Towards Explainable Conversational Recommendation

Zhongxia Chen[1,2] , **Xiting Wang**[2*] , **Xing Xie**[2] , **Mehul Parsana**[3] ,
Akshay Soni[3] , **Xiang Ao**[4] and **Enhong Chen**[1]

[1]School of Computer Science and Technology, University of Science and Technology of China
[2]Microsoft Research Asia [3]Microsoft Bing Ads   [4]IIP, ICT, Chinese Academy of Sciences
{czx87@mail., cheneh@}ustc.edu.cn, {xitwan, xing.xie, mparsana, Akshay.Soni}@microsoft.com,
aoxiang@ict.ac.cn

Abstract

Recent studies have shown that both accuracy and explainability are important for recommendation. In this paper, we introduce explainable conversational recommendation, which enables incremental improvement of both recommendation accuracy and explanation quality through multi-turn user-model conversation. We show how the problem can be formulated, and design an incremental multi-task learning framework that enables tight collaboration between recommendation prediction, explanation generation, and user feedback integration. We also propose a multi-view feedback integration method to enable effective incremental model update. Empirical results demonstrate that our model not only consistently improves the recommendation accuracy but also generates explanations that fit user interests reflected in the feedbacks.

1 Introduction

Recommender systems, which predict users' personalized preferences to items, have become one of the most effective techniques for overcoming information overload. Recently, researchers are reaching a consensus that both accuracy and explainability are essential for recommendation. Explainability requires that a system provides explanations about why items are recommended. It has been shown that enhanced explainability leads to improved user satisfaction, trust, and efficiency [Tintarev and Masthoff, 2007].

Serving as a bridge between users and recommender systems, explanations not only help users understand the working mechanisms of the models, but also trigger potential user feedbacks, e.g., inspiring users to inform the system when it is wrong [Tintarev and Masthoff, 2007]. Existing methods focus on providing single-turn explanations and lack the capability to incorporate user feedbacks [Sharma and Cosley, 2013; Zhang *et al.*, 2014; He *et al.*, 2015; Li *et al.*, 2017; Wang *et al.*, 2018; Chen *et al.*, 2019; Gao *et al.*, 2019; Zhao *et al.*, 2020]. For example, the explanation "*This is a good documentary about the battle of thermopylae*" for recommending a movie may help a user realize why the recommendation is wrong, i.e., the model provides the recommen-

---

*Xiting Wang is the corresponding author

Model: <u>I recommend</u> Pulp Fiction. This is a dark comedy with a great cast.
User: I don't want to watch a comedy right now.
Model: <u>How about</u> 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: <u>I recommend</u> Mission Impossible. This is by far the best of the action series.
User: Sounds great. Thanks for the recommendation!

- -
<u>Predefined Template</u>　　Recommended Item　　Generated Explanation

Figure 1: Conversation excerpts between a user and our explainable conversational recommendation model.

dation based on his/her previous interest *documentary*. However, the user cannot communicate his/her findings with the system, e.g., his/her interest has recently shifted to *thrillers*.

In this paper, we introduce **explainable conversational recommendation**, which integrates user feedbacks into explainable recommendation to enable bidirectional user-model communications through conversations. As shown in Fig. 1, explainable conversational recommendation provides explanations to help users understand the model, and collects user feedbacks to understand and integrate user needs. During the multi-turn user-model collaboration process, both recommendations and explanations are iteratively refined.

Explainable conversational recommendation provides an alternative interaction paradigm between users and recommendation models. Prior works on conversational search and recommendation lack the capability for providing explanations [Christakopoulou *et al.*, 2016; Zhang *et al.*, 2018; Sun and Zhang, 2018; Li *et al.*, 2018; Bi *et al.*, 2019]. Since they cannot trigger feedbacks through explanations, these methods often collect user feedbacks by asking the **what** questions, e.g., "*What category of movies do you like?*". While these methods are effective for users who have a clear search target, they are less friendly for users wandering for interesting items. In comparison, users of explainable conversational recommendation can simply inform the system **whether** they like or dislike the features mentioned in the explanations, which reduces user cognitive load. The features in the explanations may also trigger users to come up with related features that they like or help users understand model imperfections. Both types of findings can be communicated to the system through user feedbacks.

While explainable conversational recommendation is promising, designing such a model is a challenging multi-objective problem. The model needs to effectively integrate user feedbacks to ensure significant and stable improvement

of recommendation accuracy (*O1*) and explainability (*O2*). Moreover, the model should seldom violate the user requirements reflected in the feedbacks (*O3*). For example, if a user claims in the conversation that she does not like *documentary*, the model should take this explicit request seriously and avoids recommending documentaries. While objective *O3* requires the model to satisfy specific user requirements (**satisfaction**), objectives *O1* and *O2* additionally request effective understanding of users by generalizing feedbacks to similar items and features (**generalization**).

In this paper, we aim to develop an **E**xplainable **C**onversational **R**ecommendation (**ECR**) model that fulfills objectives *O1–O3*. Our contributions are three-fold.

First, we **design an incremental multi-task learning framework** for explainable conversational recommendation. In our framework, multiple objectives can be simultaneously achieved through tight collaboration among the recommendation prediction task, the explanation generation task, and the user feedback integration module. The collaboration is achieved through context-aware modeling of item concepts extracted by using Microsoft Concept Graph[1]. Modeling the key concepts that a user likes about an item enables us to derive the cross knowledge between the two tasks (recommendation and explanation), trigger feedbacks about concepts, and integrate the feedbacks for incremental model update.

