Co-adaptation in Spoken Dialogue Systems

Senthilkumar Chandramohan [1,3], Matthieu Geist [1],
Fabrice Lefèvre [3], Olivier Pietquin [1,2]

Abstract Spoken Dialogue Systems are man-machine interfaces which use speech as the medium of interaction. In recent years, dialogue optimization using reinforcement learning has evolved to be a state of the art technique. The primary focus of research in the dialogue domain is to learn some optimal policy with regard to the task description (reward function) and the user simulation being employed. However, in case of human-human interaction, the parties involved in the dialogue conversation mutually evolve over the period of interaction. This very ability of humans to co-adapt attributes largely towards increasing the naturalness of the dialogue. This paper outlines a novel framework for co-adaptation in spoken dialogue systems, where the dialogue manager and user simulation evolve over a period of time; they incrementally and mutually optimize their respective behaviors.

1 Introduction

Spoken Dialogue Systems (SDS) are man-machine interfaces which use spoken language (most often speech but can also be multi-modal interaction) as the medium of interaction. The dialogue management module is responsible for navigating the system to accomplish a specific task. Proper functioning of the dialogue management module can be attributed to the so called dialogue policy. Given the state of dialogue progress (context) and the most recent user response the dialogue policy determines the next action to be performed by the dialogue system. Manually choosing a dialogue policy can perhaps be an option if the problem studied is simple enough. However, for many real world dialogue problems hand-crafting a dialogue policy is often a complex task. Considering the stochastic behavior of users and various

[1] Supelec, MaLIS - IMS Research Group, Metz, France
[2] UMI 2958 (CNRS - GeorgiaTech), France
[3] Université d'Avignon et des Pays de Vaucluse, LIA-CERI, France
[1] firstname.lastname@supelec.fr [3] firstname.lastname@univ-avignon.fr

uncertainties involved (for example: speech recognition errors and channel noise), the complexity of dialogue management increases exponentially with increase in the size of the dialogue problem. To address this issue, dialogue management problems are casted as a Markov Decision Process (MDP) [13, 21, 3] or Partially Observable Markov Decision Process (POMDP) [23, 2], following which dialogue policy optimization can be performed using Reinforcement Learning (RL) [22].

Generally speaking, in case of human-human interaction, the parties involved in the conversation tend to co-adapt and thus mutually evolve over a period of time. Most often, when the conversation begins, the parties assess each others ability to understand and then continue to interact based on their initial assessment. Let us term this aspect of human-human communication as *dialogue-initiation*. However, once the dialogue progresses humans tend to evolve mutually (based on the history of conversation). Using an acronym such as MDP (during subsequent references) after defining the Markov Decision Process in oral or written communication is an example for such evolution. Let us term the later stage of human-human communication as *dialogue-evolution*. It may be useful to note that the dialogue-evolution occurs at several levels (such as: the amount of information exchanged, terminologies used during the conversation, *etc.*).

In case of man-machine interaction, dialogue-initiation has been carried out by policy optimization. Most existing approaches for dialogue management [7] focuses on retrieving some optimal (with respect to the reward function) and user adaptive (with respect to user simulation) dialogue policy [6]. This can be perceived as the dialogue-initiation stage of human-human interaction. This aspect of man-machine interaction has been studied in detail and is now state of the art. However, from the authors perspective, much less attention has been paid to dialogue-evolution in man-machine interaction. One of the most relevant works done towards dialogue-evolution is to perform on-line policy optimization (for instance, [5]). There are two primary drawbacks with regard to on-line policy optimization: (i) when the dialogue manager tries to evolve or optimize its behavior, the human user also tends to adapt to the dialogue manager (instead of speaking normally, the user tries to provide only information asked by the system). These contradicting efforts when applied simultaneously may often result in sub-optimal policies and thus blocking the possibility for dialogue-evolution, (ii) even if dialogue-evolution occurs it may bias the dialogue management to act over confidently [8, 5] and thereby resulting in inferior policies. Changes made directly to the policy used for dialogue management cannot always be guaranteed to improve performance. In the worst case scenario this may induce very bad user experience. Also in case of on-line optimization the speed of adaptation is relatively slow considering the fact that users (and hence behaviors) encountered are random in nature.

