This document is published in:

Molina, J. M. et al. (eds.) (2011) *User-Centric Technologies and Applications: Proceedings of the CONTEXTS 2011 Workshop*. (Advances in Intelligent and Soft Computing, 94). Springer, 29-36. DOI: http://dx.doi.org/10.1007/978-3-642-19908-0_4

Context-Aware Conversational Agents Using POMDPs and Agenda-Based Simulation

David Griol and Jose´ M. Molina
Group of Applied Artificial Intelligence (GIAA). Computer Science Department. Carlos III University of Madrid
e-mail: {david.griol,josemanuel.molina}@uc3m.es

Abstract. Context-aware systems in combination with mobile devices offer new opportunities in the areas of knowledge representation, natural language processing and intelligent information retrieval. Our vision is that natural spoken conversation with these devices can eventually become the preferred mode for managing their services by means of conversational agents. In this paper, we describe the application of POMDPs and agenda-based user simulation to learn optimal dialog policies for the dialog manager in a conversational agent. We have applied this approach to develop a statistical dialog manager for a conversational agent which acts as a voice logbook to collect home monitored data from patients suffering from diabetes.

1 Introduction

Ambient Intelligence (AmI) systems usually consist of a set of interconnected computing and sensing devices which surround the user pervasively in his environment and are invisible to him, providing a service that is dynamically adapted to the interaction context. In this framework, spoken interaction can be the only way to access information in some cases, like for example when the screen is too small to display information (e.g. hand-held devices) or when the eyes of the user are busy in other tasks (e.g. driving). It is also useful for remote control of devices and robots, specially in smart environments. Finally, one of the most demanding applications for fully natural and understandable dialogs, are embodied conversational agents and companions. This way, conversational agents have became a strong alternative to provide computers with intelligent and natural communicative capabilities.

Funded by projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485), and DPS2008-07029-C02-02.

A conversational agent is a software that accepts natural language as input and generates natural language as output, engaging in a conversation with the user. To successfully manage the interaction with the users, conversational agents usually carry out five main tasks: automatic speech recognition (ASR), natural language understanding (NLU), dialog management (DM), natural language generation (NLG) and text-to-speech synthesis (TTS).

Learning statistical approaches to model the different modules that compose a conversational agent has been of growing interest during the last decade [7, 2]. The motivations for automating dialog learning are focused on the time-consuming process that hand-crafted design involves and the ever-increasing problem of dialog complexity. The most extended methodology for machine-learning of dialog strategies consists of modeling human-computer interaction as an optimization problem using Partially Observable Markov Decision Process (POMDPs) and reinforcement methods [5]. The main drawback of this approach is due to the large state space of practical spoken dialog systems, whose representation is intractable if represented directly. An approach that scales the POMDP framework for implementing practical spoken dialog systems by the definition of two state spaces and the use of an agent is presented in [6].

In this paper, we apply this approach to develop a conversational agent which acts as a voice logbook to collect home monitored data from patients suffering from Type-2 diabetes [1], integrated in a context-aware multiagent architecture [3]. In this architecture, a Context Manager in the conversational agents deals with loading the context information provided by the User and Positioning Agents, and communicates it to the different modules of the Conversational Agent during the interaction. Context information is managed by means of a data structure called *user profile*. The conversational agent developed using the POMDP and agenda-based user simulation has been compared with a dialog management technique based on a rule-based model. The results of this comparison show how the POMDP methodology achieves satisfactory task completion rates and improves the selection process of goal directed actions.

2 The DI@L-Log System

DI@L-log is a spoken dialog system which acts as a voice logbook to collect home monitored data from patients suffering from Type-2 diabetes [1]. The data collected by the system are the patient's weight, blood pressure (systolic and diastolic values) and sugar levels. The system validates and analyses the data, providing some immediate feedback to the patients regarding their current progress as well as communicating the results to doctors at the hospital who are able to review the patient's progress graphically and deal with any alerts generated by the system concerning abnormal developments.

To apply our methodology to this task, the definition of the semantics was carried out considering the information that is required to monitor the patients and inform them about their condition. Taking into account the requirements of the task, an

initial strategy was designed for the DM. Figure 1 shows an example of a dialog acquired using this strategy.

S1: Welcome to the DI@L-log System. Please say or enter your four digit pin.
U1: 1356.
S2: Are you ready to give your measurements?
U2: Yes.
S3: Let's begin with your weight. How many stones?
U3: 12.
S4: I have understood 12 stones. Is it correct?
U4: Yes.
S5: What is your blood sugar?
U5: 11.
S6: I have understood 11. Is it correct?
U6: Yes.
S7: And finally, for your blood pressure you will be asked for the top number and then the lower number. What is the top number of your blood pressure?
U7: 160.
S8: I have understood 160. Is it correct?
U8: Yes.
S9: And what is the lower number?
U9: 86.
S10: I have understood 86. Is it correct?
U10: Yes.
S11: Your readings have been successfully updated. According to them, your weight is higher than last week, be very careful. Thanks for contacting DI@L-log. Goodbye.

