REINFORCEMENT LEARNING WITH SIMULATED USER FOR AUTOMATIC DIALOG STRATEGY OPTIMIZATION

Minh-Quang Nguyen[1], Philip H.P. Nguyen[2], Tho-Hau Nguyen[3], Jean-Guy Meunier[4], Douglas O'Shaughnessy[5]

[1,5]*Institut national de recherche scientifique, EMT*
800, De la Gauchetière Ouest, #6900, Montréal, H5A 1K6, Canada. E-mail:{ nguyenmq,dougo}@emt.inrs.ca

[2]*Justice Technology Services, Department of Justice, Government of South Australia,*
30, Wakefield St., Adelaide, SA 5000, Australia. E-mail: nguyen.philip@saugov.sa.gov.au

[3]*Université de Québec à Montréal, Dép. d'informatique*
201, Ave. du Président-Kennedy, PK-4150, Montréal, H2X 3Y7, Canada. E-mail: nguyen.tho-hau@uqam.ca

[4]*Université de Québec à Montréal, Dép. de philosophie*
455, Blv. René-Lévesque Est, W-5440, Montréal, H2L 4Y2, Canada. E-mail: meunier.jean-guy@uqam.ca

Abstract: In this paper, we propose a solution to the problem of formulating strategies for a spoken dialog system. Our approach is based on reinforcement learning with the help of a simulated user in order to identify an optimal dialog strategy. Our method considers the Markov decision process to be a framework for representation of speech dialog in which the states represent history and discourse context, the actions are dialog acts and the transition strategies are decisions on actions to take between states. We present our reinforcement learning architecture with a novel objective function that is based on dialog quality rather than its duration.

1. INTRODUCTION

Speech recognition and speech synthesis techniques have become increasingly efficient and robust, facilitating implementation of human-machine spoken dialog systems. In these applications, a machine *speaks* to a human by imitating human communication acts. However, human-machine dialogs still lack naturalness and flexibility. One of the most important issues in this domain is the management of conversational interactions between human and machine, which do not occur randomly, but rather follow precise rules of the communication acts. While some research is focused on the acoustic and semantic aspects of speech signals (*what to say*), other is directed towards dialog strategies (*how to say*) in order to control those interactions. A number of machine learning approaches for the design of such strategies have been proposed in literature [2][3][4][5][7][8][9]. One recent promising technique is reinforcement learning (RL) with the help of a simulated user, involving semi-supervised learning and trials-and-errors with a return value (negative or positive) for each decision. A machine could develop an optimal strategy from observation examples, provided that they are comprehensive. However, in the current state of the art, it is not possible to produce such a strategy by directly learning from corpora of dialog data (Schatzman et al., 2006) [8], mainly due to their small sizes, which are insufficient to permit exploration of all possible states and actions pertinent to a dialog. In addition, it is not certain that an optimal strategy is present in those corpora even if they are of reasonable sizes. Hence the idea of creating a simulated user to assist learning [4][7][8][9]. In our implementation, we model dialog acts on Markov properties (actions, states, and transitions) [4][10] and if these properties are satisfied, the resulting dialog strategy is called a Markov Decision Process (MDP) (Sutton and Barto, 1998) [10].

Currently, we have a demonstration version of a spoken dialog system for hotel reservation (Hotel-Demo) from Nuance[1] (2006). The system was designed according to traditional approaches, with *manual* (as opposed to automatic) optimization of dialog strategies. The problem with such a system is that it becomes cumbersome and expensive to maintain (any significant modifications is hard to make). Furthermore, it is difficult to identify an optimal strategy that could cope with all the different behaviors of users. In this paper, we propose a learning architecture that enables automation of the design of a model for dialog strategy optimization, in which the cost of maintenance and the timeframe for

[1] Nuance Communications Inc. - www.nuance.com

development of new applications are minimized. The main feature of our architecture resides in a novel objective function that achieves optimal dialog strategy based on quality of conversation [2], rather than its "quantity" (or duration), similar to what is proposed in [4]. This quality could be measured via the questions that the machine poses to the hotel reservation customer, which could be: implicit (e.g., W*hen do you want to reserve a room for two persons?*), explicit (e.g., *Did you say two persons?*), or repetitive (e.g., *Please repeat your reservation date?*).

