2018 IEEE International Conference on Data Mining Workshops (ICDMW)

Why You Should Listen to This Song: Reason Generation for Explainable Recommendation

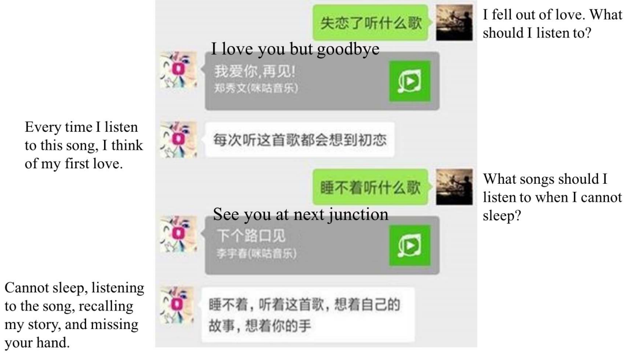
Guoshuai Zhao[⋆¶], Hao Fu[†], Ruihua Song[†], Tetsuya Sakai[‡], Xing Xie[∗], Xueming Qian[⋆§]
[⋆]Xi'an Jiaotong University, China. Email: {zgs2012@stu; qianxm@mail}.xjtu.edu.cn
[†]Microsoft XiaoIce, China. Email: {fuha; song.ruihua}@microsoft.com
[‡]Waseda University, Japan. Email: tetsuyasakai@acm.org
[∗]Microsoft Research Asia, China. Email: xing.xie@microsoft.com

***Abstract*—Explainable recommendation, which makes a user aware of why such items are recommended has received a lot of attention as a highly practical research topic. The goal of our research is to make the users feel as if they are receiving recommendations from their friends. To this end, we formulate a new challenging problem called reason generation for explainable recommendation in conversation applications, and propose a solution that generates a natural language explanation of the reason for recommending an item to that particular user. Evaluation with manual assessments indicates that our generated reasons are relevant to songs and personalized to users. They are also fluent and easy to understand. A large-scale online experiments show that our method outperforms manually selected reasons by 8.2% in terms of click-through rate.**

***Index Terms*—Conversational recommendation, explainable recommendation, natural language generation, personalization, recommender system**

I. Introduction

Personalized recommendation is considered an effective approach to solving the "consumer product overload problem" in the digital era. Besides the problem of *what* should be recommended, *why* they should be recommended has received a lot of attention. Explainable recommendation has received a lot of attention helps improve the effectiveness, efficiency, persuasiveness, and user satisfaction of recommender systems [1], [2]. However, existing recommender systems provide recommendations with a generic explanation ("Customers who bought this item also bought...") or some feature-based explanations ("You may like this item because it is good at these features...") which do not necessarily encourage the user to accept (i.e., click on) the recommended item. In light of the above consideration, we formulate a problem which we call *reason generation* for explainable recommendation in conversation applications, where our goal is to increase the click through rate of the recommended items by automatically generating recommendation reasons tailored to this goal. In the present study, we focus on the song recommendation domain, as we are working to improve a conversational song recommendation functionality of XiaoIce chatbot[1], which has over 100 million users as of May 2018.



失恋了听什么歌

I fell out of love. What should I listen to?

I love you but goodbye

我爱你,再见!
郑秀文(咪咕音乐)

每次听这首歌都会想到初恋

Every time I listen to this song, I think of my first love.

睡不着听什么歌

What songs should I listen to when I cannot sleep?

See you at next junction

下个路口见
李宇春(咪咕音乐)

睡不着，听着这首歌，想着自己的故事，想着你的手

Cannot sleep, listening to the song, recalling my story, and missing your hand.

Fig. 1. A snapshot of XiaoIce chatbot recommending a user songs followed by corresponding reasons during conversations.

Fig. 1 demonstrates how our explainable song recommendation actually works in XiaoIce chatbot. In this example, the user says "I fell out of love. What should I listen to?" XiaoIce chatbot responds by recommending a song named *I love you but goodbye*, while providing a reason for the recommendation: "Every time I listen to this song, I think of my first love." When the user cannot sleep, it recommends a song with providing a reason: "Cannot sleep, listening to the song, recalling my story, and missing your hand."

There are several challenges to generating effective reasons in explainable song recommendation via conversations. First, we do not have any existing data that consists of actual song recommendations from friends; instead, what we have are the comments on songs posted to a music website, and only some of them can be regarded as recommendation reasons. Second, we aim to give a personalized reason to a specific user according to the description of his/her general interests, and current status, although music websites lack these kinds of user tags. Furthermore, even if we are able to leverage the comments from the music website to generate recommendations for users, simply retrieving and reusing the comments would not be able to handle recommendations of *new* songs (i.e., songs for which we do not have any comment data from the music website) at all.