This paper presents a novel framework for co-adaptation in spoken dialogue systems. The primary focus of this work is to introduce co-adaptation in man-machine interaction and thereby facilitate dialogue-evolution. The dialogue manager and the user simulation (both casted as an MDP and optimized using RL) are made to interact with each other and subsequently co-adapt. Optimizing the dialogue manager and user simulation alternatively over a period of time, results in generalization of

dialogue policy and user behavior (as a result of back propagation of rewards). Thus co-adaptation provides an opportunity to learn policies which can cope with situations unobserved in the dialogue corpus. The layout of the paper is as follows: Section 2 formally defines the MDP and presents an overview on dialogue optimization. Section 3 outlines the process of co-adaptation and explains how it can be introduced in spoken dialogue systems. Section 4 describes the experimental set-up and analyze the results. Eventually Section 5 concludes and outlines the future directions of work with regard to co-adaptation.

2 Spoken dialogue optimization

Given a specific task to accomplish (such as: town-information) the dialogue management engine has to perform a sequence of decisions. Thus in itself the task of dialogue management can be perceived as a sequential decision making problem. Considering this fact, in recent years dialogue management problems are often casted as an MDP and policy optimization is carried out using RL.

2.1 Markov Decision Process

Statistical frameworks such as MDPs provide a well defined mathematical paradigm for modeling sequential decision making problems such as dialogue management. Formally, an MDP [3] is defined as a tuple $\{S, A, P, R, \gamma\}$ where S is the state space, A is the action space, $P : S \times A \to \mathscr{P}(S)$ is a set of Markovian transition probabilities, $R : S \to \mathbb{R}$ the reward or the utility function and γ is the discount factor for weighting long-term rewards. A learning agent has to perform a sequence of decisions and move from one state to another to accomplish the task (succinctly defined by the reward function). At any given time step, the agent is in a state $s_i \in S$ and transits to s_{i+1} according to $p(.|s_i, a_i)$ upon (choosing and) performing an action $a_i \in A$ according to a policy $\pi : S \to A$. After each transition the agent receives a reward $r_i = R(s_i)$. The quality of the policy π followed by the agent can be quantified by the state-action value function or Q-function ($Q^\pi : S \times A \to \mathbb{R}$) defined as:

$$Q^\pi(s,a) = E[\sum_{i=0}^{\infty} \gamma^i r_i | s_0 = s, a_0 = a] \tag{1}$$

The optimal policy π^* is the one for which the Q-function is maximum for each state-action pair: $\pi^* \in \arg\max_\pi Q^\pi(s,a)$. The optimal Q-function $Q^*(s,a)$ leads to an optimal policy: $\pi^*(s) = \arg\max_a Q^*(s,a)$. There exist several RL algorithms to compute the optimal policy π^* [22] and the associated $Q^*(s,a)$.

2.2 RL based dialogue optimization

Once the task of dialogue management is modeled as an MDP [13, 21], RL can be used to retrieve the optimal dialogue policy. Since RL based dialogue optimization is data intensive, user simulations are introduced to cope with the data requirement and also to evaluate the resulting dialogue policies. Dialogue optimization using RL and user simulation is shown in Figure 1. However, there exists a set of sample efficient algorithms for dialogue policy optimization [17]. Despite all these methods, user simulations continue to play a key role in evaluating the quality of the dialogue policy. An overview of dialogue optimization using number of machine-learning-techniques can be found in [12].

