EXPERT-BASED REWARD SHAPING AND EXPLORATION SCHEME FOR BOOSTING POLICY LEARNING OF DIALOGUE MANAGEMENT

Emmanuel Ferreira and Fabrice Lefèvre

LIA - University of Avignon
BP1228 - 84911 Avignon Cedex 9 - France
{emmanuel.ferreira,fabrice.lefevre}@univ-avignon.fr

ABSTRACT

This paper investigates the conditions under which expert knowledge can be used to accelerate the policy optimization of a learning agent. Recent works on reinforcement learning for dialogue management allowed to devise sophisticated methods for value estimation in order to deal all together with exploration/exploitation dilemma, sample-efficiency and non-stationary environments. In this paper, a reward shaping method and an exploration scheme, both based on some intuitive hand-coded expert advices, are combined with an efficient temporal difference-based learning procedure. The key objective is to boost the initial training stage, when the system is not sufficiently reliable to interact with real users (e.g. clients). Our claims are illustrated by experiments based on simulation and carried out using a state-of-the-art goal-oriented dialogue management framework, the Hidden Information State (HIS).

Index Terms— dialogue management, reinforcement learning, reward shaping, value function approximation

1. INTRODUCTION

Goal-oriented Interactive Systems (such as Spoken or Multimodal Dialogue Systems, SDSs) are designed to help a human to achieve a task. The latter is generally formalised as a goal related to an information retrieval problem, for instance hotel reservation, or flight booking service. The research on the topic has seen tremendous breakthroughs proposed these recent years. However, the residual lack of both efficiency (timely offer of the desired information) and naturalness (behaviour not distinguishable from human) of current systems can still end up in a frustrating user experience.

Dialogue Manager (DM) is the main component of SDSs. It is designed to make appropriate decisions to fulfil the user goal based on uncertain dialogue contexts. Since its first description as a Markov Decision Process (MDP) [1] a DM can be seen as an agent interacting with an environment using a Reinforcement Learning (RL) algorithm so as to maximise some expected cumulative discounted reward [2]. The system's objective design criteria, based on the task completion and the overall system efficiency (e.g. interaction length), can be coded in the reward function. For a better account of the high level of uncertainty in the information conveyed inside SDSs the MDP mathematical framework scheme was extended to the Partially Observable Markov Decision Process (POMDP) framework [3, 4, 5].

In the case where a new system is developed from scratch, in-domain dialogue corpora are seldom readily available and collecting such data is both time consuming and expensive. Indeed, it requires either a working prototype or to resort to a Wizard-of-Oz configuration. Furthermore, such collected data often provide information about a small part of possible system dynamics (e.g. user typical behaviours, situations, etc.). To alleviate this problem a recent trend of research in DM considers the use of user simulations to generate additional samples. Resulting policies trained with such an approach showed some satisfying performance on user trials [6]. However, while doing so, a modelling bias is introduced. Indeed, the learnt behaviour may consequently not be fitted for any real user but the simulated average user [7]. That is why, the capacity of an RL algorithm to learn online while interacting with the user may still be interesting. However, current approaches in this line of thought assume that an acceptable sub-optimal initial policy has been found by either some sample-efficient, off-policy RL method or by hand.

Recent attempts were made to address this problem by using sample-efficient learning algorithms in order to limit the need for such a "bootstrap step". Gaussian Processes SARSA [8], incremental sparse Bayesian method [9] and KTD [10] are among the most promising approaches to tackle this issue. The three of them provide frameworks, pretty comparable in some aspects, offering convenient solutions to the sparse value function estimation problem which arises in the context of MDP with large state space as in the dialogue management case.

Even if some good policies can be obtained with much fewer training examples using such methods, they still imply a first step of training with a high level of exploration during which the system acting mostly randomly can not reasonably be confronted to "real" users. Thus, lowering the length of this warm-up phase is still an open problem when such systems are to be used in real-world situations (and not only tested with user simulators as what is done currently). One solution can be to introduce some subgoal-based heuristic with a reward shaping mechanism such as in [11], or to find ways to directly exploit a limited and fixed dataset as was proposed in [12] with hybrid models, or else in [13] with human-agent transfer rules. The two latter methods combine RL methods with supervised learning ones, such as in Learning from Demonstration (LfD) methods (classification, regression, planning, etc.). However the level of performance achieved by such approaches are closely related to the overall quality of the seed dataset, but it often provides insufficient, noisy and suboptimal information about how the system should perform in real conditions. Another solution is to introduce soft or hard constraints based on expert knowledge in the POMPD state-action space. In [14], a hand-crafted dialogue manager nominates a set of one or more "valid" actions considering the current dialogue state and the POMDP-based learning agent selects the optimal action from this restricted set. However, in this approach the expert knowledge is not supposed to be trivial or incomplete. Our claim is that a simple vision of the nature of the problem, which can be easily produced by a human expert, may be sufficient to bootstrap

the initial performance of a dialogue system.

