Available online at www.sciencedirect.com

ScienceDirect

Computer Speech and Language 34 (2015) 256–274

COMPUTER SPEECH AND LANGUAGE

www.elsevier.com/locate/csl

Reinforcement-learning based dialogue system for human–robot interactions with socially-inspired rewards☆

Emmanuel Ferreira*, Fabrice Lefèvre

LIA-CERI, University of Avignon, Avignon, France

Received 27 May 2014; received in revised form 11 February 2015; accepted 25 March 2015
Available online 2 April 2015

Abstract

This paper investigates some conditions under which polarized user appraisals gathered throughout the course of a vocal interaction between a machine and a human can be integrated in a reinforcement learning-based dialogue manager. More specifically, we discuss how this information can be cast into socially-inspired rewards for speeding up the policy optimisation for both efficient task completion and user adaptation in an online learning setting. For this purpose a potential-based reward shaping method is combined with a sample efficient reinforcement learning algorithm to offer a principled framework to cope with these potentially noisy interim rewards. The proposed scheme will greatly facilitate the system's development by allowing the designer to teach his system through explicit positive/negative feedbacks given as hints about task progress, in the early stage of training. At a later stage, the approach will be used as a way to ease the adaptation of the dialogue policy to specific user profiles. Experiments carried out using a state-of-the-art goal-oriented dialogue management framework, the Hidden Information State (HIS), support our claims in two configurations: firstly, with a user simulator in the tourist information domain (and thus simulated appraisals), and secondly, in the context of man–robot dialogue with real user trials.

Keywords: Human–robot interaction; POMDP-based dialogue management; Reinforcement learning; Reward shaping

1. Introduction

In a goal-oriented vocal interaction between a machine and a human, the dialogue manager (DM) is responsible for making appropriate dialogue decisions to fulfil the user goal based on uncertain dialogue contexts. The Partially Observable Markov Decision Process (POMDP) framework (Kaelbling et al. (1998)) has been successfully employed in the Spoken Dialogue System (SDS) field (Young et al., 2010; Thomson and Young, 2010; Pinault and Lefèvre, 2011) as well as in human robot interaction (HRI) context (Roy et al., 2000; Lucignano et al., 2013), due to its capacity to explicitly handle parts of the inherent uncertainty of the information which the system has to deal with (e.g. erroneous speech recognizer, falsely recognised gestures, etc.). In this setup, the agent maintains a distribution over possible dialogue states, referred to as the belief state in the literature, and interacts with its perceived environment using a

☆ This paper has been recommended for acceptance by Roger K. Moore.

* Corresponding author. Tel.: +33 490843500.
E-mail addresses: emmanuel.ferreira@univ-avignon.fr (E. Ferreira), fabrice.lefevre@univ-avignon.fr (F. Lefèvre).

http://dx.doi.org/10.1016/j.csl.2015.03.007
0885-2308/

reinforcement learning (RL) algorithm so as to maximise some expected cumulative discounted reward (Sutton and Barto, 1998).

Recent studies in SDS have shown the possibility to learn a dialogue policy from scratch with a limited number (several hundreds) of interactions (Gašić et al., 2010; Sungjin and Eskenazi, 2012; Daubigney et al., 2012) and the potential benefit of this approach compared to the classical use of a Wizard-of-Oz or developing a well-calibrated user simulator (Gašić et al., 2010). Following this idea, sample-efficient learning algorithms, as for instance the Kalman Temporal Differences (KTD) framework (Geist and Pietquin, 2010; Daubigney et al., 2012), can be employed to learn and adapt a system behaviour in an online setup, i.e. while interacting with users. The main shortcoming of this approach is the very poor initial performance. Lowering the length of this warm-up learning phase, defined as the phase when the system can hardly interact with real users due to a high level of exploration and poor performance, is still an open problem when such systems are to be applied to real-world domains. Some solutions can be to introduce some initial expert knowledge (Williams, 2008) or to find ways to collect more hints from the environment which will accelerate the policy learning.

Moreover, problems addressed by RL generally introduce non-stationa-rity at several levels. Indeed, as in many real-world machine learning applications, adaptation to non-stationary environments is a desired feature. In the DM case, users with various levels of expertise (from novice to advanced) and characteristics (restless, bad pronunciations, bad hearing, etc.) can interact with the system. So, the latter should be able to cope with a wide range of behaviours, which may also change over time (switch to new users but also user self-adaptation to the system). Another source of non-stationarity arises when the policy iteration scheme (Sutton and Barto, 1998) is adopted. Policy iteration is an iterative procedure which aims at discovering the optimal policy by generating a sequence of monotonically improving policies. Each iteration consists of two stages: policy evaluation which computes the value function of a given policy and policy improvement which defines the improved policy over the value function. The fact that the value function changes together with the policy makes it non-stationary. In all non-stationary contexts (e.g. environment, optimization method) tracking the value function instead of converging to it seems preferable. A more detailed discussion about the advantages of tracking versus converging, even in stationary environments, can be found in Sutton et al. (2007). Most existing RL algorithms assume stationarity of the problem at hand and aim at converging to a fixed solution. Actually, few attempts to handle non-stationarity can be found in the literature. Among them, we can mention a class of methods which combine RL and planning paradigms such as the Dyna-Q algorithm (Sutton and Barto, 1998).

In most works, the reward function used to learn the dialogue agent is exclusively based on objective features, such as duration and full completion of the user goal. The overall quality of such a function plays a crucial role in finding the optimal solution. However, recent studies have shown that such features, although objective, could not be collected with entire reliability from users (Gašić et al., 2010; Sungjin and Eskenazi, 2012). Anyhow, if the user's point of view is totally ignored or reduced to a rather simple satisfaction questionnaire, naturalness of the overall system can be impacted. In the PARADISE evaluation paradigm (Walker et al., 1997), subjective and objective features are correlated through linear regression. It is worth noting that subjective information is more easily produced by the user. Therefore, it may be interesting to gather some subjective features during the course of the dialogue in order to accelerate the policy learning instead of relying exclusively on an imprecise final appraisal.

This idea could be linked to some works in social and human sciences (e.g. psychology, anthropology) which have shown to which extent acts as simple and spontaneous as facial expressions or gestures can convey social meaning affecting our perception and shaping our daily interactions (Richmond et al., 1991; Kunda, 1999; Custers and Henk, 2005). Also Vinciarelli et al. (2009) present a wide coverage survey of an emerging domain aiming to endow computers with social intelligence abilities. And among these abilities some of the most important are correct perception, accurate interpretation and appropriate generation of social signals. So in the same line of thought, in this proof of concept study, we are focusing on the potential interest of considering a subclass of these social signals with which a user conveys some raw assessments of the current situation during the course of the interaction. Indeed, we claim that positive/negative user appraisals gathered during the course of the dialogue can be used to partially address the two aforementioned bootstrap and tracking problems.

By the fact that user appraisals can be gathered all along the dialogue, we intend to directly exploit them in a socially-inspired diffuse and interim reward function employed in online learning strategy. In that sense, the formulated problem can be closely related to the reward shaping one. In RL, reward shaping consists in supplying meaningful diffuse rewards to a learning agent with the objective to speed up the learning towards the same optimal policy than the one that we could reach with a sparse reward function, giving the meaningful reward only at the end of an episode (e.g. task

completion). For example, El Asri et al. (2013) have investigated how a corpus of evaluated dialogues can be used to estimate a posterior diffuse reward function based on dialogue features representative of the system usability (e.g. dialogue length, task completion) in a way to address the temporal credit assignment problem in an offline setting (i.e. offline batch learning).

Despite some recent attempts to use emotion as a judgement of the task progress with RL (e.g. Broekens and Haazebroek, 2007), little has been done in the goal-oriented DM problem context. Two of the reasons explaining this are that it requires both non-stationary and sample-efficient RL algorithms able to cope with this additional variability and reliable mechanisms to correctly estimate these appraisals from real behavioural cues (such as smiles or nods). Even if the latter point is not addressed (although discussed) in the present study, we propose a potential-based shaping reward method (Ng et al., 1999) to integrate some socially-inspired aspects in the RL scheme in combination with the use of the unified KTD framework (Geist and Pietquin, 2010). The latter expresses the problem of value function approximation as a filtering problem (Kalman filtering). This framework has several advantages and desirable properties for the DM problem (Daubigney et al., 2012). Among others, it is sample-efficient (being based on second-order statistics), it allows online/batch on-policy/off-policy learning, it offers ways to fit the exploration/exploitation dilemma through uncertainty estimation and it supports linear or non-linear parametrisation. Furthermore, as shown in Geist et al. (2009) on toy examples the KTD framework tracks the optimal solution rather than converging to it which is a desirable property for the targeted issues here.

