to learn an optimal policy that achieves a good balance between both objectives. Formally, we define the MOMDP as follows.

In this section, the authors introduce three important components of the CRS model - the State, the Action Space, and the Rewards.

1. \*\*State **s***𝑡* = **x**state: This vector is computed by the (UIM) module. The state is represented based on implicit semantic features of the user's descriptive response and user preferences estimated from the current dialogue as well as historical records.

2. \*\*Action Space\*\*: The size of the action space is equal to one plus the number of questions remaining in the candidate pool.

3. \*\*Rewards\*\*: In this system, the reward is defined to be a vector where each component represents feedback from the environment with respect to a specific objective. Three factors are considered - recommendation performance, user experience, and item popularity bias. A reward vector R\_t∈R^3 is assigned to the action when the user's response is collected, R𝑡 = [𝑅rec,𝑡, 𝑅dial,𝑡, 𝑅bias,𝑡]..

- 𝑅rec,𝑡 refers to the recommendation success state. It has a value of r\_fail if the user rejects the recommendation or leaves, or a value of r\_success if the user accepts the recommendation.

𝑅dial,𝑡 is associated with the user experience in each dialogue. It has a value of r\_reply if the user provides a preference answer to an attribute-based question asked by the agent, or r\_empty if the user does not provide a response, indicating that the user does not have a specific preference over the attribute.

𝑅bias,𝑡 = exp(−E𝐴𝑡[ˆ𝑉𝑢,𝑡]) provides a signal regarding the popularity bias. It is defined over the remaining attributes A \A\_t not associated with the past questions.

Objective

In this section, the authors discuss how the Dialogue State Management (DSM) module's goal is to learn an optimal policy {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"normal\">&#x3C0;</mi><mi mathvariant=\"normal\">&#x3B8;</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"}, parameterized by {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>&#x3B8;</mi></math>","origin":"MathType for Microsoft Add-in"}, that simultaneously maximizes three objectives: recommendation performance, user experience, and item popularity bias.

