Leveraging POMDPs trained with User Simulations and Rule-based Dialogue Management in a Spoken Dialogue System

Sebastian Varges, Silvia Quarteroni, Giuseppe Riccardi, Alexei V. Ivanov, Pierluigi Roberti

Department of Information Engineering and Computer Science

University of Trento

38050 Povo di Trento, Italy

{varges|silviaq|riccardi|ivanov|roberti}@disi.unitn.it

Abstract

We have developed a complete spoken dialogue framework that includes rule-based and trainable dialogue managers, speech recognition, spoken language understanding and generation modules, and a comprehensive web visualization interface.

We present a spoken dialogue system based on Reinforcement Learning that goes beyond standard rule based models and computes on-line decisions of the best dialogue moves. Bridging the gap between handcrafted (e.g. rule-based) and adaptive (e.g. based on Partially Observable Markov Decision Processes - POMDP) dialogue models, this prototype is able to learn high rewarding policies in a number of dialogue situations.

1 Reinforcement Learning in Dialogue

Machine Learning techniques, and particularly Reinforcement Learning (RL), have recently received great interest in research on dialogue management (DM) (Levin et al., 2000; Williams and Young, 2006). A major motivation for this choice is to improve robustness in the face of uncertainty due for example to speech recognition errors. A second important motivation is to improve adaptivity w.r.t. different user behaviour and application/recognition environments.

The RL approach is attractive because it offers a statistical model representing the *dynamics* of the interaction between system and user. This contrasts with the supervised learning approach where system behaviour is learnt based on a fixed corpus. However, exploration of the range of dialogue management strategies requires a simulation environment that includes a simulated user (Schatzmann et al., 2006) in order to avoid the prohibitive cost of using human subjects.

We demonstrate various parameters that influence the learnt dialogue management policy by using pre-trained policies (section 5). The application domain is a tourist information system for accommodation and events in the local area. The domain of the trained DMs is identical to that of a rule-based DM that was used by human users (section 4), allowing us to compare the two directly.

2 POMDP demonstration system

The POMDP DM implemented in this work is shown in figure 1: at each turn at time t, the incoming N user act hypotheses $a_{n,u}$ split the state space S_t to represent the complete set of interpretations from the start state ($N=2$). A belief update is performed resulting in a probability assigned to each state. The resulting ranked state space is used as a basis for action selection. In our current implementation, belief update is based on probabilistic user responses that include SLU confidences. Action selection to determine system action $a_{m,s}$ is based on the best state (m is a counter for actions in action set A). In each turn, the system uses an ϵ-greedy action selection strategy to decide probabilistically if to exploit the policy or explore any other action at random. (An alternative would be softmax, for example.) At the end of each dialogue/session a reward is assigned and policy entries are added or updated for each state-action pair involved. These pairs are stored in tabular form. We perform Monte Carlo updating similar to (Levin et al., 2000):

$$Q_t(s,a) = R(s,a)/n + Q_{t-1} \cdot (n-1)/n \quad (1)$$

where n is the number of sessions, R the reward and Q the estimate of the state-action value.

At the beginning of each dialogue, a user goal U_G (a set of concept-value pairs) is generated randomly and passed to a user simulator. The user simulator takes U_G and the current dialogue context to produce plausible SLU hypotheses. These
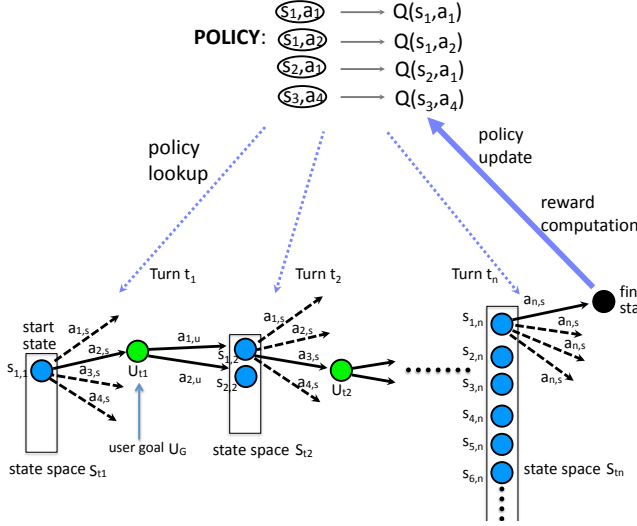
Figure 1: POMDP Dialogue Manager

are a subset of the concept-value pairs in U_G along with a confidence estimate bootstrapped from a small corpus of 74 in-domain dialogs. We assume that the user 'runs out of patience' after 15 turns and ends the call.

The system visualizes POMDP-related information live for the ongoing dialogue (figure 2). The visualization tool shows the internal representation of the dialogue manager including the the N best dialogue states after each user utterance and the reranking of the action set. At the end of each dialogue session, the reward and the policy updates are shown, i.e. new or updated state entries and action values. Moreover, the system generates a plot that relates the current dialogue's reward to the reward of previous dialogues.

3 User Simulation

To conduct thousands of simulated dialogues, the DM needs to deal with heterogeneous but plausible user input. We designed a User Simulator (US) which bootstraps likely user behaviors starting from a small corpus of 74 in-domain dialogs, acquired using a rule-based version of the system (section 4). The role of the US is to simulate the output of the SLU module to the DM during the whole interaction, fully replacing the ASR and SLU modules. This differs from other user simulation approaches where n-gram models of user dialog acts are represented.

