Design and Evaluate Conversational Restaurant Recommender System

Chengxi Luo
University of Waterloo
Waterloo, ON, Canada
chengxi.luo@uwaterloo.ca

ABSTRACT
Conversational agents are used widely in e-commerce websites to interact with customers and help them solve questions and find products. Recommendation is also an essential feature of e-commerce websites, which help customers discover more interesting products. In this paper, we implement a Conversational Restaurant Recommender System(CRRS), which combines the conversational agent and recommender system together. We also evaluate the performance of CRRS by a laboratory experiment. Our result shows that CRRS is an efficient tool, which always recommends suitable restaurants based on our users' needs.

Author Keywords
Conversational Agents; Recommender Systems; E-commerce; Faceted Search.

INTRODUCTION
Conversational agents are emerging as an important way of users' daily life, such as Apple Siri, Microsoft Cortana and Amazon Alexa. From the purpose of conversational agents, there are three types, goal-based dialog agents, informational dialog agents, and chatbots. Goal-based dialog agents aim to help users finish a specific goal, such as call someone or set up an alarm. Informational dialog agents are helpful to provide information when users have questions. Chatbots offer social conversations with users. This paper is related to goal-based dialog agents and informational dialog agents.

Recently, the e-commerce industry has grown rapidly. Interactions between buyers and sellers, such as conversations and payments, are no longer face-to-face meetings. Sellers save their budgets by not renting a shop, while buyers can get the products they like with minimal effort. Before buyers decide to order a product, they might have questions about multiple aspects of the product. Since all the interactions between buyers and sellers are online, it is possible that buyers ask questions anytime so that sellers have to deal with these questions all day. There are always some unavoidable situations that sellers cannot converse with them. Companies, such as Amazon, Alibaba, and eBay, implement goal-based dialog agents and informational dialog agents to solve this issue, which interact with users and help them find products.

Recommendation becomes an essential part of e-commerce websites[4, 6]. Prior work in this area uses similarity to predict what products users might be interested in[5,7]. However, more and more researchers believe and focus that combining recommendation features with conversational agents provides better services to users to find product information[1, 2, 3]. Conversational recommender system extracts users' preferences through conversation to recommend suitable products. Conversations are able to offer more detailed information about users such as users' acceptable prices, preferable colors, which builds complete and accurate user profiles and makes recommendations more precise. This paper tries to build a multi-round conversational restaurant recommender system(CRRS). CRRS will communicate with users and learn users' needs from the conversations. Based on what CRRS learns from the conversations, it will generate a list of recommended restaurants. This paper also focuses on evaluating CRRS's performance and addressing the following research questions:

- **RQ1:** How is the overall performance of CRRS(using reinforcement learning method), compared with another conversational recommendation system using MaxEntropy method?
- **RQ2:** How well do users interact with CRRS?
- **RQ3:** How well can faceted search contribute to CRRS's performance?

Specifically, we will investigate how accurate and efficient CRRS using reinforcement learning method can recommend restaurants, compared with the MaxEntropy system. Then, we will evaluate users' experience with CRRS. We also want to find whether our faceted values applied in CRRS play an important role when the users decide a restaurant they want to visit.

To better evaluate CRRS and address research questions, we performed a pre-questionnaire about their demographic information. We also conducted a laboratory study that the participants were asked to interact with CRRS and select several restaurants in the recommendation list. We also used a

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04. . . $15.00

DOI: https://doi.org/10.1145/3313831.XXXXXXX

questionnaire to interview the participants after the experiment. Preliminary results indicate that CRRS spends less chatting time and provides more accurate restaurants based on users' needs than the MaxEntropy system. Users think CRRS is efficient and helpful in recommending restaurants, and they are willing to use CRRS in the future.

RELATED WORK

In this section, we discuss related work on dialogue system, recommender system, faceted search and deep reinforcement learning.

Dialogue System

There are some types of conversational agents, such as goal-based dialog agents, informational dialog agents, and chatbots.In early works of goal-based dialog agents, building models is computationally intractable and requires a large number of labeled data. Young et al. [22] proposed a statistical spoken dialogue system based on partially observable Markov decision processes.However, this system is complex and involves efficient algorithms, carefully constructed approximations and large amounts of labeled data. Recently, researchers have improved this issue and focus on using deep learning to develop dialogue systems.

Recently, with the rapid development of E-commerce industry, some researchers applied dialogue systems into E-commerce websites. Putri et al. [14] proposed an interactive personalized chatbot to provide all services in the hotels to the customers. Most of these researches, however, are not related to recommendations. They mainly focus on natural language process and natural language understanding. They did not use users' preference profiles to recommend items to users.

