Reinforcement Learning of Question-Answering Dialogue Policies for Virtual Museum Guides

Teruhisa Misu[1]*, Kallirroi Georgila[2], Anton Leuski[2], David Traum[2]
[1]National Institute of Information and Communications Technology (NICT), Kyoto, Japan
[2]USC Institute for Creative Technologies, Playa Vista, CA, USA
teruhisa.misu@nict.go.jp, {kgeorgila,leuski,traum}@ict.usc.edu

Abstract

We use Reinforcement Learning (RL) to learn question-answering dialogue policies for a real-world application. We analyze a corpus of interactions of museum visitors with two virtual characters that serve as guides at the Museum of Science in Boston, in order to build a realistic model of user behavior when interacting with these characters. A simulated user is built based on this model and used for learning the dialogue policy of the virtual characters using RL. Our learned policy outperforms two baselines (including the original dialogue policy that was used for collecting the corpus) in a simulation setting.

1 Introduction

In the last 10 years Reinforcement Learning (RL) has attracted much attention in the dialogue community, to the extent that we can now consider RL as the state-of-the-art in statistical dialogue management. RL is used in the framework of Markov Decision Processes (MDPs) or Partially Observable Markov Decision Processes (POMDPs). In this paradigm dialogue moves transition between dialogue states and rewards are given at the end of a successful dialogue. The goal of RL is to learn a dialogue policy, i.e. the optimal action that the system should take at each possible dialogue state. Typically rewards depend on the domain and can include factors such as task completion, dialogue length, and user satisfaction. Traditional RL algorithms require on the order of thousands of dialogues to achieve good performance. Because it is very difficult to collect such a large number of dialogues with real users, instead, simulated users (SUs), i.e. models that simulate the behavior of real users, are employed (Georgila et al., 2006). Through the interaction between the system and the SUs thousands of dialogues can be generated and used for learning. A good SU should be able to replicate the behavior of a real user in the same dialogue context (Ai and Litman, 2008).

Most research in RL for dialogue management has been done in the framework of slot-filling applications (Georgila et al., 2010; Thomson and Young, 2010), largely ignoring other types of dialogue. In this paper we focus on the problem of learning dialogue policies for question-answering characters. With question-answering systems (or characters), the natural language understanding task is to retrieve the best response to a user initiative, and the main dialogue policy decision is whether to provide this best response or some other kind of move (e.g. a request for repair, clarification, or topic change), when the best answer does not seem to be good enough. Note that often in the literature the term question-answering is used for slot-filling dialogue systems as well, in the sense that the user asks some questions, for example, about restaurants in a particular area, and the system answers by providing a list of options, for example, restaurants. We use the term "question-answering" for systems where user questions can be independent of one another (follow-up questions are possible though) and do not have the objective of reducing the search space and retrieving results from a database of e.g. restaurants, flights, etc. Thus examples of question-answering

* This work was done when the first author was a visiting researcher at USC/ICT.

Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pages 84–93,
Seoul, South Korea, 5-6 July 2012.

characters can be virtual interviewees (that can answer questions, e.g. about an incident), virtual scientists (that can answer general science-related questions), and so forth.

For our experiments we use a corpus (Aggarwal et al., 2012) of interactions of real users with two virtual characters, the Twins, that serve as guides at the Museum of Science in Boston (Swartout et al., 2010). The role of these virtual characters is to entertain and educate the museum visitors. They can answer queries about themselves and their technology, generally about science, as well as questions related to the exhibits of the museum. An example interaction between a museum visitor and the Twins is shown in Figure 1. The dialogue policy of the Twins was arbitrarily hand-crafted (see section 7 for details) and many other policies are possible (including Baseline 2, presented in section 7, and taking more advantage of question topics and context). We propose to use RL for optimizing the system's response generation. This is a real-world application for which RL appears to be an appropriate method.

Although there are similarities between question-answering and slot-filling dialogues there are also a number of differences, such as the reward function and the behavior of the users. As discussed later in detail, in question-answering the users have a number of questions that they are planning to ask (stock of queries), which can be increased or decreased depending not only on whether they received the information that they wanted but also on how satisfied they are with the interaction. The system has to plan ahead in order to maximize the number of successful responses that it provides to user queries. At the same time it needs to avoid providing incorrect or incoherent responses so that the user does not give up the interaction.

