Modeling Spoken Decision Making Dialogue and Optimization of its Dialogue Strategy

Teruhisa Misu, Komei Sugiura, Kiyonori Ohtake,
Chiori Hori, Hideki Kashioka, Hisashi Kawai and Satoshi Nakamura
MASTAR Project, NICT
Kyoto, Japan.
teruhisa.misu@nict.go.jp

Abstract

This paper presents a spoken dialogue framework that helps users in making decisions. Users often do not have a definite goal or criteria for selecting from a list of alternatives. Thus the system has to bridge this knowledge gap and also provide the users with an appropriate alternative together with the reason for this recommendation through dialogue. We present a dialogue state model for such decision making dialogue. To evaluate this model, we implement a trial sightseeing guidance system and collect dialogue data. Then, we optimize the dialogue strategy based on the state model through reinforcement learning with a natural policy gradient approach using a user simulator trained on the collected dialogue corpus.

1 Introduction

In many situations where spoken dialogue interfaces are used, information access by the user is not a goal in itself, but a means for decision making (Polifroni and Walker, 2008). For example, in a restaurant retrieval system, the user's goal may not be the extraction of price information but to make a decision on candidate restaurants based on the retrieved information.

This work focuses on how to assist a user who is using the system for his/her decision making, when he/she does not have enough knowledge about the target domain. In such a situation, users are often unaware of not only what kind of information the system can provide but also their own preference or factors that they should emphasize. The system, too, has little knowledge about the user, or where his/her interests lie. Thus, the system has to bridge such gaps by sensing (potential) preferences of the user and recommend information that the user would be interested in, considering a trade-off with the length of the dialogue.

We propose a model of dialogue state that considers the user's preferences as well as his/her knowledge about the domain changing through a decision making dialogue. A user simulator is trained on data collected with a trial sightseeing system. Next, we optimize the dialogue strategy of the system via reinforcement learning (RL) with a natural policy gradient approach.

2 Spoken decision making dialogue

We assume a situation where a user selects from a given set of alternatives. This is highly likely in real world situations; for example, the situation wherein a user selects one restaurant from a list of candidates presented by a car navigation system. In this work, we deal with a sightseeing planning task where the user determines the sightseeing spot to visit, with little prior knowledge about the target domain. The study of (Ohtake et al., 2009), which investigated human-human dialogue in such a task, reported that such consulting usually consists of a sequence of information requests from the user, presentation and elaboration of information about certain spots by the guide followed by the user's evaluation. We thus focus on these interactions.

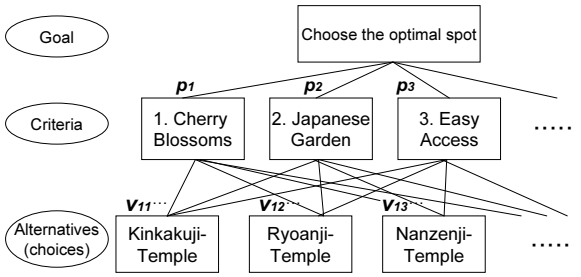

Figure 1: Hierarchy structure for sightseeing guidance dialogue

Several studies have featured decision support systems in the operations research field, and the typical method that has been employed is the Analytic Hierarchy Process (Saaty, 1980) (AHP). In AHP, the problem is modeled as a hierarchy that consists of the decision goal, the alternatives for achieving it, and the criteria for evaluating these alternatives. An example hierarchy using these criteria is shown in Figure 1.