To address the aforementioned challenges, we build data

[¶]The work was done when the first author was an intern at Microsoft.
[§]Corresponding author
[1]http://www.msxiaoice.com/

2375-9259/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDMW.2018.00187

sets that consist of (song, user tag, reason) triplets, and propose a method that learns to generate recommendation reasons. First, from a music website, we extract song comments that can be regarded as recommendation reasons. Second, we collect reasons related to a particular user tag. Third, we use an encoder-decoder framework with attention [3]–[6], to generate a recommendation reason for a particular song and a user. Our experimental results indicate that our proposed method significantly improves on a Factorization Machine baseline. Furthermore, we deploy our proposed methods on XiaoIce chatbot, and observe that the click-through rate of recommended songs improves by at least 8.2% over four different baselines.

Our main contributions are as follows:

- We formulate a novel problem, namely, reason generation for explainable recommendation in conversation applications. The task is to generate a personalized reason to make the user feel as if she is receiving a recommendation from her friends.
- We demonstrate how to overcome the problem posed by a lack of training data for generating conversational reasons in the song recommendation domain. Moreover, our method is extensible to other recommendation domains.
- We propose fusing user tags with the information of songs into the encoder-decoder model with attention to generate personalized reasons. Experiments show the effectiveness of our method, and its deployment on XiaoIce chatbot improves the click-through rate substantially.

The rest of this paper is organized as follows. We start with an overview of related work in Section II. Section III presents the details of our approach. Experiment results and discussions are given in Section IV, and Section V concludes this paper.

II. Related Work

Our work is related to two groups of work: explainable recommender systems and conversation systems.

A. Explainable Recommender Systems

Instead of simply presenting recommended items to the user in traditional recommender systems [7]–[19], some researchers have tried to mine the reasons behind recommendations [20]–[26]. Explainable recommender systems can be grouped into four categories: 1) User-based or item-based explainable systems; 2) Social-based explainable systems; 3) Feature-based explainable systems; 4) Review-based explainable systems.

The fundamental idea of collaborative filtering is looking for similar users and recommending the items they are interested in. Schafer et al. [27] state that a recommender system would be used to explain to a user what type of thing a product is, such as "this product you are looking at is similar to these other products that you have liked in the past", which is the main idea of item-based collaborative filtering [12] [28]. Among social-based explainable systems [22] [29], Wang et al. [29] generate social explanations such as "A and B also like the item". They propose generating the persuasive social explanation by recommending the optimal set of users to be put in the explanation. Feature-based explanations can also be seen as content-based explanations. They provide recommendations by matching user preference with the available item content features [21], [24], [30]–[33]. For example, Zhang et al. [30] use phrase-level sentiment analysis to mine the explicit features of items and the corresponding sentiment polarity of the user. They propose Explicit Factor Models (EFM) to fit user-item ratings by the latent representations. The features can also be displayed in different forms [32]–[34]. In review-based explainable systems [26] [35], Chen et al. [26] introduce an attention mechanism to explore the usefulness of reviews, and propose a neural attentional rating regression model for recommendation. It can not only predict ratings, but also learn the usefulness of each review simultaneously.

B. Conversation Systems

The availability of large conversational data has enabled rapid development of conversation systems [36]–[38] based on data-driven approaches. One approach is retrieval-based [39]: match a user input with existing question and answer pairs to retrieve the most appropriate responses. Another approach is generation-based [6]: learn a response generation model within a Statistic Machine Translation (SMT) framework from large scale conversation data.

A common generation-based approach treats posts as user inputs, and comments as responses. Response generation can be regarded as translation from posts to comments. Ritter et al. [40] find that SMT techniques are more suitable than information retrieval approaches for the task of response generation. A basic sequence-to-sequence model proposed by Cho et al. [3] consists of two recurrent neural networks (RNNs): an encoder that processes the input and a decoder that generates the output. Multi-layer cells have been successfully used in sequence-to-sequence models by Sutskever et al. [4]. To allow the decoder more direct access to the input, Bahdanau et al. [5] introduce an attention mechanism. Shang et al. [6] propose a neural network-based response generator for Short Text Conversation using the encoder-decoder framework.

III. Our Approach

In this section, we first define the problem and describe our solution to generating a reason that explains why a particular user should listen to a particular song.

