INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

A Stochastic Model For Computer-Aided Human-Human Dialogue

Merwan Barlier[1,2], *Romain Laroche*[1], *Olivier Pietquin*[2,3]

[1]NaDia Team, Orange Labs
[2]Univ. Lille, CNRS, Centrale Lille, Inria, UMR 9189 CRIStAL - Lille, France
[3] Institut Universitaire de France

merwan.barlier@orange.com, romain.laroche@orange.com, olivier.pietquin@univ-lille1.fr

Abstract

In this paper we introduce a novel model for computer-aided human-human dialogue. In this context, the computer aims at improving the outcome of a human-human task-oriented dialogue by intervening during the course of the interaction. While dialogue state and topic tracking in human-human dialogue have already been studied, few work has been devoted to the sequential part of the problem, where the impact of the system's actions on the future of the conversation is taken into account. This paper addresses this issue by first modelling human-human dialogue as a Markov Reward Process. The task of purposely taking part into the conversation is then optimised within the Linearly Solvable Markov Decision Process framework. Utterances of the Conversational Agent are seen as perturbations in this process, which aim at satisfying the user's long-term goals while keeping the conversation natural. Finally, results obtained by simulation suggest that such an approach is suitable for computer-aided human-human dialogue and is a first step towards three-party dialogue.

1. Introduction

Spoken dialogue is the most common way for communicating between humans. When an interaction aims at achieving a task, the dialogue is said to be task-oriented. For example, negotiating an appointment, interacting with an operator to solve a land-line problem or making decisions in a meeting are all task-oriented dialogues. In some cases, an interlocutor needs external information to perform the task successfully. In the land-line troubleshooting case for instance, the operator may need technical information about the customer's problem to provide useful support. Time taken to search for this information may deteriorate the quality and the naturalness of the dialogue. The goal of this paper is to provide a general framework for computer-aided human-human dialogue, where the task of providing contextual information to interlocutors is automated. The way information is provided by the *Conversational Assistant* (CA) may take any form like speech, pop ups on a screen *etc*.

Addressing this problem first requires analysing human-human spoken dialogue. Even if it has been studied for a long time from a cognitive/social science perspective [1, 2], its automatic processing from the computing point of view is relatively recent. Works from the mid 90's like [3, 4] presented systems able to passively listen to human-human conversations and process them in background (the former performing call-routing and the latter automatically summarizing meetings). Since then, research on human-human dialogue mainly focused on Spoken Language Understanding (SLU) tasks such as dialogue act tagging [5], topic detection [6], dialogue state tracking [7], *etc*.

In this work, we address a different aspect by considering the sequential nature of the task, taking into account the fact that utterances of the CA may have an impact on the future of the conversation. In both human-machine and computer-aided human-human dialogue, Dialogue Management (DM) may be seen as a sequential decision making problem, where decisions are made on which dialogue act has to be taken according to the dialogue context. Since its introduction for DM [8, 9], Reinforcement Learning (RL) has successfully proven its efficiency for optimising human-machine interaction strategies [10, 11].

In this paper, we propose to model for the first time the optimisation of the computer-aided human-human dialogue task as an RL problem. To do so, several issues are addressed. First, RL requires providing a reward function quantifying the quality of each intervention decision. It has to account for the naturalness of the conversation between both humans while ensuring a gain in the outcome of the dialogue. An original cost function is therefore introduced, balancing the intrusiveness and the success of the conversation. Second, an efficient way of computing the intervention strategy is provided by casting the problem into the Linearly Solvable Markov Decision Processes framework [12]. Third, to take Automatic Speech Recognition (ASR) and Natural Language Understanding (NLU) errors into account, experiments are made in the presence of noise. Empirical results show that the learned intervention strategy improves the general outcome of the conversation without much perturbation and while being robust to noise.