Fig. 1 An example of a dialog for the DI@L-log task

As can be observed, three different phases are present in every dialog. Firstly, there is an identification phase in which the system asks the user about his login and password and then waits until the user says that he is ready to provide the control data (S1 and S2 system turns). Secondly, the system analyzes which data is required for the current user, taking into account that the weight and sugar values are mandatory and the blood control is only carried out for specific patients (S3 to S10 system turns). In this phase, the system requires the user to provide this data. Every item is confirmed after the user has provided its value. The user can only provide one item at a time. In the last phase, the system consults the information that the patient has provided during the current dialog and compares it with the data that is present in a database that contains the values that he provided in previous dialogs. By means of this comparison, the system is able to inform the user about his condition and provide him with instructions that take this into account (S11 system turn).

A corpus of 100 dialogs was acquired using this strategy. In order to learn statistical models, the dialogs of the corpus were labeled in terms of dialog acts. In the case of user turns, the dialog acts correspond to the classical frame representation of the meaning of the utterance. For the DI@L-log task, we defined three task-independent concepts (*Affirmation*, *Negation*, and *Not-Understood*) and four attributes (*Weight*, *Sugar*, *Systolic-Pressure*, and *Diastolic-Pressure*).

The labeling of the system turns is similar to the labeling defined for the user turns. A total of 12 task-dependent concepts was defined, corresponding to the set of concepts used by the system to acquire each of the user variables (*Weight*,

Sugar, *Systolic-Pressure*, and *Diastolic-Pressure*), concepts used to confirm the values provided by the user (*Confirmation-Weight*, *Confirmation-Sugar*, *Confirmation-Systolic*, and *Confirmation-Diastolic*), concepts used to inform the patient about his condition (*Inform*), and three task-independent concepts (*Not-Understood*, *Opening*, and *Closing*).

3 POMDPs and Agenda-Based User Simulation

Formally, a POMDP is defined as a tuple $\{S, A, T, R, O, Z, \lambda, b_0\}$ where:

- S is a set of the agent states.
- A is a set of actions that the agent may take.
- T defines the transition probability $P(s'|s,a)$.
- R defines the immediate reward obtained from taking a particular action in a particular state $r(s,a)$.
- O is a set of possible observations that the agent can receive.
- Z defines the probability of a particular observation given the state and machine action $P(o'|s',a)$.
- λ is a geometric discount factor $0 \leq \lambda \leq 1$.
- b_0 is an initial belief state $b_0(s)$.

The operation of a POMDP is as follows. In each moment, the agent is in an unobserved state s. The agent selects an action a_m, receives a reward r, and transits to a state (unobserved) s', where s' only depends on s and a_m. The agent receives an observation o' which depends on s and a_m. Although the observation allows the agent to have some evidences about the state s in which the agent is now, s is not exactly known, and $b(s)$ (*belief state*) is defined to indicate the probability of the agent being in the state s. In each moment, this probability is updated taking into account o' and a_m:

$$b'(s') = P(s' \left| o', a_m, b \right.) = k \cdot P(o' \left| s', a_m \right.) \sum_{s \in S} P(s'|a_m, s) b(s) \tag{1}$$

where $k = P(o'|a_m, b)$ is a normalization constant [4].

At each time t the agent receives a reward $r(b_t, a_{m,t})$ which depends on b_t and the selected action $a_{m,t}$. The reward accumulated during the dialog (*return*) can be calculated by means of:

$$R = \sum_{t=0}^{\infty} \lambda^t R(b_t, a_{m,t}) = \sum_{t=0}^{\infty} \lambda^t \sum_{s \in S} b_t(s) r(s, a_{m,t}) \tag{2}$$

Each action $a_{m,t}$ is determined by the policy $\pi(b_t)$ and the construction of the POMDP model implies to find the strategy π^* which maximizes *return*. The goal of POMDP policy optimization is to find the policy that maximizes the value function at every point b. Due to the vast space of possible belief states, however, the use of POMDPs for any practical system is far from straightforward. Exact algorithms for

solving POMDPs do exist, but have been shown to be intractable except for domains limited to a few states. Instead, the belief state and actions are mapped down to a summarized form where optimization becomes tractable. In this context, the original belief space and actions are called master space and master actions, while the summarized versions are called summary space and summary actions. The summarized space consists of the N-best states (s_u) from the original space (N is usually 1 or 2) and a simplified codification of the user's action a_u and the dialog history s_d. The main idea of this summarized space is to explore only the actions that has sense for the current situation in the dialog (e.g. do not begin the conversation with a confirmation, do not say *Welcome* except at the start, etc.).

