Improving Search through A3C Reinforcement Learning based Conversational Agent

Milan Aggarwal[1], Aarushi Arora[2], Shagun Sodhani[1], and Balaji Krishnamurthy[1]

[1] Adobe Systems Inc., Noida, India,
milan.ag1994@gmail.com, kbalaji@adobe.com
[2] IIT Delhi, Hauz Khas, Delhi, India

Abstract. We develop a reinforcement learning based search assistant which can assist users through a sequence of actions to enable them realize their intent. Our approach caters to subjective search where user is seeking digital assets such as images which is fundamentally different from the tasks which have objective and limited search modalities. Labeled conversational data is generally not available in such search tasks, to counter this problem we propose a stochastic virtual user which impersonates a real user for training and obtaining bootstrapped agent. We develop A3C algorithm based context preserving architecture to train agent and evaluate performance on average rewards obtained by the agent while interacting with virtual user. We evaluated our system with actual humans who believed that it helped in driving their search forward with appropriate actions without being repetitive while being more engaging and easy to use compared to conventional search interface.

Keywords: Subjective search, Reinforcement Learning, Virtual user model, Context aggregation

1 Introduction

Within the domain of "search", the recent advances have focused on personalizing the search results through recommendations [28, 17]. While the quality of recommendations have improved, the conventional search interface has not innovated much to incorporate useful contextual cues which are often missed. Conventional search interface enables the end user to perform a keyword based faceted search where the end user types in her search query, applies some filters and then modifies the query based on the results. This iterative interaction naturally paves way for incorporating conversations in the process. Instead of the search engine just retrieving the "best" result set, it can interact with the user to collect more contextual cues. For example, if a user searches for "birthday gift", the search engine could follow-up by asking "who are you buying the gift for". Such information and interaction can provide more human-like and engaging search experience along with assisting user in discovering their search intent. In this work we address this problem by developing a Reinforcement

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

Learning (RL) [18] based conversational search agent which interacts with the users to help them in narrowing down to relevant search results by providing them contextual assistance.

RL based dialogue agents have been designed for tasks like restaurant, bus and hotel reservation [16] which have limited and well-defined objective search modalities without much scope for subjective discussion. For instance, when searching for a restaurant, the user can specify her preferences (budget, distance, cuisines etc) due to which the problem can be modeled as a slot filling exercise. In contrast, suppose a designer is searching for digital assets (over a repository of images, videos etc) to be used in a movie poster. She would start with a broad idea and her idea would get refined as the search progresses. The modified search intent involves an implicit cognitive feedback which can be used to improve the search results. We train our agent for this type of search task where it is modeled as a sequence of alternate interactions between the user and the RL agent. The extent to which the RL agent could help the user depends on the sequence and the type of actions it takes according to user behavior. Under the RL framework, intermediate rewards is given to the agent at each step based on its actions and state of conversational search. It learns the applicability of different actions through these rewards. In addition to extrinsic rewards, we define auxiliary tasks and provide additional rewards based on agent's performance on these tasks. Corresponding to the action taken by the agent at each turn, a natural language response is selected and provided to the user. Since true conversational data is not easily available in search domain, we propose to use query and session log data to develop a stochastic virtual user environment to simulate training episodes and bootstrap the learning of the agent.

Our contributions are three-fold: 1) formulating conversational interactive search as a reinforcement learning problem and proposing a generic and easily extendable set of states, actions and rewards; 2) developing a stochastic user model which can be used to efficiently sample user actions while simulating an episode; 3) we develop A3C (Asynchronous Advantage Actor-Critic) [13] algorithm based architecture to predict policy and state value functions of RL agent

2 Related Work

There have been various attempts at modeling conversational agents, as dialogue systems [4, 26, 20, 10] and text-based chat bots [5, 11, 12, 21, 24]. Some of these have focused on modeling goal driven RL agent such as indoor way finding system [5] to assist humans to navigate to their destination and visual input agents which learn to navigate and search object in 3-D environment space [27].

