Cost-level integration of statistical and rule-based dialog managers

Shinji Watanabe[1], John R. Hershey[1], Tim K. Marks[1], Youichi Fujii[2], and Yusuke Koji[2]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA
[2]Information Technology R&D Center, Mitsubishi Electric, Kamakura, Japan

{watanabe,hershey,tmarks}@merl.com,
{Fujii.Yoichi@bk,Koji.Yusuke@bx}.MitsubishiElectric.co.jp

Abstract

Statistical dialog managers can potentially make more robust decisions than their rule-based counterparts, because they can account for uncertainties due to errors in speech recognition and natural language understanding. In practice, however, statistical dialog managers can be difficult to use, as they may require a large number of parameters to be inferred from limited data. Consequently, hand-crafted rule based systems are still effective for practical use. This paper proposes a method to integrate an existing rule-based dialog manager with a statistical dialog manager based on Bayes decision theory, by incorporating the rule-based dialog manager into the cost function of the statistical dialog manager. The cost function has two parts: an efficiency cost that penalizes inefficient actions, as in conventional statistical dialog approaches, and a regularization cost that slightly penalizes system actions that differ from those that would be chosen by the rule-based system. Our experiments, which use a destination-setting task in an automobile dialog scenario, demonstrate that the integrated system produces system actions that are similar to those of an existing rule-based dialog manager but enable task completion using fewer turns than the rule-based system.

Index Terms: spoken dialog system, statistical dialog manager, rule based system, cost-level integration, goal estimation

10.21437/Interspeech.2014-77

1. Introduction

A core component of any spoken dialog system is the dialog manager. Based on the history of a dialog in progress, the dialog manager provides an appropriate system action to help a user to complete a task [1]. For each utterance made by the user, the input to the dialog manager is the *user intention*, which represents the user's intended meaning as extracted from the user utterance by the Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) components of the system. These components often produce errors in the obtained user intentions.

Statistical dialog managers have been widely studied [2–7] as a way to handle these errors. By modeling the uncertainties in the user intentions obtained from the ASR and NLU outputs, statistical dialog managers can potentially make more robust decisions than rule-based dialog managers. Statistical dialog managers are generally formulated based on Bayes decision theory, where the optimal system action can be determined by minimizing the expected cost function (or, equivalently, maximizing the expected reward function in the reinforcement learning context [8]). Despite the uncertainty in the observations, a statistical dialog manager can produce a robust decision by optimizing the expected value of the cost function over the distribution of all possible past, present, and future user intentions and system actions. Partially Observable Markov Decision Processes (POMDPs) [9] have been studied as a promising framework for statistical dialog managers in spoken dialog systems [2, 6]. However, statistical dialog managers can be difficult to use in practice because they may have a large number of parameters to be inferred from limited data, and this problem of sparse data can make a statistical dialog manager unstable [10, 11]. Consequently, even though hand-crafted rule based systems are not as robust in the face of uncertain observations, rule-based dialog managers remain a practical choice and are used in many real-world systems [12–15].

This paper proposes a method for integrating a statistical dialog manager with an existing rule-based dialog manager. The relationships between user intentions and system actions that are explicitly encoded into the rule-based system are used as a means of regularizing the mapping between user intentions and system actions that is inferred by the statistical dialog manager. Our formulation is based on Bayes decision theory, and the integration with the rule-based system is accomplished by means of a combined cost function. The cost function is composed of two parts: an efficiency cost that penalizes redundant system actions, and a regularization cost that penalizes system actions that are different from those that would be selected by the rule-based dialog manager.

As in conventional statistical dialog managers, the efficiency cost encourages short dialogs by giving higher penalties for system actions that increase the length of the dialog. In this paper, we represent an existing rule-based dialog manager using a graph structure and assign to each potential action an efficiency cost, which is defined as the average number of dialog turns needed to reach a hypothesized goal from that system action [16, 17]. To compute this cost, we first represent the rule-based dialog manager using a (deterministic) finite state automaton, where the nodes represent the system actions and the arcs represent the user intentions, as shown in Figure 1. The cost is computed as the average distance from the node representing a potential system action to the node representing the goal. This distance is computed by considering the graph structure as a directed unweighted graph (assuming equal probability for possible user intentions) and using a breadth-first dynamic programming search to compute the length of all paths to each hypothesized goal state, given each hypothesized system action. The length is averaged over all paths to get the efficiency cost. In addition to this efficiency cost, there is a small regularization cost, which is higher for system actions that are not identical to those that would be chosen by the rule-based dialog manager.