This paper investigates how unelaborated expert cues can be used to boost the initial training stage. Two options are considered to deal with such additional information. First, an exploration scheme to safely explore the state space exploiting the expert advices. Second, a potential-based reward shaping method to integrate expert hints as an additional reinforcement signal in the RL problem.

In order to illustrate the benefits of our proposals, the KTD framework is employed with regard to its interesting properties [10] in the context of the HIS system [3]. Before going live with real user trials, although this is our overall goal, this preliminary study is carried out with simulated dialogues. In this context, a better control over the experimental conditions, such as the simulated Concept Error Rate (CER) level, is possible and comparison between several techniques is facilitated.

The remainder of the paper is organised as follows. In Section 2 some background on MDP/POMDP, RL paradigm, DM problem and KTD method are given. Then, in Section 3 the considered options to introduce expert knowledge in a RL problem are detailed. Section 4 is dedicated to the presentation of the experimental setup. Then the following section details and discusses the results. Finally Section 6 opens a discussion on some considerations relevant to the use of expert knowledge, before concluding in Section 7 with some perspectives.

2. BACKGROUND

2.1. MDP and RL paradigm

A MDP is a tuple $\{S, A, T, R, \gamma\}$ where S is the state space (discrete, continuous or mixed), A is the discrete action space, T is a set of Markovian transition probabilities, R is the immediate reward function, $R : S \times A \times S \rightarrow \Re$ and $\gamma \in [0, 1]$ the discount factor (discounting long term rewards). At each time step t the environment is in a state s_t and the agent chooses an action a_t according to some mapping from state to actions called a policy, $\pi : S \rightarrow A$. The state then changes to s_{t+1} according to the Markovian transition probability $s_{t+1} \sim T(.|s_t, a_t)$ and, according to this, the agent received from the environment a reward $r_t = R(s_t, a_t, s_{t+1})$.

The core problem of MDPs is to find a "policy" that maximises some cumulative function of the rewards. Typically the expected discounted sum over a potentially infinite horizon is used: $\sum_{t=0}^{\infty} \gamma^t r_t$ (the return). For a given policy and start state s, this quantity is called the value function: $V^\pi(s) = E[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, \pi] \in \Re^S$. V^* corresponds to the value function of any optimal policy π^*. As an alternative to the value function one may define the Q-function, adding a degree of freedom on the first action to be chosen: $Q^\pi(s, a) = E[\sum_{t\geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi] \in \Re^{S\times A}$. Like V^*, Q^* corresponds to the action-value function of any optimal policy π^*. If it is known, an optimal policy can be directly computed by being greedy according to this Q-function: $\pi^*(s) = \arg\max_a Q^*(s, a) \forall s \in S$.

2.2. Dialogue Management as a POMDP

Dialogue management problem has first been described in [1] as a Markov Decision Process to determine an optimal mapping between situations and actions. The POMDP framework [15], as a generalization of the fully-observable MDP, maintains a belief distribution $b(s)$ over user states, assuming the true one is unobservable. Thereby, POMDP explicitly handles parts of the inherent uncertainty of the DM problem (word error rate, concept error rate, etc.). A POMDP policy maps the belief state space into the action space. That is why, the optimal policy can be understood as the solution of a continuous space MDP.

In practice, POMDP problems are intractable to solve exactly due to the curse of dimensionality (belief state/action spaces). Among other techniques, the HIS model [3] circumvents the RL scaling problem by organising the belief space into partitions (grouping states sharing the same probability) and then mapping the full belief space (partitions) into a much reduced summary space where RL algorithms work reasonably well.