To illustrate the potential benefit of the proposed approach we first carry out a preliminary study on a tourist information retrieval task where a simulation setting is available to statistically validate the impact of adding this additional information to the learning stage. Then, in the context of the MaRDi project,[1] we consider a Pick-Place-Carry HRI task involving real users to obtain the first results in a more realistic setup. More exactly, a 3D simulation software is used where the human can interact with a robot through an avatar involving multimodal dialogues. Although objectively artificial, this platform provides an interesting test-bed module for online dialogue learning. Indeed, a better control over the global experimental conditions can be achieved (e.g. environment instantiation, robot's sensors, etc.). Hence, comparisons between different approaches and configurations are facilitated. Furthermore, this solution reduces the subject recruitment costs without strongly hampering their natural expressiveness (due to the capacities offered by the simulator).

Preliminary results have been already presented in Ferreira and Lefèvre (2013a, 2013b). This extended journal paper offers a unified presentation of the proposed approach along with a direct application on a robotic task with real user trials.

The remainder of the article is organised as follows. In Section 2 some background on the POMDP-based Dialogue Management problem, the RL paradigm and the KTD method are given. Then, in Section 3 the socially-inspired interim reward principle is detailed. Section 4 is dedicated to present the two considered tasks and the experimental setup. Then the following section details and comments on the various evaluation results obtained. Section 6 discusses some considerations relevant to the use of socially-inspired reinforcement, before concluding in Section 7 with some perspectives.

2. Background on machine-learning dialogue management

This section briefly recaps some of the main notions required to follow the novelties proposed in the paper. Readers not familiar with machine-learning approaches for dialogue management are invited to glance at the given references to have a more precise picture of the current state-of-the-art of the field.

2.1. POMDP-based dialogue management

The dialogue management problem has first been described in Levin et al. (1997) as a Markov Decision Process (MDP) to determine an optimal mapping between situations and actions. 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

[1] Man–Robot Dialogue project, funded by the French National Agency for Research.

long term rewards). The environment evolves at each time step t to a state s_t and the agent picks an action a_t according to a policy mapping states to actions, $\pi : S \rightarrow A$. Then state changes to s_{t+1} according to the Markovian transition probability $s_{t+1} \sim T(.|s_t, a_t)$ and, following this, the agent received a reward $r_t = R(s_t, a_t, s_{t+1})$ from the environment. The overall problem of MDP is to derive an optimal policy maximising the reward expectation. Typically the averaged discounted sum over a potentially infinite horizon is used, $\sum_{t=0}^{\infty} \gamma^t r_t$. Thus, for a given policy and start state s, this quantity is called the 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^*. The Q-function may be defined as an alternative to the value function. It adds a degree of freedom on the first selected action, $Q^{\pi}(s, a) = E[\sum_{t \geq 0} \gamma^t r_t | s_0 = s, a_0 = a, \pi] \in \Re^{S \times A}$. As well as 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 Q^*, $\pi^*(s) = arg\max_a Q^*(s, a) \forall s \in S$.

The POMDP framework (Kaelbling et al., 1998), as a generalization of the fully-observable MDP, maintains a belief distribution $b(s)$ over user states, assuming the true one is unobservable. Indeed, POMDP explicitly handles parts of the inherent uncertainty of the DM problem (e.g. word error rate, concept error rate). 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 (i.e. belief state/action spaces). Among other techniques, the HIS model (Young et al., 2010) circumvents the scaling problem for the DM by organising the belief space into partitions, grouping states sharing the same probability, and then mapping the master belief space (partitions) into a much reduced summary space where RL algorithms work reasonably well.

Although variants have been proposed and tested, e.g. Pinault and Lefèvre (2011), HIS remains a reference. However, the choice of a Monte Carlo Control RL algorithm (Sutton and Barto, 1998) is still questioned and recent studies showed the interest of considering sample-efficient algorithms for the DM problem (Gašić et al., 2010; Daubigney et al., 2012). More especially Daubigney et al. (2012) showed that Kalman Temporal Differences (KTD) framework (Geist and Pietquin, 2010) offers a unified framework able to cope with all DM required properties. Indeed, it is sample-efficient, it allows on-policy/off-policy learning through two algorithms (respectively KTD-Q and KTD-SARSA) which can both perform online and offline learning, it provides ways to deal with the "exploration/exploitation" dilemma using uncertainty on value estimates, it allows value tracking, and it supports linear and non-linear parametrisation. Furthermore, KTD algorithms 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.

2.2. Kalman temporal differences framework

The KTD framework (Geist and Pietquin, 2010) is derived from the well-known Kalman filter algorithm (Kalman, 1960) 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.

In the considered linear case, a parametric representation of the Q-function is 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. Notice that a non-linear representation of the Q-function could be employed. However, just the very basic explanations are recalled here, for further details please refer to Geist and Pietquin (2010) and Daubigney et al. (2012). The components of the parameter vector θ are the hidden variables which are modelled as a random vector. Such parameter vector is considered to evolve following a random walk though an 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):

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

Rewards 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. Two algorithms can be defined: KTD-SARSA which denotes the use of the sampled evaluation Bellman equation and KTD-Q, the use of the sampled optimality one.

3. Socially-inspired reinforcement

This section describes the socially-inspired reward principle employed in our proposition and how this kind of reward can be estimated from both simulated and real user appraisals.

3.1. Definition and formalisation

"Social signal" is a generic term which encompasses all signals that convey socially relevant information like (dis-)agreement, empathy, hostility. In human–human interaction they are expressed by means of behavioural cues (e.g. blinks, smiles, crossed arms, laughter, nods and the like). The term socially-inspired RL is employed here to denote a learning process exploiting a subclass of these signals. However this information can be used in multiple ways in the RL scheme (e.g. as part of user state in the dialogue model, as a meta-parameter influencing the exploration/exploitation scheme, or even as part of the system response for an emotional agent). In the present work we exclusively consider this information as a way to gather an additional reinforcement signal for speeding up dialogue policy optimisation in an online learning setting.

So this work focuses on exploiting socially-inspired rewards (also denoted for short as social rewards afterwards) based on positive and negative appraisals emitted by the user and using them as additional interim feedbacks perceived by the system at each dialogue turn. In this scenario these appraisals are considered to be the user's assessments of the interaction evolution and thus implicitly about the overall task progress. Ergo, the social reward itself corresponds to the associated positiveness or negativeness of an appraisal and can be represented as a signed real value.

We propose to use this newly defined social reward function as a shaping reward function. This kind of function is dedicated to help the learning algorithm by giving additional "shapings" to guide it towards a good policy faster. The memoryless shaping reward function, which is one of the most general shaping patterns, is adopted here. So, the overall reward function is the sum of the basic environment reward function R_{env} (objective) and the new social one R_{social} (subjective). The resulting transformed MDP M' is defined by the tuple (S, A, T, γ, R') where R' is the reward function defined as: $R'(s_t, a_t, s_{t+1}) = R_{env}(s_t, a_t, s_{t+1}) + R_{\text{social}}(s_t, a_t, s_{t+1})$ where $R_{\text{social}} : S \times A \times S \rightarrow \Re$ is a bounded real-valued function considered here as the social 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 social shaping reward function R_{social} can guarantee that the optimal policy in M' will be optimal in M? In the case where no further knowledge of the T and R dynamics is available (no expert), a potential-based shaping reward leaves (near-)optimal policies unchanged (Ng et al., 1999). Hence, a potential-based shaping reward function is adopted for R_{social}, corresponding to function F in Ng et al.'s paper, and can be defined as follows:

$$R_{\text{social}}(s_t, a, s_{t+1}) = \gamma\psi(s_{t+1}) - \psi(s_t) \tag{1}$$

where ψ is a real-valued function (denoted shaping potential function) used to evaluate each state and here associated to the user appraisal valence.

3.2. Simulated user appraisals

In a preliminary study to evaluate the impact of social rewards on policy convergence the agenda-based user simulator from Schatzmann et al. (2006) is used wherein the user is simulated at the intentional semantic level (i.e. dialogue act level). This approach factors the user state into an agenda A and a goal G: $S = (A, G)$, where $G = (C, R)$.

The goal G ensures that the simulated user reacts in an appropriate, consistent and goal-oriented manner. It consists of a set of constraints C specifying the required properties that the system should satisfy (they are the objects of the negotiation) and a set of requests R which represent the desired pieces of information (e.g. address, phone number, available schedules). The agenda A is a stack-like structure containing the pending user acts that are deemed necessary to elicit the information specified in the goal. For further details on this simulation method please refer to Schatzmann et al. (2006) and Keizer et al. (2010).