Our paper is organized as follows: Section 2 describes the Markov Decision Process (MDP). Section 3 summarizes the RL technique, Section 4 details our proposed RL architecture with a simulated user, including the parameters for our objective function and the initialization of the reward variables, all necessary for satisfactory learning. And finally, Section 5 concludes our proposal and suggests new directions for research.

2. DIALOG AS A MARKOV DECISION PROCESS

Recent research suggests that the formalism of the Markov Decision Process (MDP) could be used in the representation of dialog acts and in the modeling of problems relating to dialog strategy optimization [4][5][8].

As per [6][10], a MDP is a 4-tuple: (S, A, P(.,.), R(.)) in which:

. $S = \{s_1, s_2, ..., s_n\}$ is the set of states, representing the whole dialog, i.e., the knowledge of the concerned domain. A state at time t is denoted s_t or s, and at time t+1, s_{t+1} or s'. In our hotel room reservation domain, dialog states could be: s_1 = (*date*: unknown, *nrr*: 0) and s_2 = (*date*:12-Feb-2007, *nrr*: 0) where *date* is the date of reservation and *nrr* is the number of rooms to be reserved.

. $A = \{a_1, a_2, ..., a_m\}$ is the set of actions, which are dialog acts. An action carried out at time t is denoted a_t or a, and at time t+1, a_{t+1} or a'. For example, action a_1 = (*For which date would you like to reserve?*) moves the dialog from state s_1 (*date*: unknown) to state s_2 (*date*: known).

. P: S×A→S is the transition function, which associates a state and an action, with another state (which is the outcome of the action). An important property of an MDP is that the probability $P(s_{t+1}, r_{t+1} \mid s_t, a_t)$ of transitioning to state s_{t+1} (and collecting the reward r_{t+1} at that state) depends solely on the current action a_t and the current state s_t.

. $R(s_t)$ is the reward function, representing the reward received in reaching state s. The goal of an optimal strategy is to maximize the sum of all rewards collected, discounted by a rate γ (between 0 and 1), which could be expressed by the following mathematical formula:

$$\sum_{t=0}^{\infty} \gamma^t R(s_t) \qquad \text{(eq.1)}$$

MDP permits visualization of a dialog strategy π as a path connecting different states reached through different actions. An optimal strategy π* is a strategy that maximizes the discounted cumulative sum of all rewards collected on that path. The Markov decision problem is to identify that optimal strategy after some learning, and RL algorithms help us solve that problem.

3. REINFORCEMENT LEARNING

RL is the best choice for machine learning when the environment is uncertain, unknown or complex. In the case of a human-machine spoken dialog system, the machine cannot fully *understand* all what is said by a human. This is due to a variety of limitations, such as degraded speech recognition (e.g., signals distorted by the environment), deficient semantic interpretation, etc. Sometimes, the machine must interact with the environment without being certain about the coherence and/or correctness of its choice of dialog acts. It must learn by trials and errors, by analyzing all the responses from the user and the outcomes of its actions. In this perspective, the reward function defined in an MDP permits the machine to progress in its learning despite an uncertain environment. Dialog acts are translated into a sequence of states and actions, with each action leading to a state where a reward is collected. The cumulative reward can be expressed by a generalized formula that extends (eq. 1).

$$R = \sum_{t=1}^{T} \gamma^t R(s_{t+1}, a_t, s_t) \qquad \text{(eq.2)}$$

Here the learning task consists of optimizing the interaction between human and machine, and the goal is to find a strategy that maximizes the value of R. That value could be recursively calculated from the state-value function V^π (s), and the state-action or Q-learning function Q^π (s,a) of the strategy π. The associated optimization functions are $V^*(s)$ and $Q^*(s,a)$, defined as:

$$V^*(s) = \max_\pi V^\pi(s) \qquad \text{(eq.3)}$$

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \qquad \text{(eq.4)}$$

A number of algorithms exist for the determination of these optimal values [9]. However, the simplest and most efficient algorithm for RL is Q-learning, which consists of maintaining the Q-value, i.e. the set of all Q(s,a) values for all pairs of state s and action a.