Second, **we propose a multi-view feedback integration method** to achieve effective incremental model update. Our method combines two views of incremental learning. The first view focuses on satisfying user requirements through *local* propagation of user needs (satisfaction), and the second view better generalizes user feedbacks by updating *global* model parameters, e.g., addressing model imperfections by learning a better concept embedding (generalization).

Third, we **evaluate our method with different settings of simulated users and human assessors**. The experiments demonstrate that our method consistently improves recommendation accuracy, increases explainability, and seldom violates user requirements (*O1-O3*).

2 Problem Formulation

Fig. 2 shows the pipeline of explainable conversational recommendation. The model takes as input a user id $u \in U$, an item id $v \in V$, and side information $\mathcal{I}_u$ and $\mathcal{I}_v$ about the user and the item. The model then outputs 1) a score $r_{u,v}$ which predicts how much u likes v and 2) an linguistic explanation $Y_{u,v}$ for the recommendation that consists of a word sequence. Item v will be displayed to the user together with the explanation if $r_{u,v}$ is the largest among the candidates V. After the user checks the recommendation and the explanation, s/he will provide a feedback $\mathcal{F}$, which helps to refine the recommendation model. This iterative process continues until 1) the user is satisfied with the recommendation or 2) the maximum number of communication turns is reached.

Different choices of side information and user feedbacks result in different types of recommendation models.
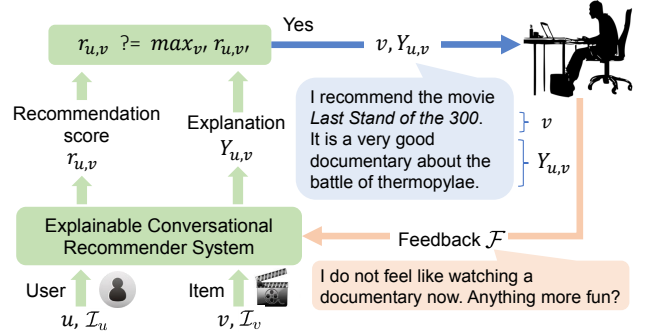
---

[1]https://concept.research.microsoft.com/

Figure 2: Pipeline of explainable conversational recommendation.

Side information $\mathcal{I}_u$ and $\mathcal{I}_v$. Many explainable recommendation methods take review comments as inputs to facilitate explanation generation [Zhang *et al.*, 2014; Chen *et al.*, 2018]. Following these methods, we consider $\mathcal{I}_u$ and $\mathcal{I}_v$ as review comments. Specifically, $\mathcal{I}_u$ denotes reviews user u writes and $\mathcal{I}_v$ is the set of reviews about item v:

$$\mathcal{I}_u = \{\mathcal{D}_{u,1}, ..., \mathcal{D}_{u,n_d}\}, \ \mathcal{I}_v = \{\mathcal{D}_{v,1}, ..., \mathcal{D}_{v,n_d}\} \quad (1)$$

where n_d is the maximum number of reviews. Each review $\mathcal{D}_{u,i}$ or $\mathcal{D}_{v,i}$ is represented by a sequence of words. We also derive the **concepts** from reviews by utilizing Microsoft Concept Graph [Wu *et al.*, 2012; Wang *et al.*, 2015]. The concepts are a subset of the words that correspond to important explicit features mentioned in the review (e.g., *documentary*).

User feedbacks $\mathcal{F}$. While free-form natural-language feedbacks empower users with the most flexibility, it is very likely that a model does not have the knowledge to process such feedbacks. For example, a model cannot correctly handle feedback "*please recommend a movie that is currently showing in the local cinemas*", because it does not know the location of the user or what movies are showing in the cinemas. We believe that a model should clearly define feasible feedbacks and inform the users. In our framework, concepts are the key for connecting different components. Thus, we allow users to provide feedbacks about what concepts they are (not) interested in together with their item interests:

$$\mathcal{F} = \{c_1^+, ..., c_{n_g}^+\} \cup \{c_1^-, ..., c_{n_b}^-\} \cup \{v^-\} \quad (2)$$

$\mathcal{F}$ is the feedback provided at the current turn, c_i^+ is a concept that u likes, and c_i^- (or v^-) is a concept (or an item) that u is not interested in. To collect feedbacks about concepts, we can ask questions like "*Are you interested in [CONCEPT]?*") and parse feedbacks with aspect-level sentiment analysis tools [Zhang *et al.*, 2014]. We may also collect concept-level feedbacks by requiring users to respond with pre-defined feedback templates. We set v^- to the last item recommended to the user. If u likes the last recommended item, the conversation ends and no feedbacks will be provided.