Table 1 gives a sample dialogue for the TownInfo task (Schatzmann et al., 2006) and illustrates how the simulation works and how tracking both the agenda and the goal evolution can serve to give the system some hints on what would be the current user's judgement on the interaction course. For instance, in A_2, the presence of a *negate* act at the top of the agenda means that a user constraint has been violated (here `drinks=beer`) and thus can be perceived as a

Table 1
Sample dialogue for the TownInfo task and its corresponding agenda sequence.

$C_0 = \begin{bmatrix} \text{type=bar} \\ \text{drinks=beer} \\ \text{area=east} \end{bmatrix}$

$R_0 = \begin{bmatrix} \text{name=} \\ \text{addr=} \\ \text{phone=} \end{bmatrix}$

Sys_1 Hello, how may I help you ?

$A_1 = \begin{bmatrix} \text{inform(type=bar)} \\ \text{inform(drinks=beer)} \\ \text{inform(area=east)} \\ \text{request(name)} \\ \text{request(addr)} \\ \text{request(phone)} \\ \text{bye()} \end{bmatrix}$

Usr_1 I'm looking for a nice bar serving beer

Sys_2 Ok, a wine bar. What part of town do you have in mind?

$A_2 = \begin{bmatrix} \text{negate(drinks=beer)} \\ \text{inform(area=east)} \\ \text{request(name)} \\ \text{request(addr)} \\ \text{request(phone)} \\ \text{bye()} \end{bmatrix}$

Usr_2 No, beer please !

Sys_3 A bar serving beer, correct?

$A_3 = \begin{bmatrix} \text{affirm()} \\ \text{inform(area=east)} \\ \text{request(name)} \\ \text{request(addr)} \\ \text{request(phone)} \\ \text{bye()} \end{bmatrix}$

Usr_3 Yes something in the east part of town please !

Sys_4 There is no bar serving beer in the east part of town. But the Kings Shilling, near the Castle, is a very nice bar serving beer.

$C_4 = \begin{bmatrix} \text{type=bar} \\ \text{drinks=beer} \end{bmatrix}$

$R_4 = \begin{bmatrix} \text{name=Kings Shilling} \\ \text{addr=Castle Loop} \\ \text{phone=} \end{bmatrix}$

$A_4 = \begin{bmatrix} \text{request(phone)} \\ \text{bye()} \end{bmatrix}$

Usr_4 Ok, That is just the way it is. What's the phone number?

Sys_5 Could you please rephrase that?

$A_5 = \begin{bmatrix} \text{request(phone)} \\ \text{bye()} \end{bmatrix}$

Usr_5 What's the phone number ?!

Sys_6 The number is728-64-32.

$R_6 = \begin{bmatrix} \text{name = Kings Shilling} \\ \text{addr = Castle Loop} \\ \text{phone=728-64-32} \end{bmatrix}$

$A_6 = \begin{bmatrix} \text{bye()} \end{bmatrix}$

Usr_6 Thanks, goodbye!

Table 2
List of positive and negative cues collected from agenda and goal.

Positive cues		Negative cues	
1	Positive top dialogue act type (affirm, confirm, etc.)	1	Negative top dialogue act type (negate, deny, etc.)
2	Number of slots filled	2	Agenda size
3	Partial completion flag	3	Dialogue length
4	Final completion flag	4	Top agenda act contains already transmitted item

negative hint. In the same way, the `affirm` act in A_3 underlines a positive situation. Hence the nature of the top dialogue act of the agenda can be considered as a useful cue to determine the valence of the user's appraisal with realistic outcomes (i.e. correctly reflecting what a real user appraisals would be in such situations).

Table 2 presents some simple positive and negative cues extracted from the agenda and goal internal structures in the user simulator during dialogue simulations. The underlying assumption is that combinations of such kind of cues might result in extracting some situation where a user is prone to generate some form of positive/negative appraisal to the system such as shouting "no no no that's wrong" and grimacing in a real setup. Hence, each of them is weighted in order to give more or less emphasis on specific features in the socially-inspired reward computation. Thus, the considered social signals are directly encoded into the reward function rather than in the simulated user behaviours. Although a continuous scale is possible, a five-point agreement scale (Likert scale) is adopted here for ψ with regard to the way subjective measures are gathered in PARADISE (Walker et al., 1997). Each level is associated with a representative real number associated with an agreement scale, from strongly negative (−−) to strongly positive (++). So, after a

Table 3
Example of social reward computation during simulation.

s_t	Positive cues	Negative cues	$\psi(s_t)$	R_{social}
s_3	1(1)	2(−6), 3(−4)	0.5	0.45
s_4	2(2/3), 3(1)	2(−2), 3(−5)	1	

normalisation step the sum of all the weighted cues gives an overall score C_{s_t} which is rescaled on a five-point Likert scale using a threshold ξ. Thus, at each time step t, a potential-based social shaping reward is computed using Eq. (1) and ψ function:

$$\psi(s) = \begin{cases} -1, & \text{if } C_s < -\xi & (--) \\ -0.5, & \text{if } -\xi \leq C_s < 0 & (-) \\ 0, & \text{if } C_s = 0 & (\text{neutral}) \\ 0.5, & \text{if } 0 < C_s \leq \xi & (+) \\ 1, & \text{if } C_s > \xi & (++) \end{cases}$$

The process of social reinforcement reward computation can be decomposed into two steps:

1. gathering of positive and negative cues from the factored user state;
2. estimate of the social reward using the potential-based shaping reward function.

An example of such a process is summarised in Table 3. The first column represents the analysed user state s_t (i.e. the corresponding agenda A_t and goal G_t in Table 1). The second and the third columns are respectively the lists of positive and negative cues which have been detected (using the id from Table 2) and their associated value in brackets. For example, in the first row and third column, cue 2 corresponds to the number of items in the agenda and the value 6 is extracted from A_3, the minus sign merely indicates the negativeness of the cue. The fourth column corresponds to the ψ value (i.e. the Likert score). It is computed applying some weights on the detected cue values. As an illustration, for the negative cue 3, 1/30 is chosen as weight because the maximum number of turns allowed by the system is 30. Consequently, 1/30 can be viewed as a normalisation value. Indeed, the objective at hand is to range each cue value in the interval $[-1, 1]$ according to their valence and their relative importance. It is important to notice that such normalisation weights have been determined following some expert intuitions and are not considered as optimal. They have been chosen to deliver an average user appraisal of the dialogue progress. Different user profiles can be designed by varying these weights as well as the global patience (i.e. tolerance of inconsistent system behaviours) and initiative (i.e. number of conveyed pieces of information) levels of the simulated user to study to what extent user appraisals can help user adaptation capacities of a learning agent. The last column shows the resulting social reward, applying Eq. (1) with $\gamma = 0.95$. The positive score 0.45 denotes a quite favourable evolution between s_3 and s_4. To compete with the environment reward the social reward can be rescaled using a weight coefficient before being added to R_{env} in R'.

3.3. Real user appraisals

In real applications user appraisals must be detected from observable behavioural cues employed by the user to convey her judgement on her own state evolution. This could be done using several multimodal social detectors (e.g. emotion face tracking, gesture classification, social keyword spotting, etc.). These latter may produce a set of positive and negatives cues for each detector. For instance, the face tracker may produce a cue dedicated to smile detection the value of which is the probability of its inner model consisting in a positive cue. Then, a similar weighted interpolation mechanism as the one presented in Section 3.2 could be used to infer $\psi(s)$ from the valued cues output by the various detectors. This could be done in a more principled and data-driven way than the previously described handcrafted settings, for instance using regression methods exploiting an annotated corpus comparable to Rieser and Lemon (2011) or El Asri et al. (2013) for reward estimation. However, here we intend to test the benefits of using socially-inspired rewards related to the user appraisal. And as such we mean to be able to control the level of uncertainty

Table 4
Example of a multimodal dialogue in the MaRDi task.

R1	DA	hello()
	NLG/TTS	How can I help you ?
U1	ASR	Can you put the book in my bedroom?
	SLU	inform(action=move,desc=in,room=bedroom)
R2	DA	confreq(type=book,position)
	NLG/TTS	Sorry, but where is the book you are talking about?
U3	ASR	I am talking about this one
	SLU	inform(idobj=?)
	GRU	pointsAt BLUE_BOOK 1395848705.31
R3	DA	execute(action=move,destination=bedroom_bedsidetable, idobj=BLUE_BOOK,position=livingroom_table,type=book, colour=blue)
	NVBP/MC	move(BLUE_BOOK,livingroom_table,bedroom_bedsidetable)
	NLG/TTS	Ok, I will put the blue book on your bedside table

of the social cue detection in the system and not to depend on the intrinsic quality of the detection components. So, a simple workaround used in the HRI user trials consists in asking the user to make explicit her appraisal through an interface which allows her to rate the current state evolution towards the task completion after each transition on a five-point agreement scale which is directly considered as the ψ value.