One of the challenges of our task is to define an appropriate reward function. Unlike slot-filling dialogues, it is not clear what makes an interaction with a question-answering system successful. A second challenge is that in a museum setting it is not clear what constitutes a dialogue session. Often two or more users alternate in asking questions, which further complicates the problem of defining a good reward function. A third challenge is that the domain is not well defined, i.e. users do not know in advance what the system is capable of (what kind of questions the characters can answer). Moreover, there are many cases of "junk" user questions (e.g. "are you stupid?") or even user prompts in languages other than English (e.g. "hola").

User: What are your names? (ASR: `what are your names`)

Ada: My name's Ada.

Grace: And I'm Grace. We're your Virtual Museum Guides. With your help, we can suggest exhibits that will get you thinking! Or answer questions about things you may have seen here.

Ada: What do you want to learn about?

User: Artificial intelligence. (ASR: `is artificial intelligence`)

Grace: One example of AI, or Artificial Intelligence, is 20Q, an online computer activity here at Computer Place that asks you questions to guess what you're thinking.

Ada: I wish we'd been programmed to do that. Nah... on second thought, I prefer just answering your questions.

Grace: That takes AI too.

Figure 1: Example dialogue between the Twins virtual characters and a museum visitor.

We first analyze our corpus in order to build a realistic model of user behavior when interacting with the virtual characters. A SU is built based on this model and used for learning the dialogue policy of the virtual characters using RL. Then we compare our learned policy with two baselines, one of which is the dialogue policy of the original system that was used for collecting our corpus and that is currently installed at the Museum of Science in Boston. Our learned policy outperforms both baselines in a simulation setting.

To our knowledge this is the first study that uses RL for learning this type of question-answering dialogue policy. Furthermore, unlike most studies that use data collected by having paid subjects interact with the system, we use data collected from real users, in our case museum visitors.[1] We also compare our learned dialogue policy with the dialogue policy of the original system that is currently installed at the Museum of Science in Boston.

The structure of the paper is as follows. In sec-

[1]Note that the CMU "Let's Go!" corpus is another case of using real user data for learning dialogue policies for the Spoken Dialogue Challenge.

tion 2 we present related work. Section 3 provides a brief introduction to RL and section 4 describes our corpus. Then in section 5 we explain how we built our SU from the corpus, and in section 6 we describe our learning methodology. Section 7 presents our evaluation results. Finally section 8 presents some discussion and ideas for future work together with our conclusion.

2 Related Work

To date, RL has mainly been used for learning dialogue policies for slot-filling applications such as restaurant recommendations (Jurčíček et al., 2012), sightseeing recommendations (Misu et al., 2010), appointment scheduling (Georgila et al., 2010), etc., largely ignoring other types of dialogue. Recently there have been some experiments on applying RL to the more difficult problem of learning negotiation policies (Heeman, 2009; Georgila and Traum, 2011a; Georgila and Traum, 2011b). Also, RL has been applied to tutoring domains (Tetreault and Litman, 2008; Chi et al., 2011).

There has been a lot of work on developing question-answering systems with dialogue capabilities, e.g. (Jönsson et al., 2004; op den Akker et al., 2005; Varges et al., 2009). Most of these systems are designed for information extraction from structured or unstructured databases in closed or open domains. One could think of them as adding dialogue capabilities to standard question-answering systems such as the ones used in the TREC question-answering track (Voorhees, 2001). Other work has focused on a different type of question-answering dialogue, i.e. question-answering dialogues that follow the form of an interview and that can be used, for example, for training purposes (Leuski et al., 2006; Gandhe et al., 2009). But none of these systems uses RL.

To our knowledge no one has used RL for learning policies for question-answering systems as defined in section 1. Note that Rieser and Lemon (2009) used RL for question-answering, but in their case, question-answering refers to asking for information about songs and artists in an mp3 database, which is very much like a slot-filling task, i.e. the system has to fill a number of slots (e.g. name of band, etc.) in order to query a database of songs and present the right information to the user. As discussed in section 1 our task is rather different.

3 Reinforcement Learning

A dialogue policy is a function from contexts to (possibly probabilistic) decisions that the dialogue system will make in those contexts. Reinforcement Learning (RL) is a machine learning technique used to learn the policy of the system. For an RL-based dialogue system the objective is to maximize the reward it gets during an interaction. RL is used in the framework of Markov Decision Processes (MDPs) or Partially Observable Markov Decision Processes (POMDPs).