4. PROPOSED LEARNING ARCHITECTURE

There exist several learning approaches for dialog strategy optimization such as the non-supervised method from Pietquin (2004) [5] and the hybrid (reinforcement and supervised) method from Henderson et al. (2005) [3]. Our proposed approach is based on an architecture described in [7][8][9]. It consists of two steps:

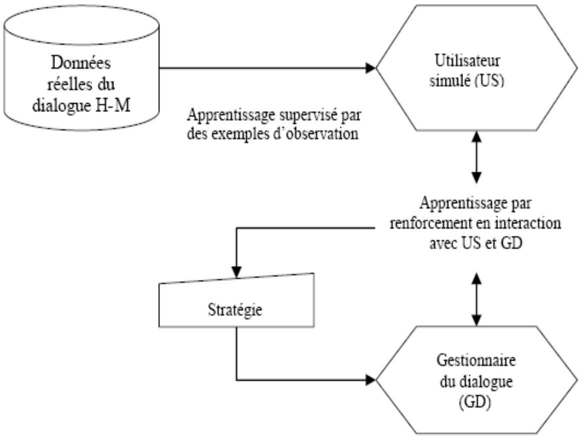
Données réelles du dialogue H-M
Apprentissage supervisé par des exemples d'observation
Utilisateur simulé (US)
Apprentissage par renforcement en interaction avec US et GD
Stratégie
Gestionnaire du dialogue (GD)

Fig. 1. RL Architecture (inspired from [8])

First a simulated user is created (according to an algorithm given in [1]) and trained by a number of dialog examples, selected among real dialogs taken from the current Hotel-Demo application. Then, a learning agent is built based on a Q-learning algorithm [6][10], This agent represents the dialog manager, which is the heart of the system. In direct interaction with the simulated user, the learning agent learns its strategy by examining the answers and remarks of the simulated user (represented by the values of the rewards).

The Q-value is calculated in each training session. After a number of sessions (in general, in the order of a million [7]), training stops and the objective function is called to evaluate the value of max Q^{π} (s,a), which represents the optimal strategy. This strategy will be implemented in a new version of the Hotel-Demo application.

4.1 Objective Function

Levin et al. (2000) [4] defines an objective function $C = \sum C_i$ where C represents the sum of all performance measures, such as the number of interactions, the numbers of errors and attributes non completed, etc. An example is a dialog system aiming to obtain the day, month and year in a minimum number of interactions.

Since our optimal dialog strategy is based on the quality of the conversation rather than its quantity, we select for the objective function, the following evaluation parameter:

$$. \; C_i = W_{imp}(N_{imp}) + W_{exp}\,(N_{exp}) + Wrep\,(N_{rep}) + W_{inc}\,(N_{inc}) \qquad \text{(eq.5)}$$

The variable C_i represents the total cost of the reward parameters. The variables W_{imp}, W_{exp}, W_{rep}, W_{inc} are the weights associated respectively with the variables N_{imp}, N_{exp}, N_{rep} and N_{inc} , and represent the relative importance between these costs. The variables N_{imp}, N_{exp}, N_{rep} et N_{inc} represent the numbers of implicit, explicit, and repetitive questions of the system, and the number of non-completed fields (e.g., at the end of the dialog, if the reservation date is not determined, the variable N_{inc} will have the value of -1.) The ratios $N_{imp}/N_{exp,}$ $N_{imp}/N_{rep,}$ N_{exp}/N_{rep} give us an idea about the performance of the system. For example, if the ratio N_{imp}/N_{exp} is equal to 1, this indicates that there were as many implicit questions as explicit ones produced by the system. This conveys some idea about the manner with which the dialog was conducted, but in itself this ratio does not constitute a decisive factor for the choice of an optimal strategy. We must compare that ratio with other ratios such as N_{imp}/N_{rep} or N_{exp}/N_{rep} which are rather typical factors to measure the quality of the dialog, because a high level of repetitive questions may also indicate a possible deficiency of the speech recognition module.