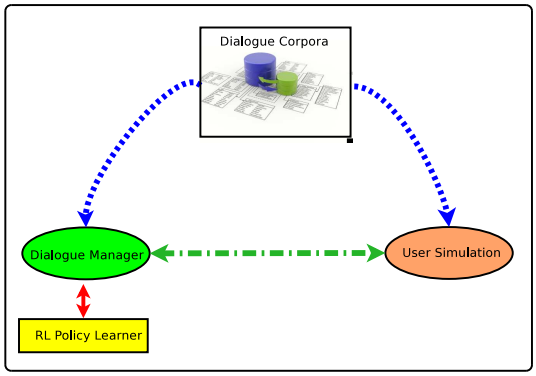

Fig. 1 *Dialogue optimization using RL & user simulation*

User simulation in dialogue systems [6] aims at generating synthetic dialogue corpus and are often trained using the available dialogue corpus. Most existing methods for building user simulations [9, 18, 16] focus on generating a synthetic dialogue corpus which has the same statistical consistency as observed in the dialogue corpus. Even-though there exists goal directed [15] and agenda based [19] user simulations, determining the goal or agenda of the user in itself is a complex task. It is important to note that the performance of the user simulation has a direct impact on the optimized dialogue policy as shown in [20]. One possible solution to this problem is to employ Inverse Reinforcement Learning (IRL) [14] to retrieve the utility function of the agent and then use it to perform RL based optimization. Recently the task of user simulation was casted as an MDP and IRL [14, 1] was used for imitating the user behavior [4]. This provides a unique opportunity for the user simulation to generalize (using the reward function learned from the data) and thus evolve over a period of time. Such an evolution would be very similar to that of the dialogue-evolution happening in the human-human interaction. Let us term the task of learning the user behavior as user (behavior) optimization.

3 Co-adaptation in dialogue systems

Naturalness of human-human dialogues can be attributed to several unique abilities of humans (such as: mutual evolution during the period of interaction). However, research on man-machine interaction (more specifically on dialogue optimization), primarily focused on retrieving some (initial) optimal policy. The reward function used for dialogue optimization is often hand-crafted to retrieve a decent policy which can accomplish the dialogue task in a robust and user adaptive manner. Thus dialogue managers built using such methods often lag-behind in terms of naturalness of the dialogue and thus makes it unpleasant for the end users. Co-adaptation in dialogue systems can provide an opportunity for the dialogue manager and user simulation to evolve mutually. To begin with the task of dialogue management (mdp-sds) and user simulation (mdp-user) are casted as an MDP. In order to perform policy optimization, reward functions for mdp-sds and mdp-user have to be defined in advance. As a pre-processing step for co-adaptation, these reward functions can be learned from the dialogue corpus using IRL. Unlike dialogue optimization shown in Figure 1, co-adaptation is an iterative procedure where policy optimization of mdp-user and mdp-sds is performed repeatedly. As shown in Figure 2 in Step 1: policy optimization for mdp-user is performed using a hand-crafted dialogue manager and the available dialogue corpus. Using the policy learned in Step 1; dialogue optimization for mdp-sds is performed in the next step. Following which policy optimization for mdp-user is performed using the optimal dialogue policy retrieved in the previous step. Step N and Step N+1 are repeated iteratively until convergence (in other words until the resulting policies cease to evolve).

Fig. 2 *Co-adaptation framework for dialogue evolution*

Even-though the resulting policies are deterministic, some amount of stochasticity can be introduced using Gibbs sampling based on state-action values *i.e.* Q(s,a). Let the probability of choosing a dialogue act or user act $a_i \in A$ be λ_i such that $\sum_{i=1}^{n} \lambda_i = 1$. The values of $\lambda_{1...n}$ can be heuristically determined using a Gibbs distribution: $\lambda_i = e(Q(s,a_i)/\tau)/\sum_{j=1}^{n} e(Q(s,a_j)/\tau)$. RL based optimization is subjective to that of the reward function and the environment the agent is interacting with. In

case of co-adaptation even-though the reward function remains the same, employing Gibbs sampling scheme results in changes in the dynamics of the environment. This very change in the dynamics of the dialogue manager or the user simulation will yield different user or dialogue policies respectively. It may be useful to note that during the process of co-adaptation the rewards obtained by the agents are back propagated and thus results in some degree of generalization of dialogue manager as well as user simulation modules. The immediate effect of this would be an opportunity for the dialogue manager and user simulation to cope with unseen situations (which are not observed in the dialogue corpus). This in turn will help the dialogue manager to retrieve the real optimal policy from its own capacity (policy which originally may not be present in the corpus but at the same time can cope with noise and changes in user behavior).