In this paper we follow a POMDP-based approach. A POMDP is defined as a tuple (S, A, P, R, O, Z, γ, b_0) where S is the set of states (representing different contexts) which the system may be in (the system's world), A is the set of actions of the system, $P : S \times A \rightarrow P(S, A)$ is the set of transition probabilities between states after taking an action, $R : S \times A \rightarrow \Re$ is the reward function, O is a set of observations that the system can receive about the world, Z is a set of observation probabilities $Z : S \times A \rightarrow Z(S, A)$, and γ a discount factor weighting long-term rewards. At any given time step i the world is in some unobserved state $s_i \in S$. Because s_i is not known exactly, we keep a distribution over states called a *belief state* b, thus $b(s_i)$ is the probability of being in state s_i, with initial belief state b_0. When the system performs an action $\alpha_i \in A$ based on b, following a policy $\pi : S \rightarrow A$, it receives a reward $r_i(s_i, \alpha_i) \in \Re$ and transitions to state s_{i+1} according to $P(s_{i+1}|s_i, \alpha_i) \in P$. The system then receives an observation o_{i+1} according to $P(o_{i+1}|s_{i+1}, \alpha_i)$. The quality of the policy π followed by the agent is measured by the expected future reward also called Q-function, $Q^\pi : S \times A \rightarrow \Re$.

There are several algorithms for learning the optimal dialogue policy and we use Natural Actor Critic (NAC) (Peters and Schaal, 2008), which adopts a natural policy gradient method for policy optimization, also used by (Thomson and Young, 2010; Jurčíček et al., 2012). Policy gradient methods do not directly update the value of state S or Q-function (expected future reward). Instead, the policy π (or parameter $\mathbf{\Theta}$, see below) is directly updated so as to increase the reward of dialogue episodes generated by the previous policy.

A system action a_{sys} is sampled based on the following soft-max (Boltzmann) policy:

$$\pi(a_{sys} = k|\Phi) = Pr(a_{sys} = k|\mathbf{\Phi}, \mathbf{\Theta}) = \frac{\exp(\sum_{i=1}^{I} \phi_i \cdot \theta_{ki})}{\sum_{j=1}^{J} \exp(\sum_{i=1}^{I} \phi_i \cdot \theta_{ji})}$$

Here, $\mathbf{\Phi} = (\phi_1, \phi_2, \ldots, \phi_I)$ is a basis function, which is a vector function of the belief state. $\mathbf{\Theta} = (\theta_{11}, \theta_{12}, \ldots \theta_{1I}, \ldots, \theta_{JI})$ consists of J (# actions) $\times$ I (# features) parameters. The parameter θ_{ji} works as a weight for the i-th feature of the action j and determines the likelihood that the action j is selected. $\mathbf{\Theta}$ is the target of optimization by RL.

During training, RL algorithms require thousands of interactions between the system and the user to achieve good performance. For this reason we need to build a simulated user (SU) (Georgila et al., 2006), that will behave similarly to a real user, and will interact with the policy for thousands of iterations to generate data in order to explore the search space and thus facilitate learning.

Topic	**Example user question/prompt**
introduction	Hello.
personal	Who are you named after?
school	Where do you go to school?
technology	What is artificial intelligence?
interfaces	What is a virtual human?
exhibition	What can I do at Robot Park?

Table 1: Topics of user questions/prompts.

4 The Twins Corpus

As mentioned in section 1 the Twins corpus (Aggarwal et al., 2012) was collected at the Museum of Science in Boston (Swartout et al., 2010). The Twins can answer a number of user questions/prompts in several topics, i.e. about themselves and their technology, about science in general, and about exhibits in the museum. We have divided these topics in six categories shown in Table 1 together with an example for each category.

An example interaction between a museum visitor and the Twins is shown in Figure 1. We can also see the output of the speech recognizer. In the part of the corpus that we use for our experiment automatic speech recognition (ASR) was performed by Otosense, an ASR engine developed by the USC SAIL lab. Natural language understanding and dialogue management are both performed as a single task by the NPCEditor (Leuski and Traum, 2010), a text classification system that classifies the user's query to a system's answer using cross-language information retrieval techniques. When the system fails to understand the user's query it can prompt her to do one of the following:

- rephrase her query (from now on referred to as off-topic response 1, OT1), e.g. "please rephrase your question";
- prompt the user to ask a particular question that the system knows that it can handle (from now on referred to as off-topic response 2, OT2), e.g. "you may ask us about our hobbies";
- cease the dialogue and check out the "behind the scenes" exhibit which explains how the virtual characters work (from now on referred to as off-topic response 3, OT3).