2. A General Model Of Human-Human Communication

2.1. Temporal Sequences

From the point of view of an external observer (like for instance a CA), dialogue may be seen as a sequence of observations o_t belonging to a set $\mathcal{O}$, namely utterances for a human observer and dialogue acts with confidence scores for a dialogue manager. Let us assume that each dialogue $d = \{o_0, o_1, ..., o_t, ...\}$ is drawn according to an underlying probability distribution:

$$\mathbb{P}(o_0, o_1, ...) = \mathbb{P}(o_0) \prod_{t>0} \mathbb{P}(o_t | h_t),$$

where $h_t = \{o_0, ..., o_{t-1}\}$ is called the *history* at time t.

2.2. Markov Chains

Because the space of histories can be very large, computing this distribution is often intractable. Relevant information from histories may however be compressed in a so-called *dialogue state*, belonging to some set $\mathcal{S}$. States are built so as to fulfill the

http://dx.doi.org/10.21437/Interspeech.2016-479

Markov assumption (the future depends only on the present independently from the past). It is always possible to represent a temporal sequence as a Markov chain by defining states as histories $s_t = h_t$. Designing more sophisticated state representations remains a major topic of research since the Markov property is heavily constraining. For some problems, it is possible to handcraft them [13] but such representations are strongly problem-dependent and do not generalise well. They can also be learned [14, 15]. Dialogue state tracking over dialogue turns has become a research problem in its own right [16] and is not yet solved. Human-human dialogue state tracking has also been the focus of recent researches [7]. Here, we assume that these states are given but that transition probabilities are unknown.

Given such a representation, a dialogue is then seen as a sequence of states $d = \{s_0, s_1, ..., s_t, ...\}$ drawn from a probability distribution:

$$\mathbb{P}(s_0, s_1, ...) = \mathbb{P}(s_0) \prod_{t>0} \mathbb{P}(s_t | s_{t-1}).$$

This process is called a *Markov chain* and the probability distribution mapping states to states is called the *transition function*.

2.3. Markov Reward Processes

Even though Markov chains are a powerful tool to model dialogue, they do not allow its evaluation. One defines the *reward function* as a mapping from states to the set of real numbers called. A reward function associated to a Markov chain defines a *Markov Reward Process*. Formally, a reward function associates a real number to each state ($r : \mathcal{S} \to \mathbb{R}$).

Assuming that each trajectory ends with probability 1, one can define the value of each state s as the expected cumulative reward that will be obtained when starting from this state:

$$\forall s \in \mathcal{S},\ V(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} r(s_t) \middle| s_0 = s \right].$$

In other words, the reward function defines the quality of each utterance while the value function measures how good it is to be in a state. The value function of the first state represents then the average value associated with the dialogue.

Reward functions measure the quality of dialogues. They have to be carefully designed to encode the optimisation criteria. Handcrafted evaluation criteria [17] may be chosen but they heavily rely on human subjectivity and may therefore be biasprone. Inverse Reinforcement Learning (IRL) [18, 19] is a more objective approach to this problem based on statistical learning. It relies on the assumption that humans are experts in interacting and that their behaviour is optimal according to some underlying unknown reward. IRL aims at finding a reward function explaining the observed behaviour. Another approach to learn this reward function, Score-Based Inverse Reinforcement Learning (SBIRL) [20], makes use of expert ratings which are commonly used to evaluate dialogue to regress a reward function.

3. Controlling Markov Chains

In this section, we suggest a way of improving the outcome of human-human dialogues through the intervention of a CA without introducing too much perturbation.

3.1. Markov Decision Process

Markov Decision Processes (MDPs) are a common paradigm to address the problem of sequential decision making under uncertainty. Formally, an MDP is a tuple $\{\mathcal{S}, \mathcal{A}, P, r\}$ where $\mathcal{S}$ is the set of states, $\mathcal{A}$ the set of actions, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ the transition function and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function. When each possible trajectory ends with probability 1, the aim is to find in each state the sequence of actions maximising the expected cumulative reward, called the value function $V(s) = \mathbb{E}[\sum_{t=0}^{\infty} r(s_t, a_t) | s_0 = s]$. The optimal value function V^* is the value function that is maximal in each state. Let L be the operator such that

$$\forall s \in \mathcal{S},\ LV(s) = \max_{a \in \mathcal{A}} \left\{ r(s, a) + \sum_{s' \in \mathcal{S}} P_a(s, s') V(s') \right\}.$$

The optimal value function V^* is the only function verifying the Bellman Optimality Equation: $\forall s \in \mathcal{S}, V^*(s) = LV^*(s)$ [21].