Thus, the cost function for our statistical dialog manager consists of two parts, both of which are derived from a rule-based dialog manager. Through this cost function design, the proposed statistical dialog manager is tightly integrated with the rule-based dialog manager. Note that unlike the offline policy

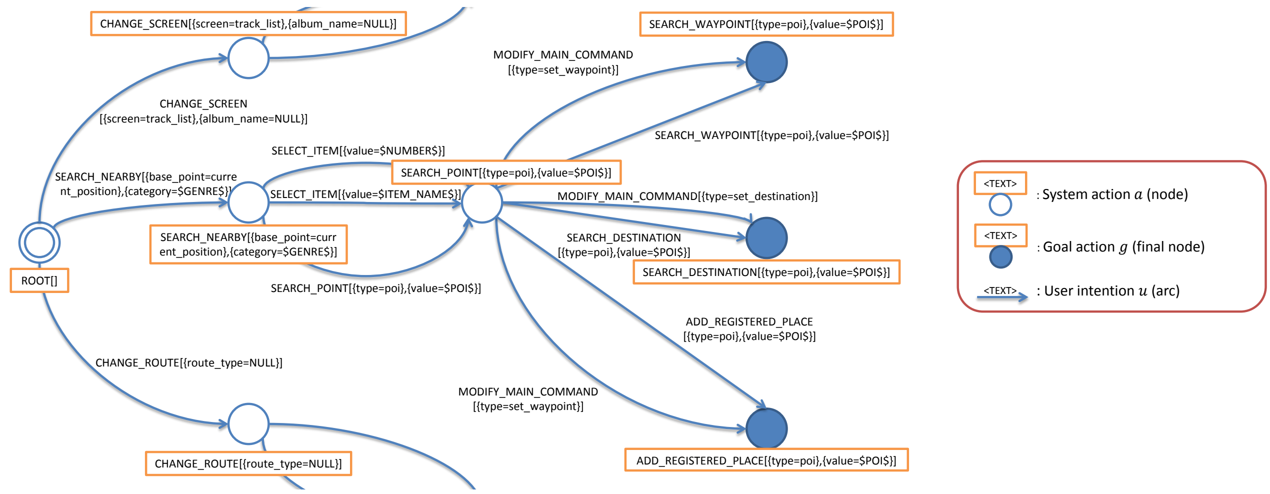

Figure 1: *Part of a finite-state-automaton representation of a rule-based dialog manager. Nodes represent system actions, and arcs are user intentions. Note that a single system action can be reached via multiple user intentions (e.g., specifying a slot value or uttering a full intention). The actual rule-based system has various additional arcs, including some exception arcs (e.g., going back to the root node), that are omitted in our study for simplicity.*

optimization required in reinforcement learning, the proposed approach can compute the expected costs on the fly during dialog management.

The proposed approach was evaluated using database of dialog logs from user interactions with the existing rule-based system, in a destination-setting task in an automobile scenario.

2. Formulation

2.1. Bayes decision theory for dialog manager

We first define our statistical decision problem [18] as that of choosing an optimum system action $\hat{a}_t$ at dialog turn t that minimizes the objective function $\mathcal{E}(a_t)$, which we define below. The problem is then to provide the expected cost function given observations based on Bayes decision theory. The observations consist of the following variables:

- $u_{1:t} = \{u_1, \cdots, u_t\}$: sequence of user intentions from the beginning of the dialog to the current turn, obtained from user utterances by the ASR and NLU components. We can also consider a list of the N best intentions at each time step, with confidence scores.
- $a_{1:t-1} = \{a_1, \cdots, a_{t-1}\}$: sequence of decided system actions from the beginning to the previous turn.
- a_t^{rule}: current system action that would be chosen by a *rule-based* system.