A. Problem Formulation and System Overview

In the problem of reason generation for explainable recommendation in conversation applications, we assume that a user U has asked the chatbot to recommend a song and that a recommendation algorithm has returned a song S that is appropriate for the user. Our target is to generate a reason as a sequence of words $Y = (y_1, y_2, \cdots, y_M)$ to explain why the user U should listen to the song S. The generation model maximizes the probability of Y conditioned on S and U: $p(Y|S,U)$.

To train a generation model, we need a data set $\mathcal{D} = (S_i, U_i, Y_i)_{i=1}^{N}$, but there is no existing data of this kind. We

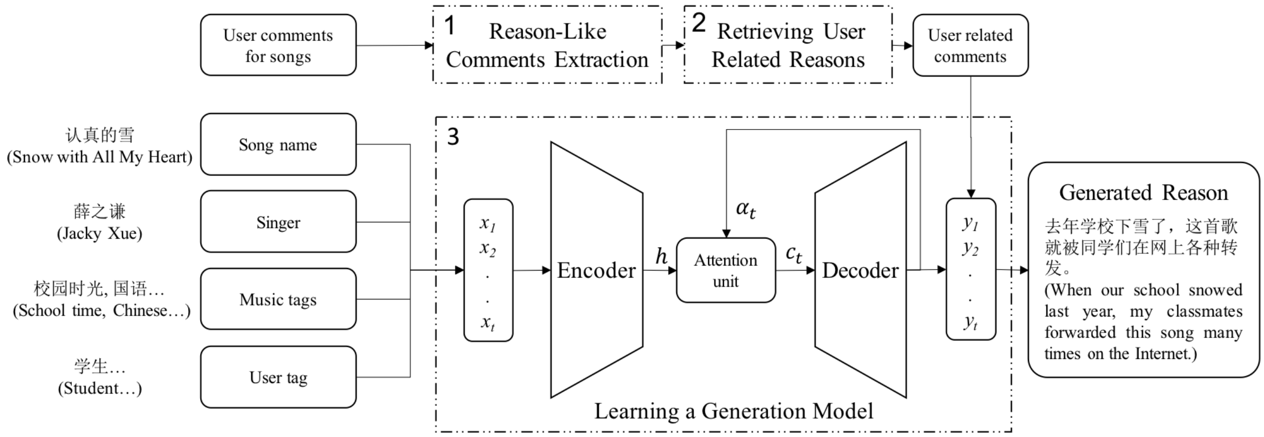


Fig. 2. The flowchart of our proposed solution to generate personalized reasons for a particular song and a particular user.

propose extracting (S_i, Y_i) from comments that users post to songs on a music website. Then we use a method to retrieve relevant (S_i, Y_i) for U_i and thereby to compose the required data set $\mathcal{D}$. For example, in Fig. 2, S is *Snow with All My Heart* sung by Jacky Xue and it has some tags like *school time*, *mandarin*, etc. A user U has the tag *student*, which represents his/her current status and is mined from the user's chat logs. We manage to connect them with the target reason "When our school snowed last year, my classmates forwarded this song many times on the Internet."

When we concatenate the representations of U with S, we can regard them as the source sequence X of a statistic machine translation model. The target sequence is Y. Hence we apply an encoder-decoder framework with attention to maximize the generation probability $p(Y|X)$.

B. Reason-Like Comments Extraction

We crawl 80 million comments of 1.2 million pieces of music from NetEase Music[2]. Most of them are songs. We find that the comments of the following categories are useful as recommendation reasons:

Fact and Opinion Comments of this kind focus on certain facts of the music and its related entities, e.g. "The music reminds me of Totoro and sounds like Joe Hisaishi." These comments are informative recommendation reasons because they provide insight into the commented music.

Emotion and Experience Users may express their personal emotions and past experiences when listening to the music, e.g. "I was about to sleep but now too excited to sleep." These comments can easily resonate with other users.

Joke and Story These comments consist of jokes and stories made up by users, e.g. "Only three persons in this world think you're beautiful: your mom, your dad, and James Blunt." As recommendation reasons, they arouse user's curiosity by making fun of certain facts.

