Learning to ground in spoken dialogue systems

Olivier Pietquin

LEARNING TO GROUND IN SPOKEN DIALOGUE SYSTEMS

Olivier Pietquin

École Supérieure d'Électricité - SUPÉLEC
Metz Campus - IMS Team
2 rue Édouard Belin - F-57070 Metz - FRANCE
e-mail: olivier.pietquin@supelec.fr

ABSTRACT

Machine learning methods such as *reinforcement learning* applied to dialogue strategy optimization has become a leading subject of researches since the mid 90's. Indeed, the great variability of factors to take into account makes the design of a spoken dialogue system a tailoring task and reusability of previous work is very difficult. Yet, techniques such as reinforcement learning are very demanding in training data while obtaining a substantial amount of data in the particular case of spoken dialogues is time-consuming and therefore expansive. In order to expand existing data sets, dialogue simulation techniques are becoming a standard solution.

In this paper, we present a user model for realistic spoken dialogue simulation and a method for using this model so as to simulate the *grounding* process. This allows including grounding subdialogues as actions in the reinforcement learning process and learning adapted strategy.

Index Terms— Speech Communication, Unsupervised Learning, User Modelling

1. INTRODUCTION

During the last decade, research in the field of Spoken Dialogue Systems (SDS) has experienced increasing growth. The design of an efficient SDS does not basically consist in combining speech and language processing systems such as Automatic Speech Recognition (ASR) and Text-to-Speech (TTS) synthesis systems. It requires the development of an interaction management strategy taking at least into account the performances of these subsystems (and others), the nature of the task (i.e. form filling or database querying) and the user's behavior (i.e. cooperativeness, expertise, general knowledge). The great variability of these factors makes rapid design of dialogue strategies and reusability across tasks of previous work very complex. For these reasons, automatic learning of optimal strategies is currently a leading domain of researches [1][2][3][4]. Yet, the low amount of data generally available for learning and testing dialogue strategies does not contain enough information to explore the whole space of dialogue states (and of strategies). Simulation is most often required to expand the existing dataset and man-machine spoken dialogue stochastic modelling and simulation has become a research field by its own right [2][3][4][5][6][7].

Among simulation methods presented in the literature, one can distinguish between state-transition methods like proposed in [2] and methods based on modular simulation environments as described in [4][5][6][7]. The first type of methods is very task-dependent as well as the hybrid method proposed in [3]. Moreover, these methods are not really able to expand the dataset to unseen similar examples. Therefore, strategy learning can only lead to the learning of the best strategy used in the data corpus which is not always optimal. The second type of methods intends to be more task-independent by integrating models of each component of a SDS including the speech processing systems but also the user as shown on figure 1. One crucial point is of course the user modelling method. There exists several examples in the literature and most of them are stochastic [8][9][10][11] but it is still an open area of researches.

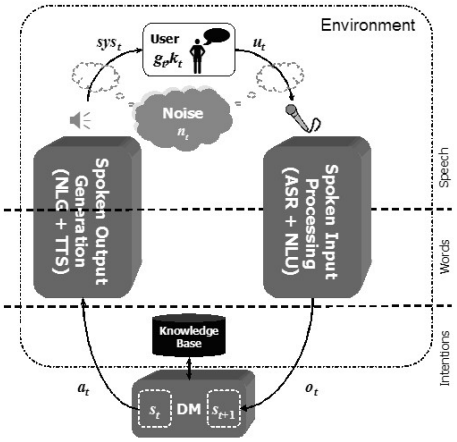

Fig. 1. Spoken Dialogue Model

In this paper we present a user model based on previous work [4][10] but we emphasize on the use of this model to simulate *grounding* [12]. In the following, grounding will be regarded as the process used by dialogue participants to ensure that they share the background knowledge necessary for the understanding of what will be said later in the dialogue. Particularly, we will use this user model in the purpose of learning to identify situations in which grounding can be necessary in average.