3 Model Description

Fig. 3 shows the incremental multi-task learning framework of our **E**explainable **C**onversational **R**ecommendation (ECR) model, which consists of two major parts. The first part, **incremental cross knowledge modeling**, learns the transferred
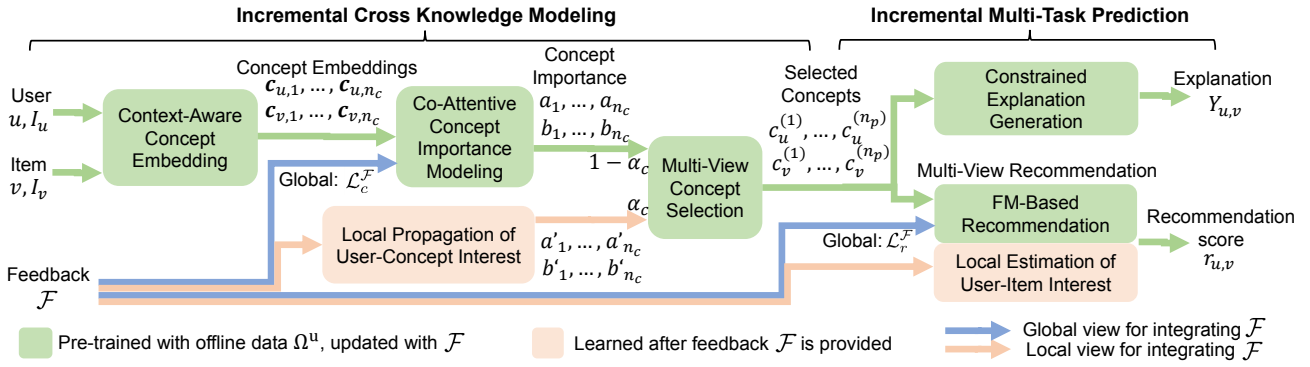
Figure 3: Our incremental multi-task learning framework for explainable conversational recommendation (ECR)

cross knowledge for the recommendation task and the explanation task, and illustrates how the cross knowledge can be updated by using incremental learning. The second part, **incremental multi-task prediction**, illustrates how we generate explanations based on the cross knowledge, and how we predict recommendation scores based on both the cross knowledge and user feedbacks.

We will introduce two parts in detail and illustrate how they can be jointly optimized to achieve end-to-end learning.

3.1 Incremental Cross Knowledge Modeling

A prior study shows that the transferred cross knowledge for the recommendation task and the explanation task can be modeled by using the key concepts that a user likes about an item [Chen *et al.*, 2019]. We first illustrate how their concept embedding method can be extended to facilitate incremental feedback and improve model performance. Then, we show how concept importance can be computed by combining two views of incremental learning. Finally, we introduce how to select key concepts by considering concept importance from both views. The selected concepts will be included in the explanations to trigger feedbacks and used for recommendation prediction. As shown in Fig. 3, the incremental cross knowledge modeling part consists of four major modules:

Context-aware concept embedding. For each user u, we collect concepts $c_{u,1}, .., c_{u,n_c}$ that appear in reviews $\mathcal{D}_{u,1}, ..., \mathcal{D}_{u,n_d}$ and calculate their latent representations. While a concept can be represented by using its word embedding [Chen *et al.*, 2019], we also find its context, i.e., related reviews, to be important for concept modeling. Thus, our embedding $\mathbf{c}_{u,i}$ of concept $c_{u,i}$ is a concatenation of its word embedding and context embedding: $\mathbf{c}_{u,i} = [\mathbf{c}_{u,i}^w; \mathbf{c}_{u,i}^n]$, where $\mathbf{c}_{u,i}^w$ is computed by using a word embedding lookup layer, and $\mathbf{c}_{u,i}^n$ is calculated by averaging the embeddings of related reviews, i.e., $\mathbf{c}_{u,i}^n = (\sum_{j \in \Gamma_{u,i}} \mathbf{d}_{u,j})/|\Gamma_{u,i}|$. Here, $\mathbf{d}_{u,j}$ is the embedding of review $\mathcal{D}_{u,j}$, and $\Gamma_{u,i}$ denotes the set of reviews that contains concept $c_{u,i}$, i.e., $\Gamma_{u,i} = \{j|c_{u,i} \in \mathcal{D}_{u,j}\}$. We follow [Tay *et al.*, 2018] to calculate the review embedding $\mathbf{d}$ by summing up its word embeddings: $\mathbf{d} = \sum_{\omega \in \mathcal{D}} \mathbf{w}$.

Given item v, we can similarly collect concepts in the reviews of v and calculate their context-aware embeddings.

Co-attentive concept importance modeling. Given concept embeddings $\mathbf{c}_{u,1}, ..., \mathbf{c}_{u,n_c}$ of user u and concept em-

beddings $\mathbf{c}_{v,1}, ..., \mathbf{c}_{v,n_c}$ of item v, we calculate co-attention weight matrix $\mathbf{\Phi} \in \mathcal{R}^{n_c \times n_c}$ to model deep user-item interactions: $\phi_{i,j} = f(\mathbf{c}_{u,i})^{\mathrm{T}} \mathbf{W}_c f(\mathbf{c}_{v,j})$, where $\mathbf{W}_c \in \mathcal{R}^{l_w \times l_w}$ is a learning weight matrix and $f(\cdot)$ is a l_f-layered feedforward neural network with activation function ReLU. We obtain user (item) concept importance vectors $\mathbf{a}$ and ($\mathbf{b}$) to find concepts with the maximum co-attentions:

$$a_i = \zeta(\max_{j=1,...,n_c} \phi_{i,j}) \text{ and } b_j = \zeta(\max_{i=1,...,n_c} \phi_{i,j})$$

Here, $\zeta(\cdot)$ is a Softmax function that ensures $\sum_i a_i = 1$ and $\sum_j b_j = 1$. a_i (or b_j) represents the probability that the concept in the reviews of u (or v) is important for the user-item pair. Concepts with larger a_i or b_j values have a larger probability to be selected and included in the explanations.