4. Experimental setup

This section describes the two considered dialogue tasks as well as the experimental setup employed for simulated and real user trials.

4.1. TownInfo and MaRDi dialogue tasks

In this study two versions of a HIS-based dialogue system are considered. The first one is dedicated to the TownInfo task (Young et al., 2010) in the tourist information domain. A user wants to obtain some information (address, phone number) about a particular venue located in a virtual town based on some constraints such as its type, area, served food, price range, etc. This system has already been tested with real users in Schatzmann et al. (2006), and in a more recent and matured version, called CamInfo (Cambridge tourist information), in Gašić et al. (2010). In our experiments only the simulated version of the task will be used, where simulated user appraisal cues are obtained as described in Section 3.2. Thanks to an already available user simulator this task offers a convenient way for a preliminary evaluation and tuning of the proposed approaches.

The second version, the MaRDi Dialogue System which is the true goal of our work, is designed to solve a Pick-Place-Carry task in an HRI context. In this setup, the robot and the user share a three rooms flat environment, in which there are different kinds of objects varying in terms of colour, type, and position (e.g. a blue mug on the kitchen table, a red book on the living room table). The user interacts with the robot using unconstrained speech (large vocabulary speech recognition) and pointing gestures to ask the robot to perform some specific object manipulation tasks (e.g. "move the blue mug from the living room table to the kitchen table"). A multimodal dialogue is used to solve ambiguities and to request missing information until task completion (i.e. full command execution) or failure (i.e. explicit user disengagement or wrong command execution). An example of such an interaction is given in Table 4.

In our experimental setup, the MaRDi multimodal dialogue system is tightly coupled with a 3D simulation software. Here, the open-source robotics simulator MORSE (Echeverria et al., 2011) is used. This tool provides a realistic rendering (see Fig. 1) through the Blender Game Engine, supports a wide range of middleware (e.g. ROS, YARP), and proposes reliable implementations of realistic sensors and actuators which ease the integration on real robotic platforms afterwards. It also provides the operator with an immersive control (see Fig. 2) of a virtual human avatar in terms of displacement, gaze and interactions with the environment, such as object manipulation (e.g. grasp/release of an object).

As shown in Fig. 3, 12 components are involved in the overall functioning of the MaRDi system. The four orange ones are those implicated in the user's input management, speech and gesture modalities in our case. The combination

Fig. 1. MaRDi simulated environment.

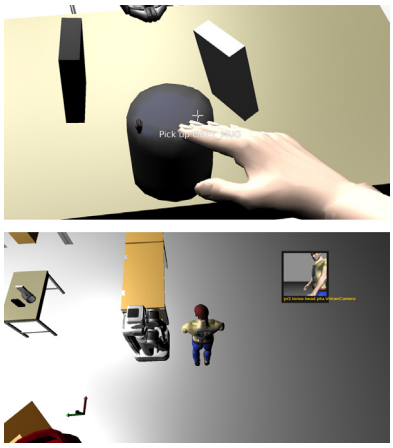

Fig. 2. Human avatar in the first (top) and the third person perspective (bottom).

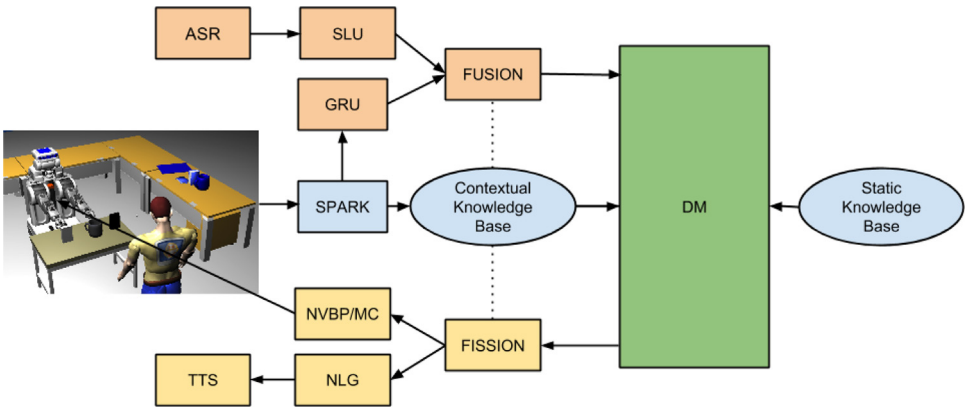

Fig. 3. Architecture of the multimodal and situated dialogue system. (For interpretation of the references to colour in this figure citation, the reader is referred to the web version of this article.)

of the Google Web Speech API[2] for Automatic Speech Recognition (ASR) and a custom-defined grammar parser for Spoken Language Understanding (SLU) is used to perform speech recognition and understanding. The Gesture Recognition and Understanding (GRU) module simply catches the gesture-events generated by the spatial reasoner during the course of the interaction. Then, the Fusion module temporally aligns the monomodal inputs and then merges them with custom-defined rules. Finally, the result of the fusion (i.e. N-best list of interpretation hypotheses and their related confidence scores) becomes the input of the multimodal DM.

The three blue components are responsible of the context modelling. SPARK (Milliez et al., 2014) both detects user gestures and generates the per-agent spatial facts (perspective taking) which are used to dynamically feed the Contextual Knowledge Base. Taking into account the environment during the course of the interaction makes it a situated system. These two modules are responsible of the per-agent knowledge modelling which allows the robot to reason over different perspectives on the world, not only its own but also these of the users. Furthermore, a Static Knowledge Base containing the list of all available objects (even those not perceived) and their related static properties (e.g. colour) is used. More details on these aspects, for instance the notion of perspective taking, and their influence on the dialogue system can be found in Milliez et al. (2014) and Ferreira et al. (2015).

The four yellow components are dedicated to the output restitution. The Fission module splits the abstract system action into verbal and non-verbal actions. The spoken output is produced by chaining a template-based Natural Language Generation (NLG) module with a Text-To-Speech Synthesis (TTS) component based on the commercial Acapela TTS system.[3] The Non-verbal Behaviour Planning and Motor Control (NVBP/MC) module produces arm gestures and head and body poses for the robot by translating the non-verbal action into a sequence of abstract actions such as *grasp*, *moveTo*, *release*.

Finally, the green component is the DM, responsible for updating the internal belief state and making the next robot decision.

4.2. User simulation and real user trials

To assess the performance of introducing socially-inspired information as an additional reinforcement signal, the off-policy KTD-Q algorithm (denoted BASELINE) is employed as our baseline due to its high performance in the conditions at hand (Daubigney et al., 2012). A setup close to the one described in Daubigney et al. (2012) is adopted here. Thus, the Q-function is parametrised using linear-based Radial Basis Function (RBF) networks, one per action and the Bonus-Greedy scheme (Daubigney et al., 2011) is adopted in training conditions, with $\beta = 1000$ and $\beta_0 = 100$. β and β_0 are meta parameters used to scale the variance-based bonus added to the Q-value estimates to deal with the exploration/exploitation dilemma. The discount factor γ is set to 0.95 in all experiments.

In this study, we distinguish two kinds of conditions to present our results. Firstly, the online training conditions, also denoted as *controlled case*, where the results are gathered with a continuously improving policy (and thus also exploring policy). Secondly, the testing conditions, denoted as *no control*, where an already trained policy acts greedily (without exploration ability and no additive bonus) to conduct the interactions.

For the TownInfo task, a user simulator is employed and set to interact with the DM at a 10% concept error rate. The weight coefficient used to rescale the social reward is set empirically (grid search) to 4 to compete against the basic environment rewards. Similarly, ξ is set to 0.3 in all experiments. As presented in Section 3.2 all individual cues are manually weighted by following expert intuitions according to their nature and their considered importance in the appraisal valence determination of the targeted user profiles. However the weights assigned to each cue presented in Table 2 have not been tested individually during the evaluation. Hence, these handcrafted parameters are not considered as optimal or specifically tuned for the task at hand.

Considering MaRDi, as already mentioned, some trials have been carried out with real users in a 3D virtual environment. At the beginning of each dialogue, a specific goal (a command with arguments in our case) is randomly generated taking into account the simulated environment settings and the current interaction history in order to select a possible command to perform. For example, “You want the robot to give you the white book on the kitchen table”. After each completed interaction the users evaluated the system in terms of task completion. In case of socially-inspired

[2] https://www.google.com/intl/en/chrome/demos/speech.html.
[3] http://www.acapela-group.com/index.html.