A mapping from states to actions is called a policy (noted π). The policy π^* is said optimal if it induces the optimal value function V^*. Several policies may be optimal and share V^*.

3.2. Kullback-Leibler Cost

Our goal is to improve the outcome of a human-human dialogue by intervening during the course of the conversation. From models defined in the previous section, we decide to cast the process of optimising the intervention strategy into the MDP framework. To do so, a reward function has to be defined.

We assume that dialogue states are naturally sampled from a probability distribution P_0 when the human-human conversation is going on. If the systems takes the action a in a state s_t, the next state will be sampled according to another distribution P_a. It is assumed that humans are quasi-experts at interacting and only need small state shifts to be optimal. In other words, we consider that we build on the capacity of a human to interact naturally by augmenting his/her interactionnal context but not by modifying totally this context so as to keep the conversation on track. One would then like to have P_a quite close to P_0. The Kullback-Leibler (KL) divergence is commonly used to measure the dissimilarity between two probability distributions. Formally, if p and q are two probability distributions on some discrete set $\mathcal{S}$, the KL divergence between p and q is defined by $\text{KL}(p||q) = \sum_{s \in \mathcal{S}} p(s) \log \frac{p(s)}{q(s)}$. Roughly speaking, KL divergence measures how likely a sample drawn from the distribution p could have been drawn from the distribution q.

From this, we build a reward function combining the outcome quality measure and an additional cost measuring how much the conversation will change by intervening. Formally, taking the action a in a state s induces a reward $r(s, a) = r(s) - KL(P_a(s, .)||P_0(s, .))$. A trade-off has then to be found between perturbing the conversation and improving the dialogue's general outcome. To our knowledge, the only research work balancing an action utility and the dialogue naturalness [22] concerns listening-oriented human-machine dialogue and does not make use of the KL divergence.

3.3. Linearly-Solvable Markov Decision Problems

We now describe an efficient framework to solve the optimisation problem defined just before: the Linear Markov Decision Processes (LMDPs) [12]. LMPDs are a special class of MDPs addressing the issue of perturbing a Markov Reward Process under a KL cost. Their efficiency has been shown on many different problems [23, 24, 25, 26] since by making one more assumption that we will later relax, they make the Bellman Equation linear. Powerful tools exist then to solve it.

LMDPs operate in a framework similar to ours. They however make the assumption that the action space is the whole transition probability simplex $\mathcal{P}$, that is actions consist in modifying directly the transition probabilities. Writing V_c^* as the optimal value function with those continuous actions, the Bellman Optimality Equation may be rewritten as follows:

$$\begin{aligned} V_c^*(s) =& \max_{P \in \mathcal{P}} \{r(s,P) + \sum_{s' \in \mathcal{S}} P(s,s')V_c^*(s')\}, \\ =& \max_{P \in \mathcal{P}} \{r(s) - \text{KL}(P(s,.)||P_0(s,.)) \\ &+ \sum_{s' \in \mathcal{S}} P(s,s')V_c^*(s')\}. \end{aligned}$$

In each state, the optimal transition function is given by:

$$\begin{aligned} \pi^*(s) =& \operatorname*{argmax}_{P \in \mathcal{P}} \Big\{ - \sum_{s' \in \mathcal{S}} P(s,s') \log \frac{P(s,s')}{P_0(s,s')} \\ &+ \sum_{s' \in \mathcal{S}} P(s,s')V_c^*(s') \Big\}, \\ =& \operatorname*{argmin}_{P \in \mathcal{P}} \Big\{ \sum_{s' \in \mathcal{S}} P(s,s') \log \frac{P(s,s')}{P_0(s,s')e^{V_c^*(s')}} \Big\}, \\ =& \operatorname*{argmin}_{P \in \mathcal{P}} \text{KL}\Big(P(s,.) \Big|\Big| \frac{P_0(s,.)e^{V_c^*(.)}}{Z(s)}\Big). \end{aligned}$$