For the user, the problem of making an optimal decision can be solved by fixing a weight vector $\mathbf{P}_{user} = (p_1, p_2, \ldots, p_M)$ for criteria and local weight matrix $\mathbf{V}_{user} = (v_{11}, v_{12}, \ldots, v_{1M}, \ldots, v_{NM})$ for alternatives in terms of the criteria. The optimal alternative is then identified by selecting the spot k with the maximum priority of $\sum_{m=1}^{M} p_m v_{km}$. In typical AHP methods, the procedure of fixing these weights is often conducted through pairwise comparisons for all the possible combinations of criteria and spots in terms of the criteria, followed by weight tuning based on the results of these comparisons (Saaty, 1980). However, this methodology cannot be directly applied to spoken dialogue systems. The information about the spot in terms of the criteria is not known to the users, but is obtained only via navigating through the system's information. In addition, spoken dialogue systems usually handle several candidates and criteria, making pairwise comparison a costly affair.

We thus consider a spoken dialogue framework that estimates the weights for the user's preference (potential preferences) as well as the user's knowledge

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 221–224,
The University of Tokyo, September 24-25, 2010.

about the domain through interactions of information retrieval and navigation.

3 Decision support system with spoken dialogue interface

The dialogue system we built has two functions: answering users' information requests and recommending information to them. When the system is requested to explain about the spots or their determinants, it explains the sightseeing spots in terms of the requested determinant. After satisfying the user's request, the system then provides information that would be helpful in making a decision (e.g., instructing what the system can explain, recommending detailed information of the current topic that the user might be interested in, etc.). Note that the latter is optimized via RL (see Section 4).

3.1 Knowledge base

Our back-end DB consists of 15 sightseeing spots as alternatives and 10 determinants described for each spot. We select determinants that frequently appear in the dialogue corpus of (Ohtake et al., 2009) (e.g. cherry blossoms, fall foliage). The spots are annotated in terms of these determinants if they apply to them. The value of the evaluation e_{nm} is "1" when the spot n applies to the determinant m and "0" when it does not.

3.2 System initiative recommendation

The content of the recommendation is determined based on one of the following six methods:

1. **Recommendation of determinants based on the currently focused spot (Method 1)**
   This method is structured on the basis of the user's current focus on a particular spot. Specifically, the system selects several determinants related to the current spot whose evaluation is "1" and presents them to the user.
2. **Recommendation of spots based on the currently focused determinant (Method 2)**
   This method functions on the basis of the focus on a certain specific determinant.
3. **Open prompt (Method 3)**
   The system does not make a recommendation, and presents an open prompt.
4. **Listing of determinants 1 (Method 4)**
   This method lists several determinants to the user in ascending order from the low level user knowledge $\mathbf{K}_{sys}$ (that the system estimates). ($\mathbf{K}_{sys}$, $\mathbf{P}_{sys}$, p_m and $Pr(p_m = 1)$ are defined and explained in Section 4.2.)
5. **Listing of determinants 2 (Method 5)**
   This method also lists the determinants, but the order is based on the user's high preference $\mathbf{P}_{sys}$ (that the system estimates).
6. **Recommendation of user's possibly preferred spot (Method 6)**
   The system recommends a spot as well as the determinants that the users would be interested in based on the estimated preference $\mathbf{P}_{sys}$. The system selects one spot k with a maximum value of $\sum_{m=1}^{M} Pr(p_m = 1) \cdot e_{k,m}$. This idea is based on collaborative filtering which is often used for recommender systems (Breese et al., 1998). This method will be helpful to users if the system successfully estimates the user's preference; however, it will be irrelevant if the system does not.

We will represent these recommendations through a dialogue act expression, ($ca_{sys}\{sc_{sys}\}$), which consists of a communicative act ca_{sys} and the semantic content sc_{sys}. (For example $Method1\{(Spot_5), (Det_3, Det_4, Det5)\}$, $Method3\{NULL, NULL\}$, etc.)