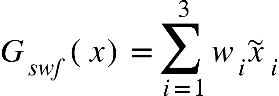
一張含有 文字, 字型, 白色, 行 的圖片

自動產生的描述

The objective function is represented as J({"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"normal\">&#x3C0;</mi><mi mathvariant=\"normal\">&#x3B8;</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"}), where s J({"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"normal\">&#x3C0;</mi><mi mathvariant=\"normal\">&#x3B8;</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"}) {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><semantics><mstyle mathsize=\"16px\"><mo>&#x2208;</mo><msup><mi mathvariant=\"normal\">&#x211D;</mi><mn>3</mn></msup></mstyle><annotation encoding=\"application/json\">{\"x\":[[77,39,11,13,34,82],[13,65],[111,102,109,130,146,147,135,111,113,144],[97,102],[172.5,172.5,173.5,174.5,176.5,179.5,182.5,183.5,185.5,186.5,187.5,188.5,188.5,188.5,188.5,188.5,188.5,188.5,188.5,188.5,186.5,183.5,179.5,176.5,172.5,167.5,165.5,162.5,162.5,163.5,165.5,167.5,170.5,171.5,174.5,176.5,178.5,180.5,182.5,182.5,183.5,183.5,183.5,184.5,184.5,184.5,183.5,182.5,179.5,178.5,177.5,175.5,174.5,174.5,173.5,172.5,171.5,170.5,167.5,166.5,164.5,162.5,161.5,159.5,159.5,159.5,158.5]],\"y\":[[39,37,68,98,111,111],[77,77],[113,8,7,7,17,40,50,50,55,116],[7,111],[-26.5,-27.5,-28.5,-28.5,-28.5,-27.5,-25.5,-24.5,-22.5,-21.5,-20.5,-19.5,-18.5,-16.5,-13.5,-11.5,-10.5,-8.5,-8.5,-7.5,-5.5,-4.5,-4.5,-3.5,-2.5,-1.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,-0.5,1.5,2.5,3.5,5.5,6.5,9.5,10.5,11.5,13.5,14.5,15.5,16.5,16.5,17.5,17.5,17.5,17.5,18.5,18.5,18.5,18.5,16.5,15.5,14.5,13.5,12.5,11.5,10.5,10.5]],\"t\":[[0,0,0,0,0,0],[0,0],[0,0,0,0,0,0,0,0,0,0],[0,0],[1690346084102,1690346084122,1690346084147,1690346084156,1690346084164,1690346084172,1690346084181,1690346084189,1690346084197,1690346084206,1690346084214,1690346084222,1690346084231,1690346084239,1690346084247,1690346084256,1690346084264,1690346084273,1690346084281,1690346084289,1690346084298,1690346084306,1690346084314,1690346084322,1690346084331,1690346084339,1690346084347,1690346084356,1690346084431,1690346084456,1690346084464,1690346084472,1690346084481,1690346084490,1690346084498,1690346084506,1690346084514,1690346084522,1690346084531,1690346084539,1690346084547,1690346084556,1690346084564,1690346084572,1690346084581,1690346084589,1690346084597,1690346084606,1690346084614,1690346084622,1690346084631,1690346084639,1690346084647,1690346084656,1690346084664,1690346084672,1690346084681,1690346084689,1690346084700,1690346084708,1690346084716,1690346084724,1690346084732,1690346084740,1690346084748,1690346084756,1690346084764]],\"version\":\"2.0.0\"}</annotation></semantics></math>","origin":"MathType for Microsoft Add-in"}and {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mi>&#x3B3;</mi></math>","origin":"MathType for Microsoft Add-in"} is a discount factor that reduces the influence of long-term rewards. A naive approach to optimize this vectorized objective is to use a weighted sum to aggregate all three objectives. However, such linear combinations fail to provide sufficient control to balance between different objectives. Hence, a non-linear mapping over multiple objectives is preferred.

一張含有 文字, 字型, 白色, 行 的圖片

自動產生的描述

The authors introduce a non-linear generalized Gini social welfare function  and cast the learning objective of CRS as:{"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mi>a</mi><mi>r</mi><mi>g</mi><mi>m</mi><mi>a</mi><msub><mi>x</mi><mrow><mi>&#x3C0;</mi><mi>&#x3B8;</mi><mo>&#xA0;</mo></mrow></msub><msub><mi>G</mi><mrow><mi>s</mi><mi>w</mi><mi>f</mi><mo>&#xA0;</mo></mrow></msub><mi>J</mi><mfenced><msub><mi mathvariant=\"normal\">&#x3C0;</mi><mi mathvariant=\"normal\">&#x3B8;</mi></msub></mfenced></mstyle></math>","origin":"MathType for Microsoft Add-in"}, where {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><mover><mi>x</mi><mo>~</mo></mover></math>","origin":"MathType for Microsoft Add-in"} is the vector whose components are copied from x but sorted in ascending order, and w {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mo>&#x2208;</mo></mstyle></math>","origin":"MathType for Microsoft Add-in"}[0, 1] is a weight vector satisfying 保持三种objectives的平衡

The optimal policy {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><msub><mi mathvariant=\"normal\">&#x3C0;</mi><mi mathvariant=\"normal\">&#x3B8;</mi></msub></mstyle></math>","origin":"MathType for Microsoft Add-in"}\* in Eq. 14 can provably achieve Pareto optimality, that is, a good balance among recommendation performance, user experience, and item popularity bias.