4 Experiment

This section outlines a simple experiment to exhibit the outcome of co-adaptation in spoken dialogue systems. Our primary motivation is to show how co-adaptation can be performed in case of man-machine interfaces and analyze the possibility of dialogue-evolution. Generally speaking, more appropriate experiment would have been a two step process: (i) perform IRL on the available dialogue corpus to retrieve the reward functions and (ii) use the reward function obtained from the data to perform co-adaptation. However, for the sake of simplicity a much smaller dialogue problem and RL based optimization is performed in the following experiments.

4.1 Town-Information dialogue system (2 Slots)

The dialogue problem studied in this paper is a 2 slot sub-problem from the town-information domain [11]. The dialogue manager has to seek and obtain user preferences for price-range and location of restaurants in the city. The state of the mdp-sds (dialogue manager casted as an MDP) involves 3 dimensions: (i) 0-2 (corresponding to the price-range; 0 - slot is not filled, 1 - slot is filled but yet to be confirmed, 2 - slot is filled and confirmed) (ii) 0-2 (corresponding to the location) and (iii) 0-1 (indicates whether the user has performed negation: indirect measure of channel noise). The list of dialogue acts includes: ask-slot1, ask-slot2, ask-all-slots, confirm1, confirm2, confirm-both, close-dialogue and two implicit confirmation acts. The state of the mdp-user (user simulation casted as an MDP) involves 4 dimensions: (i) 0-9 (represents the action performed by the dialogue manager), (ii) 0-2 (corresponding to the price-range), (iii) 0-2 (corresponding to the location) and (iii) 0-1 (indicates whether speech recognition errors have occurred). The list of user acts includes: provide-slot1, provide-slot2, provide-all-slots, confirm1, confirm2, confirm-all-slots, negate, remain-silent and hangUp. The same reward function was

used for both mdp-sds and mdp-user (positive reward of 20 for each correctly filled slot and bonus reward of 60 for successful task completion). The discount factor of both the MDPs are set to 0.95.

4.2 Experimental set-up and results

To begin with, hand-crafted (HC) policies for both mdp-sds and mdp-user are defined (these policies attempt to ask/provide and confirm/ascertain one slot after the other) so as to generate data from which a user simulation will be learnt (simulating the acquisition of a corpus). During the experiments dialogue manager and user policy optimization are carried out using Least Squares Policy Iteration (LSPI) [10] considering its sample efficiency and generalization abilities. Co-adaptation is performed as explained in Figure 2. First user optimization is performed using the mdp-sds with HC policy. The resulting user policy was then used to perform dialogue optimization. During the initial stages of the experiment, the channel noise is set to zero. At the end of each step a dialogue/user policy is generated. The following are a set of dialogue episodes generated from these retrieved policies:

```
==========================================
Step 1: HC-DialogueManager vs Train-RL-User
==========================================
  UserState: AskSlot_1 0 0 0
    UserAct: provide_slot_1
  UserState: ExpConfirm_1 1 0 0
    UserAct: confirm_slot_1
  UserState: AskSlot_2 2 0 0
    UserAct: provide_slot_2
  UserState: ExpConfirm_2 2 1 0
    UserAct: confirm_slot_2
  UserState: CloseDia 2 2 0
    UserAct: hangUp
==========================================
Step 2: Train-RL-DialogueManager vs RL-User
==========================================
  DiaState: 0 0 0   DiaAct: AskSlot_1
    UserResponse: provide_slot_1
  DiaState: 1 0 0   DiaAct: ExpConfirm_1
    UserResponse: confirm_slot_1
  DiaState: 2 0 0   DiaAct: AskSlot_2
    UserResponse: provide_slot_2
  DiaState: 2 1 0   DiaAct: ExpConfirm_2
    UserResponse: confirm_slot_2
  DiaState: 2 2 0   DiaAct: CloseDia
    UserResponse: hangUp
```

```
==============================================
Step 3: RL-DialogueManager vs Train-RL-User
==============================================
  UserState: AskSlot_1 0 0 0
    UserAct: provide_slot_1
  UserState: ExpConfirm_1 1 0 0
    UserAct: confirm_slot_1
  UserState: AskSlot_2 2 0 0
    UserAct: provide_slot_2
  UserState: ExpConfirm_2 2 1 0
    UserAct: confirm_slot_2
  UserState: CloseDia 2 2 0
    UserAct: hangUp %
==============================================
Step 4: Train-RL-DialogueManager vs RL-User
==============================================
  DiaState: 0 0 0   DiaAct: AskAllSlots
    UserResponse: provide_all_slots
  DiaState: 1 1 0   DiaAct: ExpConfirmAll
    UserResponse: confirm_all_slots
  DiaState: 2 2 0   DiaAct: CloseDia
    UserResponse: hangUp
==============================================
```