Figure 1 illustrates part of our existing rule-based dialog manager, where arcs are user intentions and nodes are system actions. For instance, the system action `SEARCH_WAYPOINT[{type=poi},{value=$POI$}]` in the upper right of the graph is a goal state, meaning that the system understands the user's goal (to set a particular waypoint) and executes the command, presenting the result. This command's slot information is given by `type=poi` and `value=$POI$`. The user intentions are defined using a similar representation, with some additional syntax (e.g., `SELECT_ITEM[{value=$ITEM_NAME$}]` for specifying a slot value). A set of these observation variables is denoted as a *history* $\mathcal{H}_{1:t} \triangleq \{a_{1:t-1}, a_t^{\text{rule}}, u_{1:t}\}$ as explained below.

These following variables are not observed, and must be inferred from the observations:

- g: goal action, which is constant across all turns t. Examples are given by the filled nodes in Figure 1.
- $a_{t:t+T}$: sequence of system actions from the current turn t to a future turn $t+T$ (where $T > 0$).

Based on the observed and unobserved variables, we provide the following empirical Bayes risk function $\mathcal{E}(a_t)$, which is the expected value of the cost function $C(g, a_{t:t+T})$, given the observations $\mathcal{H}_{1:t}$ and the hypothesized system action a_t.

$$\mathcal{E}(a_t) = \sum_{g, a_{t+1:t+T}} C(g, a_{t:t+T}) p(g, a_{t+1:t+T} | \mathcal{H}_{1:t}, a_t). \quad (1)$$

Here the cost function is marginalized over the posterior distribution $p(g, a_{t+1:t+T} | \mathcal{H}_{1:t})$. Thus, we obtain a general form of the empirical Bayes risk for our dialog manager. The following sections provide a more detailed expression for this empirical Bayes risk, along with some efficient approximations.

2.2. Empirical Bayes risk for proposed method

For the rest of the discussion, we assume a finite T (actually, $T = 1$), which means that our dialog manager only considers one step into the future in our dialog management. Although this is different from previous statistical dialog managers that are based on the reinforcement learning (where $T = \infty$), we believe this is an efficient approximation, since the number of turns in task-oriented dialogs is usually small. An additional advantage of this finite-future approximation is that we can compute the expected cost function on-the-fly during dialog management [11,19,20], unlike standard reinforcement learning approaches that must pre-compute a policy function in advance.

To estimate the cost, we first predict a_{t+1} using f_{rule}, a deterministic function defined by a rule-based dialog manager:

$$\hat{a}_{t+1} = f_{\text{rule}}(a_t, u_{t+1}) \ \text{ if } \ a_t \neq g. \quad (2)$$

This function represents a transition in the finite state automaton in Figure 1, where the next node a_{t+1} is determined given a current node a_t and a transition arc u_{t+1} from the graph. This approximation enables us to compute the expectation over user intention u_{t+1}, rather than over a_{t+1} as in the original cost function. Note that if a_t is a goal action, we do not need to consider the next system action, and hence f_{rule} is not used. In addition, we include a dependency on the current rule-based action a_t^{rule} in the cost function, to allow for regularization using the rule-based system. Therefore, by substituting Eq. (2) into the cost function, the expected cost function is rewritten as:

$$\mathcal{E}(a_t) \approx \sum_{g,u_{t+1}} C(g, a_t^{\text{rule}}, f_{\text{rule}}(a_t, u_{t+1}))p(g, u_{t+1}|\mathcal{H}_{1:t}, a_t). \tag{3}$$

Here the posterior distribution $p(g, u_{t+1}|\mathcal{H}_{1:t}, a_t)$, which is a distribution over the joint probability of g and u_{t+1}, should be factorized for practical use. By assuming g and u_{t+1} are approximately conditionally independent, we can factorize the posterior distribution as $p(g, u_{t+1}|\mathcal{H}_{1:t}, a_t) \approx p(g|\mathcal{H}_{1:t})p(u_{t+1}|g, a_t)$. Thus, the empirical Bayes risk $\mathcal{E}(a_t)$ is approximated as the following objective function of a_t:

$$\mathcal{E}(a_t) \approx \sum_{u_{t+1},g} C(g, a_t^{\text{rule}}, f_{\text{rule}}(a_t, u_{t+1}))p(u_{t+1}|g, a_t)p(g|\mathcal{H}_{1:t}) \tag{4}$$

This is composed of $p(g|\mathcal{H}_{1:t})$, which is a goal estimation model, $p(u_{t+1}|g, a_t)$, which can be regarded as a user simulation model, and the cost function. The following sections describe these components in detail.