一張含有 文字, 字型, 白色, 印刷術 的圖片

自動產生的描述

To solve the optimization of Eq. 14, they define a variant of the policy gradient, where {"mathml":"<math style=\"font-family:stix;font-size:16px;\" xmlns=\"http://www.w3.org/1998/Math/MathML\"><mstyle mathsize=\"16px\"><mover><mi>w</mi><mo>~</mo></mover></mstyle></math>","origin":"MathType for Microsoft Add-in"} is a vector with components derived and sorted from weight vector w. Note that {"mathml":"<math xmlns=\"http://www.w3.org/1998/Math/MathML\" style=\"font-family:stix;font-size:16px;\"><msub><mo>&#x2207;</mo><mi>&#x3B8;</mi></msub><mi>J</mi><mfenced><msub><mi mathvariant=\"normal\">&#x3C0;</mi><mi mathvariant=\"normal\">&#x3B8;</mi></msub></mfenced><mo>&#x2208;</mo><msup><mi mathvariant=\"normal\">&#x211D;</mi><mrow><mn>3</mn><mo>&#xD7;</mo><mfenced open=\"|\" close=\"|\"><mi>&#x3B8;</mi></mfenced></mrow></msup></math>","origin":"MathType for Microsoft Add-in"} is equivalent to the classic policy gradient over 3 objectives. The DSM module thereby gradually learns to select the optimal actions based on the current state considering multiple factors.

5.3 Recommendation

In our model, the recommendation module (REC) is invoked when the DSM-PD module decides to make a recommendation. REC measures the relevance between the user and the candidate items based on the conversation history and item features, and selects the top items most likely to satisfy the user’s needs. Both the user preferences and the item features are captured by understanding the semantics of natural language inputs. The user representation is generated based on the conversation history and the item representation is generated based on its textual description. Suppose the DSM-PD module decides to make a recommendation at turn 𝑡 given

the collected 𝑡 responses {𝑝𝑖 } 𝑖 ∈ [𝑡].

**Item Representation**. Different from dialogue utterances that tend to be fairly short within a single conversation, the number of candidate items could be very large in real-world scenarios and the REC module has to measure the relevance between the user and every candidate item. Therefore, we aim to generate the item representation from its description in a simpler way to balance efficiency and effectiveness. Specifically, we first encode the description of an

item 𝑣 ∈ V into a sequence of word embeddings {d𝑗 ∈ R𝑑}，𝑗 ∈ [𝑙].

Then, a weighted average of word embeddings yields the item representation ˜v with an attention mechanism, which has been shown effective in previous work。

By calculating the dot product between vs and the transformed word embedding, the model is essentially measuring how relevant each word in the item's description is to the user's preferences. This relevance measure is then used as the attention weight in the weighted average calculation to create the final item representation.

In other words, words that are more relevant to the user's preferences will have a higher weight and will contribute more to the final item representation. This allows the model to focus more on the important words when making recommendations.

一張含有 字型, 文字, 白色, 印刷術 的圖片

自動產生的描述 Here, W𝑠 ∈ R^𝑑×𝑑, v𝑠 ∈ R^𝑑 are parameters of the attention module.

**User Representation**.

However, not all words in the response are equally important in expressing the user’s preferences,and the importance of a single word may also depend on the particular attribute. Therefore, when generating the representation of response 𝑝𝑖, we apply another attribute-aware attention mechanism to enable the model to focus on more informative words:

一張含有 字型, 文字, 行, 白色 的圖片

自動產生的描述

Here, h𝑖,𝑗 ∈ R𝑑 is the hidden state of the 𝑗-th word in the user’s response and a𝑖 ∈ R𝑑a is the attribute embedding at turn 𝑖. W𝑟 ∈R 𝑑×𝑑, W𝑎 ∈ R 𝑑a×𝑑, v𝑟 ∈ R 𝑑 are parameters of the attention module.The output r𝑖 ∈ R 𝑑 is the representation of the user’s response at

turn 𝑖.

To summarize the information in the conversation history up to turn 𝑡, all responses {r𝑖 }𝑖 ∈ [𝑡] of user 𝑢 are encoded into a dense representation via item-aware attention with a candidate item 𝑣:

一張含有 文字, 字型, 行, 白色 的圖片

自動產生的描述

Here, W𝑢 ∈ R 𝑑×𝑑 , W𝑖 ∈ R 𝑑×𝑑, v𝑢 ∈ R𝑑 are parameters of the attention module.