Based on the above observations, we design a two-phase method to extract recommendation reasons from the crawled comments. In the first phase, we extract seed comments with manually crafted rules. The seed comments are designed to be a clean set of recommendation reasons. We craft the rules in an iterative manner. In each iteration, we partition the crawled comments into two sets, namely seed comments and non-seed comments. We then look for new rules that improve the partition. The rules make use of various textual features, e.g., length of comment, usage of punctuations and numbers, language, repeating phrases, and pre-defined keywords. In the second phase, we train a classifier which takes a comment as input and predict if it can be used as a recommendation reason. We take seed comments as positive samples and a random equal-sized set of non-seed comments as negative samples. Features for a comment consist of the features used in the first phase and a feature vector containing character uni-grams. Finally, 4.9 million comments on 0.4 million pieces of music are extracted as recommendation reasons.

C. Retrieving User Related Reasons

As mentioned in Section III-A, a user U is represented as a set of user tags. For simplicity, we consider only one tag per user; for users having more than one tag, we randomly draw one from them. Given a user with a tag, e.g., *student*, how can we obtain recommendation reasons that are suitable for him/her?

NetEase Music does not provide user tags. Hence we rely on the user tags that we mine from chat logs. The user tags are a set of predefined keywords, covering users' status and interests. Status tags describe the current state or lifestyle of a user, e.g., *break-up*, *student*, and *sleep late*. Interest tags represent if a user likes or dislikes certain kind of entities, e.g., color, food, and type of music. The tags are pre-defined and extracted from chat logs with a set of rules. For example, those who chat more often in the mid-night are assigned with

[2] http://music.163.com/

sleep late, and a user message with pattern "I like/love..." indicates an interest tag. Also we filter out user tags that are not suitable for song recommendation: for example, those that express dislike (e.g., *dislike working*). The remaining user tags are used for searching NetEase Music to construct the training data (S_i, U_i, Y_i).

To each user tag, we apply query expansion to enhance recall. Specifically, we first project the user tag into a pre-trained Word2Vec model[3] [41] and discover similar words in terms of cosine similarity. For example, given the user tag *student*, we discover similar words including *teacher*, *worker*, *campus*, and *study*. Next, we manually review the expanded words and filter the irrelevant ones, such as *worker* in the above example. Finally, we retrieve reasons by using these queries.

D. Learning a Generation Model

We choose the recurrent attention network to model the generation probability $p(Y|X)$, where $X = (S, U)$. For S_i, we have music tags that are mined from playlists (denoted by $(t_{i,1}, \cdots, t_{i,L})$. We use the top five music tags, i.e., $L = 5$), its singer names (denoted by g_i), and song names (denoted by a sequence $(q_{i,1}, \cdots, q_{i,K_i}$, where K_i is the number of words in song names). As mentioned earlier, each user is represented by exactly one user tag mined from chat logs: $U_i = (u_{i,1})$. Finally we concatenate S and U to compose X.

An encoder reads X into vector h. Here, a bidirectional RNN [42] is utilized. Given an input sequence with ordering from x_1 to x_T, the forward RNN calculates a sequence of its forward hidden states $\{\overrightarrow{h_1}, \cdots, \overrightarrow{h_T}\}$. Meanwhile, reversing the input as the order from x_T to x_1, the backward RNN calculates a sequence of its backward hidden states $\{\overleftarrow{h_1}, \cdots, \overleftarrow{h_T}\}$. Then we obtain the final hidden states by concatenating them as $h_j = \{\overrightarrow{h_j}, \overleftarrow{h_j}\}$, which saves the summaries of both the preceding words and the following words.

The attention mechanism [5] aims to find the parts of inputs that should be focused on. Thus, the context vector c is calculated by a weighted sum of the final hidden states as in [5]. Given the predicted preceding words $\{y_1, y_2, \cdots, y_{t-1}\}$, context vector c_t, and the RNN hidden state s_t, the decoder calculates a probability of the next word y_t:

$$p(y_t|x_1, \cdots, x_T, y_1, \cdots, y_{t-1}) = g(y_{t-1}, s_t, c_t), \quad (1)$$

where $g(\cdot)$ is a softmax activation function.

Finally, we leverage a beam search strategy to generate the reason given by $x_1, x_2, \cdots, x_T$. For the first node, we go through the softmax activation function and calculate the Top K candidates by $p(y_1|x_1, x_2, \cdots, x_T)$. Then for each candidate y_1, we calculate its next sequences by

$$y_t = \arg\max_{y_i : i \geqslant 2} p(y_i|x_1, x_2, \cdots, x_T, y_1, y_2, \cdots, y_{i-1}), \quad (2)$$

until y_t is equal to the "End of Sentence" symbol. Finally, we randomly select one from the K generated candidates to ensure diversity of our output.