Interface for E-Commerce Recommender Systems

Recommender systems have been widely applied in commercial websites. Leading E-commerce websites such as Amazon, Taobao and Netflix extract user's past views to provide a personalized recommendation of new interesting products or items. Most researchers focus on building recommender systems based on content, collaborative filtering or hybrid. Lops et al. [11] used a content-based recommender system to match user's preferences with objects' attributes. Musto et al. [12] presented a deep content-based recommender system based on textual description of items. Koren et al. [8] introduced a collaborative-filtering-based recommender system that analyzes relationships between users and interdependencies among products to identify new user-item associations. Strub et al. [17] developed a hybrid recommender model that combines collaborative filtering and content-based filtering techniques.

More and more researchers have proved that combining recommendation features with conversational agents provides better services to users to find product information[1, 5, 9]. Conversational recommender system extracts users' preferences through conversation to recommend suitable products. Conversations offer more detailed information about users such as users' acceptable prices, preferable colors, which builds complete and accurate user profiles and makes recommendation more precise.

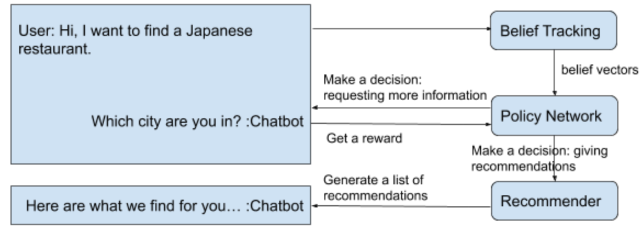


Figure 1. Conversational restaurant recommender system overview.

Previous work about conversational recommender systems, however, mainly are under simplified settings. Christakopoulou et al. [3] proposed a conversational recommender model using probabilistic matrix factorization models to recommend items without interacting and asking users about their preferences. Their later work [2] still restricts the conversation to two stages of a single round, one stage for asking questions to users, another stage for recommendation based on users' responses. Zhang et al. [23] used multi-memory network architecture that learns user reviews to build a System Ask-User Respond system, while this system is unable to determine the suitable occasion to ask or make decisions due to lack of a dialogue policy.

Motivated by previous work, we will focus on asking users about their facets and preferences, using reinforcement learning to implement a multi-stage conversational restaurant recommender system.

Faceted Search

Faceted search is a technique that combines text search and faceted navigation techniques to find in document collections [19]. Adding facets or constraints, users are able to narrow down search results. Faceted browsing is widely used in e-commerce websites, which buyers can constrict product lists by applying products' facets. This approach has two main issues, one is creating effective facets list that requires a large amount of time for both customers and developers, another is a facet might become useless when all products match this facet. Fortunately, researchers have been dedicated to solving these problems. Vandic et al. [21] presented a dynamic facet ordering based on specificity and dispersion of facet values to group facets with similar properties. Koren et al. [7] developed a faceted search engine to personalize facets list based on user's behaviors. In our work, we will integrate faceted search into my conversational recommender system. This system will take a facet based on context, and ask users about their prefered facet value. Examples will be "what price range do you prefer?", "What size are you wearing?", "What brand do you prefer?". The main difference between my model and the existing ones is the deep reinforcement learning algorithm that offers better service to users by combining a personalized recommender system with sequential decision making.

SYSTEM DESIGN

System Overview

To better understand and aid recommendations, we followed [18] and designed a conversational recommender system. The purpose of this system is to engage a user in a one-to-one text-based chat and recommend suitable items to the user. This system calculates what the user has rated or visited before and current dialogue to analyze the conversations between a user and system, ask questions to decide the user's preferences, and recommend items specialized for the user.

This system can be divided into three steps. Firstly, the system has to recognize and understand what a user wants. Secondly, the system has to make sequential decisions whether it chooses to recommend or continuously ask questions. Thirdly, the system has to offer recommendations that satisfies the user's expectation. In this way, this system contains three parts, belief track, recommender system, policy network. The overview of the system is shown as Figure 1. A user tells the system about his needs, the belief track converts the user's intent into vector representation or "belief". These vectors are passed into the policy network to decide which actions the system will take. If the policy network decides to request, then the system will ask user follow-up questions to obtain more requirements from the user, and get a reward to train the policy network. If the policy network believes that it has gained enough information, then it will pass all the information to recommender and recommender will predict a list of good items to the user.