2.3. Goal estimation model

This paper uses a discriminative, rather than generative, approach to represent the goal estimation model, since we can flexibly incorporate various information via feature engineering. We use multivariate logistic regression to compute $p(g|\mathcal{H}_{1:t})$:

$$p(g|\mathcal{H}_{1:t}) \propto \exp\left(\mathbf{w}_g^{\intercal}\boldsymbol{\phi}(\mathcal{H}_{1:t})\right), \tag{5}$$

where $^{\intercal}$ is the transpose operation, $\mathbf{w}_g$ is a weight vector for goal g, and $\boldsymbol{\phi}(\mathcal{H}_{1:t})$ is a feature vector extracted from the history. We can also include the N-best NLU outputs, along with their confidence scores, in the features. This makes the system's decisions more robust to ASR and NLU errors, which is a well-known advantage of using a statistical dialog manager.

2.4. Cost function

The cost function is designed to encourage dialogs that are efficient and that resemble those of a rule-based system. For this purpose, we split the cost function into three cases:

$$C(g, a_t^{\text{rule}}, f_{\text{rule}}(a_t, u_{t+1})) = \begin{cases} \theta_g, & a_t = g \\ \theta_r C'(g, f_{\text{rule}}(a_t, u_{t+1})), & a_t = a_t^{\text{rule}} \\ C'(g, f_{\text{rule}}(a_t, u_{t+1})), & \text{otherwise.} \end{cases} \tag{6}$$

In the first case, the system action is a goal action. Since the rule-based dialog manager $f_{\text{rule}}(a_t, u_{t+1})$ of Eq. (2) is not well-defined in this case, we define a special cost using control parameter θ_g. Setting a smaller (or larger negative) value for θ_g will encourage the manager to skip intermediate steps in order to reach the goal more efficiently. The second case assigns a regularization weight θ_r to the action that would be selected by a rule-based system, to encourage the manager to mimic the rule-based system. The third case's cost function, $C'(g, f_{\text{rule}}(a_t, u_{t+1}))$, is discussed in the next section.

2.5. Graph-based cost and user simulation

Using Eq. (2), the remaining cost function can be rewritten as $C'(g, f_{\text{rule}}(a_t, u_{t+1})) = C'(g, \hat{a}_{t+1})$. In this paper, this value is obtained from the number of arcs in a path from the node $\hat{a}_{t+1}$ to the estimated goal g in the state graph of a rule-based system. There are multiple paths from $\hat{a}_{t+1}$ to g, and the path length is averaged by assuming that each path has equal probability. The following cost is an example from Figure 1 in which there are two paths (0.5 probability each) from state $\hat{a}_{t+1}$ to the goal g, and each path's cost is its length (1 arc):

$$C'(g, \hat{a}_{t+1}) = 1 \times 0.5 + 1 \times 0.5 = 1 \quad \text{when:}$$
$$\begin{cases} \hat{a}_{t+1} : \text{SEARCH\_POINT[\{type=poi\},\{value=\$POI\$\}]} \\ g \quad : \text{SEARCH\_WAYPOINT[\{type=poi\},\{value=\$POI\$\}]} \end{cases}$$

If there is no path from $\hat{a}_{t+1}$ to g, then we compute the cost as the cost from the root node to g.

In summary, the cost is basically computed from the distance between two nodes in the directed unweighted graph. We can also consider variations in which we use other graph distance measures (e.g., longest path or random walk), or in which we use a weighted graph [16, 21].

We can also use the graph to compute the user intention probability $p(u_{t+1}|g, a_t)$ in Eq. (4) by assuming equal probability for all possible actions (outdegree of a_t) to reach goal g:

$$p(u_{t+1}|g, a_t) = \frac{1}{\text{outdegree of } a_t \text{ to reach } g}. \tag{7}$$

This part can be also use a weighted graph [16], if we have sufficient data to learn the user model, or we can also simulate some uncertainty from the alternative hypotheses in the ASR and NLU components using posterior lattices or N-best lists.