The optimization of the policy in these two spaces is usually carried out by using techniques like the Point-based Value Iteration or Q-learning, in combination with a user simulator. Q-learning is a technique for online learning traditionally used in an MDP framework. It is an iterative Monte-Carlo style algorithm where a sequence of sample dialogs are used to estimate the Q functions for each state and action. This way, the summarized Q-learning algorithm discretizes summary space and uses Q-learning on the resulting MDP-like grid. Using this algorithm, at each point, the master belief space is mapped down to the summary level as described and the nearest summary point in the grid is found and the optimal summary action given by that point is chosen. This optimal action for each point p is given by

$$\bar{a}_p = \underset{\bar{a}}{\operatorname{argmax}} \bar{Q}(a, p)$$

After a set of dialogs has been completed, the estimates of the Q-functions are updated with the new dialog scores. At the end of the dialog, the discounted future reward is known for each stage where a choice was taken (i.e., the Q-function evaluated at this grid point). A good estimate of the true Q-value is obtained if sufficient dialogs are done. User simulation is then introduced to reduce the too time-consuming and expensive task to obtain these dialogs with real users. Simulation is usually done at a semantic dialog act level to avoid having to reproduce the variety of user utterances at the word or acoustic levels. At the semantic level, at any time t, the user is in a state s_u, takes action a_u, transitions into the intermediate state s'_u, receives machine action a_m, and transitions into the next state s''_u.

Agenda-Based state representations, like the one described in [6], factors the user state into an agenda A and a goal G. The goal G consists of constraints C which specify the detailed venue of the dialog, and requests R which specify the desired pieces of information.

$$s_u = (A;G) \qquad G = (C;R)$$

The user agenda A is a stack-like structure containing the pending user dialog acts that are needed to elicit the information specified in the goal. At the beginning of each dialog, a new goal G is randomly selected. Then, the goal constraints C are converted into user and system *inform acts* (a_u and a_m acts) and the requests R into *request acts*. A *bye* act is added at the bottom of the agenda to close the dialog once the goal has been fulfilled. The agenda is ordered according to priority, with A[N]

denoting the top and A[1] denoting the bottom item. This way, $a_u[i]$ denotes the *ith* item in the user act a_u:

$$a_u[i] := A[N-n+i] \quad \forall i \in [1..n];\ 1 \leq n \leq N$$

$$P(a_u|s_u) = P(a_u|A,G) = \delta(a_u, A[N-n+1..N])$$

By denoting A' to the agenda after selecting the act a_u, the first state transition depending on a_u and the second state transition based on a_m can be respectively expressed by means of the following equations [6]:

$$P(s'_u|a_u,s_u) = P(A',G'|A,G) = \delta(A', A[1..N'])\delta(G',G)$$

$$P(s''_u|a_m,s'_u) = P(a_m|A',G'')P(G''|a_m,G')$$

The first probability in the latter equation denotes the agenda update model. By assuming that every dialog act a_m triggers one push-operation from the agenda, this probability can be expressed as follows:

$$P(a_m|A',G'') = \prod_{M}^{i=1} P(A''[N'+i]|a_m[i],G'')\ \ \delta(A''[1..N'], A'[1..N'])$$

The second probability denotes the goal update model. Assuming that R'' is conditionally independent of C' given C'', it can be expressed as follows:

$$G''|a_m,G') = P(R''|a_m,R',C'')P(C''|a_m,R',C')$$

where the first probability can be approximated as follows:

$$P(R''|a_m,R',C'') = \prod_k P(R''[k]|a_m,R'[k],\mathscr{M}(a_m,C''))$$

4 Evaluation

The summary Q-learning algorithm and agenda-based user simulation described in the previous section were used to develop a POMDP-based conversational agent for the Di@L-log task. To do this, we took into account the benefits of using standards like VoiceXML and also include a specific module for the statistical dialog model to avoid the effort of manually defining the dialog strategy. This module selects the following system response using the dialog policy obtained by means of the POMDP. By means of the incorporation of this module, developers only have to define a set of VXML files, each one including a system prompt and the associated grammar to capture users answers for it. This way, the statistical dialog manager automatically decides the following file (i.e. system prompt) that has to be selected.