2. FORMALIZING DIALOGUE

2.1. Dialogue as a turn taking process

As depicted on figure 1, a task-oriented man-machine dialogue can be seen as a turn-taking process in which a human user and a Dialogue Manager (DM) exchange information through different channels processing speech inputs and outputs (ASR, TTS,...). At each

Part of this work has been realized when the author was with the Faculty of Engineering, Mons (FPMs, Belgium) and was funded by the DGTRE, the Walloon Region and the SIMILAR European Network of Excellence.

turn t the DM chooses an action a_t according to its internal state s_t and its strategy π_t so as to accomplish the task it has been designed for. These actions can be greetings, spoken utterances (i.e. constraining questions, confirmations, relaxation, data presentation), database queries, dialogue closure etc. They result in a response from the DM environment (i.e. user speech input, database records), considered as an observation o_t, which usually leads to a DM internal state update ($s_t \rightarrow s_{t+1}$). During the interaction, the DM action has been transformed in synthesized speech sys_t if needed. This acoustic signal is mixed up with noise n_t before reaching the user. According to his/her goal (g_t), knowledge (k_t) and understanding of sys_t, the user will produce a spoken utterance u_t also mixed up with noise before reaching the ASR system.

2.2. Intention-based Communication

In our vision of a simulated environment, communication between modules takes place at the intention level rather than at the word sequence or speech signal level (like proposed in [6]). An intention is regarded as the minimal unit of information that a dialogue participant can express independently. Indeed, concept-based communication allows error modelling of all the parts of the system, including natural language understanding [13]. Pragmatically, it is easier to automatically generate concepts compared with word sequences (and certainly speech signals), as a large number of utterances can express the same intention.

2.3. Markov Decision Processes

In the (discrete) Markov Decision Processes (MDP) formalism, a stochastic system is described by a finite number of states $\{s_i\}$ and an action set $\{a_j\}$. To each state-action pair is associated a transition probability $\mathcal{T}^a_{ss'}$ giving the probability of stepping from state s at turn t to state s' at turn $t+1$ after having performed action a when in state s. To this transition is also associated a reinforcement signal (or reward) r_{t+1} describing how good was the result of action a when performed in state s. Formally, an MDP is thus completely defined by a 4-tuple $\{S, A, \mathcal{T}, \mathcal{R}\}$ where S is the state space, A is the action set, $\mathcal{T}$ is a transition probability distribution over the state space and $\mathcal{R}$ is the expected reward distribution. The couple $\{\mathcal{T}, \mathcal{R}\}$ defines the dynamics of the system:

$$\mathcal{T}^a_{ss'} = P(s_{t+1} = s' | a_t = a, s_t = s) \quad (1)$$

$$\mathcal{R}^a_{ss'} = E^{\pi}(r_{t+1} | a_t = a, s_t = s, s_{t+1} = s') \quad (2)$$

These last expressions assume that the Markov property is met, which means that the system's functioning is fully defined by its one-step dynamics and that the functioning from state s will be identical whatever the path followed until s. This assumption can be met by including historical information into the state representation. To control a system described as an MDP, one would need a strategy or policy π mapping states to actions: $\pi(s) = P(a|s)$ (or $\pi(s) = a$ if the strategy is deterministic). In this framework, a RL agent aims at optimally mapping states to actions, that is finding the best strategy π^* so as to maximize an overall reward R which is a function (most often a weighted sum) of all the immediate rewards r_t. If the probabilities of equations 1 and 2 are known, an analytical solution can be computed [14], otherwise the system has to learn the optimal strategy by a trial-and-error process. In the most challenging cases, actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards. Different techniques are described in the literature, in section 4 the Watkin's $Q(\lambda)$ algorithm [14] will be used.

2.4. Dialogue as an MDP

In the context of SDS, the DM strategy has to be optimized, thus the DM will be the learning agent. The environment modelled by the MDP comprises everything but the DM: the human user, the communication channels (ASR, TTS), and any external information source (i.e. database, sensors). To fit to the MDP formalism, a reinforcement signal r_{t+1} is required. In [2] it is proposed to use the contribution of an action to the user's satisfaction. Although this seems very subjective, some studies have shown that such a reward could be approximated by a linear combination of objective measures such as the duration of the dialogue (D), the ASR performances (ASR) or the task completion (TC) [15]:

$$r_{t+1} = w_D.D_t - w_{ASR}.ASR_t - w_{TC}.TC_t, \quad (3)$$

where the w_x are positive weights

2.5. Stochastic Modelling of Spoken Dialogue

The user model proposed in this paper is based on a simplified version of the statistical description of man-machine spoken communication described in [4] and [7]. Referring to section 2.1, the interaction can be described by the following probability:

$$P(s_{t+1}, o_t, a_t | s_t, n_t) = \underbrace{P(s_{t+1}|o_t, a_t, s_t, n_t)}_{Task\ Model} . \underbrace{P(o_t|a_t, s_t, n_t)}_{Environment} . \underbrace{P(a_t|s_t, n_t)}_{DM} \quad (4)$$