where $Z(s) = \sum_{s' \in \mathcal{S}} P_0(s,s')e^{V_c^*(s')}$ is a normalisation constant. Using the fact that the KL divergence is minimal if and only if the two probability distributions are equal, the optimal transition function is given by

$$\pi^*(s) = P^*(s) = \frac{P_0(s,.)e^{V_c^*(.)}}{Z(s)} \tag{1}$$

The Bellman Equation then may be rewritten as

$$\begin{aligned} V_c^*(s) =& r(s,P^*) + \sum_{s' \in \mathcal{S}} P^*(s,s')V_c^*(s') \\ =& r(s) + \sum_{s' \in \mathcal{S}} P^*(s,s') \log Z(s) \\ =& r(s) + \log Z(s). \end{aligned}$$

Exponentiating this equation in vector form, it becomes linear:

$$\forall s \in \mathcal{S},\ e^{V_c^*(s)} = e^{r(s)}Z(s) = e^{r(s)}P_0(s,.)e^{V_c^*}.$$

3.4. Solving LMDPs with Stochastic Gradient Descent

Solving the previous equation via linear algebra methods may be intractable for large state spaces. An original method has been suggested in [26] in which, instead of solving the equation $e^{V_c^*} = e^{LV_c^*}$, one minimises the following quantity $|e^{V_c^*} - e^{LV_c^*}|$. The exponentiated value function is then linearly parametrised: $\forall s \in \mathcal{S},\ e^{V_c^*(s)} = \langle \theta, \phi(s) \rangle$ where $\phi(s)$ is a feature vector, θ a parameter vector that has to be learned and $\langle \theta, \phi \rangle$ the dot product between vectors θ and ϕ.

The following value $e^{V_c^*(s)} - e^{r(s)}P_0(s,.)e^{V_c^*}$ may then be rewritten as $\langle \theta, \phi(s) \rangle - e^r P_0(s,.)\langle \theta, \phi \rangle$ which is linear and thus convex. Given H some positive regularisation constant, $\mathcal{T}$ a set of trajectories, and v a probability distribution over those trajectories, it is shown that the optimiser θ^* of the following cost function:

$$\begin{aligned} c(\theta) =& -\log(\langle \theta, \phi(s_0)) \\ &+ H \sum_{t \in \mathcal{T}} v(T) \sum_{s \in T} \Big| \langle \theta, \phi(s) \rangle - e^{r(s)}P_0(s,.)\langle \theta, \phi \rangle \Big| \end{aligned}$$

induces a near optimal value function and thus a near optimal policy defined by (1). Since $c(\theta)$ is convex, it is possible to optimise it through Stochastic Gradient Descent [27], where an unbiased estimate of the cost function is given by:

$$\begin{aligned} r(\theta) =& -\frac{1}{\langle \theta, \phi(s_0) \rangle}\phi(s_0) \\ &+ H \sum_{s \in T} \Big[\text{sign}\Big(\langle \theta, \phi(s) \rangle - e^{r(s)}P_0(s,.)\langle \theta, \phi \rangle \Big) \\ &\times \Big(\phi(s) - e^{r(s)}P_0(s,.)\phi \Big) \Big]. \end{aligned}$$

3.5. LMDP with discrete actions

We now restrict the action space to a discrete one $\mathcal{A}$. Following the same equations, the optimal action is given by:

$$a^*(s) = \operatorname*{argmin}_{a \in \mathcal{A}} \text{KL}\Big(P_a(s,.) \Big|\Big| \frac{P_0(s,.)e^{V(.)}}{Z(s)}\Big).$$

This quantity is however not computable since in this case V is unknown. It has been however empirically shown [23] that, if we always take the action a greedy with respect to the optimal value function of the continuous MDP V_c^*, the obtained value function V_d will not be far from the optimal one V^*.

In the following sections, we will show the efficiency and the relevance of this method on a simulated dialogue task.

4. Experimental setting

The dialogue problem addressed in this paper is a variation on the negotiation dialogue game presented in [28, 29] where two humans interact in order to settle an appointment. We added one CA to this human-human conversation. It belongs to a human and is able to intervene during the dialogue to optimise its general outcome, defined as the outcome of its owner.