Dialogue episodes generated using policies learned from Step 3 and Step 4, clearly indicates that co-adaptation of dialogue management engine and user simulations has indeed happened. A more interesting aspect of this result is the fact that such optimal policies can be learned even-though they were not observed in the dialogue corpus (recall that the hand-crafted policies used in Step 1 only used simple dialogue acts and user acts). This result can be attributed to the generalization ability of dialogue manager and user simulation when casted as interacting MDPs. It may be useful to note that Gibbs sampling is not introduced in Step 2, since the focus was to learn a basic dialogue policy (similar to the hand-crafted dialogue manager used in Step 1). Even-though Gibbs sampling of dialogue policy (from Step 2) was employed in Step 3, the dialogue episode may look similar to that of Step 1. In this case the evolution of the user simulation is invisible because the episode presented here is pure greedy interaction between the dialogue policy from Step 2 and user policy from Step 3. However, this invisible evolution of user simulation evolution of the dialogue policy in Step 4. Changes in transitions caused due to co-adaptation of the dialogue management engine is shown in Figure 3.

As a next step artificial noise (error model) is introduced. Ideally speaking if there is some amount of channel noise, performing complex user action (where more information is exchanged) will exponentially increase the possibility for a speech recognition error. Having this in mind, 40% error is introduced when the user simulation performs a complex user action (*i.e.* provide-all-slots). The binary field in the user-state is set to 1 if both the user and dialogue manager performs

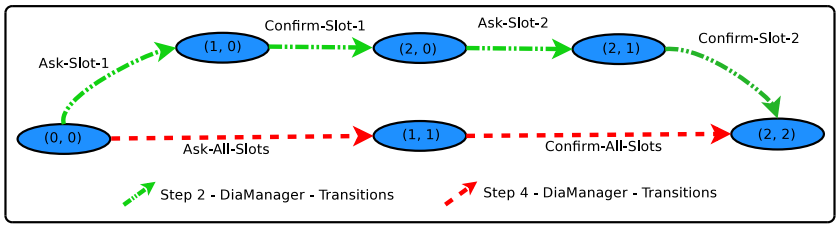

Fig. 3 *Changes in dialogue transitions due to co-adaptation*

(which provides opportunity to interact and evolve mutually). One interesting direction of future work would be to investigate the effectiveness of such generalizations and quantify the resulting dialogue and user policies; (ii) one other direction of future work is to explore the possibility of performing IRL based co-adaptation as discussed in Section 4.

6 Acknowledgements

This research was partly funded by the EU INTERREG IVa project ALLEGRO and by the Rgion Lorraine (France).

References

[1] P. Abbeel and A. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proc. of ICML, Banff, Alberta (Canada)*, 2004.

[2] Karl Johan Astrom. Optimal control of Markov decision processes with incomplete state estimation. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.