User feedbacks $\{c_i^+\}$ and $\{c_j^-\}$ provide ground-truth labels for important and unimportant concepts. These labels can be naturally incorporated by using a concept-level feedback loss:

$$\mathcal{L}_c^{\mathcal{F}} = \sum_i \log(1 - a_i^+) + \sum_j \log(a_j^-) + \\ \sum_i \log(1 - b_i^+) + \sum_j \log(b_j^-) \quad (3)$$

Here, a_i^+, b_i^+ (or a_j^-, b_j^-) are the entries in $\mathbf{a}$ and $\mathbf{b}$ that correspond to c_i^+ (or c_j^-) and denote the probabilities for c_i^+ (or c_j^-) being considered important.

We consider minimizing $\mathcal{L}_c^{\mathcal{F}}$ as a **global** view of incremental learning, since it fixes model imperfections that *affect all user-item pairs*, e.g., it refines concept embeddings. While this method is intuitive and useful for parameter refining, it usually fails to ensure satisfaction of user needs, e.g., significantly reduce the importance of c_j^- and remove it from the explanations. This is because $\mathcal{L}_c^{\mathcal{F}}$'s impact on a_i and b_j is indirect, i.e., can only be achieved by changing model parameters. Since the number of feedbacks is small, the influence of this indirect method is limited.

Local propagation of user-concept interest. To solve the aforementioned issue of indirect feedback integration, we use local label propagation, in which importance of concepts are measured according to their distance with c_i^+ and c_j^- (the labels). To this end, we first calculate the aggregated embedding $\widehat{\mathbf{c}}^+ = \frac{\alpha_t}{n_g}(\sum_{i=1}^{n_g} \mathbf{c}_i^+) + (1 - \alpha_t)\mathbf{p}^+$, where $\mathbf{p}^+$ records the value of $\widehat{\mathbf{c}}^+$ at the last turn, and $0 < \alpha_t \leq 1$ balances current and previous feedbacks. Similarly, we have

$\widehat{\mathbf{c}}^- = \frac{\alpha_t}{n_b}(\sum_{i=1}^{n_b}\mathbf{c}^-)+(1-\alpha_t)\mathbf{p}^-$. We then compute a preference score β_c of the concept c based on the 2-norm distances between its embedding and the aggregated embeddings:

$$\beta_c = (||\mathbf{c} - \widehat{\mathbf{c}}^+|| - ||\mathbf{c} - \widehat{\mathbf{c}}^-||)/(||\widehat{\mathbf{c}}^+ - \widehat{\mathbf{c}}^-||) \quad (4)$$

Accordingly to the triangle inequality, $-||\widehat{\mathbf{c}}^+ - \widehat{\mathbf{c}}^-|| \leq (||\mathbf{c} - \widehat{\mathbf{c}}^-|| - ||\mathbf{c} - \widehat{\mathbf{c}}^+||) \leq ||\widehat{\mathbf{c}}^+ - \widehat{\mathbf{c}}^-||$. Thus, $\beta_c \in [-1, 1]$ and obtains the maximum (or minimum) value when $\mathbf{c} = \widehat{\mathbf{c}}^+$ (or $\mathbf{c} = \widehat{\mathbf{c}}^-$). The local concept importance vectors can be obtained by: $a_i' = \zeta(\beta_{c_{u,i}})$ and $b_j' = \zeta(\beta_{c_{v,j}})$.

Multi-view concept selection. The multi-view concept importance are computed with:

$$\bar{a}_i = (1-\alpha_c)a_i + \alpha_c a_i', \text{ and } \bar{b}_j = (1-\alpha_c)b_j + \alpha_c b_j' \quad (5)$$

where $\alpha_c \in \mathcal{R}$ is a learnable parameter. We further ensure that the multi-view importance match user-concept interests reflected in the feedbacks by using a loss $\mathcal{L}_d^{\mathcal{F}}$ similar with $\mathcal{L}_c^{\mathcal{F}}$:

$$\mathcal{L}_d^{\mathcal{F}} = \sum_i \log(1-\bar{a}_i^+) + \sum_j \log(\bar{a}_j^-) + \\ \sum_i \log(1-\bar{b}_i^+) + \sum_j \log(\bar{b}_j^-) \quad (6)$$

Based on $\bar{a}_i$ and $\bar{b}_j$, we select a set of key concepts that will be included in the explanations and considered in recommendation prediction. To avoid the model being non-differentiable, we use the Straight-Through Gumbel-Softmax function $Gumbel(\cdot)$ [Tay $et\ al.$, 2018]:

$$\mathbf{z}_u = Gumbel(\log\bar{\mathbf{a}}) \text{ and } \mathbf{z}_v = Gumbel(\log\bar{\mathbf{b}}) \quad (7)$$

Here, $\mathbf{z}_u = (z_{u,1}, ..., z_{u,n_c}) \in \{0,1\}^{n_c}$. $z_{u,i}$ is 1 (or 0) if $c_{u,i}$ is (or is not) selected. Similarly, $\mathbf{z}_v = (z_{v,1}, ..., z_{v,n_c}) \in \{0,1\}^{n_c}$ denotes whether concepts $\{c_{v,j}\}$ are selected. Each time we calculate $Gumbel(\cdot)$, only one concept will be selected, i.e., $\sum_i z_{u,i} = 1$. However, a user may consider multiple concepts of an item. To select n_p concepts, we run $Gumbel(\cdot)$ n_p times with different Gumbel noises and co-attention weight matrices $\mathbf{W}_c$. As a result, we obtain multiple selected concepts $\{c_u^{(1)}, ..., c_u^{(n_p)}\}$ and $\{c_v^{(1)}, ..., c_v^{(n_p)}\}$.