RL based dialogue systems have been explored in the past. For example, [20] uses User Satisfaction (US) as the sole criteria to reward the learning agent and completely disregards Task Success(TS). But US is a subjective metric and is much harder to measure or annotate real data with. In our formulation, we provide a reward for task success at the end of search along with extrinsic and auxiliary rewards at intermediate steps (discussed in section 3.4). Other RL

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

based information seeking agents extract information from the environment by sequentially asserting questions but these have not been designed on search tasks involving human interaction and behavior[2].

RL has also been used for improving document retrieval through query reformulation where the agent sequentially reformulates a given complex query provided by the user [15, 14]. But their work focuses on single turn episodes where the model augments the given query by adding new keywords. In contrast, our agent engages the user directly into the search which comprises of sequence of alternate turns between user and agent with more degrees of freedom (in terms of different actions the agent can take).

To minimize human intervention while providing input for training such agents in spoken dialogue systems, simulated speech outputs have been used to bypass spoken language unit [4]. This approach enables to reduce the system's dependence on hand engineered features. User models for simulating user responses have been obtained by using LSTM which learns inter-turn dependency between the user actions. They take as input multiple user dialogue contexts and outputs dialogue acts taking into account history of previous dialogue acts and dependence on the domain [1].

Often task oriented dialogue systems are difficult to train due to absence of real conversations and subjectivity involved in measuring shortcomings and success of a dialogue [7]. Evaluation becomes much more complex for subjective search systems due to absence of any label which tells whether the intended task had been completed or not. We evaluate our system through rewards obtained while interacting with the user model and also on various real world metrics (discussed in experiments section) through human evaluation.

3 System Model

3.1 Reinforcement Learning

Reinforcement Learning is the paradigm to train an agent to interact with the environment in a series of independent episodes where each episode comprises of a sequence of turns. At each turn, the agent observes state s of the environment ($s \in \mathbf{S}$ - set of possible states) and performs an action from $\mathbf{A}$ - set of possible actions which changes the state of the environment and the agent gets the corresponding reward [18]. An optimal policy maximizes cumulative reward that the agent gets based on the actions taken from start till the final terminal state.

3.2 Agent action space

Action space $\mathbf{A}$ is designed to enable the search agent to interact with the user and help her in searching the desired assets conveniently. The agent actions can be divided into two sets - the set of probe intent actions - $\mathbf{P}$ and general actions - $\mathbf{G}$ as described in Table 1 and Table 2 respectively. The agent uses the probe intent actions $\mathbf{P}$ to explicitly query the user to learn more about her context.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

Table 1. Probe intent actions

Action	Description
probe use case	ask about where assets will be used
probe to refine	ask the user to further refine query if less relevant search results are retrieved
cluster categories	ask the user to select from categorical options related to her query

Table 2. General actions

Action	Description
show results	display results corresponding to most recent user query
add to cart	suggest user to bookmark assets for later reference
ask to download	suggest user to download some results if they suit her requirement
ask to purchase	advise the user to buy some paid assets
provide discount	offer special discounts to the user based on search history
sign up	ask the user to create an account to receive updates regarding her search
ask for feedback	take feedback about the search so far
provide help	list possible ways in which the agent can assist the user
salutation	greet the user at the beginning; say goodbye when user concludes the search

For instance, the user may make a very open-ended query resulting in a diverse set of results even though none of them is a good match. In such scenarios, the agent may prompt the user to refine her query or add some other details like where the search results would be used. Alternatively, the agent may cluster the search results and prompt the user to choose from the clustered categories. These actions serve two purposes - they carry the conversation further and provide various cues about the search context which is not evident from input query.

The set **G** consists of generic actions like displaying assets retrieved corresponding to the user query, providing help to the user etc. The set **G** comprises of actions for carrying out the functionality which the conventional search interface provides like "presenting search results". We also include actions which promote the business use cases (such as prompting the user to signup with her email, purchase assets etc). The agent is rewarded appropriately for such prompts depending on the subsequent user actions.