To guarantee that a generated reason is easy to read and understand, we need to filter out noisy text. Thus we propose learning a linear regression function of a score based on generation probability, a score based on N-gram language models, a score based on POS (Part-Of-Speech) RNN language models, and a score based on dependency parsing. All scores are rescaled into 0 to 1 before combination. Finally, we filter out reasons that do not pass a threshold.

IV. Experiments

In this section, we describe our dataset, our offline and online experiments to compare different methods.

A. Dataset

Our training data are crawled from NetEase Music, one of the most popular music websites in China. As described in Section III-B, we extract 4.9 million reason-like comments for 0.4 million of songs. Among the crawled songs, we manage to mine song tags for about 7,932 songs. On the other hand, we obtain 22 user tags to retrieve personalized reasons for our experiments. By joining the three sets, we finally obtain a data set of the form (S_i, U_i, Y_i), which contains 2,778 songs (S_i), 206 thousands of reasons (Y_i) corresponding to 22 user tags (U_i). We use this data set for training our reason generation model.

B. Offline Comparison of Reasons

1) Evaluation Metrics: We conduct offline evaluation to compare the effectiveness of different methods in generating reasons for personalized song recommendation. We randomly sample 30 songs that are not included in training data for testing purposes. For each method for comparison, we collect top five results to make a pool for assessments. As user satisfaction is somehow difficult to explain or decompose, for a given song and a reason, we first ask six assessors to independently give an overall rating on whether the text is bad (i.e. rating 1), acceptable (i.e. rating 2), or attractive (i.e. rating 3) as a reason. Then we ask them to give detailed ratings ranged from 1 (bad) to 3 (good) according to the following three criteria:

- *Fluency*. The generated reason is easy to read and understand as a natural language text.
- *Relevance*. It is relevant to the recommended song, so that the user can understand why that particular song is recommended.
- *Personalization*. It is relevant to the personality, interests, and situation of that particular user.

2) Compared Methods: In offline evaluation, we compare the following methods:

FM Factorization Machines (FM) [43] are widely used in recommendation systems due to their effectiveness and rich functionality.

Retrieval The comments retrieved in Section III-C are used directly as recommendation reasons of associated music.

Generation w/o userTag To assess the effect of user tags, we consider a simplified version of the proposed method. It

[3] https://code.google.com/archive/p/word2vec/

TABLE I
Comparing methods of obtaining recommendation reasons by human ratings. The numbers are average ratings.

	Method	FM	Retrieval	Generation w/o userTag	Generation	Generation w/ Scoring
Top-1	Fluency	2.80	2.83	2.60	2.67	**2.93**
	Relevance	1.97	1.90	1.80	1.93	**2.03**
	Personalization	1.97	2.30	2.00	2.30	**2.33**
	Overall	1.90	1.93	2.00	1.97	**2.07**
Top-3	Fluency	2.84	2.81	2.40	2.86	**2.90**
	Relevance	1.91	1.91	1.93	2.03	**2.06**
	Personalization	1.94	2.26	1.87	2.32	**2.33**
	Overall	1.83	1.92	1.87	2.03	**2.06**
Top-5	Fluency	**2.88**	2.85	2.32	2.78	2.85
	Relevance	1.91	1.89	1.92	1.97	**2.05**
	Personalization	1.93	2.28	1.84	2.25	**2.29**
	Overall	1.85	1.90	1.80	1.97	**2.05**

just uses $\{S_i, Y_i\}$ to train a generator without user tags and without automatic scoring.

Generation This is our proposed method, which utilizes a generation model to produce the personalized reasons with user tags for given songs, but without using any automatic scoring function.

Genenration w/ Scoring This is the ranked results of *Generation* by using the automatic scoring method presented in the last paragraph of Section III-D.

3) Performance Comparison: Table I shows the comparison result based on human ratings. Our experimental results indicate that our proposed method significantly improves on a Factorization Machine baseline in terms of overall rating (by 9.0%), relevance to a song (by 8.7%), and personalization (by 12.9%). The average fluency score of generated reasons is as high as 2.67, where 2 means *acceptable* and 3 means *good*. Overall, our proposed generation method with scoring is the best.

4) Visualization of Attention Weights : Fig. 3 visualises the attention over a particular generated reason. The song, *Love Transfer* by Eason Chen, is tagged with *classic*, *school time*, ..., and *time*. The user is tagged with *student*. The color of each cell represents the attention weight between a word in X and a word in Y. It can be observed, for example, that the attention weight for the pair *school time* and *classmate* and that for the pair *publicize* and *sang* are high, as the words are semantically related. If we sum the weights for each input word (i.e., each line), two song tags, i.e., *school time* and *publicize*, the song name *Love Transfer*, and the user tag *student* are the most salient. This shows that our proposed model can utilize both song information and user tags in generating reasons.