A total of 25 dialogs were recorded from interactions of six users employing the initial dialog strategy defined for the DI@L-log system and the POMDP-based systems presented in this paper. Rewards in this system were given based on the

task completion rate and the number of turns in the dialog. Using the definitions described in [6], the POMDP system was given 20 points for a successful dialog and 0 for an unsuccessful one, One point was subtracted for each dialog turn. We considered the following measures for the evaluation:

1. Dialog success rate (%success). This is the percentage of successfully completed tasks. In each scenario, the user has to obtain one or several items of information, and the dialog success depends on whether the system provides correct data (according to the aims of the scenario) or incorrect data to the user.
2. Average number of turns per dialog (nT).
3. Confirmation rate (%confirm). It was computed as the ratio between the number of explicit confirmations turns (nCT) and the number of turns in the dialog (nCT/nT).
4. Average number of corrected errors per dialog (nCE). This is the average of errors detected and corrected by the dialog manager. We have considered only those errors that modify the values of the attributes and that could cause the failure of the dialog.
5. Average number of uncorrected errors per dialog (nNCE). This is the average of errors not corrected by the dialog manager. Again, only errors that modify the values of the attributes are considered.
6. Error correction rate (%ECR). The percentage of corrected errors, computed as nCE/ (nCE + nNCE).

The results presented in Table 1 show that the initially defined rule-based conversational agent and the POMDP-based conversational agent could interact correctly with the users in most cases. The success rate in the POMDP system is reduced with regard to the initial rule-based system. This is due to the introduction in this system of confidence scores to indicate the cases for which a confirmation must be done. This means that it is possible for the system to assign a high level confidence to an incorrectly recognized value. For this reason, the error correction rate in this system is also slightly lower. The average number of required turns is reduced in the POMDP-based system from 10.4 to 7.0. This is again due to the initial rule-based strategy is based on confirming every item provided by the user in the following system turn. The POMDP-based system reduces the number of required dialog turns by reducing the number of confirmations, as it can be observed in the lower confirmation rate achieved for this system.

Table 1 Results of the evaluation of the rule-based and POMDP-based conversational agents

	%success	nT	%confirm	%ECR	nCE	nNCE
Rule-based Conversational Agent	97%	10.4	41%	92%	0.81	0.07
POMDP-based Conversational Agent	93%	7.0	28%	89%	0.86	0.11

5 Conclusions

Modeling human-computer interaction by means of POMDPs and reinforcement methods are the most extended methodology for machine-learning of dialog strategies in conversational agents. Due to the main drawback of this approach is the large state space of practical spoken dialog systems, whose representation is intractable if represented directly, we have applied the proposal described in [6] to deal with this and also introduce an agenda-based model user simulation technique to learn the dialog model. The application of this approach to develop a conversational agent which collects home monitored data from patients suffering from diabetes show how it allows the dialog manager to tackle new situations and generate new coherent answers for the situations already present in the initial corpus. Due to the new learning process, the dialog manager can now ask for the required information using different orders, confirm these information items taking into account the confidence scores, reduce the number of system turns for the different kinds of dialogs, automatically detect different valid paths to achieve each of the required objectives, etc. As a future work, we want to compare this approach with other statistical and corpus-based methodologies for dialog management.

References

1. Black, L., McTear, M.F., Black, N.D., Harper, R., Lemon, M.: Appraisal of a conversational artefact and its utility in remote patient monitoring. In: Proc. of the 18th IEEE Symposium CBMS 2005, Dublin, Ireland, pp. 506–508 (2005)
2. Griol, D., Hurtado, L., Segarra, E., Sanchis, E.: A Statistical Approach to Spoken Dialog Systems Design and Evaluation. Speech Communication 50(8-9), 666–682 (2008)
3. Griol, D., Sánchez-Pi, N., Carbó, J., Molina, J.: An Architecture to Provide Context-Aware Services by means of Conversational Agents. Advances in Intelligent and Soft Computing 79, 275–282 (2010)
4. Kaelbling, L., Littman, M., Cassandra, A.: Planning and acting in partially observable stochastic domains. Artificial Intelligence 101, 99–134 (1998)
5. Levin, E., Pieraccini, R., Eckert, W.: A stochastic model of human-machine interaction for learning dialog strategies. IEEE Transactions on Speech and Audio Processing 8(1), 11–23 (2000)
6. Thomson, B., Schatzmann, J., Weilhammer, K., Ye, H., Young, S.: Training a real-world pomdp-based dialogue system. In: Proc. of NAACL-HLT 2007, pp. 9–16 (2007)
7. Young, S.: The Statistical Approach to the Design of Spoken Dialogue Systems. Tech. rep., CUED/F-INFENG/TR.433, Cambridge University Engineering Department, Cambridge, UK (2002)