Belief Tracking

Similar to [4], a belief track is to identify a user's intent by extracting product facets and its values from conversations and constructing facet-value pairs. Each facet-value pair represents a constraint of product. For example, if the system asks "what color do you prefer", then the facet will be "color". Then, if a user answers with white, then facet value will be "white" and facet-value pair will be (color, red).

Basically, fed by all the conversations between a user and system, the belief track recognizes all the possible facets and all their values in the conversations and produces a probability distribution over these pairs. Before passing inputs to the belief track, the system preprocesses each dialogue into a n-gram vector. LSTM network takes a list of n-gram vector as input, and its output will be fed into a softmax activation layer. The result of the softmax activation layer is a list of the agent's current belief of dialogue state for one facet. If there are multiple facets, then there are multiple learned vector representations.

Recommender

There are two decisions that the system can make, recommendations and requesting more information from a user. The state is called dialogue state when the system is confident to recommend based on current belief. Rendle [15] introduces a new model called Factorization Machines(FMs), which are a good general predictor dealing with problems with huge sparsity like recommender systems. In this paper, since there are multiple learner vector representations for multiple facets, FMs are a good choice to predict the recommendation. Inputs for FMs contain multiple vector representations mentioned previously, user information and item information. The relationship between user and item information is that the user in the user information has rated item in the item information. Following the model equation in the [15], the output for recommender is a rating score for a specific item and user.

Specifically, the recommender takes argmax of each facet's belief to form a new probability distribution. Next, for some facets with high probability, the recommender will retrieve items from the item set based on corresponding facet values. Finally using FMs to calculate the rating score and re-rank the candidate items.

Policy Network

For the policy network, one of the reinforcement learning techniques is used, policy gradient methods of reinforcement learning. According to [13], policy gradient methods are to optimize parametrized policies regarding the expected long-term reward. It works perfectly without value function and resulting from uncertain state information. This method is used to manage the conversational system. There are two decisions in the dialogue state, returning recommendations and requesting more information. Policy gradient method always chooses the one with highest reward in the long term.

A reinforcement learning agent interacts with its environment in discrete time steps. At each time t, the agent receives an observation, which typically includes the reward. It then chooses an action from the set of available actions, which is subsequently sent to the environment. The environment moves to a new state and the reward associated with the transition is determined. In this series of steps, the agent consists of states, actions, rewards, policies.

State is the current environment, which in my case is the output of the belief track.

Actions are the decisions the agent can make. In my case, there are two actions, recommendation and asking a question to a user. For asking questions, if there are multiple facets, then the system has to ask multiple questions, one per facet.

Reward is the immediate reward after transition from a state to another state with an action. In my case, the system gets reward immediately when it chooses to recommend or ask questions. Reward is determined by following Normalized Discounted Cumulative Gain(NDCG)[20]. NDCG calculates the ratio of recommended order to ideal order, which assumes a user prefers the item with a higher rank than the item with a lower rank.

Policy is the map of agent's action selections, which represents the probability of taking actions from states. The goal of this policy network is to maximize the rewards from starting states to ending states. Ending states means conversation is finished, for example, an user quits the conversation, or an user is satisfied with recommendation. In order to achieve this goal, the policy network is applied by two fully connected layers with ReLU activation function and a softmax layer.

EXPERIMENT SETUP

Before doing experiments, CRRS has to be set up.

The target restaurant has the following facets: {category: Spanish, state: NC, city: Charlotte, price: low, stars: 3.5}

CRRS: Hi!

User: Hi, I'm looking for Spanish food in Charlotte.

CRRS: Which state are you in?

User: I'm in North Carolina.

CRRS: Great! Do you have a preferable price range?

User: Low price please.

CRRS: What rating range do you want?

User: At least 3.5.

CRRS: Here are the restaurants I found:

Restaurant 1:
Category: Spanish
State:NC
City: Charlotte
Price: Low
Stars: 4.5
...

User: Thanks! I'm interested in Restaurant 3.

Figure 2. A conversation example between CRRS and an user.

Dataset

The experiment is conducted based on Yelp challenge recommendation dataset. In order to reduce the sparsity of data, we followed the setting of recommendation evaluation [6, 16], and preprocessed our Yelp dataset so that each user have rated at least 5 restaurants and each restaurant have received 5 reviews or ratings. There are 62047 users, and 21350 items/restaurants in our dataset. We choose 5 item attributes as candidate facets, including category, state, city, price and rating range. Category is the food categories of restaurants, such as Mexican and Burger. Price includes lowest, low, medium and high. Rating range is from 0 to 5. The location is limited to the U.S. and Canada.

User Simulation

CRRS uses reinforcement learning to select actions. Previous work used simulated users to pre-train the reinforcement learning model [10]. Following the previous studies, we created a simulated user to interact and pretrain the system.