3.2 Incremental Multi-Task Prediction

We predict recommendation scores and generate explanations based on the selected concepts. Considered as concepts that user u likes about item v, the selected concepts can be used to 1) calculate user and item embeddings and 2) determine the key words that must appear in the explanations.

We first leverage the context-aware embeddings of the selected concepts to learn user and item embeddings $\mathbf{e}_u = \sigma(\mathbf{W}_p[\mathbf{c}_u^{(1)}, ..., \mathbf{c}_u^{(n_p)}] + \mathbf{b}_p)$ and $\mathbf{e}_v = \sigma(\mathbf{W}_p[\mathbf{c}_v^{(1)}, ..., \mathbf{c}_v^{(n_p)}] + \mathbf{b}_p)$. Here $\mathbf{W}_p \in \mathcal{R}^{l_p \times 2n_p l_w}$, $\mathbf{b}_p \in \mathcal{R}^{l_p}$ and σ is the sigmoid function. Since explicit factors such as concepts and related reviews may fail to include all information about a user or an item, we additionally compute implicit user and items representations. In particular, a lookup layer is used to transform u (v) into implicit representations $\mathbf{h}_u \in \mathcal{R}^{l_u}$ ($\mathbf{h}_v \in \mathcal{R}^{l_v}$). The final user and embeddings are $\mathbf{x}_u = [\mathbf{e}_u; \mathbf{h}_u]$ and $\mathbf{x}_v = [\mathbf{e}_v; \mathbf{h}_v]$, which capture both explicit and implicit factors about a user or an item.

Multi-view recommendation. We predict recommendation scores based on two views. The global view learns a recommendation model based on all training samples. The local view ensures satisfaction of user feedbacks by considering local user-item interest.

Global FM-based recommendation. Factorization machine (FM) [Rendle, 2010] is used to predict recommendation score

$$r_{u,v} = w_0 + \sum_{i=1}^{n} w_i q_i + \sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle\mathbf{m}_i, \mathbf{m}_j\rangle q_i q_j \quad (8)$$

where $q_i \in \mathcal{R}$ is the i-th entry of $\mathbf{q} = [\mathbf{x}_u; \mathbf{x}_v]$. $w_i \in \mathcal{R}$ and $\mathbf{m}_i \in \mathcal{R}^{l_m}$ are parameters to be learned.

The FM can be pre-trained with offline training instances Ω_u by using the BPR loss [Rendle $et\ al.$, 2009].

$$\mathcal{L}_r^{\Omega_u} = -\frac{1}{|\Omega_u|}\sum_{v' \in \Omega_u}\ln\sigma(r_{u,v'} - r_{u,v''}) \quad (9)$$

$v' \in \Omega_u$ denotes an item that u likes. For each v', an item v'' that u does not like is obtained through negative sampling.

Items v^- provided in $\mathcal{F}$ can be integrated similarly:

$$\mathcal{L}_r^{\mathcal{F}} = -\frac{1}{|\Omega_u|}\sum_{v' \in \Omega_u, v^- \in \mathcal{F}}\ln\sigma(r_{u,v'} - r_{u,v^-}) \quad (10)$$

Local estimation of user-item interest. Given $\mathcal{F}$, we can compute a recommendation score based purely on whether feedbacks are satisfied. Specifically, estimated score $\widehat{r}_{u,v}$ is set to 0 if $v=v^-$ or if none of $c_1^+, ...c_{n_g}^+$ is contained in the reviews of v. Otherwise, $\widehat{r}_{u,v}$ is set to 1. We then calculate the local recommendation score $r_{u,v}'$ with: $r_{u,v}' = \alpha_t\widehat{r}_{u,v} + (1-\alpha_t)\tau_{u,v}$, where $\tau_{u,v}$ is the $r_{u,v}'$ score at the previous turn.

Multi-view combination. The final recommendation score is obtained by combining the two views: $\bar{r}_{u,v} = (1-\alpha_r)r_{u,v} + \alpha_r r_{u,v}'$, where α_r is a learnable parameter. We use $\mathcal{L}_s^{\mathcal{F}}$ to learn $\bar{r}_{u,v}$:

$$\mathcal{L}_s^{\mathcal{F}} = -\frac{1}{|\Omega_u|}\sum_{v' \in \Omega_u, v^- \in \mathcal{F}}\ln\sigma(\bar{r}_{u,v'} - \bar{r}_{u,v^-}) \quad (11)$$

Constrained explanation generation. We generate an linguistic explanation $Y_{u,v}$ by considering two types of constraints. The first type is a hard constraint, which requires that the first selected concept $c_v^{(1)}$ must appear in the explanations. This better ensures that a concept that u likes will be included in the explanations. Note that $c_v^{(1)}$ can only be selected if it appears in the item reviews. Thus, requiring $c_v^{(1)}$ to appear in the explanation will not introduce false claims about item v in the explanation. The second type is soft constraints, which punish the model if other selected concepts are not included.

To satisfy the first constraint, we use a bi-directional generation method based on Gated Recurrent Unit (GRU) [Chung $et\ al.$, 2014]. In particular, two GRUs are learned: a backward GRU_b and a forward GRU_f. GRU_b considers $c_v^{(1)}$ as the first generated word, and outputs all words before $c_v^{(1)}$. When GRU_b reaches the start of the explanation (<SOS>), GRU_f starts to generate words after $c_v^{(1)}$ by considering all words output by GRU_b. In both GRUs, the initial state is set to $\tanh(\mathbf{W}_u\mathbf{x}_u + \mathbf{W}_v\mathbf{x}_v + \mathbf{b}_s) \in \mathcal{R}^{l_s}$, where $\mathbf{W}_u, \mathbf{W}_v, \mathbf{b}_s$ are parameters to be learned. The two GRUs can be trained by using the widely-adopted negative log-likelihood loss

$$\mathcal{L}_n^{\Omega_u} = \frac{1}{|\Omega_u|}\sum_{v' \in \Omega_u}\sum_{t=1}^{T}(-\log o_{t,y_{u,v',t}*}) \quad (12)$$

where $y_{u,v',t}*$ is the ground truth word at time t for the (u, v') pair and $o_{t,y}$ is the probability of word y being generated.

