DYNAMIC BAYESIAN NETWORKS FOR NLU SIMULATION WITH APPLICATIONS TO DIALOG OPTIMAL STRATEGY LEARNING

Olivier Pietquin[†]

École Supérieure d'Électricité (Supélec)
Metz Campus – STS Team
2, rue Édouard Belin
F-57070 Metz – FRANCE
olivier.pietquin@supelec.fr

Thierry Dutoit

Faculty of Engineering, Mons (FPMs)
TCTS Lab.
Avenue Copernic, 1
B-7000 Mons – BELGIUM
thierry.dutoit@fpms.ac.be

ABSTRACT

In this paper, we propose to add a model for NLU-related error generation in a modular environment for computer-based simulation of man-machine spoken dialogs. This model is jointly designed with a user model. Both of them are based on the same underlying Bayesian Network used with different parameters in such a way that it can generate a consistent user behavior, according to a goal and the interaction history, and been used as a concept classifier. The proposed simulation environment was used to train a reinforcement-learning algorithm on a simple form-filling task and the results of this experiment show that the addition of the NLU model helps pointing out problematic situations that may occur because of misunderstandings and modifying the dialog strategy accordingly.

1. INTRODUCTION

Today, man-machine spoken dialog simulation for optimal strategy learning is a real field of research which was instigated in the mid 90's. Systems developed so far used either a state-transition model [1] or a modular simulation environment reproducing the user behavior [2] and sometimes Automatic Speech Recognition (ASR) errors [3]. Yet, no real attempt to explicitly model Natural Language Understanding (NLU) errors in such a modular environment exists in the literature. Although state-transition models and to some extent the hybrid environment of [4] implicitly take NLU errors into account, they use very task-dependent techniques. We describe here a more complete modular simulation environment including models for a consistent goal-directed User Model (UM) and task-independent ASR and NLU.

The field of NLU can not be compared to the one of ASR in which there exist standard implementations (like the GMM/HMM or the hybrid ANN/HMM approaches). This allows taking those implementations as a basis for ASR simulation. At the opposite, lots of different implementations of NLU systems for Spoken Dialog Systems (SDS) can be found in the literature such as grammar-based techniques [5], HMM-based approaches [6] and more recently, systems based on belief networks [7]. So, it is difficult to model NLU thanks to a general implementation method. Yet modeling NLU errors is a crucial issue for task-independent optimal dialog strategy learning.

The main idea proposed in this paper is to develop a generative model for NLU simulation which would be similar to the user model. Indeed, if we are able to capture the user behavior in such a statistical model, this model should be also usable for concept classification after the generated sentence was possibly modified by the ASR error model. In this purpose, the Bayesian Networks paradigm seemed to be the most appropriate as it provides generative models suitable for classification as well. It also allows introducing *a priori* knowledge about the task which is very helpful when designing a SDS. Finally, the graphical representation is also very convenient

2. HUMAN-MACHINE DIALOG MODEL

Formally speaking, a Human-Machine Dialog (HMD) can be regarded as a sequential turn-taking process in which a user and a Dialog Management (DM) system interact.

This interaction takes place through speech and language processing sub-systems (Figure 1) of two main types: input processing modules (such as ASR and NLU sub-systems) and output generation modules (such as Natural Language Generation (NLG) and Text-To-Speech (TTS) synthesis sub-systems). As, the purpose of a SDS is often to provide information to a user, the DM is usually connected to an external information source, the Knowledge Base (KB). As shown on Figure 1, in our view, everything but the DM is part of the environment which has to be simulated for strategy learning.
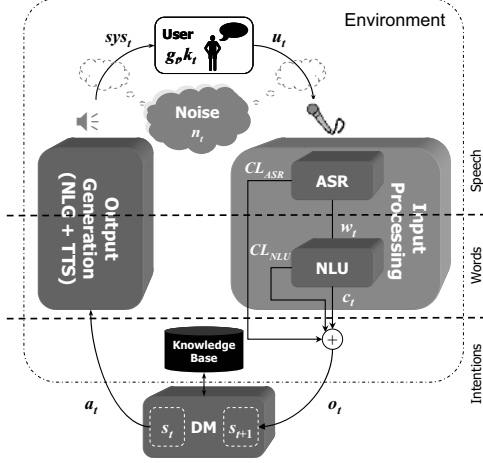