4.1. The negotiation dialogue game

In the negotiation dialogue game, two humans indexed by $i = 1, 2$ are involved. They have to agree on some time slot τ among N available slots. Booking time slot τ^i costs $c^i_{\tau^i}$ to human i while agreeing on the time slot $\tau^i = \tau^j$ makes her/him earn R^i, inducing then a total reward $r^i_\tau = \mathbb{1}_{\tau^i = \tau^j} R^i - c^i_\tau$.

It is a turn-based game and the first player is randomly chosen. At each turn, 3 actions are available to her/him. S/He can accept the last proposal by choosing the action `Accept`. This action ends the dialogue and each player receives her/his corresponding reward. Otherwise, the human may refuse the last proposal and suggest a new slot by the use of the `RefProp` action. Finally, s/he can take the `EndDial` action, ending then the dialogue while giving no reward to each one.

To avoid side effects, two states are added, an initial state in which all dialogues begin and a final state in which they end.

4.2. Strategies

As in [28], we assume that strategies of humans are fixed and not necessarily optimal, so we handcrafted them. Decisions are made according to each agent's set of available positive slots, which is the set of all slots inducing a positive reward and have never been refused by the opponent. When this set is empty, the agent ends the dialogue. Otherwise, s/he compares the reward induced by the last offer with the one associated with the best available slot. If this reward is smaller, the human refuses the proposition and suggests this best available slot. Otherwise, s/he accepts the offer. In order to add some variability to the simulator, those policies are randomised. If the last action was RefProp, with probability ϵ_1, the user takes a random action between Accept, Enddial and Refprop, in that case, the human suggests his/her best available slot.

4.3. State Representation

A Markovian representation of the conversation has to be designed. This representation will correspond to the point of view of the CA. It has first access to the timetable of its owner, inducing then the reward function of the dialogue.

It also tracks the last spoken utterance. Since ASR and NLU systems are error prone, the inputs of the CA do not necessarily match what has been said. The type of each action is always understood but when an offer is made, there may be misunderstandings on the time slot. Let SER be the Sentence Error Rate of the system. With probability SER, the system understands a wrong slot, while with probability $1 - SER$ the utterance is received undisrupted. We adopt the way [30] generates speech recognition confidence scores: $score_{reco} = \frac{1}{1+e^{-X}}$ where $X \sim \mathcal{N}(c, 1)$, where $c = 1$ if the player understood the right option, and $c = -1$ otherwise.

To get a fully Markovian state representation, the number of utterances said during the dialogue is also tracked.

4.4. Actions of the system

In that game, the CA has four actions available: ShouldAccept, ShouldEnddial, ShouldRefProp or doing nothing. Those actions are taken just before the turn of its owner. They represent suggestions of the system. If the system suggests an action, its user takes that action with probability $1 - \epsilon_2$ and take randomly another action otherwise.

5. Results

In our experiments, we fixed the number of slots to 3. The costs for each slot are randomly sampled between 0 and 4, while the reward obtained by setting an appointment is 3, leading to approximately 40000 states. Humans act randomly with probability $\epsilon_1 = 0.3$ and follow the suggestions of the dialogue system with probability $1-\epsilon_2 = 0.9$. To deal with discrete state spaces, we split the confidence into 6 categories around the cut points 0.05, 0.15, 0.5, 0.85, 0.95 and 1. Components of the feature vector are: initial state; final state; exponentiated rewards associated to each slot as well as the turn of the beginning state; the exponentiated obtained reward if the last action was Accept; the exponentiated differences of rewards between the suggested slot and the other slots; the exponentiated values of those slots; the value of accepting the suggested slot if the last action was RefProp and if it is the main user's turn; otherwise, the exponentiated reward of each available slot and the value of the last suggested action; finally a bias feature is added. We applied Stochastic Gradient Descent during 20000 iterations with a constant learning rate $\eta = 1$ during the first 5000 iterations and $\eta = 0.1$ for the others. At each step, this learning rate is divided by the norm of the gradient. The regularisation parameter is set to $h = 5$. To speed-up convergence, we used the mini-batch trick which averages at each iteration the gradient computed on 5 samples. The transition matrix P_0 and KL divergences between distributions are estimated according to the Monte-Carlo method over 20000 trajectories for each state.