To satisfy the second constraint, we use the concept relevance loss [Chen *et al.*, 2019]. Let us denote the selected concepts as $\mathbf{s} \in \mathcal{R}^{|\mathcal{V}|}$, where $|\mathcal{V}|$ is vocabulary size. s_k is 1 when the k-th word is a concept and has been selected for the (u, v') pair and is 0 otherwise. The concept relevance loss is

$$\mathcal{L}_c^{\Omega_u} = \frac{1}{|\Omega_u|} \sum_{v' \in \Omega_u} \sum_{t=1}^{T} (\max_k \ (-s_k \log o_{t,k})) \quad (13)$$

3.3 Learning Process

As shown in Fig. 3, the model is first pre-trained with offline data Ω_u. This is achieved by minimizing the following loss

$$\mathcal{L}^{\Omega} = \sum_{u \in U} (\mathcal{L}_r^{\Omega_u} + \lambda_n \mathcal{L}_n^{\Omega_u} + \lambda_c \mathcal{L}_c^{\Omega_u}) + \lambda_\theta \|\Theta\|_2^2 \quad (14)$$

where λs denote the importance of different losses and Θ represents the model parameters. During pre-training, local views are disabled, i.e., α_c and α_r are set to 0. We choose Adam [Kingma and Ba, 2014] as the optimizer.

When a feedback $\mathcal{F}$ is provided by u, we incrementally update the previous model parameters and learn local use interests. This is achieved by minimizing

$$\mathcal{L}^{\mathcal{F}} = \mathcal{L}_c^{\mathcal{F}} + \lambda_d \mathcal{L}_d^{\mathcal{F}} + \lambda_r \mathcal{L}_r^{\mathcal{F}} + \lambda_s \mathcal{L}_s^{\mathcal{F}} + \lambda_\theta' \|\Theta - \Theta'\|_2^2 \quad (15)$$

Here, Θ' represents the previous model parameters. When a sequence of feedbacks $\mathcal{F}_1, ..., \mathcal{F}_T$ are given, we incrementally update the model by processing $\mathcal{F}_t$ one by one.

4 Experiment

4.1 Experimental Settings

Dataset. As shown in Table 1, we use three publicly available datasets. **Electronics** and **Movie&TV** are two categories of the Amazon dataset[2], and **Yelp** contains restaurant reviews from Yelp Challenge 2016[3]. Each dataset is split into a training set (80%), a validation set (10%) and a test set (10%). The training set is used to derive Ω^u, which consists of items that u provided reviews for. Following [Li *et al.*, 2017; Chen *et al.*, 2019], we consider the **ground-truth explanations** as the first sentence in the reviews. The validation set is used for model hyperparameter tuning and the test set is leveraged for simulating conversations. The goal of the model is to correctly predict the ground-truth items and explanations in the test set. If it fails to recommend the ground-truth item, user feedbacks will be provided to help it refine the results.

Feedback simulation. Most works on conversational search and recommendation directly use ground-truth concepts (or features) as user feedbacks [Bi *et al.*, 2019]. In our setting, the ground-truth concepts are the ones mentioned in the ground-truth explanations (reviews). Different from previous works, we assume that a user will only give a ground-truth concept c_i^+ when the generated explanation contains a concept that is similar with c_i^+. We consider the cosine similarity between BERT embeddings [Devlin *et al.*, 2018] of the concepts and assume that the 20 most similar concepts of c_i^+ will trigger c_i^+. We further assume that the user will point out a unimportant concept c_i^- if it appears in the explanation. Effects of different simulation settings are evaluated in Sec. 4.3.

[2] http://jmcauley.ucsd.edu/data/amazon/
[3] https://www.yelp.com/dataset/challenge

Dataset	Users	Items	Reviews	Concepts
Electronics	146,481	50,526	844,702	652
Movies&TV	90,227	42,553	830,004	791
Yelp	96,304	39,596	769,991	1,004

Table 1: Statistics of three experiments datasets.

Baselines. We find that no existing works can be used directly for explainable conversational recommendation. To evaluate our method, we design two baselines by extending existing explainable recommendation methods **NRT** [Li *et al.*, 2017] and **CAML** [Chen *et al.*, 2019]. We train both models with the BPR loss. NRT is updated with $\lambda_r \mathcal{L}_r^{\mathcal{F}} + \lambda_\theta' \|\Theta - \Theta'\|_2^2$. Other feedback losses cannot be used because NRT does not model concepts and cannot perform multi-view incremental learning. CAML is updated similarly, except that the concept-level loss $\mathcal{L}_c^{\mathcal{F}}$ is also used.

Evaluation measures. We evaluate recommendation accuracy by using three widely-adopted measures: **HR** (hit ratio), **NDCG** (normalized discounted cumulative gain), and **MRR** (mean reciprocal rank). Following [Zhang *et al.*, 2018], we calculate HR, NDCG and MRR based on the top 1, 10 and 100 recommended item(s), respectively.