5) Some Examples : Table II shows the song *I love you but goodbye* by the singer Pushu, is tagged with *ballad*, *college years*, *rock*, etc. For a user whose tag is *lovelorn*, our model generates the reason: "Every time I listen to this song, I think of my first love". Although there is no specific comment on the song in training set, such a reason may cause resonance for the user who has just been bereft of love. The examples in Table II indicate that our method can generate recommendation reasons that apply to songs outside the training data.

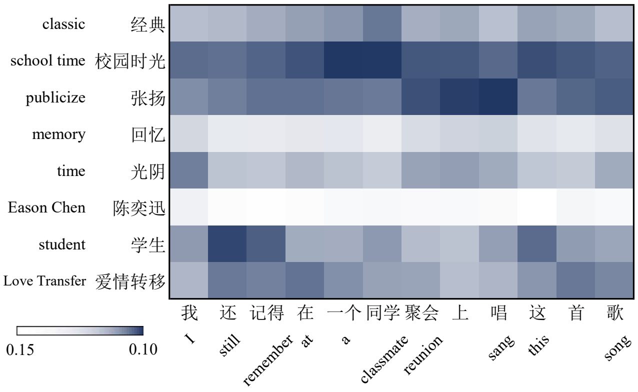


Fig. 3. Visualization of attention weights between an input (S, U) and the output reason Y. They reflect the importance or contribution of a word in generating a reason. Darker color means more important.

C. Online Evaluation

Our objective is to increase the click-through rate of our recommended songs. The aforementioned offline experiments rely on assessors, who are not the actual users of our recommendation service. Therefore, we conduct an online evaluation of methods by comparing the Click-Through Rate (CTR), i.e. the number of clicked songs divided by the number of songs shown to users. We generate 40 thousand reasons for CTR comparison on about 1,400 recommended songs which are the most popular songs in our service.

1) Compared Methods: We conduct online experiments by deploying the following reason generation methods:

Chat Responses We use the query that a user asks for a song recommendation to retrieve a chat response as the recommendation reason.

TABLE II
OUR METHOD GENERATES REASONS FOR USERS WITH DIFFERENT TAGS AND AN UNSEEN SONG IN THE TEST SET.

Input	User Tag	Generated Reason
Singer: 朴树 (Pushu) **Song name**: 我爱你，再见 (I love you but goodbye) **Tags**: 民谣 (Ballad) 大学时光 (College Years) 校园时光 (School Time) 光阴 (Time) 摇滚 (Rock)	失恋 (Lovelorn)	分手后，听着这首歌，感觉自己也是醉了 (After breaking up, listening to this song, I feel drunk) 每次听这首歌都会想到初恋 (Every time I listen to this song, I will think of my first love)
	电音 (Electronic Music)	每次听到这首歌都会有一种震撼的感觉 (Every time I hear this song, I have a feeling of shock) 每次听到这首歌都会热血沸腾 (Every time I hear this song, I feel my blood boils with indignation)
	民谣 (Ballad)	每次听这首歌都会有一种很安静的感觉 (Every time I listen to this song, I have a very quiet feeling) 很喜欢这首歌，很喜欢民谣 (I love the song very much and I love ballad)
	学生 (Student)	以前学校每天中午都会放这首歌 (In the past, the school played this song at noon every day) 校园十佳歌手，我就听了这首歌。[可爱] (I heard this song at the competition of top ten singers of campus. [Cute])
	晚睡 (Sleep Late)	每天晚上睡觉前听这首歌，越听越有感觉，越听越有感觉 (Listen to this song before going to bed every night. The more you listen, the more you feel. The more you listen, the more you feel.) 这首歌是我最喜欢的一首歌，晚安 (The song is my favorite. Good night.)

Mined Reason-Like Comments We use one of the mined reason-like comments (See Section III-B) that are posted to the recommended song as the reason.

Manually Selected Reason-like Comments We ask assessors to review all the mined reason-like comments and select those are acceptable recommendation reasons. We then randomly draw one from them.

Generated Reasons w/o userTag We use $\{S_i, Y_i\}$ to train our generator. Given a song, we generate ten reasons offline and randomly draw one from them.