3.3 State space

We model the state representation in order to encapsulate facets of both search and conversation. The state s at every turn in the conversation is modeled

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

using the history of user actions - *history_user*,[3] history of agent actions - *history_agent*, relevance scores of search results - *score_results* and *length_conv* which represents number of user responses in the conversation till that point.

The variables *history_user* and *history_agent* comprises of user and agent actions in last k turns of the conversational search respectively. This enables us to capture the context of the conversation (in terms of sequence of actions taken). Each user-action is represented as one-hot vector of length 9 (number of unique user actions). Similarly, each agent-action has been represented as a one-hot vector of length 12. The history of the last 10 user and agent actions is represented as concatenation of these one-hot vectors. We use zero padded vectors wherever current history comprises of less than 10 turns.

The variable *score_results* quantifies the degree of similarity between most recent query and the top 10 most relevant search assets retrieved. They have been used to incorporate the dependency between the relevance of probe intent actions and quality of search results retrieved. *length_conv* has been included since appropriateness of other agent actions like *sign up* may depend on the duration for which the user has been searching.

3.4 Rewards

Reinforcement Learning is concerned with training an agent in order to maximize some notion of cumulative reward. In general, the action taken at time t involves a long term versus short term reward trade-off. This problem manifests itself even more severely in the context of conversational search. For instance, let us say that the user searches for “nature”. Since the user explicitly searched for something, it would seem logical to provide the search results to the user. Alternatively, instead of going for immediate reward, the agent could further ask the user if she is looking for “posters” or “portraits” which would help in narrowing down the search in the long run.

Since we aim to optimize dialogue strategy and do not generate dialogue utterances, we assign the rewards corresponding to the appropriateness of the action considering the state and history of the search. We have used some rewards such as task success (based on implicit and explicit feedback from the user during the search) which is also used in PARADISE framework [22]. We model the total reward which the agent gets in one complete dialogue as:

$$R_{total} = r_{Task\ Completion}(search) + \sum_{t \in turns} (r_{extrinsic}(t) + r_{auxiliary}(t))$$

Task Completion and Extrinsic Rewards First kind of reward (r_{TC}) is based on the completion of the task (Task Completion TC) which is download and purchase in the case of our search problem. This reward is provided once at the end of the episode depending on whether the task is completed or not. As second kind of rewards, we provide instantaneous extrinsic rewards [6] -

[3] *history_user* includes most recent user action to which agent response is pending in addition to remaining history of user actions.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

($r_{extrinsic}$) based on the response that the user gives subsequent to an agent action. We categorize the user action into three feedback categories, namely good, average or bad. For example, if the agent prompts the user to refine the query and the user does follow the prompt, the agent gets a high reward while if the user refuses, a low reward is given to the agent. A moderate reward will be given if the user herself refines the query without the agent's prompt.

Auxiliary Rewards Apart from the extrinsic rewards, we define a set of auxiliary tasks T_A specific to the search problem which can be used to provide additional reward signals, $r_{auxiliary}$, using the environment. We define $T_A = \{$# click result, # add to cart, # cluster category click, if sign up option exercised$\}$. $r_{auxiliary}$ is determined and provided at every turn in the search based on the values of different auxiliary tasks metrics defined in T_A till that turn in the search. Such rewards promotes a policy which improves the performance on these tasks.

3.5 Stochastic user model details

The RL agent is trained to learn the optimal action policy requiring actual conversational search data which is not available as conversational agents have not been used for search task we defined. To bypass this issue and bootstrap training, we propose a user model that simulates user behavior to interact with the agent during training and validation. Our methodology can be used to model a virtual user using any query and log sessions data.

We developed a stochastic environment where the modeled virtual human user responds to agent's actions. The virtual human user has been modeled using query sessions data from a major stock photography and digital asset marketplace which contain information on queries made by real users, the corresponding clicks and other interactions with the assets. This information has been used to generate a user which simulates human behavior while searching and converses with the agent during search episode. We map every record in the query log to one of the user actions as depicted in Table 3. Figure 1 shows an example mapping from session data to user action. To model our virtual user, we used the query and session log data of approximately 20 days.