2.3. Sample-efficient TD Learner

The Kalman Temporal Differences (KTD) is derived from the well-known Kalman filter algorithm [16] aiming at inferring some hidden variables from related past observations and applied to the estimation of the temporal differences for the action-value function optimisation. Notice that just the very basic explanations are recalled here, for further details please refer to [17, 10]. In this framework, a parametric representation of the Q-function can be chosen: $\hat{Q}_\theta = \theta^T \phi(s, a)$, where the feature vector $\phi(s, a)$ is a set of n basis functions to be designed by the practitioner and $\theta \in \Re^n$ the parameter vector to be learnt. The components of the parameter vector θ are the hidden variables which are modeled as a random vector. Such parameter vector is considered to evolve following a random walk though this evolution equation: $\theta_t = \theta_{t-1} + v_t$, with v_t a white noise of covariance matrix P_{v_t}. The latter allows to take into account the possible non-stationarity of the function. The observations correspond to the environment rewards which are linked to the hidden parameter vector through one of the sampled Bellman equations $g_t(\theta_t)$ depending on the RL scheme employed i.e. evaluation for on-policy or optimality for off-policy learning:

$$g_t(\theta_t) = \begin{cases} \hat{Q}_{\theta_t}(s_t, a_t) - \gamma \hat{Q}_{\theta_t}(s_{t+1}, a_{t+1}) & \text{(evaluation)} \\ \hat{Q}_{\theta_t}(s_t, a_t) - \gamma \max_a \hat{Q}_{\theta_t}(s_{t+1}, a) & \text{(optimality)} \end{cases}$$

They are supposed to follow the observation equation: $r_t = g_t(\theta_t) + n_t$ where a white noise n_t with covariance matrix P_{n_t} is also considered. KTD-SARSA denotes the use of the sampled evaluation Bellman equation in the KTD algorithm, KTD-Q, the use of the sampled optimality one.

In [10], the authors showed that KTD framework proposes a unified framework able to cope with all DM required properties. In fact, it is sample-efficient, it allows online/offline and on-policy/off-policy learning, it provides ways to estimate uncertainty to deal with the exploration/exploitation dilemma, it fits tracking issue, and it supports linear and non-linear parametrisation. Furthermore, KTD algorithms (KTD-Q/KTD-SARSA) were favourably compared to different state-of-the-art algorithms able to deal with one single property at once, such as Q-learning, LSPI or GP-SARSA. For further details on this technique please refer to [17, 10].

3. INTRODUCTION OF EXPERT KNOWLEDGE IN RL

Two main options are considered here to introduce some expert knowledge in the learning process. The first one consists in an exploration/exploitation scheme which uses expert advices to guide the initial exploration and to boost the initial performance. The second considers the expert knowledge as an additional reinforcement signal (i.e. reward).

3.1. Option 1: Expert-Greedy Scheme

The first proposed method concerns the exploration/exploitation technique employed during the learning. It is derived from the Bonus-Greedy scheme [18] which corresponds to one of the most efficient available scheme for the DM problem [18, 10]. This latter uses both the mean and the variance of the estimated Q-values, noted respectively $\mu_Q(s_t, a)$ and $\sigma^2_Q(s_t, a)$, to fit the exploration/exploitation dilemma. Here, an additional term $\upsilon(s_t, a)$ is introduced compared to [18], it corresponds to an expert advice function, returning an additional bonus β_1 when the corresponding action is also chosen by applying a set of simple hand-crafted expert rules considering s_t and 1 otherwise. At any system turn t, the following value is computed for each $a \in A$:

$$\mu_Q(s_t, a) + \beta \frac{\sigma^2_Q(s_t, a)\ \upsilon(s_t, a)}{\beta_0 + \sigma^2_Q(s_t, a)\ \upsilon(s_t, a)} \quad (1)$$

where β and β_0 are some meta-parameters. The action which maximize this value is chosen as system response. Thereby, when the uncertainty about the action-value function estimate for the state-action couple (s_t, a) is high, the associated variance is high and a bonus (of value up to β) is given for the choice of this action, favouring exploration. Furthermore, an action which benefits of an expert approval is also scaled up according to its associated variance (high bonus for high variance) to incite initial expert guided exploration. As the state space is explored the uncertainty decreases and the variance tends to be low favouring the choice of greedy actions.