[3] Richard Bellman. A markovian decision process. *Journal of Mathematics and Mechanics, vol. 6, pp. 679–684*, 1957.

[4] S. Chandramohan, M. Geist, F. Lefèvre, and O. Pietquin. User Simulation in Dialogue Systems using Inverse Reinforcement Learning. In *Proc. of Interspeech 2011*, Florence (Italy), 2011.

[5] Lucie Daubigney, Milica Gasic, Senthilkumar Chandramohan, Matthieu Geist, Olivier Pietquin, and Steve Young. Uncertainty management for on-line optimisation of a POMDP-based large-scale spoken dialogue system. In *Proc. of Interspeech 2011*, pages 1301–1304, Florence (Italy), August 2011.

[6] W. Eckert, E. Levin, and R. Pieraccini. User Modeling for Spoken Dialogue System Evaluation. In *Proc. of ASRU*, pages 80–87, 1997.

[7] Matthew Frampton and Oliver Lemon. Recent research advances in reinforcement learning in spoken dialogue systems. *Knowledge Engineering Review*, 2009.

[8] M. Gasic, F. Jurcicek, B. Thomson, K. Yu, and S. Young. On-line policy optimisation of spoken dialogue systems via live interaction with human subjects". In *Proc. of ASRU 2011*, Hawaii (USA), 2011.

[9] K. Georgila, J. Henderson, and O. Lemon. Learning User Simulations for Information State Update Dialogue Systems. In *Proc. of Eurospeech, Lisbon (Portugal)*, 2005.

[10] Michail G. Lagoudakis and Ronald Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003.

[11] O. Lemon, K. Georgila, J. Henderson, and M. Stuttle. An ISU dialogue system exhibiting reinforcement learning of dialogue policies: generic slot-filling in the TALK in-car system. In *Proc. of EACL'06*, Morristown, NJ, USA, 2006.

[12] O. Lemon and O. Pietquin. Machine learning for spoken dialogue systems. In *Proc. of InterSpeech'07*, Belgium, 2007.

[13] E. Levin and R. Pieraccini. Using markov decision process for learning dialogue strategies. In *Proc. ICASSP'98, Seattle (USA)*, 1998.

[14] Andrew Y Ng and Stuart Russell. Algorithms for inverse reinforcement learning. In *Proc. of ICML, Stanford, CA, (USA)*, 2000.

[15] O. Pietquin. Consistent goal-directed user model for realistic man-machine task-oriented spoken dialogue simulation. In *Proc. of ICME'06*, pages 425–428, Toronto (Canada), July 2006.

[16] O. Pietquin and T. Dutoit. A probabilistic framework for dialog simulation and optimal strategy learning. *IEEE Transactions on Audio, Speech & Language Processing, 14(2): 589-599*, 2006.

[17] O. Pietquin, M. Geist, S. Chandramohan, and H. Frezza-Buet. Sample-Efficient Batch Reinforcement Learning for Dialogue Management Optimization. *ACM Transactions on Speech and Language Processing*, 7(3):7:1–7:21, May 2011.

[18] O. Pietquin, S. Rossignol, and M. Ianotto. Training Bayesian networks for realistic man-machine spoken dialogue simulation. In *Proc. of IWSDS 2009*, Irsee (Germany), December 2009.

[19] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc. of HLT NAACL, Rochester, NY (USA)*, 2007.

[20] Jost Schatzmann, Matthew N. Stuttle, Karl Weilhammer, and Steve Young. Effects of the user model on simulation-based learning of dialogue strategies. In *Proc. of ASRU, Puerto Rico*, 2005.

[21] Satinder Singh, Michael Kearns, Diane Litman, and Marilyn Walker. Reinforcement learning for spoken dialogue systems. In *Proc. of NIPS, Denver, USA*. Springer, 1999.

[22] R. Sutton and A. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 3rd edition, March 1998.

[23] Jason D. Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language, vol. 21(2), pp. 393–422.*, 2007.