The Twins corpus contains about 200,000 spoken utterances from museum visitors (primarily children) and members of staff or volunteers. For the purposes of this paper we used 1,178 dialogue sessions (11,074 pairs of user and system utterances) collected during March to May 2011. This subset of the corpus contains manual transcriptions of user queries, system responses, and correct responses to user queries (the responses that the system should give when ASR is perfect).

5 User Simulation Model

In order to build a model of user behavior we perform an analysis of the corpus. One of our challenges is that the boundaries between dialogue sessions are hard to define, i.e. it is very hard to automatically calculate whether the same or a new user speaks to the system, unless complex voice identification techniques are employed. We make the reasonable assumption that a new dialogue session starts when there are no questions to the system for a time interval greater than 120 sec.

From each session we extract 30 features. A full list is shown in Table 7 in the Appendix. Our goal is to measure the contribution of each feature to the user's decision with respect to two issues: (1) whether the user will cease the dialogue or not, and (2) what kind of query the user will make next, based on what has happened in the dialogue so far. To do that we use the Chi-squared test, which is commonly used for feature selection.

So to measure the contribution of each feature to whether the user will cease the dialogue or not, we give a binary label to each user query in our corpus, i.e. 1 when the query is the last user query in the dialogue session and 0 otherwise. Then we calculate the contribution of each feature for estimating this label. In Table 8, column 1, in the Appendix, we can see the 10 features that contribute the most to predicting whether the user will cease the dialogue. As we can see the dominant features are not whether the system correctly responded to the user's query, but mostly features based on the dialogue history (e.g. the number of the system's off-topic responses so far) and user type information. Indeed, a further analysis of the corpus showed that children tend to have longer dialogue sessions than adults.

Our next step is the estimation of the contribution of each feature for predicting the user's next query. The label we predict here is the topic of the user's utterance (personal, exhibition, etc., see Table 1). We can see the 10 most predictive features in Table 8, column 2, in the Appendix. The contribution of the most recent user's utterance (previous topic category) is larger than that of dialogue history features. This tendency is the same when we ignore repeated user queries, e.g. when the system makes an error and the user rephrases her query (see Table 8, column 3, in the Appendix). The user type is important for predicting the next user query. In Figure 2 we can see the percentages of user queries per user type and topic.

Based on the above analysis we build a simulated user (SU). The SU simulates the following:

- User type (child, male, female): a child user is sampled with a probability of 51.1%, a male with 31.1%, and a female with 17.8%. These probabilities are estimated from the corpus.
- Number of questions the user is planning to ask (stock of queries): We assume here that the user is planning to ask a number of questions. This number may increase or decrease. For example, it can increase when the system prompts the user to ask about a particular topic (OT2 prompt), and it may decrease when the user decides to cease the dialogue immediately.

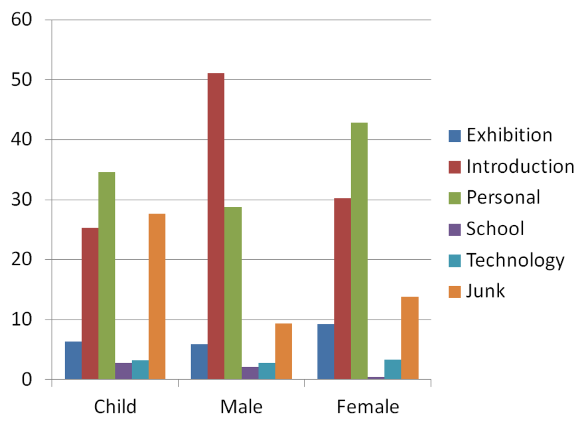

Figure 2: Percentages of user queries per user type and topic.

  The number of questions is sampled from a user type dependent Zipf distribution (strictly speaking the continuous version of the distribution; Parato distribution) the parameter of which is estimated from the corpus using the maximum likelihood criterion. We chose Zipf because it is a long-tail distribution that fits our data (users are not expected to ask a large number of questions). According to this distribution a child user is more likely to have a larger stock of queries than a male or female adult.