**Prediction**. We predict the relevance score between a user 𝑢 and an item 𝑣 via concatenation of their representations followed by a linear layer.



where v𝑦 ∈ R2𝑑, 𝑏𝑝 ∈ R are parameters for this linear layer.

We use the objective of Bayesian Personalized Ranking (BPR) to learn the parameters in the recommendation module, i.e.,

一張含有 文字, 字型, 白色, 行 的圖片

自動產生的描述

where 𝑣+ and 𝑣− refer to a positive item and a sampled negative item for the user u

6.1 Experimental Setup

two datasets: Movie lens dataset, Yelp, which provides user reviews of hotels and restaurants. They transform each review into a conversation with questions related to attributes and user responses as naturally occurring sentences referencing corresponding aspects.

The datasets were augmented with additional conversational turns using a template-based question generation process with manual verification. Unlike previous work, this system does not depend on prior knowledge of attribute-value pairs, and the User Interaction Modeling (UIM) module is optimized together with the policy network without a pretraining process. The authors split each dataset into training, development, and testing subsets with an 80%, 10%, 10% split.

In the implementation, the embedding size in the UIM module is 300. The policy network receives rewards with specific values for success, failure, reply, and empty responses. Three modules are jointly trained with Adam optimization and a learning rate of 2 × 10^-4. The Recommendation (REC) module is pretrained with Adam optimization with an initial learning rate of 10^-3.

the policy network receives rewards 𝑟success = 1, 𝑟fail = −1, 𝑟reply = −0.1, 𝑟empty =−0.5. The weight vector in the function 𝐺swf is [0.5, 0.25, 0.125] before normalization, corresponding to the objectives with rewards [𝑅rec, 𝑅bias, 𝑅dial].

For evaluation, they measure the success rate of the recommendation before achieving the maximum conversation turn and the average number of turns to achieve a successful recommendation. They use Success rate (SR) @10 and Mean reciprocal rank (MRR) to assess the overall recommendation quality. They also report the results of their proposed measurement PARP and the previously established PRU to evaluate the popularity bias.

They compare their model, Popcorn, with several state-of-the-art baseline approaches, including Max Entropy (MaxEnt), CRM, MMN, and EAR. Modifications were made to these models to enable an experimental comparison without leakage of prior knowledge of attribute-value pairs.

6.5 Analysis of Debiasing Strategies

The authors conducted several analyses to better understand the effectiveness of their popularity debiasing strategies within their proposed model, Popcorn. They tested the model under different conditions and compared the results. Here are the main findings:

Debiased DSM-PD: The authors found that without popularity debiasing in the Dialogue State Management (DSM) module, the recommendation results were as biased as the baselines. Even though the Conversational Recommender System (CRS) achieved the best recommendation utility, it obtained a sub-optimal popularity bias. They also observed that conducting popularity debiasing only on the final conversational turn resulted in less satisfactory popularity bias. The popularity bias is amplified during the interaction with users if no debiasing is invoked to guide the selection of attributes to ask the user about.

Effect of User Histories: The authors studied how users' historical records influence the model performance. They found that without historical information, the model couldn't accurately determine to what extent the user is expressing new desires or might have retained certain prior preferences. This resulted in lower recommendation quality. However, the popularity bias results improved when neither explicit user history, nor a user interaction modeling (UIM) module, nor any popularity debiasing in the DSM module were included.

Ablation Study: The authors found that the recommendation performance decreases as the weight of the debiasing objective in DSH-PD grows. However, the popularity bias gets alleviated as this weight increases. This indicates a trade-off between the two goals in the DSM module, so the ratio needs to be selected carefully by setting the weight of the debiasing objective within an acceptable range.