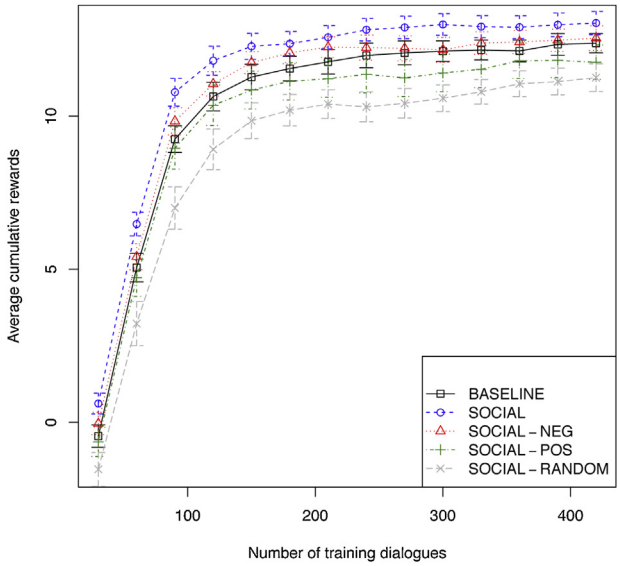

Fig. 4. Results of 4 different configurations of the social-shaped KTD-Q algorithm compared to KTD-Q baseline during the learning of the policy (controlled case).

policy learning, throughout the course of the interaction the user is instructed to rate the current state of the dialogue. This is done using a five-star rating bar, accessible on the graphical interface. All the dialogues were recorded both in terms of audio and various kinds of meta-information (e.g. ASR N-Best list, detected gestures and related timestamps, etc.) but also high level annotations (e.g. environment settings, pursued goal, task success). In the full 3D-simulated multimodal architecture, it has been observed that each interaction takes from 7 to 10 min to be achieved (object detections, robot movements and displacements, etc.). So, without loss of generality, a practical workaround to speed up the testing process consisted in using a fixed representation of the scene (a screenshot from the human point of view) and a web-based multimodal GUI instead of the fully-operational simulation setup. As for TownInfo, the weight coefficient use to rescale the social reward is set to 4.

The authors consider that the average cumulative environment rewards can be a sufficient metric to compare the different approaches. This is explained by the fact that in the environment reward function the success (full user goal completion) is rewarded by a +20 bonus and failure and elapsed time (turn) respectively punished by 0 and −1. For comparison purposes all the experiments with social rewards presented in our plots are given in terms of the environment reward, R_{env}, only (since we try to converge towards to same optimal policy, see Section 3.1).

5. Experiments and results

5.1. Online policy optimisation using social reinforcement learning

In this section the benefits of adding social rewards for optimizing the DM policy are first evaluated on the simulated TownInfo task, and then with real users on the MaRDi task.

5.1.1. TownInfo

Influence of social reward type: We consider several socially-inspired reinforcement configurations which take into account different kinds of cues for the social reward computation. The classic approach denoted SOCIAL considers both the negative and the positive cues, as described in Section 3.2. Results are shown in Fig. 4 in terms of cumulative discounted environment rewards gathered during the learning stage of the policy (controlled case) when exploration is possible. For these curves, each point is an average of 50 independent learnings with a sliding window of 100 points width.

Only the first 500 dialogues are considered here because we intend to focus on the early stage of the training for which system performance is critical. It can be observed that SOCIAL slightly outperforms BASELINE in terms of

both the final learned performance, which is better of about 0.5 turn on average, and the learning time to achieve a similar performance level, which is reduced. For example, the performance obtained performing 200 dialogues with the BASELINE algorithm are already reached with about 100 dialogues using SOCIAL. A comparison between three other configurations of the simulated social rewards is also made. The first (SOCIAL-NEG) and the second (SOCIAL-POS) configurations are respectively using only the negative or positive cues. The third configuration is a randomized social reward generator (SOCIAL RANDOM). As expected, SOCIAL-RANDOM is the worst, followed by SOCIAL-POS, BASELINE and SOCIAL-NEG. SOCIAL which combines both positive and negative cues obtains the best results. All configurations (except SOCIAL-RANDOM) are rather close if we consider the confidence radius of their results. However an important point is that even in the case of the use of random (non-informative) social rewards, the potential-based technique ensures that convergence to the near-optimal policy is still preserved. From this experiment it seems that the convergence is better guided by negative information which is an interesting finding considering that negative emotions might be easier to emit and detect in a real setup.

Influence of noise in rewards: We evaluated the impact of noise on the proposed optimization procedure. Noise robustness is studied in terms of Concept Error Rate (CER), Environment and Social Reward Error Rates, noted respectively $R_{env}ER$, $R_{soc}ER$. Although the previous experiment has shown encouraging results when socially-inspired reinforcement is considered, it should be kept in mind that in the previous conditions social rewards are perfectly perceived by the learning agent. In a more realistic setup like user trials such signals, due to their inherent complexity (e.g. multimodal aspects, context-dependent interpretation) cannot be perfectly estimated. This difficulty is introduced in the simulation by means of an artificial $R_{soc}ER$. At a given rate the cues are randomly modified to the inverse of what they should be. In the same way, when online learning is adopted the user should mark the overall dialogue in terms of task completion (objective metric). But, as shown in Gašić et al. (2010), the feedback given by a real user can be erroneous. This will be reflected by the $R_{env}ER$ in our experiments. At a certain rate the final evaluation of dialogue success (correct or not) is inverted. Wrong feedbacks can be explained by the subjectivity of the task. Although the goal is correctly achieved any inconsistent behaviour of the system during the dialogue can drive the user to penalise the system at the end. Another explanation comes from the fact that a trial user is not really committed to the task; if the system fails to fulfil its goal there is no consequence for her or if the system asks for to remove a constraint the user has no personal rationale to guide her behaviour in the negotiation. In any case, the quality of the reward function is crucial for the RL algorithms as the speed of convergence to the optimal policy relies on it. In addition, the presence of a high CER level also has a negative influence when this additional difficulty is present from the beginning of the learning (no progressive degradation).

We compared seven conditions: BASELINE and BASELINE-10$R_{env}ER$, SOCIAL, SOCIAL-10$R_{env}ER$, SOCIAL-10$R_{soc}ER$ and SOCIAL-10$R_{env}ER$-10$R_{soc}ER$. 10XER means that the corresponding error rate X is set to 10%. Results are shown in Fig. 5 in terms of cumulative rewards with respect to different CER levels. For these curves, each point is an average made over the results obtained using 50 policies learned with 400 dialogues and then tested with 1000 dialogues. In the latter test setup, the next action is chosen greedily with respect to the learnt Q-function (no exploration). Considering only the BASELINE and BASELINE-10$R_{env}ER$ the influence of CER and $R_{env}ER$ can be easily identified. Thus, as the $R_{env}ER$ and the CER increase the overall performance decreases. The BASELINE performance can be compared to both SOCIAL and SOCIAL-10$R_{soc}ER$ since they are facing the same learning conditions except for the use of social rewards (noisy or not). We observe that the two considered social methods achieve better performance than the baseline one for all CER levels. However, we can only state this for SOCIAL when the confidence radius of the results are considered (except at 20% of CER level where we can state this for the two social methods). Furthermore the observed drop in terms of performance between 10 and 40% CER is less important considering SOCIAL (-3) than BASELINE (-4).

Similarly BASELINE-10$R_{env}ER$ can be compared to both SOCIAL-10R_{env} ER and SOCIAL-10$R_{env}ER$-10$R_{soc}ER$. We also observe that the two considered social methods achieve better performance than the baseline one for all CER levels. But here we can state this at 20 and 40% CER levels for the two social methods. Hence in all the presented conditions the use of a social-based reinforcement has a positive impact on the performance of the KTD-Q algorithm. Thus, interim user guidance improves the ability to defer the impact of noise in terms of both CER and $R_{env}ER$ even when social reward is itself subject to noise. One of the reasons accounting for this is that social rewards are gathered all along the dialogue and offer a granular form of the reward function. So, in case of the user giving an erroneous final reward, collected positive and negative social rewards can counterbalance this mistake (as a hint of the overall user satisfaction). Furthermore, in case of high CER, social rewards can favour or penalize a system's local behaviour despite

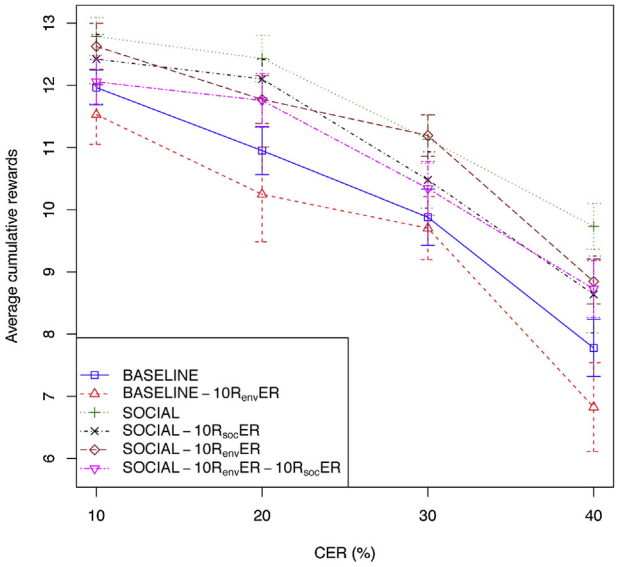

Fig. 5. Results of baseline and social-shaped KTD-Q algorithms in different noise conditions (no control) for TownInfo.