4.2 Reward Values

We draw our inspiration from [7] in our definition of the reward values applied in machine learning. To simplify, we limit our study to the case of only four fields in the hotel room reservation domain (i.e., reservation date, number of rooms, number of persons, and type of room (suite, single bed, or double bed)). Each field has three possible values (unknown, known, or confirmed), which gives a total of $3^4 = 81$ possible states. For each state, we associate three possible actions (implicit, explicit, or repetitive question). The number of combinations of the pair (state, action) thus becomes $81^3 = 531\,441$. This corresponds to the maximal number that system could explore in order to identify an optimal strategy. We could reduce this number by eliminating non-relevant actions in certain states, such as for example, at the start of the dialog, there cannot be a repetitive or implicit question. By purposely leaving this large number, we would like to exploit the learning capability of the agent in an uncertain environment and without *a priori* knowledge.

Each completed field	10
Each confirmed field	10
Each explicit question	5
Each implicit question	20
Each repetitive question	-5
Abandon	-10
Optimal value for completing 4 fields with implicit questions	(40 * 4) = 160

Tab. 1 Reward values given according to dialog strategy

To successfully train the system, we must specify the reward values (Tab. 1), which are utilized by the objective function to determine an optimal strategy. The main idea here is to define, from the start, a global value, which arbitrarily represents an optimal strategy. That value (+160) is computed by taking into account the best performance of the system, i.e., without recognition errors and assuming that the dialog progresses with only implicit questions and without incident. From a cognitive viewpoint, we believe that if a conversation is well conducted between the interlocutors (such as in the case of human-to-human), implicit questions are produced more frequently than explicit or repetitive ones (in general, having to repeat a question means that the performance of the speech recognition system degrades). The global value is then readjusted as training progresses. We give a positive value (+10, +20) when an action produces a good performance (i.e., an implicit or explicit question) and a negative value (-5, -10) when the performance is poor (i.e., a repetitive question). Finally, any abandon from any interlocutor during the dialog is considered a failure (this is rare for both the simulated user and the dialog system).

5. CONCLUSION

Our study on the design of a machine learning model is based on recent research in dialog strategy learning. The results show that learning with the help of a simulated user, implemented with MDP, RL and Q-learning techniques, could provide a reliable solution for dialog applications of the future. This type of learning is particularly suitable in complex, uncertain, or unknown contexts where the environment does not permit the determination beforehand of all possible states and actions, such as in spoken dialog applications. Our next step is to implement our proposed machine learning method in a new version of the Hotel-Demo system, with the help of software tools from Nuance, such as V-Builder V.4.0 and Open Speech Dialog.

REFERENCES

[1] Cuayahuitl, H., Renals, S., Lemon, O., Shimodaira, H., Human-Computer Dialog Simulation Using Hidden Markov Models. in *Proc. of IEEE ASRU*, Cancun, Mexico, 2005.

[2] English, M., Heeman, P., Learning Mixed Initiative Dialog Strategies By Using Reinforcement Learning On Both Conversants. In *Proc. Of HLT/EMNLP*, pp. 1011-1018, Vancouver, Canada, 2005.

[3] Henderson, J., Lemon, O., Georgila, K., Hybrid reinforcement/supervised learning for dialog policies from communicator data. In *Proc. of IJCAI on KRPDS*, Edinburgh, Scotland, 2005.

[4] Levin, E., Pieraccini, R., Eckert.,W., A Stochastic Model of Human Machine Interaction for Learning Dialog Strategies. In *Proc. of the IEEE ICASSP*, Istanbul, Turkey, pp. 1883-1886, 2000.

[5] Pietquin., O., *A Framework for Unsupervised Learning of Dialog Strategies.* Presses Universitaires de Louvain, *SIMILAR Collection*, ISBN 2-930344-63-6, 2004.

[6] Puterman, M. L., Markov Decision Processes, Wiley, 1994.

[7] Schatzmann, J., Stuttle, M., Weilhammer, K., Young, S., Effects of the user model on simulation-based learning of dialog strategies. *Proc. of ASRU*, San Juan, Puerto Rico, 2005.

[8] Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S., A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialog Management Strategies. *Knowledge Engineering Review*, 2006.

[9] Scheffler, K., Young, S., *Simulation of Human-Machine Dialogs,* Cambridge, U.K.: Engineering Dept., Cambridge University, Tech. Rep. CUED/F-INFENG/TR 355, 1999.

[10] Sutton, R., Barto, A., *Reinforcement Learning: An Introduction.* Cambridge, MA: MIT Press, 1998.