Following [Chen *et al.*, 2019], the explanation quality is evaluated by using **BLEU** [Papineni *et al.*, 2002] and **ROUGE-L** [Lin, 2004], which are widely adopted to measure the similarity between ground truth and generated texts.

We also propose criterion **CSR** to measure the **C**oncept-level feedback **S**atisfcation **R**atio. We consider c_i^+ to be satisfied if it appears in the generated explanation, and c_i^- to be satisfied if the concept is removed from the explanation.

Implementation details. For all methods, the maximum number of conversation turns is set to 5 and negative sampling is used to reduce the candidate size to 256. Most hyperparameters are set and tuned by following the papers of the baselines [Li *et al.*, 2017; Chen *et al.*, 2019]. The λs for balancing different losses are set to 1 except for λ_c=0.05. We tune n_p by performing grid search over $\{1, 2, .., 5\}$. The learning rates of L^Ω and $L^\mathcal{F}$ are set to 10^{-3} and 10^{-2}, respectively. α_t is set to 0.8, and α_c, α_r are initialized by using 0.9 and automatically tuned during the learning process.

4.2 Overall Performance

Performance after 5 turns of conversation. Table 2 compares our method with the baselines in terms of objectives *O1–O3*. To facilitate comparison, we also test Ours-G. Similar with CAML, Ours-G considers only the global view (α_c=α_r=λ_d=λ_s=0). Note that all the evaluated methods take concept feedbacks as inputs except for NRT.

The results show that our method significantly outperforms the baselines in terms of all three objectives. For example, compared with CAML, our method achieves 41.9% to 121.8% better HR (recommendation accuracy) and 42.0% average ROUGE-L improvement (explainability). Moreover, our method can almost always satisfy concept-level user feedbacks, with the CSR score ranging from 94% to 97%.

We draw three conclusions from the results. First, concept-level feedbacks are useful for model improvement, since the other methods perform better than NRT. Second, our model

Dataset		Electronics				Movie&TV				Yelp			
Metric		NRT	CAML	Ours-G	Ours	NRT	CAML	Ours-G	Ours	NRT	CAML	Ours-G	Ours
O1	HR	0.176	0.202	0.272	**0.448**	0.194	0.272	0.274	**0.386**	0.204	0.234	0.258	**0.432**
	NDCG	0.285	0.311	0.312	**0.534**	0.309	0.314	0.354	**0.474**	0.310	0.335	0.343	**0.510**
	MRR	0.244	0.279	0.303	**0.500**	0.266	0.285	0.427	**0.439**	0.272	0.298	0.313	**0.479**
O2	BLEU	1.04	1.12	1.49	**2.10**	1.49	1.47	1.53	**2.31**	1.26	1.15	1.52	**2.17**
	ROUGE-L	13.16	14.74	15.73	**19.88**	14.43	14.56	15.78	**18.92**	11.36	12.00	13.37	**19.34**
O3	CSR	0.08	0.12	0.38	**0.94**	0.20	0.19	0.45	**0.97**	0.14	0.13	0.39	**0.97**

Table 2: Comparison of recommendation accuracy (*O1*), explainability (*O2*), and feedback satisfaction ratio (*O3*) after 5 conversation turns. Ours-G is a variant of our method that considers only the global view. Ours-G and CAML use the same loss function for feedback integration.
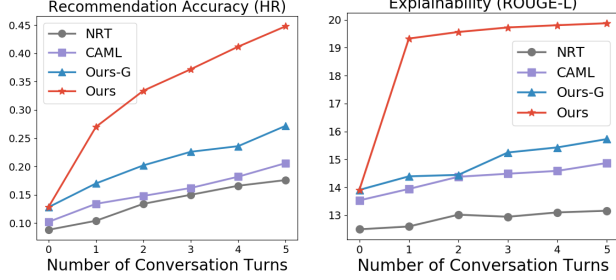

Figure 4: Recommendation accuracy and explainability on Electronics at different conversation turns.
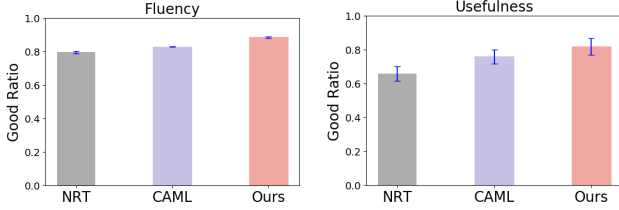

Figure 5: Human evaluation of explanation fluency and usefulness.

structure better facilitates feedback integration compared with CAML, since Ours-G outperforms CAML even when they use the same feedback integration method. This may be caused by our design of context-aware concept embeddings. Third, Ours outperforms Ours-G, which illustrates the importance of combining the global view with the local view.

Performance gain during conversation. Fig. 4 shows the recommendation accuracy and explainability of four methods at each conversation turn. As the number of conversation turns increases, the number of feedbacks increases, and the performance of all methods improves. Among all methods, ours increases the fastest, followed by Ours-G, CAML, and NRT. The explanation quality is highly related to the accuracy of selected concept and may become saturated once we find the ground-truth concept.

Human evaluation. Fig. 5 shows the human evaluation results on explanation quality, which illustrates that our explanations are considered more fluent and useful than the baselines. We hire three experienced human assessors to label the explanations generated after 5 turns of conversation. 100 test cases are sampled from the Electronics dataset, and each assessor labels whether an explanation is fluent and whether it is useful. Explanations of NRT, CAML, and Ours are provided to the assessors in random order. The Fleiss Kappa score is 0.423 for fluency and 0.303 for usefulness, which shows a moderate agreement and a fair agreement among assessors.