- User's reaction: The user has to decide on one of the following. Go to the next topic (Go-on); cease the dialogue if there are no more questions in the stock of queries (Out-of-stock); rephrase the previous query (Rephrase); abandon the dialogue (Give-up) regardless of the remaining questions in the stock; generate a query based on a system recommendation, OT2 prompt (Refill). We calculate the user type dependent probability for these actions from the corpus. But the problem here is that it is not possible to distinguish between the case in which the user asked all the questions in the stock of queries (i.e. all the questions she intended to ask) and left, from the case in which she gave up and abandoned the dialogue. We estimate the percentage of "Give-up" as the difference between the ratio of "Cease" after an incorrect response and the ratio of “Cease” after a correct response, assuming a similar percentage of “Out-of-stock” for both correct and incorrect responses. Likewise, the difference in ”Go-on” for OT2 and other responses is attributed to ”Refill”. The probability of “Rephrase” is estimated from the corpus. For example the probability that a child will rephrase after an OT1 system prompt is 54%, after an erroneous system prompt 38%, etc.

- Topic for next user query (e.g. introduction, personal, etc.): The SU selects a new topic based on user type dependent topic transition bigram probabilities estimated from the corpus.

- User utterance: The SU selects a user utterance from the corpus that matches the current user type and topic. We have split the corpus in groups of user utterances based on user type and topic and we sample accordingly.

- Utterance timing: We simulate utterance timing (duration of pause between system utterance and next user query) per user type and user change. The utterance timing is sampled based on a Gaussian distribution the parameters of which are set based on the corpus statistics. For example, the average duration of a session until the user changes is 62.7 sec with a standard deviation of 71.2 sec.

6 Learning Question-Answering Policies

Our goal is to use RL in order to optimize the system’s response generation. As we saw in the previous section the SU generates a user utterance from our corpus. We do not currently use ASR error simulation but instead a real ASR engine. So the audio file that corresponds to the selected user utterance is forwarded to 3 ASR systems, with child, male, and female acoustic models (AMs) respectively. Then these recognition results are forwarded to the NPCEditor that produces an N-best list of possible system responses (retrieval results). That is, as mentioned in section 4, the NPCEditor classifies each ASR result to a system answer using cross-language information retrieval techniques. The policy can choose one of the NPCEditor retrieval results or reject them and instead present one of the three off-topic prompts (OT1, OT2, or OT3). So the system has 10 possible actions to choose between:

- use the response with the best or the second best score retrieved from the NPCEditor based on a child AM (2 actions);
- use the response with the best or the second best score retrieved from the NPCEditor based on a male AM (2 actions);
- use the response with the best or the second best score retrieved from the NPCEditor based on a female AM (2 actions);
- use the response with the best of the 6 aforementioned scores of the NPCEditor;
- use off-topic prompt OT1;
- use off-topic prompt OT2;
- use off-topic prompt OT3.

We use the following features to optimize our dialogue policy (see section 3). We use the 6 retrieval scores of the NPCEditor (the 2 best scores for each user type ASR result), the previous system action, the ASR confidence scores, the voting scores (calculated by adding the scores of the results that agree), the system’s belief on the user type and user change, and the system’s belief on the user’s previous topic. So we need to learn a POMDP-based policy using these 42 features.

Unlike slot-filling dialogues, defining the reward function is not a simple task (e.g. reward the system for filled and confirmed slots). So in order to define the reward function and thus measure the quality of the dialogue we set up a questionnaire. We asked 5 people to rate 10 dialogues in a 5-Likert scale. Each dialogue session included 5 question-answer pairs. Then we used regression analysis to set the reward for each of the question-answer pair categories shown in Table 2. So for example, responding correctly to an in-domain user question is rewarded (+23.2) whereas providing an erroneous response to a junk question, i.e. treating junk questions as if they were in-domain questions, is penalized (-14.7).

One limitation of this reward function (Reward function 1) is that it does not take into account whether the user has previously experienced an off-topic system prompt. To account for that we define Reward function 2. Here we consider the number of off-topic responses in the two most recent system prompts. Reward function 2 is shown in Table 3.

QA Pair	Reward
in-domain → correct	23.2
in-domain → error	-12.2
in-domain → OT1	-5.4
in-domain → OT2	-8.4
in-domain → OT3	-9.6
junk question → error	-14.7
junk question → OT1	4.8
junk question → OT2	10.2
junk question → OT3	6.1
give up	-16.9

Table 2: Reward function 1.