3.2. Option 2: Expert-Based Reward Shaping

The second option focuses on using expert knowledge as a shaping reward function. This kind of function is dedicated to provide an additional reward in order to guide the learning agent towards a good (or optimal) policy faster. Indeed, it can be viewed as a solution for the temporal credit assignment problem for fine grained state-action spaces. The memoryless reward shaping function which is one of the most general shaping pattern is adopted here. So, the considered reward function is the sum of the basic environment reward function R_{env} (objective) and the new expert-based one R_{expert}. The resulting transformed MDP $M^{'}$ is defined by the tuple $(S, A, T, \gamma, R^{'})$ where $R^{'}$ is is the reward function defined as: $R^{'}(s_t, a_t, s_{t+1}) = R_{env}(s_t, a_t, s_{t+1}) + R_{expert}(s_t, a_t, s_{t+1})$ where $R_{expert} : S \times A \times S \rightarrow \Re$ is a bounded real-valued function called here the expert-shaping reward function. Since the system is learning a policy for $M^{'}$ with the idea of using it in M, the question at hand is: what form of reward function R_{expert} can guarantee that the optimal policy in $M^{'}$ will be optimal in M? In the case where no further knowledge of T and R dynamics is available, [11] showed that the potential-based shaping rewards leave (near-)optimal policies unchanged. R_{expert} is the additional reward function (corresponding to function F in Ng et al.'s paper). The potential-based shaping reward function is defined as follows:

$$R_{expert}(s_t, a, s_{t+1}) = \gamma\psi(s_{t+1}) - \psi(s_t) \quad (2)$$

where ψ is a potential function which corresponds here to an heuristic taking into account a rough estimate of the current dialogue progress (remaining dialogue effort) and a match between system act and expert advice. Ergo, a positive value is attributed to each system action which is similar to the expert prediction, determined from a set of simple hand-crafted rules as in Option 1, and more emphasis is put on similar actions which lead to a dialogue progress, i.e. actions which have good chance to bring the dialogue close to a possible task success from the expert's point of view.

4. EXPERIMENTAL SETUP

In this section the experimental setup is presented, including a brief description of the HIS-based Towninfo Dialogue System and some experimental details.

4.1. TownInfo Dialogue System

The TownInfo Dialogue System is a HIS-based dialogue system for the tourist information domain, related to a virtual town, which has been originally developed at Cambridge University [3]. It is worth noting that the HIS system has already been tested with real users in [6], and in a more recent and matured version called CamInfo (using true Cambridge tourist information), in [8]. The goal and agenda-based user simulator presented in [6] is employed to carry out our experiments. The simulation is done at the semantic (intentional) level. Nevertheless, it is possible to specify a speech understanding error rate using the error simulator so as to reproduce realistic conditions (at least in terms of observation uncertainty).

In order to deal with large state space (here 12 slot hierarchical ontology) and action space (11 typed acts) the system maintains a set of partitions which represents the overall belief state. Both the belief state and the action space are mapped into a more reduced summary state space where basic RL algorithms are tractable. Concerning the summary state space, it is the compound of two continuous values (the two-first top partitions probabilities) and three discrete values (last user act type, partition status and history status). The summary action space contains 11 actions (e.g. inform, confirm, select, etc.). Then a heuristic-based method maps the summary action back to the master state (hand-crafted part). The environment rewards penalised each dialogue turn by -1 and at the end of a dialogue the DM is rewarded by + 20 if the goal is reached, 0 otherwise.

4.2. Experimental Details

Two baselines are chosen to evaluate the two proposed options and their combination. As first baseline, the online version of the off-policy KTD-Q algorithm (noted KTD-Q BASELINE) is employed due to its high performance in the conditions at hand [10]. The Q-function is parametrised using linear-based Radial Basis Function (RBF) networks, one per action, as described in [10] and the Bonus-Greedy scheme [18] is adopted, with $\beta = 1000$ and $\beta_0 = 100$. The second baseline is a hand-crafted expert policy (noted HDC EXPERT) which maps summary states into summary actions by the use of a set of 9 rather intuitive expert rules. As an example, when the system detects that either no entities in database match with the top hypothesis, or the latter contains information denied by the user, or there is an explicit request of alternative, the rule-based expert policy chooses the summary action "ask for an alternative".

For the sake of consistency, all the results presented hereafter (with the exception of HDC EXPERT just performing "greedily") are obtained under online RL conditions. The results are averaged over 50 training runs performed in parallel and are always presented in terms of both mean discounted cumulative rewards or/and average success rates with respect to the number of training dialogues or different CER levels. These results are either gathered during the learning stage of the policy (controlled case) when exploration is possible or concern the case where policies are tested. Thus, in this latter case, the next action is chosen greedily with respect to the learnt Q-function. The associated standard deviations are added to all the results.