Based on the goal estimation model, cost function, and the user simulation model, we can compute the empirical Bayes risk for each action a_t.

2.6. Slot state

Finally, we briefly explain a slot state vector used in our dialog manager, which prevents a dialog manager from reaching a goal unless the required slot information has been provided by a user. In our example, the slot information is embedded in the user intention (e.g., `{value=$POI$}` means that the slot is filled by a specific Point-Of-Interest obtained from the NLU component). Therefore, we retain this slot information state in a dialog as a binary vector $\mathbf{b} = \{b_i \in \{0, 1\}\}$, where each element b_i corresponds to the ith slot (e.g., POI, ADDRESS, etc.) and equals 1 when the slot is filled but 0 otherwise. The current slot state is represented by this vector, with the slot contents obtained from the history of user intentions, i.e., $\mathbf{b}(u_{1:t})$. By checking the consistency between the slot vectors of a system action $\mathbf{b}(a_t)$ and user intentions in the history ($\mathbf{b}(u_{1:t})$), we can prevent the system from selecting an inappropriate action. A probabilistic treatment of this slot state (from N-best ASR and NLU outputs) can also be considered in our framework.

3. Experiments

We evaluated the proposed dialog manager in an automobile destination-setting task based on the efficiency of dialogs and how closely the system actions match those of an existing rule-based system. We used two sets of dialog log data (task1: 996 dialogs and 2031 turns; task2: 686 dialogs and 1550 turns) that were collected using our Japanese rule-based spoken dialog system, which achieved 75.5% and 89.9% task completion rates respectively during user evaluations. The log data contained estimated user intentions from the NLU component, the rule-based system actions, and user goals, as shown in Figure 1. The number of distinct user intentions, system actions, and goal actions appearing in the data sets are 219, 110, and 45, respectively.

	ASR result and system prompt (log)	Estimated intentions and rule-based actions (log)	System actions obtained by the proposed method (simulation)
u_1	(JP) ガソリンスタンド１番近いガソリンスタンドに寄って下さい (EN) Gas station, Please stop by the nearest gas station.	SEARCH_NEARBY[{base_point=current_position},{category=$GENRE$}], score=0.906196 SEARCH_NEARBY[{base_point=waypoint},{value=$NUMBER$},{category=$GENRE$}], score=0.082799	
a_1	(JP) 現在地近くのガソリンスタンドを検索しました。リストから選択してください。 (EN) Searched nearby gas stations. Please select one from the list.	SEARCH_NEARBY[{base_point=current_position},{category=$GENRE$}]	SEARCH_NEARBY[{base_point=current_position},{category=$GENRE$}]
u_2	(JP) ４番エネオスカマクラシて車に寄って下さい (EN) Number 4. Please stop by ENEOS in the Kamakura station.	SELECT_ITEM[{value=$NUMBER$}], score= 0.998939 →*SEARCH_POINT[{type=poi},{value=POI}]*	
a_2	(JP) ENEOS鎌倉ステーションを検索しました。経由地にしますか、目的地にしますか、登録地にしますか。 (EN) Searched the place of ENEOS in the Kamakura station. Do you set it as a viapoint, destination, or register it?	SEARCH_POINT[{type=poi},{value=POI}]	**SEARCH_WAYPOINT[{type=poi},{value=POI}]**
u_3	(JP) 経由地にして下さい (EN) Please set it as a viapoint.	MODIFY_MAIN_COMMAND[{type=set_waypoint}], score=0.999155 →*SEARCH_WAYPOINT[{type=poi},{value=POI}*	
a_3	(JP) ENEOS鎌倉ステーションを経由地に追加しました。 (EN) Added ENEOS in the Kamakura station as a viapoint.	SEARCH_WAYPOINT[{type=poi},{value=POI}]	

Figure 2: *An example of the log data and simulated dialogs with the Japanese ASR results and system prompts with English translation, estimated N-best intentions, and rule-based and proposed system actions. Detailed explanations are found in Section 3.2.*

Table 1: Goal estimation rates for various settings.