Figure 1 shows the average reward (computed on 1000 dialogues) by the human with the help of the CA according to an increasing SER. The dotted line is the average reward obtained without CA. We observe that the average reward gradually decreases with an increasing SER, showing that in the presence of noise, the accuracy of the system decreases. We however observe that in every case, the obtained reward is never worse than the average reward without computer-aid.

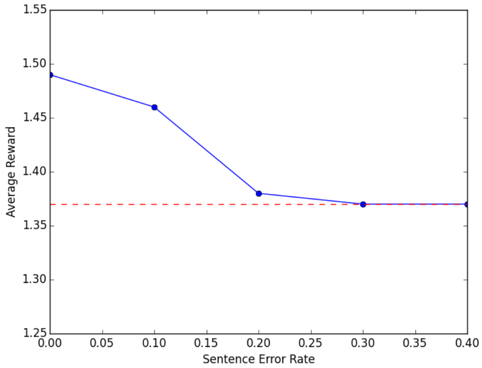

Figure 1: Average reward function of the SER

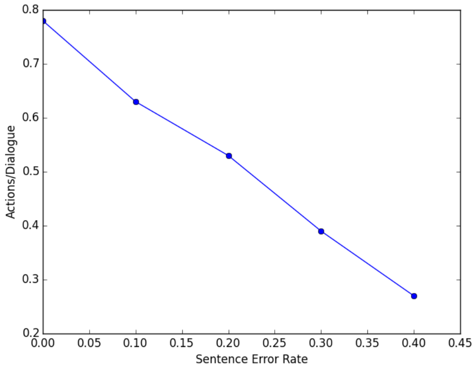

Figure 2: Interventions of the system function of the SER

Figure 2 shows the average number of actions taken by the system per dialogue according to an increasing SER. Because of the short length of the dialogues, the CA does not have always the opportunity to speak, explaining the fact that this number is smaller than 1. It decreases with an increasing SER but is always positive, even with much noise. This means that the system has learned to act only with a confidence score high enough, which is a desirable behaviour for the CA.

6. Conclusion

This paper presents a novel stochastic model for computer-aided human-human dialogue. This problem is cast as an MDP where the reward function depends on the one hand of the general outcome of the dialogue and on the other hand, on its intrusiveness. The experiment, used as a proof of concept, showed that this model leads to a gain in performance.

Future works will include more sophisticated ways to compute the transition matrices such as spectral methods [31]. It would also be interesting to introduce modifications of the algorithms to allow more general value function representations than linear parametrization. Co-adaptation between humans and CA [32] should also be studied. Finally, this approach will be tested on real human-human dialogues.

7. References

[1] J. L. Austin, *How to do things with words*. Oxford university press, 1975.

[2] H. H. Clark and E. F. Schaefer, "Contributing to discourse," *Cognitive Science*, vol. 13, no. 2, pp. 259–294, 1989.

[3] A. L. Gorin, G. Riccardi, and J. H. Wright, "How may i help you?" *Speech communication*, vol. 23, no. 1, pp. 113–127, 1997.

[4] A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen, "Meeting browser: Tracking and summarizing meetings," in *Proc. of the DARPA broadcast news workshop*, 1998.

[5] A. Stolcke, N. Coccaro, R. Bates, P. Taylor, C. Van Ess-Dykema, K. Ries, E. Shriberg, D. Jurafsky, R. Martin, and M. Meteer, "Dialogue act modeling for automatic tagging and recognition of conversational speech," *Computational linguistics*, vol. 26, no. 3, pp. 339–373, 2000.

[6] P.-Y. Hsueh, J. D. Moore, and S. Renals, "Automatic segmentation of multiparty dialogue," in *Proc. of EACL*, 2006.

[7] S. Kim, L. F. DHaro, R. E. Banchs, J. D. Williams, and M. Henderson, "The fourth dialog state tracking challenge," in *Proc. of IWSDS*, 2016.