The virtual user is modeled as a finite state machine by extracting conditional probabilities - $P(User\ Action\ u|\ History\ h\ of\ User\ Actions)$. These probabilities are employed for sampling next user action given the fixed length history of her actions in an episode. The agent performs an action in response to the sampled user action and the process continues.

The query and session log data has been taken from an asset search platform where the marketer can define certain offers/ promotions which kick in when the user takes certain actions, for instance the user can be prompted to add some images to cart (via a pop-up box). User's response to such prompts on the search interface is used as proxy to model the effect of RL agent on virtual user's sampled action subsequent to different probe actions by the agent. This ensures that our conditional probability distribution covers the entire probability space of user behavior.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

Session Data	Mapped User Action
shopping ; content_type : all ; NO_OFFSET ; search	*new query*
shopping ; content_type : all ; 100 ; search	*request more*
child while shopping ; content_type:all ; NO_OFFSET ; search	*refine query*
child while shopping ; content_type:all ; NO_OFFSET ; click	*click result*
child while shopping ; content_type: landscape ; NO_OFFSET ; search	*cluster category click*

Fig. 1. Example of mapping session data to user actions. The session data comprises of sequence of logs, each log comprises of search query, filters applied (content type), offset field and interaction performed by the user (such as search, click etc)

Table 3. Mapping between query logs and user actions

User action	Mapping used
new query	first query or most recent query with no intersection with previous ones
refine query	query searched by user has some intersection with previous queries
request more	clicking on next set of results for same query
click result	user clicking on search results being shown
add to cart	when user adds some of searched assets to her cart for later reference
cluster category click	when user clicks on filter options like orientation or size
search similar	search assets with similar series, model etc

3.6 Q-Learning

The agent can be trained through *Q-learning* [23] which consists of a real valued function $Q : S \times A \rightarrow \mathbb{R}$. This Q-function maps every state-action pair (s, a) to a Q-value which is a numerical measure of the expected cumulative reward the agent gets by performing a in state s. In order to prevent the agent from always exploiting the best action in a given state, we employ an $\epsilon-$ greedy exploration policy [25], $0 < \epsilon < 1$. The size of our state space is of the order of $\approx 10^7$. For Q-learning, we use the table storage method where the Q-values for each state is stored in a lookup table which is updated at every step in a training episode.

3.7 A3C Algorithm

In this algorithm, we maintain a value function V_π and a stochastic policy π as a function of the state. The policy $\pi : A \times S \rightarrow \mathbb{R}$ defines a probability distribution $\pi(a|s)$ over the set of actions which the agent may take in state s and is used to sample agent action given the state. The value function $V_\pi : S \rightarrow \mathbb{R}$ represents

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

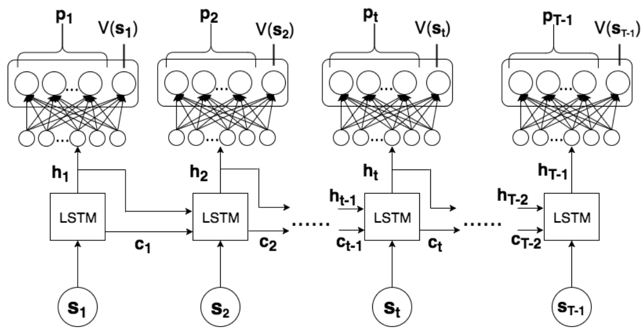

Fig. 2. A3C architecture for predicting policy $\mathbf{p_t}$ and value $V(\mathbf{s_t})$.

the expected cumulative reward from current time step in an episode if policy π is followed after observing state s i.e. $V_\pi(s) = \mathbb{E}_{a\sim\pi(.|s)}[Q_\pi(s,a)]$.