4 Optimization of dialogue strategy

4.1 Models for simulating a user

We introduce a user model that consists of a tuple of knowledge vector $\mathbf{K}_{user}$, preference vector $\mathbf{P}_{user}$, and local weight matrix $\mathbf{V}_{user}$. In this paper, for simplicity, a user's preference vector or weight for determinants $\mathbf{P}_{user} = (p_1, p_2, \ldots, p_M)$ is assumed to consist of binary parameters. That is, if the user is interested in (or potentially interested in) the determinant m and emphasizes it when making a decision, the preference p_m is set to "1". Otherwise, it is set to "0". In order to represent a state that the user has potential preference, we introduce a knowledge parameter $\mathbf{K}_{user} = (k_1, k_2, \ldots, k_M)$ that shows if the user has the perception that the system is able to handle or he/she is interested in the determinants. k_m is set to "1" if the user knows (or is listed by system's recommendations) that the system can handle determinant m and "0" when he/she does not. For example, the state that the determinant m is the potential preference of a user (but he/she is unaware of that) is represented by ($k_m = 0$, $p_m = 1$). This idea is in contrast to previous research which assumes some fixed goal observable by the user from the beginning of the dialogue (Schatzmann et al., 2007). A user's local weight v_{nm} for spot n in terms of determinant m is set to "1", when the system lets the user know that the evaluation of spots is "1" through recommendation Methods 1, 2 and 6.

We constructed a user simulator that is based on the statistics calculated through an experiment with the trial system (Misu et al., 2010) as well as the knowledge and preference of the user. That is, the user's communicative act ca^t_{user} and the semantic content sc^t_{user} for the system's recommendation a^t_{sys} are generated based on the following equation:

$$\begin{aligned} &Pr(ca^t_{user}, sc^t_{user} | ca^t_{sys}, sc^t_{sys}, \mathbf{K}_{user}, \mathbf{P}_{user}) \\ &\quad = Pr(ca^t_{user} | ca^t_{sys}) \\ &\qquad \cdot Pr(sc^t_{user} | \mathbf{K}_{user}, \mathbf{P}_{user}, ca^t_{user}, ca^t_{sys}, sc^t_{sys}) \end{aligned}$$

This means that the user's communicative act ca_{user} is sampled based on the conditional probability of $Pr(ca^t_{user}|ca^t_{sys})$ in (Misu et al., 2010). The semantic content sc_{user} is selected based on the user's preference $\mathbf{P}_{user}$ under current knowledge about the determinants $\mathbf{K}_{user}$. That is, the sc is sampled from the determinants within the user's knowledge ($k_m = 1$) based on the probability that the user requests the determinant of his/her preference/non-preference, which is also calculated from the dialogue data of the trial system.

4.2 Dialogue state expression

We defined the state expression of the user in the previous section. However the problem is that for the system, the state ($\mathbf{P}_{user}, \mathbf{K}_{user}, \mathbf{V}_{user}$) is not observable, but is only estimated from the interactions with the user. Thus, this model is a partially observable Markov decision process (POMDP) problem. In order to estimate unobservable properties of a POMDP

Priors of the estimated state:
- Knowledge: $\mathbf{K}_{sys} = (0.22, 0.01, 0.02, 0.18, \dots)$
- Preference: $\mathbf{P}_{sys} = (0.37, 0.19, 0.48, 0.38, \dots)$

Interactions (observation):
- System recommendation:
  $a_{sys} = Method1\{(Spot_5), (Det_1, Det_3, Det4)\}$
- User query:
  $a_{user} = Accept\{(Spot_5), (Det_3)\}$

Posterior of the estimated state:
- Knowledge: $\mathbf{K}_{sys} = (1.00, 0.01, 1.00, 1.00, \dots)$
- Preference: $\mathbf{P}_{sys} = (0.26, 0.19, 0.65, 0.22, \dots)$

User's knowledge acquisition:
- Knowledge: $\mathbf{K}_{user} \leftarrow \{k_1 = 1, k_3 = 1, k_4 = 1\}$
- Local weight: $\mathbf{V}_{user} \leftarrow \{v_{51} = 1, v_{53} = 1, v_{54} = 1\}$

Figure 2: Example of state update

and handle the problem as an MDP, we introduce the system's inferential user knowledge vector $\mathbf{K}_{sys}$ or probability distribution (estimate value) $\mathbf{K}_{sys} = (Pr(k_1 = 1), Pr(k_2 = 1), \dots, Pr(k_M = 1))$ and that of preference $\mathbf{P}_{sys} = (Pr(p_1 = 1), Pr(p_2 = 1), \dots, Pr(p_M = 1))$.