Table 5
Results of KTD-Q algorithm at 20% CER and 10% $R_{env}ER$ using different $R_{soc}ER$ levels (no control) for TownInfo.

Use social	$R_{soc}ER$	Rewards	Success rate
No	–	10.24 (±0.76)	91.14 (±1.58)
Yes	0	11.77 (±0.38)	93.42 (±0.80)
Yes	10	11.75 (±0.43)	93.73 (±0.58)
Yes	20	11.28 (±0.45)	92.53 (±0.88)
Yes	30	10.80 (±0.42)	91.68 (±1.10)
Yes	40	10.67 (±0.43)	91.33 (±1.01)
Yes	50	10.06 (±0.71)	89.34 (±3.70)

the overall task failure (or success). However, the benefit of socially-inspired reinforcement tends to decrease as the $R_{soc}ER$ increases. Therefore, in order to study the impact of $R_{soc}ER$ alone, Table 5 is populated with the results obtained with different $R_{soc}ER$ levels at 20% CER and 10% $R_{env}ER$, both corresponding to realistic values for field trials. Above 30% $R_{soc}ER$, taking into account social rewards seems to be unnecessary or even disadvantageous. Actually, even if the results obtained with 40% $R_{soc}ER$ are slightly better than those obtained with the baseline, they do not converge as quickly (e.g. considering 200 training dialogues the baseline outperforms the social version).

It is worth noting that $R_{env}ER$ and $R_{soc}ER$ are simulated with no specific prior assumption. Indeed a rather simple random error approach is used. In a more sophisticated framework, the error models could be learnt from data.

5.1.2. MaRDi

As in the previous setting the BASELINE performance is compared to the SOCIAL one in the MaRDi task. Results are shown in Fig. 6 in terms of cumulative environment rewards gathered during the learning stage of the policy (controlled case). For these curves, each point is an average of the performance obtained during the learning using a sliding window of 50 points width.

Here 100 training dialogues are considered for both the BASELINE and SOCIAL methods. The same expert user carried out the two policy learning and the same sequence of goals was employed to favour the comparison between the two learning curves. The WER and the CER are estimated at around 10% (after manual annotation of 50 turns).

Thus, as shown in the figure, the SOCIAL performance clearly outperforms BASELINE during the overall learning process. Nonetheless, the gap is clearly reduced at the end of the learning stage. These two observations are in line with the previous simulated experiments. It is worth noting that the high level of performance reached by the two methods

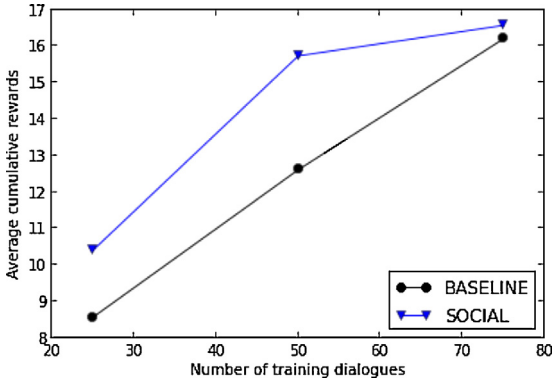

Fig. 6. Results of baseline and social-shaped KTD-Q algorithms for MaRDi (controlled case).

during this learning stage, compared to the previous setup, is partially explained by the fact that the constraints to define the command to execute are less numerous than in TownInfo, and thus dialogues are generally less ambiguous.

The learnt policy obtained considering additional social rewards has been tested by 6 distinct new subjects performing 75 dialogues in total. In this context we obtained an average of 16.1 in terms of cumulative environment rewards in test settings (no control). The estimated WER is about 10%, conform to the learning stage, but the CER is a bit higher, around 15%. The performance levels obtained in these conditions are slightly lower than those obtained during the learning stage. It can be explained by the fact that users are not aware of the system understanding ability and some tips to complete tasks. Anyhow, these results enable us to consider that a system slightly trained by a single expert user is able to cope reasonably well with new users. More importantly it is still possible to pursue the policy learning to adapt the system and to fit the new observed behaviours as shown in the next section.

5.2. *Adaptation to user profiles*

In this section the benefits of using socially-inspired reinforcement signals is evaluated for policy adaptation to user's profiles.

5.2.1. *TownInfo*

First we present the results obtained with a modified version of the goal and agenda-based user simulator (see Section 3.2), in which two types of users are simulated through different simulation settings i.e. different levels of initiative and patience. The first corresponds to a novice user and is set to be patient and to have few initiatives (as a consequence it provides information to the system one at a time and wait for instructions). The second one is the advanced/expert user. He acts in a impatient manner and has a high initiative level. As aforementioned in Section 4.2, we adapt the simulated socially-inspired reward computation process for each defined profile. For example, the dialogue length is a more important negative cue for the expert user than for the novice one since the expert want to complete the task in the fastest way. In the considered experimental setup, the KTD-Q tracking performance is compared to its social-based version with possible noise in social rewards.

The experiments can be broken down into two main steps. In the first step, the KTD-Q algorithm is employed to learn two distinct kinds of policies by interacting either with the "expert" or the "novice" simulated user. Due to the fast convergence of KTD-Q (sample-efficiency) only 400 training dialogues are sufficient to learn a user-dependent near-optimal policy. In the second step, a new batch of 400 dialogues is performed by interacting exclusively with the expert user for all the settings.

Four scenarios are identified:

- EXPERT+EXPERT: for policies previously learnt by interacting with the simulated expert user, these new interactions just pursued the previous learning (no change in behaviour). So, the resulting performance corresponds to a near-optimal mapping in the given expert user context.

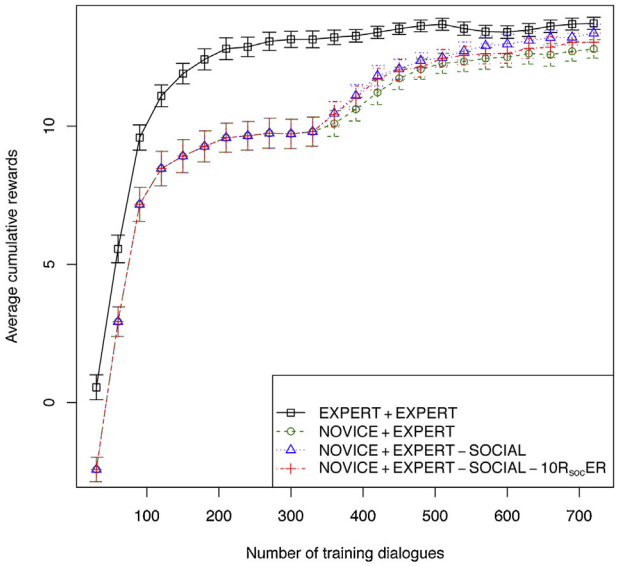

Fig. 7. Results (rewards) of KTD-Q with and without social reward shaping and with various simulated user profiles (controlled case) for TownInfo.

- NOVICE+EXPERT: the second scenario differs from the previous one by the fact that a change in the environment behaviour is considered. Indeed in this scenario, as in the following scenarios, the policies previously learnt by interacting with the simulated novice user are considered. This scenario constitutes the baseline method.
- NOVICE+EXPERT-SOCIAL: the third scenario aims to introduce social rewards with the expectation that they can help the learning agent to improve the user tracking.
- NOVICE+EXPERT-SOCIAL-10$R_{soc}ER$: the last experiment is derived from the third scenario except the fact that social rewards are scrambled with a 10% error rate. This consideration is justified by the fact that in real world application the considered cues (multimodal aspects, context-dependent interpretations, etc.) can not be perfectly estimated.