Search Context Preserving A3C Architecture We propose a neural architecture (figure 2) which preserves the context of the conversational search for approximating the policy and value functions. The architecture comprises of a LSTM [8] which processes the state at a time step t (input $\mathbf{i_t} = \mathbf{s_t}$) and generates an embedding $\mathbf{h_t}$ which is processed through a fully connected layer to predict the probability distribution over different actions using softmax function [3] and value of the input state separately. In A3C algorithm, the agent is allowed to interact with the environment to roll-out an episode. The network parameters are updated after completion of every n-steps in the roll-out. An n-step roll-out when the current state is s_t can be expressed as $(s_t, a_t, r_t, s_{t+1}, v_{s_t}) \rightarrow (s_{t+1}, a_{t+1}, r_{t+1}, s_{t+1}, v_{s_{t+1}}) \rightarrow ... \rightarrow (s_{t+n-1}, a_{t+n-1}, r_{t+n-1}, s_{t+n}, v_{s_{t+n-1}})$. The parameters are tuned by optimizing the loss function $loss_{total}$ which can be decomposed into - $loss_{policy}$, $loss_{value}$, $loss_{entropy}$. $loss_{value}$ is defined as:

$$\begin{gathered} loss_{value}(\theta) = (V_{target}(s_i) - V(s_i;\theta))^2, \quad i = t, t+1, ..., t+n-1 \\ where, V_{target}(s_i) = \sum_{k=0}^{t+n-i-1} \gamma^k r_{k+i} + \gamma^{n+t-i} V(s_{t+n};\theta) \end{gathered} \tag{1}$$

Thus an n-step roll-out allows us to estimate the target value of a given state using the actual rewards realized and value of the last state observed at the end of the roll-out. Value of a terminal state s_T is defined as 0. In a similar way, the network is trained on $loss_{policy}$ which is defined as :

$$\begin{gathered} loss_{policy}(\theta) = -\log(p(a_i|s_i;\theta)) * A(a_i, s_i;\theta), \quad i = t, t+1, ..., t+n-1, where \\ A(a_i, s_i;\theta) = \sum_{k=0}^{t+n-i-1} \gamma^k r_{k+i} + \gamma^{n+t-i} V(s_{t+n};\theta) - V(s_i;\theta) \end{gathered} \tag{2}$$

The above loss function tunes the parameter in order to shift the policy in favor of actions which provides better advantage $A(a_t, s_t, \theta)$ given the state s_t.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

This advantage can be interpreted as additional reward the agent gets by taking action a_t in state s_t over the average value of the state $V(s_t;\theta)$ as the reference. However, this may bias the agent towards a particular or few actions due to which the agent may not explore other actions in a given state. To prevent this, we add entropy loss to the total loss function which aims at maximizing the entropy of probability distribution over actions in a state.

$$loss_{entropy}(\theta) = -\sum_{a \in \mathbf{A}} -p(a|s_i;\theta)\log(p(a|s_i;\theta)), \quad i = t, t+1, ..., t+n-1 \quad (3)$$

4 Experiments

In this section, we evaluate the trained agent with the virtual user model and discuss the results obtained with the two reinforcement learning techniques, A3C and Q-learning, and compare them. For each algorithm, we simulate validation episodes after each training episode and plot the average rewards and mean value of the states obtained during the validation episodes. We also developed a chat-search interface where real users can interact with the trained agent during their search.[4]

4.1 A3C using User Model

The global model is obtained using 10 local agents which are trained in parallel threads (each trained over 350 episodes). We compare the validation results using this global model for different state representations for conversational search and hyper-parameter settings such as discount factor (γ) (which affects exploration vs exploitation trade-off) and the LSTM size which controls the context preserving capacity of our architecture.