	Task 1 (%)	Task 2 (%)
1-best	74.8	72.8
N-best	76.2	74.8
+ confidence score	78.0	77.0
+ L1 regularizer	**78.1**	**77.9**

3.1. Goal estimation

Table 1 presents the accuracy of goal prediction for each turn of a dialog before the final turn. The goal accuracies in each task were evaluated using the models that were trained using the other task's data. The training and prediction were performed by using LIBLINEAR [22]. Since the NLU component also provides N-best intentions with confidence scores, we used these values as a feature, which improved goal estimation by 5.1% from the 1-best results in Task 2. The final configuration with $L1$ regularization was used in the dialog evaluations.

3.2. Dialog evaluation

The proposed dialog manager was evaluated using the log data obtained from the rule-based system. For each turn of a dialog, the proposed dialog manager selected the optimal action according to the empirical Bayes risk (Eq. (4)) given the history data (composed of user intentions and rule-based system actions[1]). The goal cost factor θ_g in Eq. (6) was fixed at -10.0 to encourage the system to reach the goal in fewer turns.

Table 2 compares the result of the proposed system with the rule-based system. The percentage of dialogs in which our system successfully completed a task in fewer turns than the rule-based system (*Early success*) was computed for Task 1 and 2. The result shows that the proposed approach reduced the number of turns in more than half of dialogs in both tasks, which confirms the efficiency of the proposed approach. The early success rates of the N-best goal estimation results improved upon the 1-best results by around 4%, which shows the effectiveness of the proposed statistical dialog manager at mitigating the effects of errors in the NLU and ASR components. Table 2 also shows the coincidence rate, which computes the rate of the same actions obtained from the rule-based and proposed managers. In the case of N-best with $\theta_r = 1.0$ (this means we only used the graph-based cost in Eq. (6) without the regularization), we found that only 50–60% of system actions selected by the proposed dialog manager were the same as those chosen by the rule-based manager. Although these newly selected

Table 2: The rate of successful dialog completion with fewer turns than the rule-based system (Early success), and the coincidence rate of system actions from the proposed and rule-based systems.

	Early success (%)		Coincidence (%)	
	Task 1	Task 2	Task 1	Task 2
1-best ($\theta_r = 1.0$)	56.1	55.9	61.5	46.6
N-best etc. ($\theta_r = 1.0$)	**60.4**	**60.7**	59.8	49.5
N-best etc. ($\theta_r = 0.5$)	59.5	**60.7**	**70.2**	**69.2**

system actions must be different from the rule-based system's in order to improve dialog efficiency, too many actions that differ from those of the rule-based system, which already achieved very high task completion rates, may not be desirable. The setting of the regularization cost θ_r aims to avoid this situation, and N-best results with $\theta_r = 0.5$ keep the early success rates similar, while improving the coincidence rates by 10–20%. This result shows the effectiveness of the proposed dialog manager's regularization using the rule-based actions.

Figure 2 provides a typical example of how the simulated dialog of the proposed dialog manager can reach the goal with fewer turns. The 2nd column lists the Japanese ASR results and system prompts in the rule-based system's log data, with English translations. The 3rd column lists the estimated N-best user intentions with their confidence scores in the user intention rows (u_1, u_2, and u_3), and the system actions obtained from the rule-based system in the system action rows (a_1, a_2, and a_3). The 4th column lists the system actions chosen by the proposed dialog manager. In this example, the rule-based system in the 3rd turn (a_3 row) confirmed whether the command is about setting a viapoint, destination, or registering a point. However, in the 1st user turn (u_1 row), the user already specified his/her command to set a waypoint (this information was included in the 2nd-best user intention), and our goal estimator correctly estimated the command from this information. Thus, the proposed manager provided the goal action without the confirmation, which reduced the number of turns from 3 to 2.

4. Summary

This paper proposes a statistical dialog manager that can improve the efficiency of dialogs from an existing rule-based system, while choosing similar actions to those of the rule-based system, by incorporating the rule-based system into the cost function. Preliminary evaluation using dialog log data from the rule-based system demonstrates the effectiveness of the proposed dialog manager. In future work, we will perform more realistic evaluation with user data from the proposed system. In addition, we will compare the proposed system with other statistical dialog managers by using public data (e.g., [23]).