The dialogue state DS^{t+1} or estimated user's dialogue state of the step $t+1$ is assumed to be dependent only on the previous state DS^t, as well as the interactions $I^t = (a^t_{sys}, a^t_{user})$.

The estimated user's state is represented as a probability distribution and is updated by each interaction. This corresponds to representing the user types as a probability distribution, whereas the work of (Komatani et al., 2005) classifies users to several discrete user types. The estimated user's preference $\mathbf{P}_{sys}$ is updated when the system observes the interaction I^t. The update is conducted based on the following Bayes' theorem using the previous state DS^t as a prior.

$$Pr(p_m = 1|I^t) = \frac{Pr(I^t|p_m=1)Pr(p_m=1)}{Pr(I^t|p_m=1)Pr(p_m=1)+Pr(I^t|(p_m=0))Pr(1-Pr(p_m=1))}$$

Here, $Pr(I^t|p_m = 1), Pr(I^t|(p_m = 0)$ to the right side was obtained from the dialogue corpus of (Misu et al., 2010). This posterior is then used as a prior in the next state update using interaction I^{t+1}. An example of this update is illustrated in Figure 2.

4.3 Reward function

The reward function that we use is based on the number of agreed attributes between the user preference and the decided spot. Users are assumed to determine the spot based on their preference $\mathbf{P}_{user}$ under their knowledge $\mathbf{K}_{user}$ (and local weight for spots $\mathbf{V}_{user}$) at that time, and select the spot k with the maximum priority of $\sum_m k_k \cdot p_k \cdot v_{km}$. The reward $\mathbf{R}$ is then calculated based on the improvement in the number of agreed attributes between the user's actual (potential) preferences and the decided spot k over the expected agreement by random spot selection.

$$R = \sum_{m=1}^{M} p_m \cdot e_{k,m} - \frac{1}{N} \sum_{n=1}^{N} \sum_{m=1}^{M} p_m \cdot e_{n,m}$$

For example, if the decided spot satisfies three preferences and the average agreement of the agreement by random selection is 1.3, then the reward is 1.7.

4.4 Optimization by reinforcement learning

The problem of system recommendation generation is optimized through RL. The MDP **(S, A, R)** is defined as follows. The state parameter $\mathbf{S} = (s_1, s_2, \dots, s_I)$ is generated by extracting the features of the current dialogue state DS^t. We use the following 29 features [1]. 1. Parameters that indicate the # of interactions from the beginning of the dialogue. This is approximated by five parameters using triangular functions. 2. User's previous communicative act (1 if $a^{t-1}_{user} = x_i$, otherwise 0). 3. System's previous communicative act (1 if $a^{t-1}_{sys} = y_j$, otherwise 0). 4. Sum of the estimated user knowledge about determinants ($\sum_{n=1}^{N} Pr(k_n = 1)$). 5. Number of presented spot information. 6. Expectation of the probability that the user emphasizes the determinant in the current state ($Pr(k_n = 1) \times Pr(p_n = 1)$) (10 parameters). The action set $\mathbf{A}$ consists of the six recommendation methods shown in subsection 3.2. Reward $\mathbf{R}$ is given by the reward function of subsection 4.3.

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

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

Here, $\mathbf{\Theta} = (\theta_{11}, \theta_{12}, \dots \theta_{1I}, \dots, \theta_{JI})$ 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. This $\mathbf{\Theta}$ is the target of optimization by RL. We adopt the Natural Actor Critic (NAC) (Peters and Schaal, 2008), which adopts a natural policy gradient method as the policy optimization method.