The factorization of this joint probability includes a term related to the environment processing of the DM intention set (second term). Omitting the t indices, this term can in turn be factored as follow:

$$P(o|a, s, n) = \sum_{sys,k,g,u} P(o, sys, k, g, u | a, s, n).$$

$$= \sum_{sys,k,g,u} \underbrace{P(sys|a, s, n)}_{Output\ Processing} . \underbrace{P(o|u, g, sys, a, s, n)}_{Input\ Processing} . \underbrace{P(u, g, k|sys, a, s, n)}_{User\ Model} \quad (5)$$

3. USER MODELLING

3.1. Probabilistic Model

From the previous section, the user behavior can be probabilistically described by the following probability:

$$P(u, g, k|sys, a, s, n) = \underbrace{P(k|sys, s, n)}_{Knowledge\ Update} . \underbrace{P(g|k)}_{Goal\ Modification} . \underbrace{P(u|g, k, sys, n)}_{User\ Output} \quad (6)$$

To obtain the last equality, the following assumptions were made:

- the user is only informed of the DM intentions a through the system utterance sys,
- if a goal modification occurs it is because the user's knowledge has been updated by the last system utterance.

Equation 6 emphasizes on the tight relation existing between the user's utterance production process and his/her goal and knowledge, themselves linked together. The user's knowledge can be modified during the interaction according to the system's speech outputs. Yet, such a modification of the knowledge is incremental (it is an update to compare with the system state update) and it takes into account the last system utterance (which might be misunderstood, and especially in presence of noise) and the previous user's knowledge state. This can be written as follow with k^- standing for k_{t-1}:

$$\begin{aligned} P(k|sys,s,n) &= \sum_{k^-} P(k|k^-,sys,s,n).P(k^-|sys,s,n) \\ &= \sum_{k^-} P(k|k^-,sys,n).P(k^-|s) \end{aligned} \quad (7)$$

Although the user's knowledge k^- is not directly dependent of the system state s, we kept this dependency in our description so as to be able to introduce a mechanism for user knowledge inference from system state because it is supposed to contain information about the history of the dialogue.

It is this mechanism that we will use in the following to introduce *grounding* [12] subdialogs in the interaction so as to obtain a good connection between the user's understanding of the interaction and the system view of the same interaction.

3.2. Variable Representation

In practice, the use of the proposed framework is difficult without a suitable representation of variables such as u, sys, g or k. According to the intention-based communication paradigm (section 2.2), these variables can be regarded as finite sets of abstract concepts, related to the specific task, that have to be manipulated along the interactions by the SDS and the user. Consequently, we opted for an *Attribute-Value* (AV) pair variable representation based on the *Attribute-Value Matrix* (AVM) representation of the task proposed in [15]. Each communicative act is then symbolized by a set of AV pairs. From now on, we will denote $\mathcal{A}$ the set of possible attributes (concepts) according to the task, and by $\mathcal{V}$ the set of all possible values. The system utterances sys are then modelled as sets of AV pairs in which the attribute set will be denoted $Sys = \{sys^\sigma\} \subset \mathcal{A}$ and the set of possible values for each attribute sys^σ will be denoted $V^\sigma = \{v_i^\sigma\} \subset \mathcal{V}$. The user's utterance u is modelled as a set of AV pairs in which attributes belong to $U = \{u^\upsilon\} \subset \mathcal{A}$ and the set of possible values for u^υ is $V^\upsilon = \{v_i^\upsilon\} \subset \mathcal{V}$. The ASR and NLU processes (Input processing) result in an error-prone set of AV pairs c which is part of the observation o. The user's goal $G = \{[g^\gamma, gv_i^\gamma]\}$ and the user's knowledge $K = \{[k^\kappa, kv_i^\kappa]\}$ are also AV pair sets where g^γ and k^κ are attributes and where gv_i^γ and kv_i^κ are values.

3.3. Grounding Reinforcement Signal

In section 2.4, we defined the general equation of the reinforcement signal (eq. 3) for dialogue strategy learning. In this paper we argue that introducing a new term to this general equation to introduce penalties due to grounding problems will lead to a more appropriate learned strategy:

$$r_{t+1} = w_D.D_t - w_{ASR}.ASR_t - w_{TC}.TC_t - w_G.G_t, \quad (8)$$

were G_t is the new cost related to grounding problems. If we denote s_t^h (resp. k_t^h) the vectors containing the historical information of the DM state representation (resp. the user's knowledge representation), this cost is computed as a distance between both vectors ($d(s_t^h, k_t^h)$). G_t can thus be regarded as a distance between the DM knowledge about the interaction and the user's one.