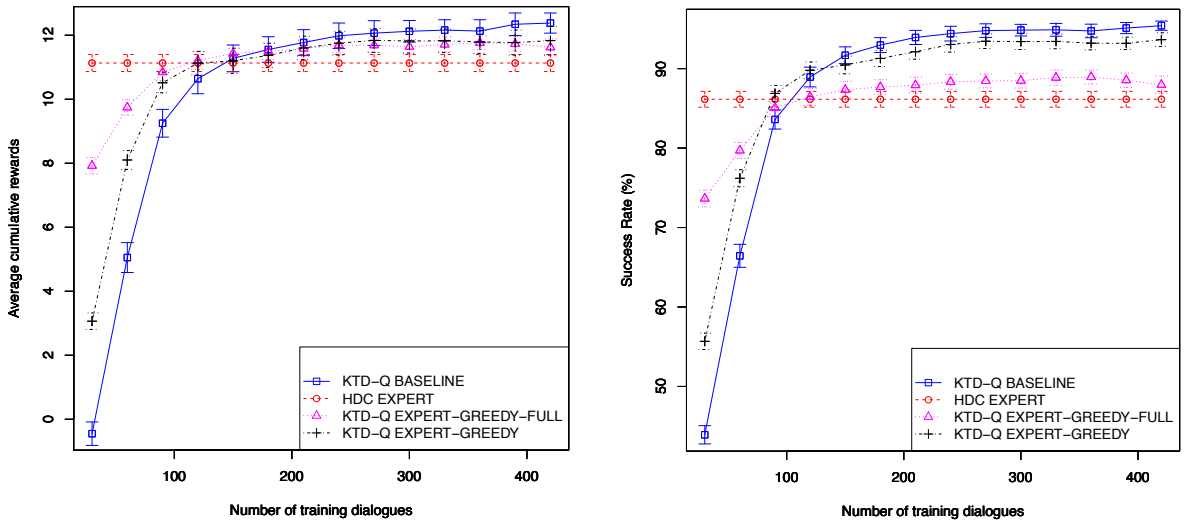

Fig. 1. Results of HDC EXPERT and KTD-Q with and without the use of the expert greedy exploration scheme (controlled case)

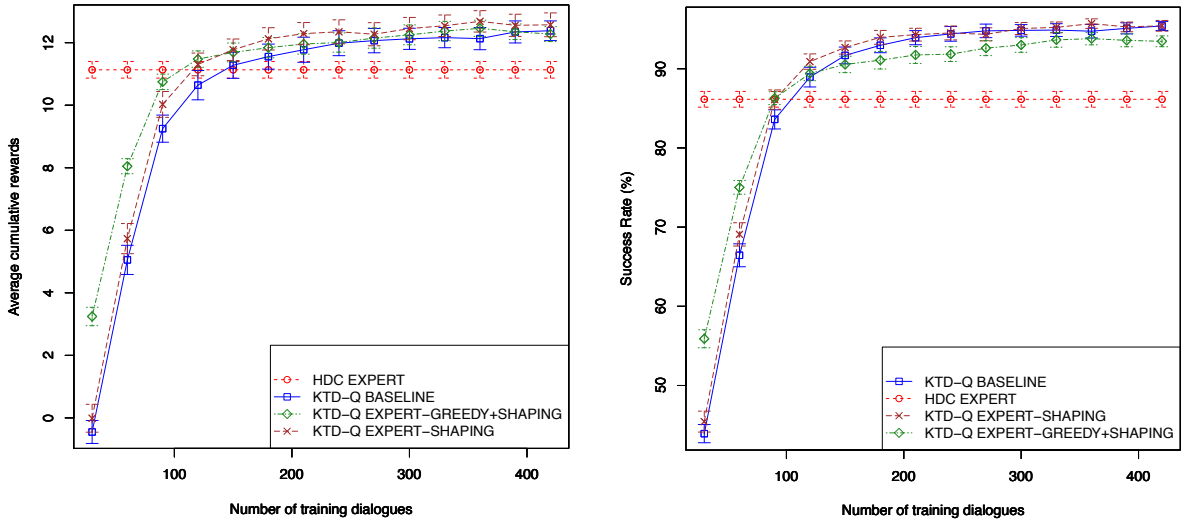

Fig. 2. Results of HDC EXPERT and KTD-Q with and without the use of the expert-based reward shaping method (controlled case)