4.5 Experiment by dialogue simulation

For each simulated dialogue session, a simulated user ($\mathbf{P}_{user}, \mathbf{K}_{user}, \mathbf{V}_{user}$) is sampled. A preference vector $\mathbf{P}_{user}$ of the user is generated so that he/she has four preferences. As a result, four parameters in $\mathbf{P}_{user}$ are "1" and the others are "0". This vector is fixed throughout the dialogue episode. This sampling is conducted based on the rate proportional to the percentage of users who emphasize it for making decisions (Misu et al., 2010). The user's knowledge $\mathbf{K}_{user}$ is also set based on the statistics of the "percentage of users who stated the determinants before system recommendation". For each determinant, we sample a random valuable r that ranges from "0" to "1", and k_m is set to "1" if r is smaller than the percentage. All the parameters of local weights $\mathbf{V}_{user}$ are initialized to "0", assuming that users have no prior knowledge about the candidate spots. As for system parameters, the estimated user's preference $\mathbf{P}_{sys}$ and knowledge $\mathbf{K}_{sys}$ are initialized based on the statistics of our trial system (Misu et al., 2010).

We assumed that the system does not misunderstand the user's action. Users are assumed to continue a dialogue session for 20 turns[2], and episodes are sampled using the policy π at that time and the user simulator

[1] Note that about half of them are continuous variables and that the value function cannot be denoted by a lookup table.

[2] In practice, users may make a decision at any point once they are satisfied collecting information. And this is the reason why we list the rewards in the early dialogue stage in

Table 1: Comparison of reward with baseline methods

	Reward ($\pm std$)			
Policy	T = 5	T = 10	T = 15	T = 20
NAC	0.96 (0.53)	1.04 (0.51)	1.12 (0.50)	1.19 (0.48)
B1	0.02 (0.42)	0.13 (0.54)	0.29 (0.59)	0.34 (0.59)
B2	0.46 (0.67)	0.68 (0.65)	0.80 (0.61)	0.92 (0.56)

Table 2: Comparison of reward with discrete dialogue state expression

	Reward ($\pm std$)			
State	T = 5	T = 10	T = 15	T = 20
PDs	0.96 (0.53)	1.04 (0.51)	1.12 (0.50)	1.19 (0.48)
Discrete	0.89 (0.60)	0.97 (0.56)	1.03 (0.54)	1.10 (0.52)

Table 3: Effect of estimated preference and knowledge

	Reward ($\pm std$)			
Policy	T = 5	T = 10	T = 15	T = 20
Pref+Know	0.96 (0.53)	1.04 (0.51)	1.12 (0.50)	1.19 (0.48)
Pref only	0.94 (0.57)	0.96 (0.55)	1.02 (0.55)	1.09 (0.53)
Know only	0.96 (0.59)	1.00 (0.56)	1.08 (0.53)	1.15 (0.51)
No Pref or Know	0.93 (0.57)	0.96 (0.55)	1.02 (0.53)	1.08 (0.52)

of subsection 4.1. In each turn, the system is rewarded using the reward function of subsection 4.3. The policy (parameter $\mathbf{\Theta}$) is updated using NAC in every 2,000 dialogues.

4.6 Experimental result

The policy was fixed at about 30,000 dialogue episodes. We analyzed the learned dialogue policy by examining the value of weight parameter $\mathbf{\Theta}$. We compared the parameters of the trained policy between actions[3]. The weight of the parameters that represent the early stage of the dialogue was large in Methods 4 and 5. On the other hand, the weight of the parameters that represent the latter stage of the dialogue was large in Methods 2 and 6. This suggests that in the trained policy, the system first bridges the knowledge gap between the user, estimates the user's preference, and then, recommends specific information that would be useful to the user.

Next, we compared the trained policy with the following baseline methods.