The simulated user found the target item through the conversational recommender system. He first decided the target item. The system asked questions about facet values of this item and he responded with corresponding answers to the system. After the system recognized and recommended, they identified whether the system recommended the correct item. An example of conversation is shown as following Figure 2:

Baseline

Our recommender system uses the reinforcement learning(RL) method, and we want to investigate how well the RL method contributes to CRRS. To achieve our goal, we choose the Maximum Entropy(MaxEntropy) rule based method as the baseline. We focus only on evaluating the prediction accuracy and the efficiency of CRRS using the RL method. The result should not be influenced by other factors, such as interface, talking style, datasets. Therefore, the MaxEntropy system is the same as CRRS except for the way of generating a recommendation list.

The MaxEntropy method calculates entropy for each facet, and decides the next facet to ask based on maximum entropy. It will continuously request facet values from a user. This process stops when all facets have been known or the number of dialogue has reached the maximum limit. Once it stops, it will return a recommendation to the user.

STUDY DESIGN

To answer the research questions, we use a laboratory experiment and two questionnaires to collect data on: 1) user experiences with CRRS; 2) performance of CRRS compared with MaxEntropy system; 3) the importance of facet value applied in CRRS. The procedure is organized in following steps:

- **Step1:** A questionnaire asks about participants' demographic information. A detailed instruction tells participants how to start the experiment. A restaurant scenario is also provided. Participants are able to read the instruction and the scenario at any time of the experiment.
- **Step2:** Participants follow the instruction to finish a laboratory experiment. They simulate an Yelp user, interact with CRRS and decide the restaurants they want to visit.
- **Step3:** A post-questionnaire asks about participants' experiences and feedback to CRRS.

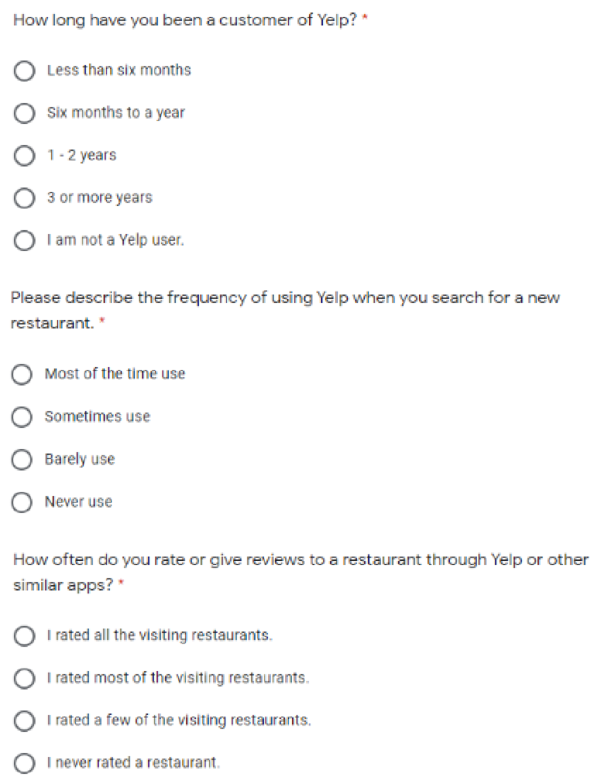
Participation

The ideal real users are the active Yelp users who have visited and rated a considerable number of restaurants and would like to share these information with the system. The system will be able to recommend based on their past interests from past visited restaurants and current interests from conversations. However, the number of participants are limited, and most of them do not satisfy these requirements. According to our pre-questionnaire, 80% of participants have used Yelp for at least 1 year. Half of them never rated a restaurant, and half of them rated only a few of the restaurants they have visited. Therefore, we will follow the experiment design [18] to simulate real scenarios as much as possible.

We recruited 5 participants from CS889 Conversational Agents W20 class and paid back our participation in their respective studies. All of them have experience with conversational systems. We obtained the research ethics TCP approval. Before the experiment, participants were given a short questionnaire using Google Forms regarding their prior experience with Yelp (Figure 3).

After filling the pre-questionnaire, participants were presented with a scenario and an instruction. We used the scenario that was used to evaluate the conversational recommender system from Sun et al. [18].

Senario: For each conversation, you are simulating a Yelp user. You have visited and rated the restaurants through Yelp. The information about your past visiting restaurants will be


How long have you been a customer of Yelp? *

- Less than six months
- Six months to a year
- 1 - 2 years
- 3 or more years
- I am not a Yelp user.