4. EXAMPLE

In this section we will consider a very simple application simulating an automatic train ticket booking system. The task will consist in filling a 5-slot form which slots are: departure city, arrival city, desired departure time, desired arrival time, class.

For the application we consider here, the AVM representation of the task is obtained by associating an attribute to each of the 5 slots. Than, to each intention or dialogue act corresponds a set of AV pairs. For instance, "departure city", "arrival city" are attributes and possible values are "Namur", "Brussels", "Paris", ... The utterance "I want to go from Namur to Brussels" can therefore be represented by the following set of AV pairs:

$$u_t = [\{dep\_city = "Namur"\}, \{arr\_city = "Bruxelles"\}] \quad (9)$$

We used 50 possible values for the cities, 48 values for the times (every half of an hour) and 2 values for the class (economy and business). So as to model the system and user's utterances, we added attributes to this description. The first is the type of system utterance (S_A in the following, $Sys \supset S_A$), which can take the following values: 'Greeting', 'Constraining Question', 'Open Question', 'Confirmation', 'Relaxation request', 'Closing'. The second is a binary attribute corresponding to the user's will of closing the dialogue ($U_C \in \{true, false\}, U \supset U_C$). Finally we also added attributes associated to user's answers to confirmation and relaxation prompts taking Boolean values. The system utterances are therefore of the form: $sys = \{[S_A = "const\_q"], [s^1 = "dep\_city"]\}$ or $sys = \{const\_q(dep\_city)\}$ in the following. The user's utterances are of the form $u = \{[U_C = false], [dep\_city = "Bruxelles"]\}$.

Figure 2 shows the task structure, the user's goal structure (AV pairs) and the knowledge structure which will be simply a set of counters associated to each goal AV pair and incremented each time the user answers to a question related to the corresponding attribute. The vector composed of these counters is the k^h vector mentioned in section 3.3. The task completion (used to compute the r_t signal) is measured as a ratio between the common values in the goal and the values retrieved by the system after the dialogue session.

Task	
Attributes ($\mathcal{A}$)	$\#\mathcal{V}$
dep	50
dest	50
t_dep	48
t_dest	48
class	2

User Goal (G)	
Att.	Value
g^{dep}	Glasgow
g^{dest}	Edinburgh
g^{t_dep}	8
g^{t_dest}	12
g^{class}	1

Knowledge (K)	
Count	init
k^{dep}	0
k^{dest}	0
k^{t_dep}	0
k^{t_dest}	0
k^{class}	0

Fig. 2. AVM description of the Task

The RL paradigm requires the definition of a state space. It will be defined by a set of state variables : 5 Booleans (one for each attribute in the task) set to $true$ when the corresponding value is known, 5 status Booleans set to $true$ if the corresponding value is confirmed and 5 binary values indicating whether the ASR confidence level associated to the corresponding value is $high$ or low (the ASR process is simulated as in [4]). The first 5 Booleans will compose the s^h vector mentioned in section 3.3. The G term of equation 8 is then computed thanks to a distance $d(s^h, k^h)$ which will be a simple edit distance between the 2 vectors (the cost of a substitution

is 1). The DM will be allowed 6 action types: greeting (greet), open question (openQ), closed question (closedQ), explicit confirmation (expC), grounding subdialog (ground), closing (close). A grounding subdialog will be initiated by the DM by showing the information included in (s^h) and ask confirmation about it to the user. If no confirmation is provided by the user, a subdialog is started to end up with an equality between s^h and k^h.

Two experiments have been realized. During the first experiment, the reinforcement signal is given by 3 and the grounding action is not included in the DM action set while it is included in the second experiment during which the reinforcement signal is given by 8. The results of the learning process tested on 10^5 dialogs shown in figure 3 can be interpreted as follow. In the first experiment, the system reaches an acceptable task completion rate of 81% in more than 9 turns in average. The majority of the turns is used for confirmations because the system is often unsure about the retrieved information. So, to reach the maximum TC rate, it asks almost systematically for confirmation after each turn. Moreover, the number of open-ended questions is quite low. In the second experiment, to reach a similar TC rate, the explicit confirmation number is much lower and it has been replaced by 1.3 grounding subdialogue turns. This reduces the average number of turns because the user has not to confirm correct information and because the systems uses more open-ended questions and therefore gathers more information in one turn. We compare experiments according to their performance in terms of number of turns and task completion rate and not according to the average return since both experiments use different reinforcement signals.