QA Pair	Reward
in-domain → correct	16.9
in-domain → error	-2.0
in-domain → OT1	13.9
in-domain → OT1(2)	7.3
in-domain → OT2	-7.9
in-domain → OT2(2)	4.2
in-domain → OT3	-15.8
in-domain → OT3(2)	-8.3
junk question → error	-4.6
junk question → OT1	4.1
junk question → OT1(2)	4.1
junk question → OT2	43.4
junk question → OT2(2)	-33.1
junk question → OT3	3.1
junk question → OT3(2)	6.1
give up	-19.5

Table 3: Reward function 2.

As we can see, providing an OT2 as the first off-topic response is a poor action (-7.9); it is preferable to ask the user to rephrase her question (OT1) as a first attempt to recover from the error (+13.9). On the other hand, providing an OT2 prompt, after an off-topic prompt has occured in the previous system prompt, is a reasonable action (+4.2).

7 Evaluation

We compare our learned policy with two baselines. The first baseline, Baseline 1, is the dialogue policy that is used by our system that is currently installed at the Museum of Science in Boston. Baseline 1 selects the best ASR result (i.e. the result with the highest confidence score) out of the results with the 3 different AMs (child, male, and female), and forwards this result to the NPCEditor to retrieve the system's response. If the NPCEditor score is higher than an emprically set pre-defined threshold (see (Leuski and Traum, 2010) for details), then the system presents the retrieved response, otherwise it presents an off-topic prompt. The system presents these off-topic prompts in a fixed order. First, OT1, then OT2, and then OT3.

We also have Baseline 2, which forwards all 3 ASR results to the NPCEditor (using child, male, and female AMs). Then the NPCEditor retrieves 3 results, one for each one of the 3 ASR results, and selects the retrieved result with the highest score. Again if this score is higher than a threshold, the system will present this result, otherwise it will present an off-topic prompt.

Each policy interacts with the SU for 10,000 dialogue sessions and we calculate the average accumulated reward for each dialogue. In Tables 4 and 5 we can see our results for Reward functions 1 and 2 respectively. In both cases the learned policy outperforms both baselines. For both reward functions the most predictive feature is the ASR confidence score when combined with the NPCEditor's retrieval score and the previous system action. Also, for both reward functions the second best feature is "voting" when combined with the retrieval score and the previous system action.

In Table 6 we can see how often the learned policy, which is based on Reward function 1 using all features, selects each one of the 10 system actions (200,000 system turns in total).

Policy	Avg Reward
Baseline 1	24.76 (19.29)
Baseline 2	51.63 (49.84)
Learned Policy - Features	
Retrieval score + system action (*)	46.74
(*) + ASR confidence score	61.59
(*) + User type probability	47.28
(*) + Estimated previous topic	47.87
(*) + Voting	59.94
All features	**60.93**

Table 4: Results with reward function 1. The values in parentheses for Baselines 1 and 2 are the rewards when the NPCEditor does not use the pre-defined threshold.

Policy	Avg Reward
Baseline 1	39.40 (38.51)
Baseline 2	55.45 (54.49)
Learned Policy - Features	
Retrieval score	
+ system action (*)	49.15
(*) + ASR confidence score	69.51
(*) + User type probability	50.15
(*) + Estimated previous topic	49.84
(*) + Voting	69.06
All features	**73.59**

Table 5: Results with reward function 2. The values in parentheses for Baselines 1 and 2 are the rewards when the NPCEditor does not use the pre-defined threshold.

System Action	Frequency
Child + 1st best score	10.33%
Child + 2nd best score	2.70%
Male + 1st best score	13.72%
Male + 2nd best score	1.03%
Female + 1st best score	39.73%
Female + 2nd best score	0.79%
Best of scores 1-6	2.38%
OT1	11.01%
OT2	6.86%
OT3	11.45%

Table 6: Frequency of the system actions of the learned policy that is based on Reward function 1 using all features.

8 Discussion and Conclusion

We showed that RL is a promising technique for learning question-answering policies. Currently we use the same SU for both training and testing the policies. One could argue that this favors the learned policy over the baselines. Because our SU is based on general corpus statistics (probability that the user is child or male or female, number of questions the user is planning to ask, probability of moving to the next topic or ceasing the dialogue, utterance timing statistics) rather than sequential information we believe that this is acceptable. We only use sequential information when we calculate the next topic that the user will choose. That is, due to the way the SU is built and its randomness, we believe that it is very unlikely that the same patterns that were generated during training will be generated during testing. Thus we do not anticipate that our results would be different if for testing we used a SU trained on a different part of the corpus, or that the learned policy is favored over the baselines. However, this is something to verify experimentally in future work.