Please describe the frequency of using Yelp when you search for a new restaurant. *

- Most of the time use
- Sometimes use
- Barely use
- Never use

How often do you rate or give reviews to a restaurant through Yelp or other similar apps? *

- I rated all the visiting restaurants.
- I rated most of the visiting restaurants.
- I rated a few of the visiting restaurants.
- I never rated a restaurant.

Figure 3. Pre-questionnaire about participants' demographic information.

provided, including the score you have rated, its category, its location (city and state), price range and overall stars. Please read carefully these information. In order to protect privacy of restaurants and users, each restaurant name or user name will be replaced by an unique id.

Then you want to find a restaurant. A specific target restaurant information is provided, including category, state, city, price and star. Please use these information to answer agent's questions.

After answering the questions, the agent will give you a list of recommended restaurants and their information. Each restaurant will be numbered from 0. Restaurants with lowest numbers are more recommended.

Please select up to three restaurants based on your needs. If one of the restaurants you have chosen is the target restaurant, then recommendation will be successful. Otherwise, the recommendation will fail.

After carefully reading instructions and scenarios, participants began the experiment. Firstly, they randomly chose a restaurant as the target, including restaurant id, user id, and its facet values. Then, in order to know this user's past interests, this user's past visiting restaurant list was shown to participants. They were told to look at these restaurants carefully, especially their facets values. In this way, they learned the preference of the sample user and simulated the sample user to make decisions. Next, the system started to chat with participants, and continuously asked questions. Participants simulated the sample user and answered questions. They were not provided with any information of the target restaurant. Finally, the system generated a recommendation list, and participants chose up to three restaurants in the list.

Each participant was required to complete 4 conversations for each agent, CRRS and MaxEntropy system. Therefore, each participant interacted with the agents 8 times in total. Each participant never used the same target restaurant and sample user each time.

After experiments, participants were asked to complete a post-questionnaire using Google Forms. This questionnaire helps us understand how participants think of this system. It contains a series of questions using scale from 1 to 5, open-ended questions asking participants about the experience with CRRS. Participants were aware that the experience could be positive, negative or neutral. The questions in the questionnaire are listed as following:

- What do you think of the agent's way of asking questions?
- Please use 3 keywords to describe your impression of CRRS, and explain.
- How would you rate CRRS on the dimension of likable, enjoyable, helpful? (1 = least likable/enjoyable/helpful, 5 = most likable/enjoyable/helpful)
- When you think about the system, do you think of it as something you need or don't need? (1 = don't need at all, 5 = really need)

Methods	Prediction Success Rate	Chosen Success Rate	Low Rank Rate	Average Number of Turns
MaxEntropy	0.9	0.9	0.15	4.6
CRRS	1.0	0.8	0.5	3.75

Table 1. Comparison of CRS and the MaxEntropy method.

- Do you satisfy its recommendation? Why or why not?
- What are the things that you like most and would most like to improve about CRRS?
- Please rate the importance of attributes that help you decide a restaurant you want to try. (1= not at all important, 5 = extremely important. Attributes: food categories, distance, price range, rating, menu photos and food photos, provides reservation, provides delivery, payment method, parking lot, number of reviews)

Measures

Several measurements are used to solve RQ1: prediction success rate, average chosen success rate, low rank rate and average number of turns.

Prediction success rate is one of the main measurements of a system. A successful conversation for prediction success rate means the restaurant recommended by CRRS is the target restaurant. Prediction success rate is the ratio of the number of successful conversations over the number of total conversations. This is directly related to the accuracy of our recommender system.

Chosen success rate is the ratio of the number of successful conversations for users over the number of total conversations. The successful conversations for users is defined as the target restaurant is in the user's selections. This is important because it measures the accuracy of CRRS when users are involved.

Low rank rate calculates the rate when the target item is ranked too low in the recommendation list.

For the average number of turns, a turn refers to the utterance that the system asks one question and then the user responds. More turns means the system needs more information to deal with user's needs. Less turns means the system performs better on collecting user's information.

The measurements solving RQ2 are questions in the post-questionnaire. We care about the speaking style of CRRS. We define likable as how much participants would like to use CRRS, enjoyable as how much participants think CRRS handles users input, and helpful as how much participants think of the way CRRS displays its recommended restaurants.

To solve RQ3, we use the scale from 1 to 5 to evaluate the importance of each attribute. If our facet values applied in CRRS obtain higher scores, it will indicate that faceted search helps CRRS improve the accuracy. If there are other attributes obtaining higher scores, it will indicate that faceted search can still be improved.