Comparisons are shown in terms of both average cumulative discounted environment rewards (Fig. 7) and success rates (Fig. 8). Each point in the plots is the result of an average made over a sliding window of 100 points width. These results are analysed as follows. The first 400 dialogues are used to explain the impact of user profiles in the learning process (i.e. the difference between interacting with an expert and a novice user), the next 400 (from 400 to 800) illustrate the tracking ability with and without social rewards, in both a noise-free setup and a 10% $R_{soc}ER$.

In this first experiment, only the first 400 training dialogues are considered. Results presented in Fig. 7 show that better environment rewards are obtained by interacting with the expert user rather than the novice (+3% on average). It is explained by the fact that an expert user acts in a turn efficient way (wrt number of pieces of information conveyed to the system in a single turn). However, in terms of success rate (see Fig. 8) the results are quite similar (so they both reach their goal even though at a different pace). Nevertheless a slightly faster convergence is observed when the novice user interacts with the learning agent (+2% on average around 100 dialogues). An explanation for this phenomenon is that the expert user is less patient than the novice, so the system has to deal with more dialogue failures. Accordingly, it seems easier to learn with a more flexible user than a demanding one.

For the remaining dialogues (400–800), an overall drop in performance is identified in terms of success rate in Fig. 8. Indeed, when some changes occur in the environment conditions (e.g. simulated user settings) the learnt policy does not fit well with the new environment dynamics (new user behaviours). Then, this drop is followed by an adaptation phase that brings back the overall performance close to the reference curve (EXPERT+EXPERT). Concerning performance in terms of average cumulative rewards, Fig. 7 shows that all considered curves are close to the reference. No such drop like the one observed in term of success rate is visible on these curves because expert users perform more efficiently than the novice ones. Indeed, they take less turns to complete their dialogue tasks and thus they gather a larger positive cumulative environment reward at the end of each successful task (less turn penalties). Hence, the loss in terms of

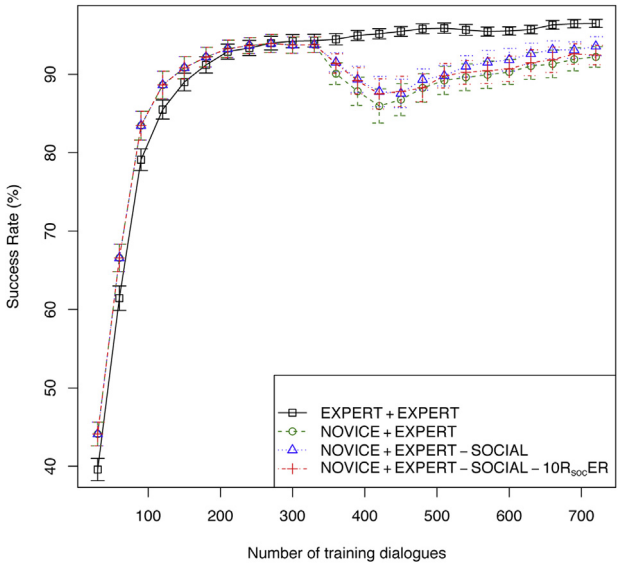

Fig. 8. Results (success rates %) of KTD-Q with and without social reward shaping and with various simulated user profiles (controlled case) for TownInfo.

success rate is somewhat hidden. As can be observed, when socially-inspired reinforcement is employed, the tracking is slightly more efficient than in the baseline. Indeed, the loss in terms of success rate is not as important as when the cues are not involved (+1.5% on average). Even more, considering both the success rate adaptation phase and the average cumulative rewards, the performance grows a little faster than what is observed in the baseline case. These results may be explained by the fact that social rewards are gathered all along the dialogue and offer a reward function with a good granularity (even more in the context of a TD-based RL algorithm). These rewards can favour or penalize a specific system behaviour (state-action reaction) on a local basis despite the overall task failure or success and thus rapidly compensate for the loss in performance due to the apparition of a new behaviour.

The aim of this last scenario is to ensure the capacity of such social-based user tracking to show improved robustness to noisy conditions. The experimental conditions are very close to those described previously. The $R_{soc}ER$ is set to 10%, so 10% of the time ψ can take every Likert score but the current true estimated one. Results obtained show that noisy social rewards still slightly improve over the baseline in terms of both average discounted cumulative rewards and success rate (resp. +0.4 and +0.8% on average). At the beginning of the adaptation, the loss in terms of success rate is not as strong as in the baseline and quite similar to the performance obtained with the noise-free social version of KTD-Q. However, for the remainder of the adaptation procedure, the noise-free method performs better than the noisy one. The benefit of socially-inspired reinforcement seems to decrease when the $R_{soc}ER$ grows. However, the convergence to the near-optimal policy is still preserved in case of very noisy social rewards (high $R_{soc}ER$) thanks to the potential-based technique which limits their effect to a close surrounding of their time occurrences.

5.2.2. *MaRDi*

For the MaRDi task 30 adaptation dialogues are considered for both the BASELINE and SOCIAL methods. In these two settings, the initial policy consists in the one learnt performing 100 dialogues using the BASELINE method (cf. Section 5.1.2). The same expert user performed the policy adaptation and the same sequence of goals was employed to favour the comparison between the two learning curves. In order to ensure that the interacting user makes use of distinct behaviours she was given some constraints regarding her dialogue strategy. As an example, the user was asked not to make use of some properties (colour, location, etc.) as long as they were not explicitly required by the robot and also to act globally in an impatient and obstinate way. For the SOCIAL method we also asked the user to rate the system accordingly.

Fig. 9 plots the results in terms of average cumulative rewards obtained during the adaptation stage of the policy (controlled case) for MaRDi. For these curves, each point is an average of the performance obtained during the learning using a sliding window of 12 points width. The SOCIAL methods seems to handle partly the user change and succeed

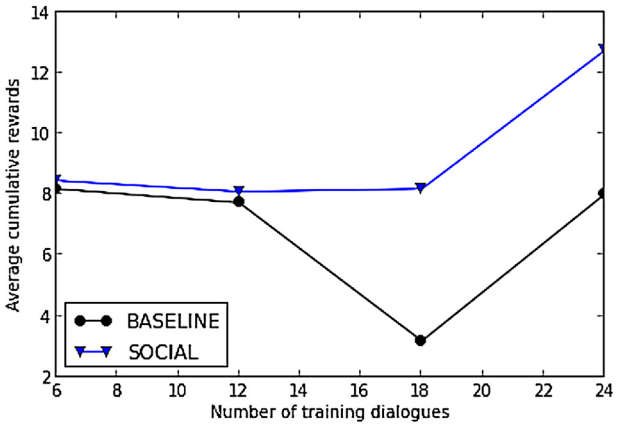

Fig. 9. Results of baseline and social-shaped KTD-Q algorithms for MaRDi (controlled case).

to learn the new behaviours efficiently (several tens of dialogues only). Indeed, less costly exploration strategies were needed to cope with the new user behaviours. Hence, these results confirm those observed in the simulated setup.

6. Discussions

In this "proof of concept" study both a simulation setup and user trials have been employed. However, user appraisals are either simulated or gathered using an explicit method which cannot be integrated straightforwardly in a more realistic on-board robotic system. Thus, mechanisms to extract correctly these appraisals through multimodal cues from real users have to be envisaged as for instance what is done in the INTERSPEECH Computational Paralinguistics Challenge (Schuller et al., 2012). Even if the capacity of these methods remains highly imperfect if these cues are gathered in an unconstrained and implicit manner (Vinciarelli et al., 2009), the experiments presented in Section 5.1 show that we can detect them with a certain level of uncertainty without jeopardizing the merits of the proposed method. Furthermore, in case of conflicting socially-inspired and environmental rewards (subjective nature of user appraisals), the potential-based reward shaping technique prevents a too strong hurt of performance in case of ill-fitted additive reward signals. Moreover, we have shown that this problem can be simplified if we consider an interaction with a cooperative and rational "seed user" (e.g. a system designer) which rates the system along the course of the interaction. A similar setting can be obtained if we restrict ourselves to a limited set of non-verbal cues (e.g. head gesture, intonation) in order to accelerate the learning process.

As flagged out in Section 3.3 one can consider a more principled and data-driven approach than the handcrafted settings depicted in Section 3.2 to deal with more sophisticated cues. For example, regression methods as the one employed in Rieser and Lemon (2011) which makes use of a modified version of the PARADISE framework Walker (2000) to learn the reward model that optimizes the task easiness criterion (determined through the answers to a user questionnaire filled at the end of each dialogue) from various input variables extracted from annotated Wizard-of-Oz data (e.g. dialogue length or task completion), via stepwise regression. In the present case the data collection effort can be shared between several tasks since the social rewards are designed to express the user's judgement about the dialogue progress which is not specially a task-dependent criterion.