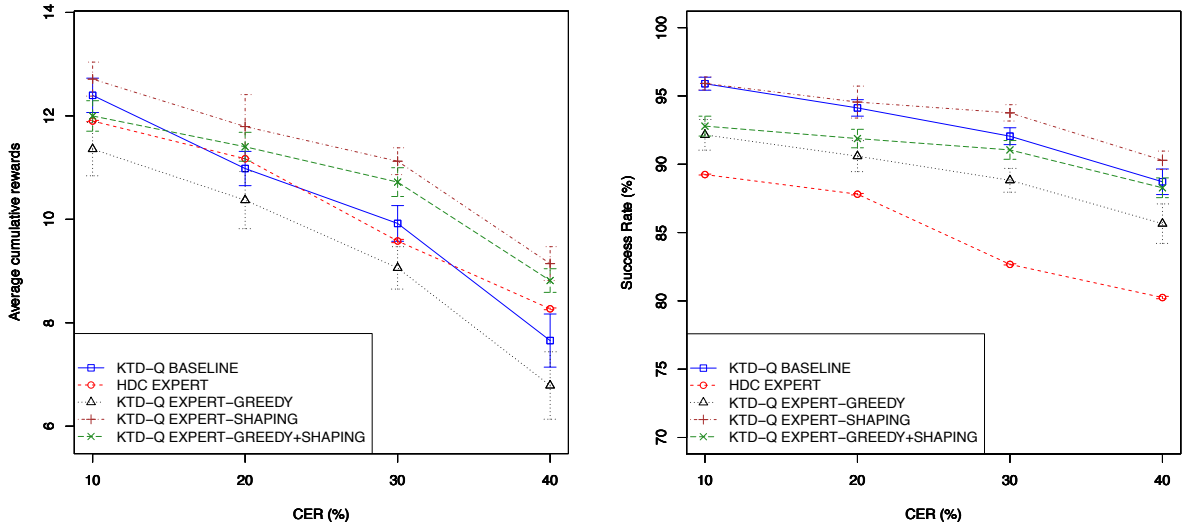

Fig. 3. Results of HDC EXPERT, KTD-Q with and without the use of expert knowledge (options) in different noise conditions (no control)

5. RESULTS

This section illustrates the advantages of using expert knowledge to boost the initial learning stage, when the system is not sufficiently consistent to interact with real users (high exploration / poor performance). First, the two baselines are compared with the two other methods presented in Section 3 (noted KTD-Q EXPERT-GREEDY, KTD-Q EXPERT-SHAPING), but also with their combination. Finally, a second experiment focuses on noise robustness of the approaches.

5.1. Expert Knowledge Transfer

In this experiment, results are presented in two figures for sake of clarity (Fig.1 and Fig.2). The two baselines are compared to four other techniques: two KTD-Q methods using Expert-Greedy scheme for exploration (Option 1) with different υ functions (different β_1), a KTD-Q algorithm with an additional expert shaping reward function (Option 2) and the combination of the two latter options (Option 1 + Option 2). In all configurations, the user simulator is set to interact with the DM at a 10 % CER. In all the curves, each point is the result of an average made over a sliding window of 100 point width of the performance gathered during the learning (controlled case).

Despite its simplicity (some intuitive rules), HDC EXPERT obtains good performance in both terms of average cumulative rewards and success rate (resp. 11.1 and 86.1% on average). KTD-Q BASELINE performance show the interest of considering RL methods rather than a hand-crafted fixed and suboptimal policy. Indeed, in only 210 dialogues it outperforms the HDC EXPERT performance both in terms of average cumulative rewards and success rate (resp. $+0.6$ and $+7.8\%$ on average, statistical significance determined by unpaired Mann-Whitney test, $p < 0.05$) and this gap is still increasing after several hundreds of dialogues. But at the beginning of the learning the results gathered from the environment are very low both in terms of average cumulative rewards and success rate. So, it is possible to identify three different phases:

• **Warm-up phase** from 0 to 100 dialogues. The agent mainly explores the action space with no prior on the systems dynamics, just using variance of this estimation (Bonus Greedy). So, the results are pretty poor and it corresponds to a phase where a system can hardly be used to interact with real people.

•**Improvement phase** from 100 to 200 dialogues. Here, the agent mainly exploits its current estimate (Q-values) and its behavior is already comparable to this of an expert-based system but also continues to explore and improves its global efficiency thanks to its uncertainty management ability.

• **Convergence phase** above 200 (as long as environment dynamics remain unchanged). The agent refines its estimate through occasional exploration and converges to a stable optimum.

It is worth noting that these phases can also be identifiable on the results obtained with real people (e.g. in [8]), of course with different bounds as they highly rely on the sample efficiency of the used RL algorithm.