User: Mexican restaurant
CRRS: Which city are you in?
User: I live in Gilbert
CRRS: Which rating range do you want?
User: 4.0
CRRS: Making recommendations...

Figure 4. CRRS asked for three facet values.

User: I want to find mexican restaurant
MaxEntropy: Which city are you in?
User: Gilbert
MaxEntropy: Which state are you in?
User: AZ
MaxEntropy: Which rating range do you want?
User: 4.0
MaxEntropy: Which price level would you like?
User: Low price level please.
MaxEntropy: Making recommendations...

Figure 5. The MaxEntropy system asked for five facet values.

RESULTS

To evaluate CRRS and answer my research questions, we have examined the chat transcripts between each user and the system and questionnaire about users' experience feedback. Here we report the findings to answer my three research questions, respectively: (a) overall performance of this system, (b) users' interaction with this system, (c) the effect of faceted search on this system's performance.

RQ1: CRRS versus MaxEntropy System

First we examined how the reinforcement learning method helps the sequential decision making in the conversation system, compared to Maximum Entropy method. Table 1 shows the experiment result of the conversational recommender system and the baseline on several important measurements.

Comparing MaxEntropy system and CRRS, we can find that CRRS provides more accurate recommendations based on users' needs. CRRS obtains a higher rate of prediction success rate, which means that there is a higher rate that target restaurant is recommended by CRRS. For chosen success rate, however, CRRS has a slightly lower rate. Participants were harder to decide the target restaurant in the recommendation list.

Possible reasons are listed as following:

- From the chat transcripts between participants and the chatbot, one possible explanation is that the system recommends a list of restaurants with the same food categories, locations and price range but slightly different rating stars. In such a situation, it is difficult for users to pick suitable restaurants. For example, one user commented: "There is no way to differentiate between two restaurants with the same star rating and cost."
- Another possible reason is that CRRS ranked the target restaurant lower than MaxEntropy agent(shown in Table 1). Users were told that all the restaurants are ranked, and the

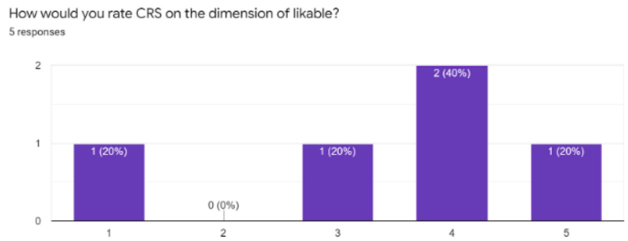


Figure 6. Post-questionnaire: likable.

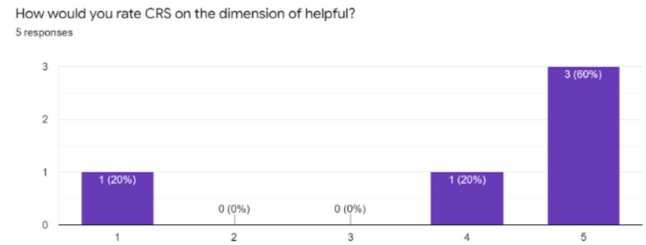


Figure 7. Post-questionnaire: helpful.

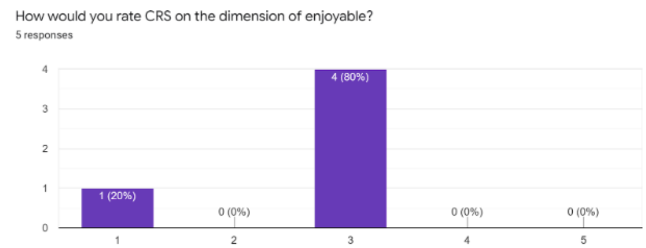


Figure 8. Post-questionnaire: enjoyable.

top one is the most recommended restaurant. In this case, users are likely to choose top restaurants instead of restaurants with lower rank. For example, one user stated when asked whether he satisfies recommendation results: "You can see a rank of all the restaurants that were filtered." It indicates that this participant prefered to choose top restaurants.

We also examined dialogue lengths between each user and systems. This is important to our performance because people might be bored if people are required to give all facet values in real conversation scenarios. CRRS has shorter average turns and Figure 4 is an example of CRRS conversation, while the MaxEntropy system has longer average turns and Figure 5 is an example of the MaxEntropy system conversation. One explanation is that the MaxEntropy system asks the facets one by one and stops when all the facets get their values, while CRRS chooses the decision (giving a recommendation or asking for facet value) based on rewards, so it sometimes uses less dialogue length to give recommendations.