Figure 1: Man-machine spoken communication process

The process being sequential, time will be discretized in turns t. At each turn, a set of communicative acts a_t (or action) is generated by the DM according the current dialog state s_t. The signal a_t is transmitted to the user thanks to the output generation sub-systems translating the communicative acts into a synthesized spoken utterance sys_t either thanks to NLG and TTS modules or to human-authored prompts or even pre-recorded prompts. According to the communicative acts s/he could extract from sys_t, to his/her knowledge k_t at time t (possibly updated according to sys_t) and to the goal g_t s/he tries to achieve while interacting with the system, the human user produces an utterance u_t. Both the user's and the system utterances are mixed with noise n_t. The ASR sub-system subsequently processes the user's utterance to produce a word sequence w_t, which is in turn processed by the NLU module providing a set of inferred concepts c_t. Both the ASR and the NLU sub-systems also generate metrics (respectively CL_{ASR} and CL_{NLU}) indicating the confidence of those modules in their processing. The combination of the metrics and the concept set builds the observation o_t, which can be considered as the result of the processing of the DM communicative acts by its environment. It is eventually used by the DM to update its internal state s_{t+1}.

2.1. Attribute-value variable representation

It is quite unclear how to model each variable present in this description (such as u_t, sys_t, g_t etc.) for computer-based HMD simulation. In the field of SDS design, it is often argued that *intention-based* communication is sufficient to internally model dialogs. Variables can then 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. For this reason, we opted for a variable representation based on Attribute-Value (AV) pairs. This representation allows very high-level considerations (attributes are regarded as concepts) while values (particular values for the concepts) allow to some extent to come back to lower levels of communication.

This variable description is founded on an Attribute-Value-Matrix (AVM) representation of the task [8].

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 modeled as sets of AV pairs in which the attribute set will be denoted $\boldsymbol{Sys}=\{sys^\sigma\} \subset \mathcal{A}$ and the set of possible values for each attribute sys^σ will be denoted $\boldsymbol{V^\sigma} = \{v_i^\sigma\} \subset \mathcal{V}$. The system utterance attribute set contains a special attribute A_S which values define the type of the embedded act. Allowed types can be constraining questions, relaxing prompts, greeting prompts, assertions, confirmation queries, etc. The user's utterance u is modeled as a set of AV pairs (transmitted to the ASR model) in which attributes belong to $\boldsymbol{U} = \{u^\upsilon\} \subset \mathcal{A}$ and the set of possible values for u^υ is $\boldsymbol{V^\upsilon} = \{v_i^\upsilon\} \subset \mathcal{V}$. The user's utterance attribute set contains a special attribute $\boldsymbol{C_U}$ which value is a Boolean indicating whether the user wants to close the dialog or not. The ASR process results in an error-prone set of AV pairs w which is in turn processed and possibly modified by the NLU model. This process provides a new AV pair set c, which is part of the observation o. The user's goal $\boldsymbol{G}$ = $\{[g^\gamma, gv_i^\gamma]\}$ and the user's knowledge $\boldsymbol{K}$ = $\{[k^\kappa, kv_i^\kappa]\}$ 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. USER MODEL

According to the conventions of Figure 1 and omitting the t indices, the user behavior is ruled by the following joined probability that can be factored and simplified [9]:

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

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

These terms emphasize on the relation existing between the user's utterance production process and his/her goal and knowledge, themselves linked together. The knowledge can be modified during the interaction through the speech outputs produced by the system. Yet, this modification of the knowledge is incremental (it is an update) and 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}:

$$P(k|sys,s,n) = \sum_{k^-} P(k|k^-,sys,s,n) \cdot P(k^-|sys,s,n)$$
$$= \sum_{k^-} P(k|k^-,sys,n) \cdot P(k^-|s)$$

From these two expressions, and using the conventions of paragraph 2.1 the user model can be realized

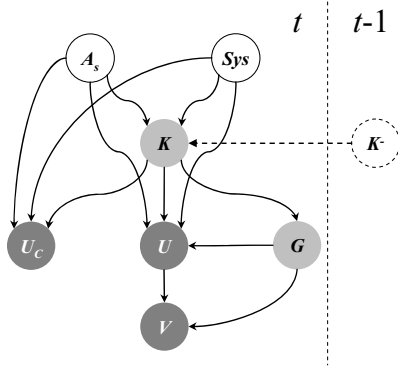thanks to the following Dynamic Bayesian Network (DBN):

Figure 2: DBN-based user model

To use this DBN as a generative model for user modeling, one should introduce the t-uple $\{A_S, Sys, K, K^-, G\}$ in the inference engine which will produce a probability distribution over the configuration space of the t-uple $\{U_C, U, V\}$. To generate a wider range of different behaviors, the actual t-uple $\{U_C, U, V\}$ transmitted to subsequent modules is drawn from the distribution returned by the inference and not the one with the highest probability.