[1]The "rule-based" system actions in the history should be replaced with the system actions obtained by the proposed dialog manager for realistic evaluation. However, since this would normally affect the subsequent user intentions, we would not be able to evaluate it using dialog log data. Although this paper evaluated our proposed approach using rule-based log data, evaluation with real users is important future work.

5. References

[1] Dan Jurafsky and James H Martin, *Speech & Language Processing*, Pearson Education India, 2000.

[2] Nicholas Roy, Joelle Pineau, and Sebastian Thrun, "Spoken dialogue management using probabilistic reasoning," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics (ACL)*, 2000, pp. 93–100.

[3] James Henderson, Oliver Lemon, and Kallirroi Georgila, "Hybrid reinforcement/supervised learning for dialogue policies from communicator data," in *Proceedings of the IJCAI workshop on knowledge and reasoning in practical dialogue systems*, 2005, pp. 68–75.

[4] Olivier Pietquin and Thierry Dutoit, "A probabilistic framework for dialog simulation and optimal strategy learning," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 2, pp. 589–599, 2006.

[5] David Griol, Lluís F Hurtado, Encarna Segarra, and Emilio Sanchis, "A statistical approach to spoken dialog systems design and evaluation," *Speech Communication*, vol. 50, no. 8, pp. 666–682, 2008.

[6] Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu, "The hidden information state model: a practical framework for POMDP-based spoken dialogue management," *Computer Speech & Language*, vol. 24, no. 2, pp. 150–174, 2010.

[7] Toyomi Meguro, Ryuichiro Higashinaka, Yasuhiro Minami, and Kohji Dohsaka, "Controlling listening-oriented dialogue using partially observable Markov decision processes," in *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, 2010, pp. 761–769.

[8] Andrew G Barto, *Reinforcement learning: An introduction*, MIT press, 1998.

[9] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[10] Jason D Williams and Steve Young, "Scaling up POMDPs for dialog management: The"summary POMDP"method," in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2005, pp. 177–182.

[11] Hao Tang, Shinji Watanabe, Tim K. Marks, and John Hershey, "Log-linear dialog manager," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, (accepted).

[12] Allen L Gorin, Giuseppe Riccardi, and Jeremy H Wright, "How may I help you?," *Speech communication*, vol. 23, no. 1, pp. 113–127, 1997.

[13] Victor Zue, Stephanie Seneff, James R Glass, Joseph Polifroni, Christine Pao, Timothy J Hazen, and Lee Hetherington, "JUPlTER: a telephone-based conversational interface for weather information," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 85–96, 2000.

[14] Roberto Pieraccini and Juan Huerta, "Where do we go from here? Research and commercial spoken dialog systems," in *Proceedings of the 6th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2005.

[15] Dan Bohus and Alexander I Rudnicky, "The RavenClaw dialog management framework: Architecture and systems," *Computer Speech & Language*, vol. 23, no. 3, pp. 332–361, 2009.

[16] Konrad Scheffler and Steve Young, "Corpus-based dialogue simulation for automatic strategy learning and evaluation," in *Proceedings of the NAACL Workshop on Adaptation in Dialogue Systems*, 2001, pp. 64–70.

[17] Koichiro Yoshino, Shinji Watanabe, Jonathan Le Roux, and John R Hershey, "Statistical dialogue management using intention dependency graph," in *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP)*, 2013.

[18] James O Berger, *Statistical decision theory and Bayesian analysis*, Springer, 1985.

[19] Teruhisa Misu and Tatsuya Kawahara, "Bayes risk-based dialogue management for document retrieval system with speech interface," *Speech Communication*, vol. 52, no. 1, pp. 61–71, 2010.

[20] Sungjin Lee, "Structured discriminative model for dialog state tracking," in *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2013.

[21] Chiori Hori, Kiyonori Ohtake, Teruhisa Misu, Hideki Kashioka, and Satoshi Nakamura, "Statistical dialog management applied to WFST-based dialog systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 4793–4796.

[22] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *The Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[23] Jason Williams, Antoine Raux, Deepak Ramachandran, and Alan Black, "The dialog state tracking challenge," in *Proceedings of the 14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2013.