For future work we would also like to do the following. First of all, currently we are in the process of analyzing user satisfaction questionnaires from museum visitors in order to define a better reward function. Second, we would like to use voice identification techniques to automatically estimate from the corpus the statistics of having more than one user or alternating users in the same session. Third, and most important, we would like to incorporate the learned policy into the system that is currently installed in the museum and evaluate it with real users. Fourth, currently our SU is based on only some of our findings from the analysis of the corpus. We intend to build a more complex and hopefully more realistic SU based on our full corpus analysis. Finally, we will also experiment with learning policies directly from the data (Li et al., 2009).

To conclude, we analyzed a corpus of interactions of museum visitors with two virtual characters that serve as guides at the Museum of Science in Boston, in order to build a realistic model of user behavior when interacting with these characters. Based on this analysis, we built a SU and used it for learning the dialogue policy of the virtual characters using RL. We compared our learned policy with two baselines, one of which was the dialogue policy of the original system that was used for collecting the corpus and that is currently installed at the Museum of Science in Boston. Our learned policy outperformed both baselines which shows that RL is a promising technique for learning question-answering dialogue policies.

Acknowledgments

This work was funded by the NSF grant #1117313. The Twins corpus collection was supported by the NSF grant #0813541.

References

Priti Aggarwal, Ron Artstein, Jillian Gerten, Athanasios Katsamanis, Shrikanth Narayanan, Angela Nazarian, and David Traum. 2012. The Twins corpus of museum visitor questions. In *Proc. of the Language Resources and Evaluation Conference (LREC)*, pages 2355–2361, Istanbul, Turkey.

Hua Ai and Diane Litman. 2008. Assessing dialog system user simulation evaluation measures using human judges. In *Proc. of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 622–629, Columbus, OH, USA.

Min Chi, Kurt VanLehn, Diane Litman, and Pamela Jordan. 2011. Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies. *User Modeling and User-Adapted Interaction*, 21(1-2):137–180.

Sudeep Gandhe, Nicolle Whitman, David Traum, and Ron Artstein. 2009. An integrated authoring tool for tactical questioning dialogue systems. In *Proc. of the IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, Pasadena, CA, USA.

Kallirroi Georgila and David Traum. 2011a. Learning culture-specific dialogue models from non culture-specific data. In *Proc. of HCI International, Lecture Notes in Computer Science Vol. 6766*, pages 440–449, Orlando, FL, USA.

Kallirroi Georgila and David Traum. 2011b. Reinforcement learning of argumentation dialogue policies in negotiation. In *Proc. of Interspeech*, pages 2073–2076, Florence, Italy.

Kallirroi Georgila, James Henderson, and Oliver Lemon. 2006. User simulation for spoken dialogue systems: Learning and evaluation. In *Proc. of Interspeech*, pages 1065–1068, Pittsburgh, PA, USA.

Kallirroi Georgila, Maria K. Wolters, and Johanna D. Moore. 2010. Learning dialogue strategies from older and younger simulated users. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGdial)*, pages 103–106, Tokyo, Japan.

Peter A. Heeman. 2009. Representing the reinforcement learning state in a negotiation dialogue. In *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Merano, Italy.

Arne Jönsson, Frida Andén, Lars Degerstedt, Annika Flycht-Eriksson, Magnus Merkel, and Sara Norberg. 2004. Experiences from combining dialogue system development with information access techniques. In *New Directions in Question Answering, Mark T. Maybury (Ed)*, pages 153–164. AAAI/MIT Press.

Filip Jurčíček, Blaise Thomson, and Steve Young. 2012. Reinforcement learning for parameter estimation in statistical spoken dialogue systems. *Computer Speech and Language*, 26(3):168–192.

Anton Leuski and David Traum. 2010. Practical language processing for virtual humans. In *Proc. of the 22nd Annual Conference on Innovative Applications of Artificial Intelligence (IAAI)*, Atlanta, GA, USA.

Anton Leuski, Ronakkumar Patel, David Traum, and Brandon Kennedy. 2006. Building effective question answering characters. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGdial)*, pages 18–27, Sydney, Australia.