Method	View		User Simulation		
Metric	Ours-G	Ours-L	Ours-P	Ours-N	Ours
HR	0.272	0.230	0.406	0.272	**0.448**
NDCG	0.312	0.316	0.501	0.321	**0.534**
MRR	0.303	0.287	0.464	0.307	**0.500**
BLEU	1.49	1.60	1.89	1.13	**2.10**
ROUGE-L	15.73	19.51	19.69	14.37	**19.88**
CSR	0.38	**1.00**	0.93	0.99	0.94

Table 3: Ablation study on Electronics with 5 conversation turns. Ours-G (Ours-L) uses only the global (local) view. In Ours-P (Ours-N), simulated users provide only $c_1^+, ...c_{n_g}^+$ ($c_1^-, ...c_{n_b}^-$).

4.3 Ablation Analysis

Effectiveness of multi-view learning. Table 3 shows that our final model (Ours) outperforms the model with only the global (Ours-G) or local (Our-L) view in terms of recommendation accuracy and explainability. This demonstrates the effectiveness of multi-view learning. The reason is that Ours-G has not fully integrated user feedbacks while Ours-L ignores users' personal interests. The CSR of Ours-L is 1.0 since it considers only user feedbacks.

Effect of different user simulations settings. Table 3 shows the result of our model when only $c_1^+, ...c_{n_g}^+$ (Ours-P) or $c_1^-, ...c_{n_b}^-$ (Ours-N) are provided as feedbacks. Ours outperforms Ours-P and Ours-N, which indicates that our method can leverage both types of user feedbacks to improve performance. By comparing this result with the result at conversation turn 0 (Fig. 5), we can find that Ours-N successfully improves HR and explainability by using only 5 negative feedbacks (no ground-truth concepts provided).

5 Conclusion

We propose a framework for explainable conversational recommendation, which enables tight collaboration between the recommendation task, the explanation generation task, and the incremental feedback integration module. A multi-view method is also proposed to effectively incorporate user feedbacks. Experiments show that our approach achieves stable and significant improvement of both recommendation and explainability, and can effectively satisfy user requirements.

Acknowledgments

References

[Bi *et al.*, 2019] Keping Bi, Qingyao Ai, Yongfeng Zhang, and W. Bruce Croft. Conversational product search based on negative feedback. In *Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM*, pages 359–368, 2019.

[Chen *et al.*, 2018] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *Proceedings of the 2018 World Wide Web Conference, WWW*, 2018.

[Chen *et al.*, 2019] Zhongxia Chen, Xiting Wang, Xing Xie, Tong Wu, Guoqing Bu, Yining Wang, and Enhong Chen. Co-attentive multi-task learning for explainable recommendation. In *IJCAI*, pages 2137–2143, 2019.

[Christakopoulou *et al.*, 2016] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. Towards conversational recommender systems. In *SIGKDD*, pages 815–824, 2016.

[Chung *et al.*, 2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Gao *et al.*, 2019] Jingyue Gao, Xiting Wang, Yasha Wang, and Xing Xie. Explainable recommendation through attentive multi-view learning. In *AAAI*, 2019.

[He *et al.*, 2015] Xiangnan He, Tao Chen, Min-Yen Kan, and Xiao Chen. Trirank: Review-aware explainable recommendation by modeling aspects. In *Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM*, pages 1661–1670, 2015.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li *et al.*, 2017] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. Neural rating regression with abstractive tips generation for recommendation. In *The International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 345–354, 2017.

[Li *et al.*, 2018] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. In *Advances in Neural Information Processing Systems, NIPS*, pages 9725–9735, 2018.

[Lin, 2004] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*, 2004.

[Papineni *et al.*, 2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the annual meeting on association for computational linguistics, ACL*, pages 311–318, 2002.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In *UAI*, pages 452–461, 2009.

[Rendle, 2010] Steffen Rendle. Factorization machines. In *IEEE International Conference on Data Mining*, 2010.

[Sharma and Cosley, 2013] Amit Sharma and Dan Cosley. Do social explanations work?: studying and modeling the effects of social explanations in recommender systems. In *Proceedings of the 2013 World Wide Web Conference, WWW*, pages 1133–1144, 2013.

[Sun and Zhang, 2018] Yueming Sun and Yi Zhang. Conversational recommender system. In *The International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2018.

[Tay *et al.*, 2018] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. Multi-pointer co-attention networks for recommendation. *SIGKDD*, 2018.

[Tintarev and Masthoff, 2007] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *2007 IEEE international conference on data engineering workshop*, pages 801–810, 2007.

[Wang *et al.*, 2015] Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao. An inference approach to basic level of categorization. In *Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM*, pages 653–662, 2015.

[Wang *et al.*, 2018] Xiang Wang, Xiangnan He, Fuli Feng, Liqiang Nie, and Tat-Seng Chua. Tem: Tree-enhanced embedding model for explainable recommendation. In *Proceedings of the 2018 World Wide Web Conference, WWW*, pages 1543–1552, 2018.

[Wu *et al.*, 2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, pages 481–492, 2012.

[Zhang *et al.*, 2014] Yongfeng Zhang, Guokun Lai, Min Zhang, Yi Zhang, Yiqun Liu, and Shaoping Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *The International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2014.

[Zhang *et al.*, 2018] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM*, 2018.

[Zhao *et al.*, 2020] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs. *The International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. *To Appear*.