Overall, CRRS has a higher prediction success rate but a lower chosen success rate. CRRS aims to make faster recommendations with less facet value. Its recommendation lists are always longer and it is harder to rank the target restaurant higher in a longer list. Therefore, CRRS gains a higher low rank rate, but shorter dialogue length.

RQ2: Users' Interaction with CRS

We examined users' impression of CRRS by examining their description and ratings of this system. Participants were asked to describe their impression of this system in three keywords. The top-3 most mentioned keywords were simple to use, straightforward and efficient. From these words, it seemed that the users perceived their interaction with the system positively. 80% of users gave all positive expressions when describing their impression, such as "clear", "standard", "effective", and "useful". The users were also asked to rate this system on three dimensions likeable, helpful, enjoyable on a scale of 1 to 5, 1 being not at all and 5 being very much. The average rating for each dimension was: likeable 3.4(Figure 6), helpfulness 4(Figure 7), enjoyable 2.6(Figure 8). Overall, participants were ambivalent about their experience with the system. To explain this, we examined the chat transcript between each user and the system. The transcripts helped explain the ratings from different aspects. It revealed that CRRS was limited at understanding users' inputs. It also showed that CRRS could not handle the mistake well if users accidentally made a typo, which made the system less enjoyable. For example, one user commented on: "To avoid making input errors it would be nice to have a radio button beside the available options to avoid accidentally typing in a typo." In this way, CRRS still needs improvements to make the system user-friendly and understand users' inputs.

On the other hand, participants felt CRRS was pretty helpful. Most users believed the agent's way of asking questions was "pretty clear", "straightforward" and "well targeted problems". 80% of users were satisfied with the recommendations. For example, a user stated: "the quality of the recommendations are good." Similarly, another answered: "I am satisfied with the recommendation I received because it matched what I asked for when I asked for it correctly." When asked whether CRRS is something you need in the future(1 = don't need at all, 5 = extremely need), participants scored 3.2 on average. This is similar to the score of likeable(3.5). This indicates that most participants were still willing to use CRRS even though it was barely user-friendly.

RQ3: Effectiveness of Faceted Search in Recommendations

The main purposes of using CRRS is to provide an accurate recommendation based on users' needs. To achieve this goal, we used faceted search to filter the restaurants by food categories, location, price range, and rating stars. Thus participants were asked to rate these facets on its importance to help them decide a restaurant, on a scale of 1 to 5, 1 being not at all important, 5 being extremely important. The average rating for each facet was: food categories 3.75, location(distance) 3.5, price range 3.0, rating stars 4.25. Although the number of participants are limited, these attributes, especially rating stars, were important factors to these participants when they decided on a restaurant.

In addition, participants were asked to rate other attributes that were not applied in CRRS. The average rating for each other facet was: menu photos and food photos 3.8, reservation service 2.5, takeout service 2.5, parking lot 2.5, number of reviews 4, payment method 2. Most participants thought photos and reviews were pretty essential. For example, when asked how CRRS could be improved, one participant stated: "The photos of the recommended restaurants are not provided." Therefore, for faceted search, these facets are essential when CRRS makes recommendations, but there are still other important factors that should be counted, such as photos and reviews of restaurants.

CONCLUSION AND FUTURE WORK

We implemented a conversational restaurant recommender system that combines a conversational chatbot and a recommender system. We have presented a laboratory experiment to evaluate the performance of CRRS. Compared to the baseline, CRRS spends less conversation length on recommending more accurate restaurants based on users' needs, but it is harder for users to pick the most suitable restaurant in the recommendation list. From the perspective of users' experience, our result indicates that participants were willing to use CRRS in the future, and believed CRRS was an efficient and helpful tool that recommended accurate restaurants within a short amount of time. Our result also shows that the facet values we chose were significant factors when participants decided restaurants, and proves faceted search improved the accuracy of CRRS.

This work has many limitations and there is so much future work we can do. Firstly, it is interesting to find that restaurants' reviews and food photos are more important than food categories and price range. We will continue improving the accuracy of CRRS by adding more facet values. Second, CRRS is a robotic and non-user-friendly tool from participants' feedback. We will focus on changing its speaking style, and making CRRS understand complex user inputs and deal with users' typo. Third, the number of participants are limited and all participants are from the same class, which makes our result less convincing. We will recruit more diverse participants to evaluate CRRS.