Varying discount factor We experiment with 3 values of discount factor and fix the LSTM size to 250. Figure 3 shows the validation trend in average rewards for different discount factors. Greater discount factor (lower value of γ) lowers weights for the future rewards due to which the agent tries to maximize the immediate rewards by taking the greedy actions. We validate this by computing the variance in the results for each case. The variance values for the 3 cases ($\gamma =$ 0.90, 0.70, 0.60) are 1.5267, 1.627, and 1.725 respectively. Since the agent takes more greedy actions with higher discount factors, the variance in the reward values also increases since the greedy approach yields good rewards in some episodes and bad rewards in others.

[4] Supplementary material containing snapshots and demo video of the chat-search interface can be accessed at https://drive.google.com/open?id=0BzPI8zwXMOiWNk5hRElRNG4tNjQ

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

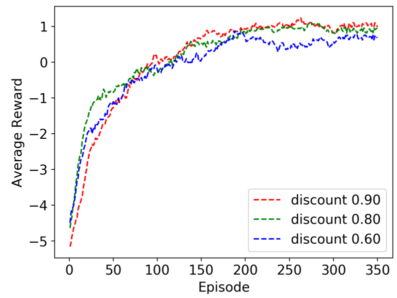

Fig. 3. Plot of average validation reward against number of training episodes for A3C agent. The size of LSTM is 250 for each plot with varying discount factor. Higher value of discount results in better average rewards.

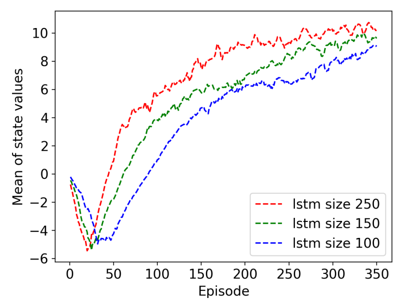

Fig. 4. Plot of mean of state values observed in an episode for A3C agent. Different curves correspond to different LSTM size. The discount value is $\gamma = 0.90$ for each curve. Better states (higher average state values) are observed with larger LSTM size since it enables the agent to remember more context while performing actions.

Varying Memory capacity We vary the size of the LSTM as 100, 150 and 250 to determine the effect of size of the context preserved. Figure 4 depicts the trend in mean value of states observed in an episode. We observe that larger size of the LSTM results in better states since average state value is higher. This demonstrates that a bigger LSTM size providing better capacity to remember the context results in agent performing actions which yield improved states.

4.2 Q-Learning using User Model

We experimented with values of different hyper-parameters for Q-learning such as discount (γ) and exploration control parameter (ϵ) determined their optimal values to be 0.70 and 0.90 respectively based on trends in average reward value at convergence. We compare the A3C agent (with LSTM size 250 and $\gamma = 0.90$ with the Q-learning agent (figure 5). It can be observed that the A3C agent is able to obtain better averaged awards (≈ 1.0) in validation episodes upon

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

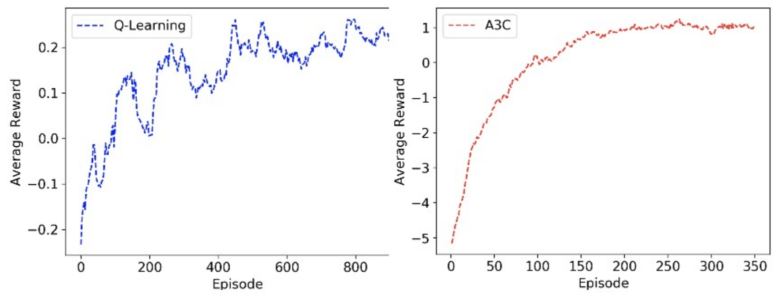

Fig. 5. Plot of average reward observed in validation episodes with Q-agent (left) with $\gamma = 0.70$ and $\epsilon = 0.90$) and A3C agent (right) with $\gamma = 0.90$ and LSTM size = 250. The average reward value at convergence is larger for A3C agent than Q-agent.

convergence as compared to the Q-agent which obtains ≈ 0.20. Since A3C algorithm performs and generalize better than Q-learning approach, we evaluated it through professional designers.