Performance		
	N	*TC*
Exp 1	9.39	0.81
Exp 2	7.60	0.82

Strategy						
	greet	constQ	openQ	expC	ground	close
Exp 1	1.0	1.85	1.23	4.31	0	1.0
Exp 2	1.0	1.25	1.85	1.20	1.30	1.0

Fig. 3. Performance of the learned strategy in terms of average number of turns for a dialogue (N) task completion (TC) and relative frequency of each action type

5. CONCLUSIONS AND PERSPECTIVES

In this paper we presented a user model enabling to simulate the grounding process. We used this model to train a reinforcement learning algorithm so as to learn an optimal dialogue strategy. We introduced a grounding cost in the reinforcement signal and a "grounding" action in the learning agent action set and showed that a strategy using this action could be learned and produced acceptable results.

This is a very preliminary work and several criticisms can be opposed to it. First, the system doesn't infer grounding problems from the interaction and it has a direct access to the user's knowledge representation which would not be possible in a real case. According to equation 7, this could be done within the proposed framework. Second, the grounding process is global here and is not specialized to particular values that could be identified as problematic. To address these two problems, the grounding information should be inserted in the learner state space as well (not only in the reinforcement signal). Yet, the purpose of this experiment is to show that grounding can be taken into account in machine learning for dialogue management. One can oppose the fact that eq. 3 already takes into account dialogue duration is enough to ensure confirmation dialogues to be avoided when non necessary. Yet, introducing grounding into the reinforcement signal leads to additional subdialogs only if grounding problems were identify while previous learned strategies inferred the necessity of a confirmation subdialog from the configuration of informational state variables such as the confidence levels. Such a configuration does not always lead to the occurrence of grounding problems and sometimes confirmation subdialogs can be avoided.

6. REFERENCES

[1] E. Levin, R. Pieraccini, and W. Eckert, "Learning dialogue strategies within the markov decision process framework," in *Proc. ASRU'97*, December 1997.

[2] S. Singh, M. Kearns, D. Litman, and M. Walker, "Reinforcement learning for spoken dialogue systems," in *Proc. NIPS'99*, 1999.

[3] K. Scheffler and S. Young, "Corpus-based dialogue simulation for automatic strategy learning and evaluation," in *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, 2001.

[4] O. Pietquin and T. Dutoit, "A probabilistic framework for dialog simulation and optimal strategy learning," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 2, pp. 589–599, March 2006.

[5] E. Levin, R. Pieraccini, and W. Eckert, "A stochastic model of human-machine interaction for learning dialog strategies," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 11–23, 2000.

[6] R. López-Cózar, A. de la Torre, J. Segura, and A. Rubio, "Assesment of dialogue systems by means of a new simulation technique," *Speech Communication*, vol. 40, no. 3, pp. 387–407, May 2003.

[7] O. Pietquin, "A probabilistic description of man-machine spoken communication," in *Proc. ICME'05*, July 2005.

[8] W. Eckert, E. Levin, and R. Pieraccini, "User modeling for spoken dialogue system evaluation," in *Proc. ASRU'97*, December 1997.

[9] J. Schatzmann, K. Georgila, and S. Young, "Quantitative evaluation of user simulation techniques for spoken dialogue systems," in *Proc. SIGdial'05*, September 2005.

[10] O. Pietquin, "Consistent goal-directed user model for realisitc man-machine task-oriented spoken dialogue simulation," in *Proc. ICME'06*, July 2006.

[11] K. Georgila, J. Henderson, and O. Lemon, "User simulation for spoken dialogue systems: Learning and evaluation," in *Proc. Interspeech'06*, September 2006.

[12] H. Clarck and E. Schaefer, "Contributing to discourse," *Cognitive Science*, vol. 13, pp. 259–294, 1989.

[13] O. Pietquin and T. Dutoit, "Dynamic bayesian networks for nlu simulation with applications to dialog optimal strategy learning," in *Proc. ICASSP'06*, May 2006.

[14] R.S. Sutton and A.G. Barto, *Reinforcement Learning : An Introduction*, MIT Press, ISBN : 0-262-19398-1, 1998.

[15] M. Walker, D. Litman, C. Kamm, and A. Abella, "Paradise: A framework for evaluating spoken dialogue agents," in *Proc. of the 35th Annual Meeting of the Association for Computational Linguistics*, 1997, pp. 271–280.