4. NLU MODEL

In this paper, the assumption that ASR errors only affect values of the AV pairs is made. It is assumed that only words occurring in a same context can be substituted with each other during speech recognition because of the use of a language model. Readers should refer to [3],[9],[10] for further details about the functioning of the ASR model. So, errors resulting in a complete mismatch between attributes and values or in an affectation of a correctly recognized value to an incorrect attribute will be considered as NLU errors and should be detected by this sub-system.

To build the evidence that allows using the DBN of Figure 2 as a concept classifier, we can only add to the word sequence w_t providing $\{V\}$, the information known by the DM which might configure the NLU system. Obviously, the DM is able to know the content of the generated sentence sys_t $\{A_S, Sys\}$. The system is not supposed to know the user's goal or knowledge. Yet, the user's knowledge is often modeled so as to contain information about the history of the interaction (we show an example in section 5). The internal state s of the system is often built so as to contain such information as well. So we also included this as a part of the evidence introduced in the inference engine which is therefore: $\{A_S, Sys, K{\sim}s, V\}$. The processing of this evidence provides a probability distribution over the configuration space of $\{U\}$ and consequently operates like a concept classifier by providing a mapping between values and attributes. Here, the attrib-

utes providing the highest probability are kept. It is important to keep in mind that the history introduced in the evidence is the history kept by the system through its state update, which is not always the same as the one included in the user's knowledge. This is the major distinction between the use of the DBN as a user model or as a NLU model. This might be a source of NLU error modeling.

This method can also provide a kind of NLU confidence level. In the case a single value has to be associated to a concept, the confidence level is simply the highest probability retrieved by the inference process. When there are several values, an individual confidence level can be affected to each retrieved AV pair or a global confidence level for the utterance can be affected by multiplying all individual probabilities. Of course, this provides confidence levels on the attributes while the ASR model provides confidence levels on the values.

5. EXPERIMENTAL RESULTS

This model was developed in the aim of being used in an optimal dialog strategy learning process. We therefore show here a use case of dialog simulation for Reinforcement-Learning (RL) agent training as proposed in [2] on a simple form-filling dialog task. To do so, a reward function (or reinforcement signal) r_t has to be defined. This reward provides information about the quality of each DM decision of performing an action a when in state s at time t. It is generally considered that the contribution of each action to the user's satisfaction is the most suitable reward function [1]. According to [8], the major contributors to user's satisfaction are the dialog time duration (which can be approximated by the number of dialog turns N), the ASR performances (which we will approximate by a confidence level CL as in [3]) and the task completion (TC). For this reason, we chose a reward function of the form:

$$r_t = w_{TC} \cdot TC + w_{CL} \cdot CL - w_N \cdot N$$

where w_x are positive tunable weights. A Watkin's Q(λ) RL algorithm has been used with a softmax action selection strategy.

The task is a simplified version of a train ticket booking system that aims at delivering train tickets corresponding to a particular travel. Users are invited to provide information about the departure city (over 50 possible options) and time (over 24 possible options) as well as the destination city and time. The desired class (2 options) is also requested. Table 1 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 a given attribute during the dialog. The task completion is therefore measured as a ratio between the common values in the goal and the values retrieved by the system after the dialog session. The simula-

tion environment includes the DBN user model, an ASR model like in [3] and the NLU model.

Task		User Goal (G)		Knowledge (K)	
Attributes ($\mathcal{A}$)	#$\mathcal{V}$	Att.	Value	Count	init
dep	50	g^{dep}	Glasgow	k^{dep}	0
dest	50	g^{dest}	Edinburgh	k^{dest}	0
t_dep	24	g^{t_dep}	8	k^{t_dep}	0
t_dest	24	g^{t_dest}	12	k^{t_dest}	0
class	2	g^{class}	1	k^{class}	0

Table 1: AVM Task description, User Goal and Knowledge representation as AV pairs

Two configurations will be used in which the NLU model is either disabled (Sim_{ASR}) or enabled (Sim_T). The used confidence level CL is either CL_{ASR} or $CL_T = (CL_{ASR} \cdot CL_{NLU})$. The RL paradigm requires the definition of a state space. It will be defined by a set of state variables which are 5 Booleans (one for each attribute in the task) set to 1 when the corresponding value is known, 5 status Booleans set to 1 if the corresponding value is confirmed and 5 binary values indicating whether the CL associated to the corresponding value is *high* or *low*. Every combination is not possible and the state space size is therefore of 52 states. The state space relying on CL_{ASR} is denoted by S_{ASR} in the following and the state space relying on CL_T is S_T. The DM will be allowed 5 action types: greeting, open question (about more than 1 attribute), closed question (about only 1 attribute), explicit confirmation, closing.