REFERENCES

[1] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. Association for Computing Machinery, New York, NY, USA, 335–344. DOI:http://dx.doi.org/10.1145/3077136.3080797

[2] Konstantina Christakopoulou, Alex Beutel, Rui Li, Sagar Jain, and Ed H. Chi. 2018. QR: A Two-Stage Approach toward Interactive Recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (KDD '18)*. Association for Computing Machinery, New York, NY, USA, 139–148. DOI: http://dx.doi.org/10.1145/3219819.3219894

[3] Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards Conversational Recommender Systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, New York, NY, USA, 815–824. DOI: http://dx.doi.org/10.1145/2939672.2939746

[4] Bhuwan Dhingra, Lihong Li, Xiujun Li, Jianfeng Gao, Yun-Nung Chen, Faisal Ahmed, and Li Deng. 2016. End-to-End Reinforcement Learning of Dialogue Agents for Information Access. *CoRR* abs/1609.00777 (2016). http://arxiv.org/abs/1609.00777

[5] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017a. Neural Collaborative Filtering. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 173–182. DOI: http://dx.doi.org/10.1145/3038912.3052569

[6] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017b. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[7] Jonathan Koren, Yi Zhang, and Xue Liu. 2008. Personalized Interactive Faceted Search. In *Proceedings of the 17th International Conference on World Wide Web (WWW '08)*. Association for Computing Machinery, New York, NY, USA, 477–486. DOI: http://dx.doi.org/10.1145/1367497.1367562

[8] Y. Koren, R. Bell, and C. Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (Aug 2009), 30–37. DOI: http://dx.doi.org/10.1109/MC.2009.263

[9] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards Deep Conversational Recommendations. *CoRR* abs/1812.07617 (2018). http://arxiv.org/abs/1812.07617

[10] Xiujun Li, Yun-Nung Chen, Lihong Li, Jianfeng Gao, and Celikyilmaz Asli. 2017. Investigation of Language Understanding Impact for Reinforcement Learning Based Dialogue Systems. (03 2017).

[11] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. 2011. *Content-based Recommender Systems: State of the Art and Trends*. 73–105. DOI: http://dx.doi.org/10.1007/978-0-387-85820-3_3

[12] Cataldo Musto, Tiziano Franza, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. 2018. Deep Content-Based Recommender Systems Exploiting Recurrent Neural Networks and Linked Open Data. In *Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization (UMAP '18)*. Association for Computing Machinery, New York, NY, USA, 239–244. DOI: http://dx.doi.org/10.1145/3213586.3225230

[13] J. Peters. 2010. Policy gradient methods. *Scholarpedia* 5, 11 (2010), 3698. DOI: http://dx.doi.org/10.4249/scholarpedia.3698 revision #137199.

[14] Farica Perdana Putri, Hira Meidia, and Dennis Gunawan. 2019. Designing Intelligent Personalized Chatbot for Hotel Services. In *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2019)*. Association for Computing Machinery, New York, NY, USA, 468–472. DOI:http://dx.doi.org/10.1145/3377713.3377791

[15] Steffen Rendle. 2010. Factorization Machines. 995–1000. DOI: http://dx.doi.org/10.1109/ICDM.2010.127

[16] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI '09)*. AUAI Press, Arlington, Virginia, USA, 452–461.

[17] Florian Strub, Romaric Gaudel, and Jérémie Mary. 2016. Hybrid Recommender System Based on Autoencoders. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems (DLRS 2016)*. Association for Computing Machinery, New York, NY, USA, 11–16. DOI:http://dx.doi.org/10.1145/2988450.2988456

[18] Yueming Sun and Yi Zhang. 2018. Conversational Recommender System. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval (SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 235–244. DOI:http://dx.doi.org/10.1145/3209978.3210002

[19] Daniel Tunkelang. 2009. *Faceted search*. Morgan Claypool, San Rafael, CA.

[20] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. 2009. Learning to Rank by Optimizing NDCG Measure. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems (NIPS'09)*. Curran Associates Inc., Red Hook, NY, USA, 1883–1891.

[21] Damir Vandic, Steven Aanen, Flavius Frasincar, and Uzay Kaymak. 2017. Dynamic Facet Ordering for Faceted Product Search Engines. *IEEE Trans. on Knowl. and Data Eng.* 29, 5 (May 2017), 1004–1016. DOI: http://dx.doi.org/10.1109/TKDE.2017.2652461

[22] S. Young, M. Gašić, B. Thomson, and J. D. Williams. 2013. POMDP-Based Statistical Spoken Dialog Systems: A Review. *Proc. IEEE* 101, 5 (May 2013), 1160–1179. DOI: http://dx.doi.org/10.1109/JPROC.2012.2225812

[23] Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W. Bruce Croft. 2018. Towards Conversational Search and Recommendation: System Ask, User Respond. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 177–186. DOI: http://dx.doi.org/10.1145/3269206.3271776