1. **No recommendation (B1)**
   The system only provides the requested information and does not generate any recommendations.
2. **Random recommendation (B2)**
   The system randomly chooses a recommendation from six methods.

The comparison of the average reward between the baseline methods is listed in Table 1. Note that the oracle average reward that can be obtained only when the user knows all knowledge about the knowledge base (it requires at least 50 turns) was 1.45. The reward by the strategy optimized by NAC was significantly better than that of baseline methods ($n = 500, p < .01$).

We then compared the proposed method with the case where estimated user's knowledge and preference are represented as discrete binary parameters instead of probability distributions (PDs). That is, the estimated user's preference p_m of determinant m is set to "1" when the user requested the determinant, otherwise it is "0". The estimated user's knowledge k_m is set to "1" when the system lets the user know the determinant, otherwise it is "0". Another dialogue strategy was trained using this dialogue state expression. This result is shown in Table 2. The proposed method that represents the dialogue state as a probability distribution outperformed ($p < .01$ (T=15,20)) the method using a discrete state expression.

We also compared the proposed method with the case where either one of estimated preference or knowledge was used as a feature for dialogue state in order to carefully investigate the effect of these factors. In the proposed method, expectation of the probability that the user emphasizes the determinant ($Pr(k_n = 1) \times Pr(p_n = 1)$) was used as a feature of dialogue state. We evaluated the performance of the cases where the estimated knowledge $Pr(k_n = 1)$ or estimated preference $Pr(p_n = 1)$ was used instead of the expectation of the probability that the user emphasizes the determinant. We also compared with the case where no preference/knowledge feature was used. This result is shown in Table 3. We confirmed that significant improvement ($p < .01$ (T=15,20)) was obtained by taking into account the estimated knowledge of the user.

5 Conclusion

In this paper, we presented a spoken dialogue framework that helps users select an alternative from a list of alternatives. We proposed a model of dialogue state for spoken decision making dialogue that considers knowledge as well as preference of the user and the system, and its dialogue strategy was trained by RL. We confirmed that the learned policy achieved a better recommendation strategy over several baseline methods.

Although we dealt with a simple recommendation strategy with a fixed number of recommendation components, there are many possible extensions to this model. The system is expected to handle a more complex planning of natural language generation. We also need to consider errors in speech recognition and understanding when simulating dialogue.

References

J. Breese, D. Heckerman, and C. Kadie. 1998. "empirical analysis of predictive algorithms for collaborative filtering". In *"Proc. the 14th Annual Conference on Uncertainty in Artificial Intelligence"*, pages 43–52.

K. Komatani, S. Ueno, T. Kawahara, and H. Okuno. 2005. User Modeling in Spoken Dialogue Systems to Generate Flexible Guidance. *User Modeling and User-Adapted Interaction*, 15(1):169–183.

T. Misu, K. Ohtake, C. Hori, H. Kashioka, H. Kawai, and S. Nakamura. 2010. Construction and Experiment of a Spoken Consulting Dialogue System. In *Proc. IWSDS*.

K. Ohtake, T. Misu, C. Hori, H. Kashioka, and S. Nakamura. 2009. Annotating Dialogue Acts to Construct Dialogue Systems for Consulting. In *Proc. The 7th Workshop on Asian Language Resources*, pages 32–39.

J. Peters and S. Schaal. 2008. Natural Actor-Critic. *Neurocomputing*, 71(7-9):1180–1190.

J. Polifroni and M. Walker. 2008. Intensional Summaries as Cooperative Responses in Dialogue: Automation and Evaluation. In *Proc. ACL/HLT*, pages 479–487.

T. Saaty. 1980. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. Mcgraw-Hill.

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. 2007. Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc. HLT/NAACL*.

the following subsections. In our trial system, the dialogue length was 16.3 turns with a standard deviation of 7.0 turns.

[3]The parameters can be interpreted as the size of the contribution for selecting the action.