Generated Reasons w/ Scoring Given a song and a user tag, we generate ten reasons offline with the proposed method of Generation w/ Scoring. We then randomly draw one from them.

2) Performance Comparison: We collect user activities on XiaoIce chatbot from July. 28, 2017 to Jan. 31, 2018. Then we compute the CTRs for the compared methods based on the same songs. We hide the number of impressions due to business confidential concern. Table III shows that our proposed method *Generation w/ Scoring* outperforms the other four methods. In particular, it outperforms even the *Manually Selected Reason-like Comments* by 8.2%, which demonstrates the power of *generating* a reason for a given song-user pair. Moreover, it can be observed that the version of our model that learns without user tags performs the worst, which shows the advantages of learning from the (song, user tag, reason) triplets and hence the importance of personalization. Our proposed method improves the mined reason-like comments, which are associated to a song by real music website users, by 11.8%. This also confirms the power of personalized recommendation reasons in conversations.

TABLE III
COMPARING FIVE METHODS BY DEPLOYING THEM IN XIAOICE CHATBOT ONLINE AND COLLECTING THEIR CLICK-THROUGH RATES

Method	Click Rate	Improves By
(1) Chat Responses	0.444	+12.8%
(2) Mined Reason-like Comments	0.448	+11.8%
(3) Manual Selected Reasons from (2)	0.463	+8.2%
(4) Generated Reasons w/o userTag	0.437	+14.6%
(5) Generated Reasons w/ Scoring	**0.501**	-

V. CONCLUSION

We formulate a new challenging problem called reason generation for explainable recommendation in conversation applications, where our goal is to increase the click-through rate of the recommended songs by generating recommendation reasons that make the users feel as if they are receiving them from their friends. To this end, we build a dataset that consists of (song, user tag, reason) triplets and construct a system that learns to generate recommendation reasons for any given song-user pair. Our experiments indicate that our method significantly improves on a the baselines. Furthermore, we deploy our proposed methods on XiaoIce chatbot, and observe that the click-through rate of recommended songs improves by at least 8.2% over four different baselines. The improvements indicate that personalized reasons do attract more end users.

ACKNOWLEDGEMENT

This work was supported in part by NSFC under Grants 61732008, 61772407, 61332018, and u1531141, in part by the National Key R&D Program of China under Grant 2017YFF0107700.

References

[1] Yongfeng Zhang and Xu Chen. Explainable recommendation: A survey and new perspectives. *CoRR*, abs/1804.11192, 2018.

[2] Jonathan L. Herlocker, Joseph A. Konstan, and John Riedl. Explaining collaborative filtering recommendations. In *In Proc. CSCW*, pages 241–250, 2000.

[3] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. EMNLP*, pages 1724–1734, 2014.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proc. NIPS*, pages 3104–3112, 2014.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014.

[6] Lifeng Shang, Zhengdong Lu, and Hang Li. Neural responding machine for short-text conversation. In *Proc. ACL*, pages 1577–1586, 2015.

[7] Xiaojiang Lei, Xueming Qian, and Guoshuai Zhao. Rating prediction based on social sentiment from textual reviews. *IEEE Trans. Multimedia*, 18(9):1910–1921, 2016.

[8] Peiliang Lou, Guoshuai Zhao, Xueming Qian, Huan Wang, and Xingsong Hou. Schedule a rich sentimental travel via sentimental POI mining and recommendation. In *Proc. IEEE BigMM*, pages 33–40, 2016.

[9] Guoshuai Zhao, Tianlei Liu, Xueming Qian, Tao Hou, Huan Wang, Xingsong Hou, and Zhetao Li. Location recommendation for enterprises by multi-source urban big data analysis. *IEEE Transactions on Services Computing*, pages 1–1, 2018.

[10] Mao Ye, Dong Shou, Wang-Chien Lee, Peifeng Yin, and Krzysztof Janowicz. On the semantic annotation of places in location-based social networks. In *Proc. ACM SIGKDD*, pages 520–528, 2011.

[11] Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. Inferring social ties between users with human location history. *J. Ambient Intelligence and Humanized Computing*, 5(1):3–19, 2014.

[12] Xiao Lin, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. Learning and transferring social and item visibilities for personalized recommendation. In *In Proc.CIKM*, pages 337–346, 2017.

[13] Longke Hu, Aixin Sun, and Yong Liu. Your neighbors affect your ratings: on geographical neighborhood influence to rating prediction. In *Proc. ACM SIGIR*, pages 345–354, 2014.