[8] E. Levin and R. Pieraccini, "A stochastic model of computer-human interaction for learning dialogue strategies." in *Proc. of Eurospeech*, 1997.

[9] S. P. Singh, M. J. Kearns, D. J. Litman, and M. A. Walker, "Reinforcement learning for spoken dialogue systems." in *Proc. of NIPS*, 1999.

[10] O. Lemon and O. Pietquin, "Machine learning for spoken dialogue systems," in *Proc. of Interspeech*, 2007.

[11] S. Young, M. Gasic, B. Thomson, and J. D. Williams, "Pomdp-based statistical spoken dialog systems: A review," *Proc. of the IEEE*, vol. 101, no. 5, pp. 1160–1179, 2013.

[12] E. Todorov, "Linearly-solvable markov decision problems," in *Proc. of NIPS*, 2006.

[13] S. Larsson and D. R. Traum, "Information state and dialogue management in the trindi dialogue move engine toolkit," *Natural language engineering*, vol. 6, no. 3&4, pp. 323–340, 2000.

[14] L. Daubigney, M. Geist, and O. Pietquin, "Model-free pomdp optimisation of tutoring systems with echo-state networks," in *Proc. of SigDial*, 2013.

[15] L. El Asri, R. Laroche, and O. Pietquin, "Compact and interpretable dialogue state representation with genetic sparse distributed memory," in *Proc. of IWSDS*, 2016.

[16] M. Henderson, "Machine learning for dialog state tracking: A review," in *Proc. of The First International Workshop on Machine Learning in Spoken Language Processing*, 2015.

[17] M. A. Walker, D. J. Litman, C. A. Kamm, and A. Abella, "Paradise: A framework for evaluating spoken dialogue agents," in *Proc. of ACL*, 1997.

[18] A. Y. Ng, S. J. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Proc. of ICML*, 2000.

[19] S. Chandramohan, M. Geist, F. Lefevre, and O. Pietquin, "User simulation in dialogue systems using inverse reinforcement learning," in *Proc. of Interspeech*, 2011.

[20] L. El Asri, B. Piot, M. Geist, R. Laroche, and O. Pietquin, "Score-based inverse reinforcement learning," in *Proc. of AAMAS*, 2016.

[21] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.

[22] T. Meguro, Y. Minami, R. Higashinaka, and K. Dohsaka, "Learning to control listening-oriented dialogue using partially observable markov decision processes," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 10, no. 4, p. 15, 2013.

[23] E. Todorov, "Efficient computation of optimal actions," *Proceedings of the national academy of sciences*, vol. 106, no. 28, pp. 11 478–11 483, 2009.

[24] H. J. Kappen, V. Gómez, and M. Opper, "Optimal control as a graphical model inference problem," *Machine learning*, vol. 87, no. 2, pp. 159–182, 2012.

[25] P. Guan, M. Raginsky, and R. M. Willett, "Online markov decision processes with kullback–leibler control cost," *IEEE Transactions on Automatic Control*, vol. 59, no. 6, pp. 1423–1438, 2014.

[26] Y. Abbasi-Yadkori, P. L. Bartlett, X. Chen, and A. Malek, "Large-scale markov decision problems with kl control cost and its application to crowdsourcing," in *Proc. of ICMl*, 2015.

[27] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[28] R. Laroche and A. Genevay, "The negotiation dialogue game," in *Proc. of IWSDS*, 2016.

[29] A. Genevay and R. Laroche, "Transfer learning for user adaptation in spoken dialogue systems," in *Proc. of AAMAS*, 2016.

[30] H. Khouzaimi, R. Laroche, and F. Lefevre, "Optimising turn-taking strategies with reinforcement learning," in *Proc. of SigDial*, 2015.

[31] H. Glaude, C. Enderli, and O. Pietquin, "Spectral learning with non negative probabilities for finite state automaton," in *Proc. of ASRU*, 2006.

[32] M. Barlier, J. Perolat, R. Laroche, and O. Pietquin, "Human-machine dialogue as a stochastic game," in *Proc. of SigDial*, 2015.