4.3 Human Evaluation of Agent Trained Through A3C

To evaluate the effectiveness of our system when interacting with real humans, we asked professional designers to search images which they will use while designing a poster on natural scenery using both our conversational search agent and conventional search interface provided by stock photography marketplace and collected feedback from 12 designers. We asked them to rate our conversational search system on following metrics. Table 4 shows average rating value of each of these metrics.

1. **Information flow** to measure the extent to which the agent provide new information and suggestions which helped in driving the search forward (on a scale of 1 to 5 where 5 represents high information flow).
2. **Appropriateness** of actions to measure the suitability of actions taken by the agent during the search in terms of coherence (on a scale of 1 to 5 where 5 denotes that it took right actions at right time during the search).
3. **Repetitiveness** to measure how repetitive was the agent's actions in providing assistance during their search (on a scale of 1-5 where 1 represents not repetitive at all).

We also asked the designers to compare our system to conventional search interface in terms of following metrics :

1. **Engagement** : This is to measure how interactive and engaging conversational search is on a scale of 1 to 5 where 1 represents same engagement as conventional search. Our system could achieve an average rating of 2.67 in this metric.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

Table 4. Human Evaluation Ratings for Agent Trained Through A3C

Metric	Average Rating
Information Flow	2.58
Appropriateness	2.67
Repetitiveness	2.50

2. **Time Required**: We asked the designers to compare two search modalities in terms of time required to reach desired results. They were asked to choose between conversational search required - 1.more time, 2. About the same time, 3. Less time. About 33.3% of designers said that it requires more time, 16.7% said that conversational search was faster while remaining 50% believed that it required about the same time.
3. **Ease of Use** : We asked them to choose one of three options - conversational search is, 1. Difficult to use and adds additional burden, 2. About the same to use, 3. Much easier to use, compared to conventional search. 33.3% of the designers believed that conversational search is easier, 41.7% said that it is the same while 25% believed that it is difficult to use it for search.

The above evaluation shows that although we trained the bootstrapped agent through user model, it performs decently well with actual users by driving their search forward with appropriate actions without being much repetitive. The comparison with conventional search shows that out system is much more engaging, requires about the same time while being easier to use than conventional search. Following is an example interaction between assistant and real human :

User : show some images of mountains
Agent : We also have many images in these categories, should I show you some?
User : nature mountains
Agent : Where will you use these images?
User : i am organizing adventure sports
Agent : Check out the images
User : can you show some more
Agent : Images can be added to cart for building a separate collection of stuff that you like. You can simply click on add to cart button
User : <User adds few assets to cart>
Agent : Want me to get you anything else?
User : i want people riding bike on mountains
Agent : We also have many images in these categories, should I show you some? - high mountains, hiking sports mountains, mountaineering mountains
User : hiking sports mountains
Agent : Signing up takes a moment, just give me your email id
User : ABC@XYZ.com

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

5 Conclusion

In this paper, we develop a Reinforcement Learning based search assistant to interact with customers to help them search digital assets suited to their use-case. We model the rewards, state space, action space and develop an A3C based architecture which leverages the context of search to predict the policy. The trained agent is able to obtain higher average rewards in the validation episodes with virtual user and observes states with better values indicative of providing better search experience. As the next step, we would deploy our system to collect true conversational data which can be used to fine tune the current model as well as to train a new model which can generate the natural language responses in addition to deciding the action. In different search domains, designing the state and action space can take significant time which makes every situation an absolutely new task to be solved. To approach this issue as a future work, another system can be designed which helps in the automation of state space characterization with the help of system query logs.