[14] Guoshuai Zhao, Xueming Qian, and Xing Xie. User-service rating prediction by exploring social users' rating behaviors. *IEEE Trans. Multimedia*, 18(3):496–506, 2016.

[15] Xueming Qian, He Feng, Guoshuai Zhao, and Tao Mei. Personalized recommendation combining user interest and social circle. *IEEE Trans. Knowl. Data Eng.*, 26(7):1763–1777, 2014.

[16] Cheng-Kang Hsieh, Longqi Yang, Honghao Wei, Mor Naaman, and Deborah Estrin. Immersive recommendation: News and event recommendations using personal digital traces. In *Proc. WWW*, pages 51–62, 2016.

[17] Guoshuai Zhao, Xueming Qian, Xiaojiang Lei, and Tao Mei. Service quality evaluation by exploring social users' contextual information. *IEEE Trans. Knowl. Data Eng.*, 28(12):3382–3394, 2016.

[18] Zhiyong Cheng and Jialie Shen. On effective location-aware music recommendation. *ACM Trans. Inf. Syst.*, 34(2):13:1–13:32, 2016.

[19] Guoshuai Zhao, Xueming Qian, and Chen Kang. Service rating prediction by exploring social mobile users' geographical locations. *IEEE Trans. Big Data*, 3(1):67–78, 2017.

[20] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. In *In Proc. ICDE*, pages 801–810, 2007.

[21] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J. Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proc. KDD*, pages 193–202, 2014.

[22] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, and Lise Getoor. User preferences for hybrid explanations. In *In Proc. ACM RecSys*, pages 84–88, 2017.

[23] Jesse Vig, Shilad Sen, and John Riedl. Tagsplanations: explaining recommendations using tags. In *In Proce. IUI*, pages 47–56, 2009.

[24] Li Chen and Feng Wang. Explaining recommendations based on feature sentiments in product reviews. In *In Proc. IUI*, pages 17–28, 2017.

[25] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *In Proc. IJCAI*, pages 2640–2646, 2016.

[26] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. Neural attentional rating regression with review-level explanations. In *In Proc. WWW*, pages 1583–1592, 2018.

[27] J. Ben Schafer, Joseph A. Konstan, and John Riedl. Recommender systems in e-commerce. In *EC*, pages 158–166, 1999.

[28] Badrul M. Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. WWW*, pages 285–295, 2001.

[29] Beidou Wang, Martin Ester, Jiajun Bu, and Deng Cai. Who also likes it? generating the most persuasive social explanations in recommender systems. In *In Proc. AAAI*, pages 173–179, 2014.

[30] 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 *Proc. ACM SIGIR*, pages 83–92, 2014.

[31] Wayne Xin Zhao, Yanwei Guo, Yulan He, Han Jiang, Yuexin Wu, and Xiaoming Li. We know what you want to buy: a demographic-based system for product recommendation on microblogs. In *In Proc. ACM SIGKDD*, pages 1935–1944, 2014.

[32] Yao Wu and Martin Ester. FLAME: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *In Proc. WSDM*, pages 199–208, 2015.

[33] Yunfeng Hou, Ning Yang, Yi Wu, and Philip S. Yu. Explainable recommendation with fusion of aspect information. *World Wide Web*, Apr 2018.

[34] Xu Chen, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. Visually explainable recommendation. *CoRR*, abs/1801.10288, 2018.

[35] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. Neural rating regression with abstractive tips generation for recommendation. In *In Proc. ACM SIGIR*, pages 345–354, 2017.

[36] Steve J. Young, Milica Gasic, Blaise Thomson, and Jason D. Williams. Pomdp-based statistical spoken dialog systems: A review. *In Proc. IEEE*, 101(5):1160–1179, 2013.

[37] Margaret Ann Boden. *Mind as machine: A history of cognitive science*. Clarendon Press, 2006.

[38] Sina Jafarpour, Christopher JC Burges, and Alan Ritter. Filter, rank, and transfer the knowledge: Learning to chat. *Advances in Ranking*, 10:2329–9290, 2010.

[39] Yu Wu, Wei Wu, Chen Xing, Ming Zhou, and Zhoujun Li. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. In *In Proc. ACL*, pages 496–505, 2017.

[40] Alan Ritter, Colin Cherry, and William B. Dolan. Data-driven response generation in social media. In *Proc. EMNLP*, pages 583–593, 2011.

[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *In Proc. NIPS*, pages 3111–3119, 2013.

[42] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Trans. Signal Processing*, 45(11):2673–2681, 1997.

[43] Steffen Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.