Performance		
Setting	N	TC
Sim_{ASR}, S_{ASR}	5.39	0.81
Sim_T, S_{ASR}	7.03	0.74
Sim_T, S_T	5.82	0.79

Strategy					
Setting	greet	constQ	openQ	expC	close
Sim_{ASR}, S_{ASR}	1.0	0.85	1.23	1.31	1.0
Sim_T, S_{ASR}	1.0	1.25	1.18	2.60	1.0
Sim_T, S_T	1.0	1.15	1.29	1.38	1.0

Table 2 : Learned strategy in terms of average number of turns for a dialog (N), task completion (TC) and relative frequency of each action type.

The results of the learning process on 10^5 dialogs shown in Table 2 can be interpreted as follow. When no NLU simulation is done (Sim_{ASR}), the CL_{ASR} introduced in the state space and rewards (S_{ASR}) is sufficient to learn a strategy that can handle ASR errors. Yet, when NLU errors are introduced in the simulation (Sim_T) as it would be in a real dialog system, the CL_{ASR} is not sufficient anymore and the system learns to ask for more confirmations since it has no clue about the quality of its understanding of the user's utterance. But when substituting CL_T to CL_{ASR} in the state space and reward function (S_T), the performances are getting better and almost reach the level of the first experiment. That means that the RL agent learned to ask questions that provide better CL_T. Having a closer

look to the evolution of the average number of open questions, one can see that it increases between the second and the last experiment. Actually, the system learned to ask open questions about several attributes but excluding open questions where the same value can be associated to different attributes. For example, it never asks for departure and destination cities in the same question but it may ask for a departure city and a departure time for example. This provides better CL_T values.

6. CONCLUSIONS AND PERSPECTIVES

In this paper we presented a modular task-independent environment for dialog simulation including a user model and a NLU simulation module based on Dynamic Bayesian Networks. Some basic experiments of optimal strategy learning using this simulation environment allowed concluding that adding the NLU model could help pointing out potentially problematic dialog scenarios.

This framework offers other interesting possibilities. For instance, grounding sub-dialogs could be added to the set of the possible actions. The need for grounding actions could be detected through inconsistency between the current system state and an inferred estimate of the user's knowledge about the history of the ongoing dialog.

7. REFERENCES

[1] S. Singh, M. Kearns, D. Litman, M. Walker, 'Reinforcement Learning for Spoken Dialogue Systems.' In *Proc. NIPS'99*, Denver, USA, 1999.

[2] E. Levin, R. Pieraccini, 'A Stochastic Model of Computer-Human Interaction for Learning Dialogue Strategies,' In *Proc. Eurospeech'97*, Rhodes, Greece, 1997.

[3] O. Pietquin, S. Renals, 'ASR System Modeling for Automatic Evaluation and Optimization of Dialogue Systems.' In *Proc. ICASSP'02*, vol. 1, Orlando (Fl), USA 2002.

[4] K. Scheffler, S. Young, 'Simulation of Human-Machine Dialogues.' *Technical Report CUED/F-INFENG/TR 355*, Cambridge University, Engineering Dept, 1999.

[5] J. Allen, *Natural Language Understanding*. Benjamin Cummings, 1987, Second Edition, 1994.

[6] R. Pieraccini, E. Levin, 'Stochastic Representation of Semantic Structure for Speech Understanding,' *Speech Communication*, vol. 11, 1992.

[7] H. Meng, W. Lam, K. Low, 'Learning Belief Networks for Language Understanding.' In *Proc. ASRU'99*, 1999.

[8] M. Walker, D. Litman, C. Kamm, A. Abella, 'PARADISE: A Framework for Evaluating Spoken Dialogue Agents.' In *Proc. 35th Annual Meeting of the ACL*, Madrid, Spain, 1997.

[9] O. Pietquin, *A Framework for Unsupervised Learning of Dialogue Strategies*, Presses Universitaires de Louvain, SIMILAR Collection, ISBN 2-930344-63-6, 2004.

[10] O. Pietquin, R. Beaufort, 'Comparing ASR Modeling Methods for Spoken Dialogue Simulation and Optimal Strategy Learning.' In *Proc. of Eurospeech'05*, Lisbon, Portugal, 2005.