References

1. Layla El Asri, Jing He, and Kaheer Suleman. 2016. A sequence-to-sequence model for user simulation in spoken dialogue systems. arXiv preprint arXiv:1607.00070 (2016).
2. Philip Bachman, Alessandro Sordoni, and Adam Trischler. 2016. Towards information-seeking agents. arXiv preprint arXiv:1612.02605 (2016).
3. John S Bridle. 1990. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Neurocomputing. Springer, 227236.
4. Heriberto Cuayhuitl. 2017. Simpleds: A simple deep reinforcement learning dialogue system. In Dialogues with Social Robots. Springer, 109118.
5. Heriberto Cuayhuitl and Nina Dethlefs. 2011. Spatially-aware dialogue control using hierarchical reinforcement learning. ACM Transactions on Speech and Language Processing (TSLP) 7, 3 (2011), 5.
6. Edward L Deci, Richard Koestner, and Richard M Ryan. 1999. A meta-analytic review of experiments examining the effects of extrinsic rewards on intrinsic motivation. (1999).
7. Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. 2015. Evaluating prerequisite qualities for learning end-to-end dialog systems. arXiv preprint arXiv:1511.06931 (2015).
8. Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. Neural computation 9, 8 (1997), 17351780.
9. Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
10. Esther Levin, Roberto Pieraccini, and Wieland Eckert. 1997. Learning dialogue strategies within the Markov decision process framework. In Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on. IEEE, 7279.
11. Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. arXiv preprint arXiv:1603.06155 (2016).

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21

12. Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016. Deep reinforcement learning for dialogue generation. arXiv preprint arXiv:1606.01541 (2016).
13. Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In International Conference on Machine Learning. 19281937.
14. Karthik Narasimhan, Adam Yala, and Regina Barzilay. 2016. Improving information extraction by acquiring external evidence with reinforcement learning. arXiv preprint arXiv:1603.07954 (2016).
15. Rodrigo Nogueira and Kyunghyun Cho. 2017. Task-oriented query reformulation with reinforcement learning. arXiv preprint arXiv:1704.04572 (2017).
16. Baolin Peng, Xiujun Li, Lihong Li, Jianfeng Gao,Asli Celikyilmaz, Sungjin Lee, and Kam-Fai Wong. 2017. Composite Task-Completion Dialogue Policy Learning via Hierarchical Deep Reinforcement Learning. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 22212230.
17. Guy Shani, David Heckerman, and Ronen I Brafman. 2005. An MDP-based recommender system. Journal of Machine Learning Research 6, Sep (2005), 1265 1295.
18. Richard S Sutton and Andrew G Barto. 1998. Reinforcement learning: An introduction. Vol. 1. MIT press Cambridge.
19. Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems. 10571063.
20. Stefan Ultes, Pawe Budzianowski, Inigo Casanueva, Nikola Mrkic, Lina Rojas-Barahona, Pei-Hao Su, Tsung-Hsien Wen, Milica Gaic, and Steve Young. 2017. Domain-independent User Satisfaction Reward Estimation for Dialogue Policy Learning. Proc. Interspeech 2017 (2017), 17211725.
21. Oriol Vinyals and Quoc Le. 2015. A neural conversational model. arXiv preprint arXiv:1506.05869 (2015).
22. Marilyn A Walker, Diane J Litman, Candace A Kamm, and Alicia Abella. 1997. PARADISE: A framework for evaluating spoken dialogue agents. In Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 271280.
23. Christopher John Cornish Hellaby Watkins. 1989. Learning from delayed rewards. Ph.D. Dissertation. Kings College, Cambridge.
24. Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. arXiv preprint arXiv:1410.3916 (2014).
25. Michael Wunder, Michael L Littman, and Monica Babes. 2010. Classes of multiagent q-learning dynamics with epsilon-greedy exploration. In Proceedings of the 27th International Conference on Machine Learning (ICML-10). 11671174
26. Tiancheng Zhao and Maxine Eskenazi. 2016. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. arXiv preprint arXiv:1606.02560 (2016).
27. Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. 2017. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 33573364.
28. Jian Wei, Jianhua He, Kai Chen, Yi Zhou, and Zuoyin Tang. 2017. Collaborative filtering and deep learning based recommendation system for cold start items. Expert Systems with Applications 69 (2017), 2939.

ICCS Camera Ready Version 2018
To cite this paper please use the final published version:
DOI: 10.1007/978-3-319-93701-4_21