The objective behind the use of expert knowledge (through the presented options) is to enhance the performance gathered during the warm-up phase and to downsize the improvement phase without delaying the optimal convergence. Thus globally shifting the learning curve towards the left. Figure 1 shows that the first issue can be addressed by giving more emphasis on the expert's guidance during the exploration. Indeed both KTD-Q EXPERT-GREEDY (β_1=12) and KTD-Q EXPERT-GREEDY-FULL (β_1=1000) obtain better results than KTD-Q BASELINE in the very beginning of the learning (resp. $+3.5$ $+12.1\%$ and $+8.2$ $+29.7\%$ at 30 dialogues). However, to set a too strong emphasis on the expert knowledge leads to policies that do not perform as well as the KTD-Q BASELINE when only the convergence phase is considered (resp. -0.57 -1.92% and -0.61 -6.56% at 390 dialogues, statistical significance determined by unpaired Mann-Whitney test, $p < 0.05$). Such results reflect the well-known exploration/exploitation dilemma. So, full expert guided exploration is not sufficient to find the optimal mapping. It is therefore necessary to appropriately defines υ as a trade-off between initial and delayed performance. Concerning the second issue, the KTD-Q EXPERT-SHAPING method (see Fig.2) achieves slightly better results than KTD-Q BASELINE in the improvement phase (reps. $+0.66$ $+1.96\%$ on average at 120 dialogues, statistical significance determined by unpaired Mann-Whitney test, $p < 0.05$). Furthermore, such benefit is conserved in the convergence phase because a more efficient policy has been learnt at the end of the improvement phase (in terms of number of turns required to reach user goal). Nevertheless, comparing to the considered Expert Greedy methods, despite the fact that both average cumulative rewards and success rate curves seem to increase slightly faster than KTD-Q BASELINE ones the overall initial performance are still low. In order to address both issues at once, the two proposed options are combined (KTD-Q EXPERT-GREEDY+SHAPING). This latter benefits of the already presented advantages of both KTD-Q EXPERT-GREEDY and KTD-Q EXPERT-SHAPING in spite of a small loss in terms of success rate during improvement and convergence phases compared to the KTD-Q BASELINE results. However, KTD-Q EXPERT-SHAPING obtains slightly better results than this method except during the warm-up phase. It is important to notice that even in the case where such shaping reward methods would be based on a worse expert estimation than the present case, the potential-based technique ensures that convergence to the near-optimal policy is still preserved.

5.2. Expert Knowledge Transfer in Noisy Conditions

This last experiment focuses on the impact of observation noise in the optimization procedure. Results are shown in Figure 3 in terms of cumulative rewards with respect to different CER levels. For these plots, each point is an average made on results obtained over 50 policies learned with 400 dialogues and then tested with 1000 dialogues using the same CER level. For all the presented curves a CER increase implies an overall performance decrease. Despite the fact that in high noise the HDC EXPERT policy obtains comparable results in terms of average cumulative rewards compared to the KTD-Q BASELINE (resp. 8.3 and 7.6 at 40% CER), results in terms of success rate are in favour of the learning method (resp. 80.2 and 88.7 at 40% CER). Thereby, a lower average cumulative rewards for the KTD results implies a lower efficiency of the policy wrt number of turns. This problem seems to be corrected when expert reward is considered. Indeed, KTD-Q EXPERT-SHAPING is above all methods whatever the noise conditions. Thus, despite its rough definition, the expert shaping reinforcement improves the ability to defer noise degradation. One of the reasons for this is that additional rewards are gathered all along the dialogue and offer a granular form of reward function. So, in case of high CER expert reward shaping can still favour or penalize a system local behaviour despite the overall task failure (or success). For the EXPERT-GREEDY case, only considering expert advices as exploration criterion does not seem to lead to improved results. As it is also illustrated in Fig.2, KTD-Q EXPERT-GREEDY+SHAPING performs better than the KTD-Q BASELINE in terms of average cumulative reward but obtains lower success rate.

6. DISCUSSION

The choice of the KTD framework in this work does not prevent the use of expert shaping reinforcement learning for other similar RL algorithms (e.g. GPTD [8]). Indeed the potential-based method presented above is amenable to all the other RL methods. Nevertheless the Expert Greedy scheme, and thus its combination with expert shaping, requires a way to estimate the variance of the current estimate of the Q-values, which is thus a constraint on the eligible RL algorithms. Another important point is that this setup allows a more granular view of the reward function rather than a mere judgement at the end of an episode. Indeed, it serves as a more specific way to avoid or strengthen some local system behaviours. So, when sample-efficient algorithms are considered, like KTD, the approach can be viewed as a good option to avoid the need of a user simulator. Hence, expert advices (rules) can serve to bootstrap the system in just 100-200 interactions and help to defer degradations due to noisy observations. As such it can be viewed as a kind of expert teaching, or Inverse RL approaches, as in [19]. In the present work, the expert knowledge is hand-crafted in order to give the basic insights on how the system must behave. Nevertheless, such rules or advices can be gathered from another already learnt policy dedicated to elicit a related task, as is done with transfer rules extracted from sampled dialogues [13].