Lihong Li, Jason D. Williams, and Suhrid Balakrishnan. 2009. Reinforcement learning for dialog management using least-squares policy iteration and fast feature selection. In *Proc. of Interspeech*, pages 2475–2478, Brighton, United Kingdom.

Teruhisa Misu, Komei Sugiura, Kiyonori Ohtake, Chiori Hori, Hideki Kashioka, Hisashi Kawai, and Satoshi Nakamura. 2010. Modeling spoken decision making dialogue and optimization of its dialogue strategy. In *Proc. of the Annual SIGdial Meeting on Discourse and Dialogue (SIGdial)*, pages 221–224, Tokyo, Japan.

Rieks op den Akker, Harry Bunt, Simon Keizer, and Boris van Schooten. 2005. From question answering to spoken dialogue: Towards an information search assistant for interactive multimodal information extraction. In *Proc. of Interspeech*, pages 2793–2796, Lisbon, Portugal.

Jan Peters and Stefan Schaal. 2008. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190.

Verena Rieser and Oliver Lemon. 2009. Does this list contain what you were searching for? Learning adaptive dialogue strategies for interactive question answering. *Natural Language Engineering*, 15(1):55–72.

William Swartout, David Traum, Ron Artstein, Dan Noren, Paul Debevec, Kerry Bronnenkant, Josh Williams, Anton Leuski, Shrikanth Narayanan, Diane Piepol, Chad Lane, Jacquelyn Morie, Priti Aggarwal, Matt Liewer, Jen-Yuan Chiang, Jillian Gerten, Selina Chu, and Kyle White. 2010. Ada and Grace: Toward realistic and engaging virtual museum guides. In *Proc. of the International Conference on Intelligent Virtual Agents (IVA)*, pages 286–300, Philadelphia, PA, USA.

Joel R. Tetreault and Diane J. Litman. 2008. A reinforcement learning approach to evaluating state representations in spoken dialogue systems. *Speech Communication*, 50(8-9):683–696.

Blaise Thomson and Steve Young. 2010. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech and Language*, 24(4):562–588.

Sebastian Varges, Fuliang Weng, and Heather Pon-Barry. 2009. Interactive question answering and constraint relexation in spoken dialogue systems. *Natural Language Engineering*, 15(1):9–30.

Ellen M. Voorhees. 2001. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378.

Appendix

Features	Features
average ASR accuracy of user queries	if system correctly answered current user query
# user queries	if system responded with off-topic prompt to current user query
# correct system responses	# times user repeated current query
# incorrect system responses	# successive incorrect system responses
# off-topic system prompts	# successive off-topic system prompts
% correct system responses	# user queries for topic "introduction"
% incorrect system responses	# user queries for topic "personal"
user type (child, male, female)	# user queries for topic "school"
if user asks example query 1	# user queries for topic "technology"
if user asks example query 2	# user queries for topic "interfaces"
if user asks example query 3	# user queries for topic "exhibition"
if user asks example query 4	# user queries for other topics
if system correctly responds to example query 1	if system correctly responds to example query 3
if system correctly responds to example query 2	if system correctly responds to example query 4
# junk user queries	previous topic category

Table 7: List of features used in predicting when the user will cease a session (Cease Dialogue), what the user will say next (Say Next 1), and what the user will say next after removing repeated user queries (Say Next 2). Example query 1 is "who are you named after?"; example query 2 is "are you a computer?"; example query 3 is "what do you like to do for fun?"; example query 4 is "what is artificial intelligence?".

Cease Dialogue	Say Next 1	Say Next 2
average ASR accuracy of user queries	previous topic category	previous topic category
user type (child, male, female)	# user queries for topic "personal"	# junk user queries
# off-topic system prompts	# user queries	# successive incorrect system responses
# successive off-topic system prompts	# junk user queries	if system correctly answered current user query
# incorrect system responses	% correct system responses	user type (child, male, female)
# user queries	% incorrect system responses	% incorrect system responses
# junk user queries	# incorrect system responses	% correct system responses
# user queries for other topics	# user queries for other topics	# incorrect system responses
if system responded with off-topic prompt to current user query	# correct system responses	# off-topic system prompts
% correct system responses	user type (child, male, female)	# user queries

Table 8: List of the 10 most dominant features (in order of importance) in predicting when the user will cease a session (Cease Dialogue), what the user will say next (Say Next 1), and what the user will say next after removing repeated user queries (Say Next 2).