Another important point is that considering the social information allows a more granular view of the reward function rather than a unique judgement at the end of the episode. It can help to avoid or strengthen some local system behaviours and can be employed to better handle the user adaptation problem because the reinforcement is not only based on objective measures but also on user-dependent subjective cues. Thus, when sample-efficient algorithms are considered the approach can be viewed as a way to avoid the need for a user simulator as illustrated in the MaRDi task. Such a setup can be assimilated to active learning like what is done in Doshi and Roy (2008) and also linked to imitation-based (Price and Boutilier, 2003) or inverse approaches to RL as in Chandramohan et al. (2011).

The choice of KTD in this work does not prevent the use of social reinforcement learning for other similar RL algorithms e.g. GPTD (Gašić et al., 2010). Indeed the potential-based method presented above is amenable to all the other frameworks.

7. Conclusion

This paper has described a method using socially-inspired rewards in RL to train a dialogue policy from scratch in just a small number of dialogues and that improves the baseline performance in terms of rapidity of convergence. It also highlighted how socially-inspired rewards can be used to better fit the user adaptation issue for dialogue management. The approach shows better robustness to noisy conditions in terms of semantic input error rate and environment reward error rate. The presented method also has interesting properties that guarantee the optimality when socially-inspired rewards are merged into an additional reinforcement learning signal using a potential-based shaping reward function to introduce the detected cues as additional reinforcement signals.

In the present work, if both simulation and user trials are considered, interactions take place in a simulated 3D environment where the user appraisal acquisition is simplified. Anyhow, on-board real settings involving more complicated scenario trials and social signal computation through multimodal cues where data driven approach are employed to estimate socially-inspired rewards are planned to uphold our claims in a realistic HRI configuration.

Acknowledgements

The authors would like to thank the Cambridge University Dialogue Systems Group for providing the TownInfo HIS System, as well as Lucie Daubigney, Olivier Pietquin and the MALIS Supélec-Metz Group for their help in using the KTD Framework. This work is partially funded by the ANR MaRDi project (ANR CONTINT 2012 ANR-12-CORD-0021).

References

Broekens, J., Haazebroek, P., 2007. Emotion and reinforcement: affective facial expressions facilitate robot learning. In: Artificial Intelligence for Human Computing, pp. 113–132.

Chandramohan, S., Geist, M., Lefèvre, F., Pietquin, O., 2011. User simulation in dialogue systems using inverse reinforcement learning. In: Interspeech.

Custers, R., Henk, A., 2005. Positive affect as implicit motivator: on the nonconscious operation of behavioral goals. Pers. Soc. Psychol. 89, 129–142.

Daubigney, L., Gašić, M., Chandramohan, S., Geist, M., Pietquin, O., Young, S., 2011. Uncertainty management for on-line optimisation of a POMDP-based large-scale spoken dialogue system. In: Interspeech.

Daubigney, L., Geist, M., Chandramohan, S., Pietquin, O., 2012. A comprehensive reinforcement learning framework for dialogue management optimization. Sel. Top. Signal Process. 6, 891–902.

Doshi, F., Roy, N., 2008. Spoken language interaction with model uncertainty: an adaptive human–robot interaction system. Connect. Sci. 20, 299–318.

Echeverria, G., Lassabe, N., Degroote, A., Lemaignan, S., 2011. Modular open robots simulation engine: Morse. In: ICRA.

El Asri, L., Laroche, R., Pietquin, O., 2013. Reward shaping for statistical optimisation of dialogue management. In: SLSP.

Ferreira, E., Lefèvre, F., 2013a. On the use of social signal for reward shaping in reinforcement learning for dialogue management. In: SemDial.

Ferreira, E., Lefèvre, F., 2013b. Social signal and user adaptation in reinforcement learning-based dialogue management. In: MLIS.

Ferreira, E., Milliez, G., Lefèvre, F., Alami, R., 2015. Users' belief awareness in reinforcement learning-based situated human–robot dialogue management. In: IWSDS, (in press).

Gašić, M., Jurčíček, F., Keizer, S., Mairesse, F., Thomson, B., Yu, K., Young, S., 2010. Gaussian processes for fast policy optimisation of POMDP-based dialogue managers. In: SIGDIAL.

Geist, M., Pietquin, O., 2010. Kalman temporal differences. Artif. Intell. Res. 39, 483–532.

Geist, M., Pietquin, O., Fricout, G., 2009. Tracking in reinforcement learning. In: ICONIP.

Kaelbling, L., Littman, M., Cassandra, A., 1998. Planning and acting in partially observable stochastic domains. Artif. Intell. J. 101, 99–134.

Kalman, R., 1960. A new approach to linear filtering and prediction problems. Basic Eng. 82, 35–45.

Keizer, S., Gašić, M., Jurčíček, F., Mairesse, F., Thomson, B., Yu, K., Young, S., 2010. Parameter estimation for agenda-based user simulation. In: SIGDIAL.

Kunda, Z., 1999. Social Cognition: Making Sense of People. MIT Press.

Levin, E., Pieraccini, R., Eckert, W., 1997. Learning dialogue strategies within the Markov decision process framework. In: ASRU.

Lucignano, L., Cutugno, F., Rossi, S., Finzi, A., 2013. A dialogue system for multimodal human–robot interaction. In: ICMI.

Milliez, G., Ferreira, E., Fiore, M., Alami, R., Lefèvre, F., 2014. Simulating human–robot interactions for dialogue strategy learning. In: SIMPAR.

Milliez, G., Warnier, M., Clodic, A., Alami, R., 2014. A framework for endowing interactive robot with reasoning capabilities about perspective-taking and belief management. In: RO-MAN.

Ng, A., Harada, D., Russell, S., 1999. Policy invariance under reward transformations: theory and application to reward shaping. In: ICML.

Pinault, F., Lefèvre, F., 2011. Unsupervised clustering of probability distributions of semantic graphs for POMDP based spoken dialogue systems with summary space. In: KRPDS.

Price, B., Boutilier, C., 2003. A Bayesian approach to imitation in reinforcement learning. In: IJCAI.
Richmond, V.P., McCroskey, J.C., Payne, S.K., 1991. Nonverbal Behavior in Interpersonal Relations. Prentice Hall, Englewood Cliffs, NJ.
Rieser, V., Lemon, O., 2011. Learning and evaluation of dialogue strategies for new applications: empirical methods for optimization from small data sets. Comput. Linguist. 37, 153–196.
Roy, N., Pineau, J., Thrun, S., 2000. Spoken dialogue management using probabilistic reasoning. In: ACL.
Schatzmann, J., Weilhammer, K., Stuttle, M., Young, S., 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. Knowl. Eng. Rev. 21, 97–126.
Schuller, B., Steidl, S., Batliner, A., Burkhardt, F., Devillers, L., Mller, C., Narayanan, S., 2012. Paralinguistics in speech and language – state-of-the-art and the challenge. Comput. Speech Lang., 4–39, Special Issue on "Paralinguistics in Naturalistic Speech and Language".
Sungjin, L., Eskenazi, M., 2012. Incremental sparse Bayesian method for online dialog strategy learning. Sel. Top. Signal Process. 6, 903–916.
Sutton, R., Barto, A., 1998. Reinforcement learning: an introduction. IEEE Trans. Neural Netw. 9, 1054.
Sutton, R., Koop, A., Silver, D., 2007. On the role of tracking in stationary environments. In: ICML.
Thomson, B., Young, S., 2010. Bayesian update of dialogue state: a POMDP framework for spoken dialogue systems. Comput. Speech Lang. 24, 562–588.
Vinciarelli, A., Pantic, M., Bourlard, H., 2009. Social signal processing: survey of an emerging domain. Image Vis. Comput. 27, 1743–1759.
Walker, M., Litman, D., Kamm, C., Abella, A., 1997. Paradise: a framework for evaluating spoken dialogue agents. In: ACL.
Walker, M.A., 2000. An application of reinforcement learning to dialogue strategy selection in a spoken dialogue system for email. Artif. Intell. Res. 12, 387–416.
Williams, J.D., 2008. Integrating expert knowledge into POMDP optimization for spoken dialog systems. In: Proceedings of the AAAI-08 Workshop on Advancements in POMDP Solvers.
Young, S., Gašić, M., Keizer, S., Mairesse, F., Schatzmann, J., Thomson, B., Yu, K., 2010. The hidden information state model: a practical framework for POMDP-based spoken dialogue management. Comput. Speech Lang. 24, 150–174.