7. CONCLUSION

This paper has presented methods to use rough expert knowledge to boost the warm-up phase of an agent reinforcement training. These methods consist of an expert-based reward shaping function and an exploration scheme both using expert knowledge as advices in their decision-making process. The presented shaping reward approaches showed a good robustness to noisy conditions and have interesting properties that guarantee the optimality when expert hints are merged into an additional reinforcement learning signal using a potential-based shaping reward function. However, this study, based on user simulations, serves as a "proof of concept" and should be complemented with real user trials.

Acknowledgments

The authors would like to thank the Cambridge University Engineering Dpt Dialogue Systems Group for providing the TownInfo HIS System and the MALIS Supélec Group for their help in using the KTD Framework. This work is partially funded by the ANR MaRDi project.

8. REFERENCES

[1] E. Levin, R. Pieraccini, and W. Eckert, "Learning dialogue strategies within the markov decision process framework," in *ASRU*, 1997.

[2] R. Sutton and A. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 1054–1054, 1998.

[3] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu, "The hidden information state model: A practical framework for pomdp-based spoken dialogue management," *Computer Speech and Language*, vol. 24, no. 2, pp. 150–174, 2010.

[4] B. Thomson and S. Young, "Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems," *Computer Speech and Language*, vol. 24, no. 4, pp. 562–588, 2010.

[5] F. Pinault and F. Lefèvre, "Unsupervised clustering of probability distributions of semantic graphs for pomdp based spoken dialogue systems with summary space," in *IJCAI 7th Workshop on knowledge and reasoning in practical dialogue systems*, 2011.

[6] J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young, "A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies," *Knowledge Engineering Review*, vol. 21, no. 2, pp. 97–126, June 2006.

[7] J. Schatztnann, M. Stuttle, K. Weilhammer, and S. Young, "Effects of the user model on simulation-based learning of dialogue strategies," in *ASRU*, 2005.

[8] M. Gašić, F. Jurčíček, S. Keizer, F. Mairesse, B. Thomson, K. Yu, and S. Young, "Gaussian processes for fast policy optimisation of pomdp-based dialogue managers," in *SIGDIAL*, 2010.

[9] L. Sungjin and M. Eskenazi, "Incremental sparse bayesian method for online dialog strategy learning," *Journal on Selected Topics in Signal Processing*, vol. 6, pp. 903–916, 2012.

[10] L. Daubigney, M. Geist, S. Chandramohan, and O. Pietquin, "A comprehensive reinforcement learning framework for dialogue management optimization," *Journal on Selected Topics in Signal Processing*, vol. 6, no. 8, pp. 891–902, 2012.

[11] A. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, 1999.

[12] J. Henderson, O. Lemon, and K. Georgila, "Hybrid reinforcement/supervised learning of dialogue policies from fixed data sets," *Computational Linguistics*, vol. 34, no. 4, pp. 487–511, Dec. 2008.

[13] M. Taylor, H. Suay, and S. Chernova, "Integrating reinforcement learning with human demonstrations of varying ability," in *The 10th International Conference on Autonomous Agents and Multiagent Systems*, 2011, AAMAS '11, pp. 617–624.

[14] J. D. Williams, "Integrating expert knowledge into pomdp optimization for spoken dialog systems," in *In Proceedings of the AAAI-08 Workshop on Advancements in POMDP Solvers*, 2008.

[15] L. Kaelbling, M. Littman, and A. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence Journal*, vol. 101, no. 1-2, pp. 99–134, May 1998.

[16] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[17] M. Geist and O. Pietquin, "Kalman temporal differences," *Journal of Artificial Intelligence Research (JAIR)*, vol. 39, no. 1, pp. 483–532, Sept. 2010.

[18] L. Daubigney, M. Gasic, S. Chandramohan, M. Geist, O. Pietquin, and S. Young, "Uncertainty management for on-line optimisation of a pomdp-based large-scale spoken dialogue system," in *Interspeech*, 2011.

[19] S. Chandramohan, M. Geist, F. Lefèvre, and O. Pietquin, "User Simulation in Dialogue Systems using Inverse Reinforcement Learning," in *Interspeech*, 2011.