For each simulated dialogue, one or more user goals are randomly selected from a list of possible user goals stored in a database table. A goal is rep-

resented as the set of concept-value pairs defining a task. Simulation of the user's behaviour happens in two stages. First, a *user model*, i.e. a model of the user's intentions at the current stage of the dialogue, is created. This is done by mining the previous system move to obtain the concepts required by the DM and their corresponding values (if any) from the current user goal. Then, the output of the user model is passed to an *error model* that simulates the "noisy channel" recognition errors based on statistics from the dialogue corpus. Errors produce perturbations of concept values as well as phenomena such as *noInput*, *noMatch* and *hangUp*. If the latter phenomena occur, they are directly propagated to the DM; otherwise, plausible confidences (based on the dialogue corpus) are attached to concept-value pairs. The probability of a given concept-value observation at time $t+1$ given the system move at time t, $a_{s,t}$, and the session user goal g_u, called $P(o_{t+1}|a_{s,t}, g_u)$, is obtained by combining the outputs of the error model and the user model:

$$P(o_{t+1}|a_{u,t+1}) \cdot P(a_{u,t+1}|a_{s,t}, g_u)$$

where $a_{u,t+1}$ is the true user action. Finally, concept-value pairs are combined in an SLU hypothesis and, as in the regular SLU module, a cumulative utterance-level confidence is computed, determining the rank of each of the N hypotheses output to the DM.

4 Rule-based Dialogue Management

A rule-based DM was developed as a meaningful comparison to the trained DM, to obtain training data from human-system interaction for the US, and to understand the properties of the domain. Rule-based dialog management works in two stages: retrieving and preprocessing facts (tuples) taken from a dialogue state database, and inferencing over those facts to generate a system response. We distinguish between the 'context model' of the first phase – essentially allowing more recent values for a concept to override less recent ones – and the 'dialog move engine' of the second phase. In the second stage, acceptor rules match SLU results to dialogue context, for example perceived user concepts to open questions. This may result in the decision to verify the application parameter in question, and the action is verbalized by language generation rules. If the parameter is accepted, application dependent task
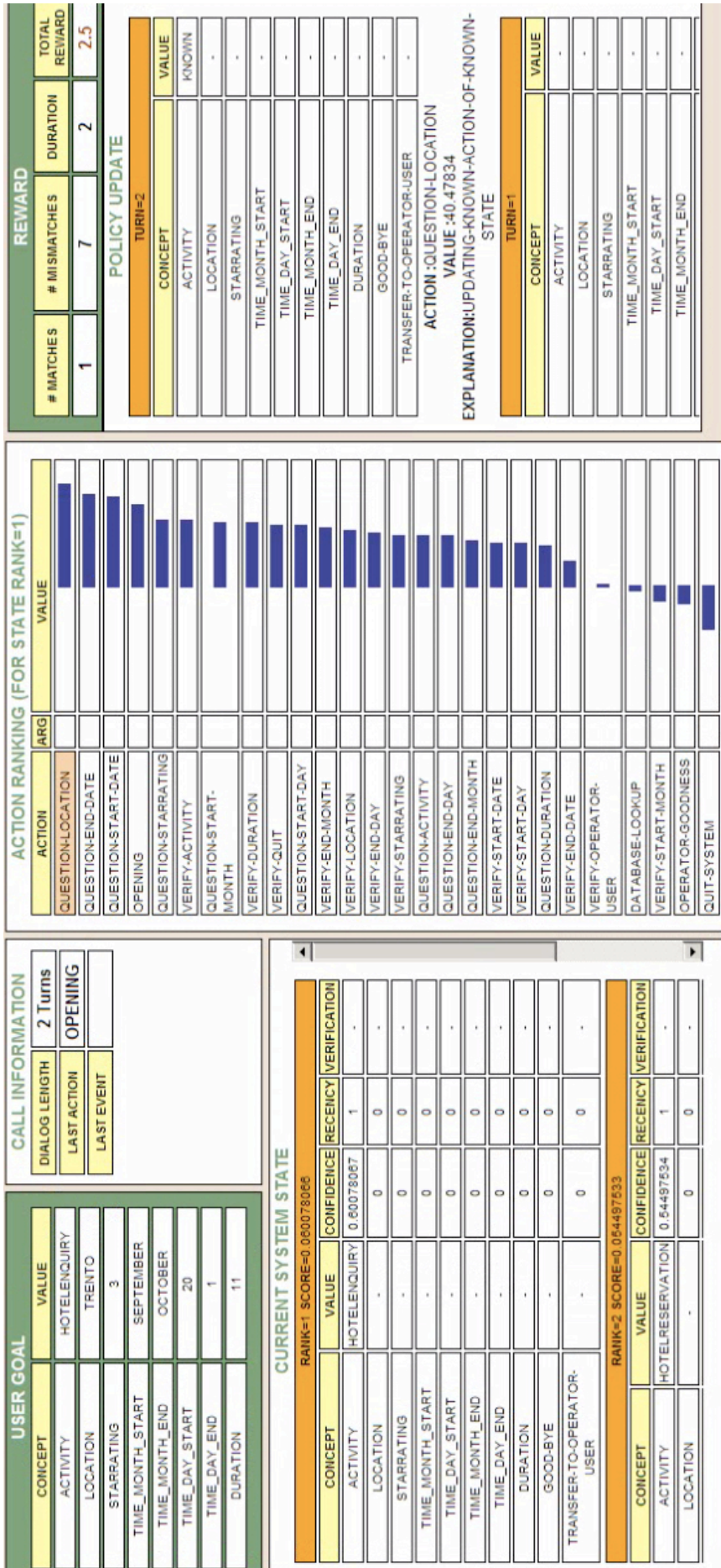
Figure 2: A screenshot of the online visualization tool. Left: user goal (top), evolving ranked state space (bottom). Center: per state action distribution at turn t_i. Right: consequent reward computation (top) and policy updates (bottom). See video at http://www.youtube.com/watch?v=69QR0tKKhCw.
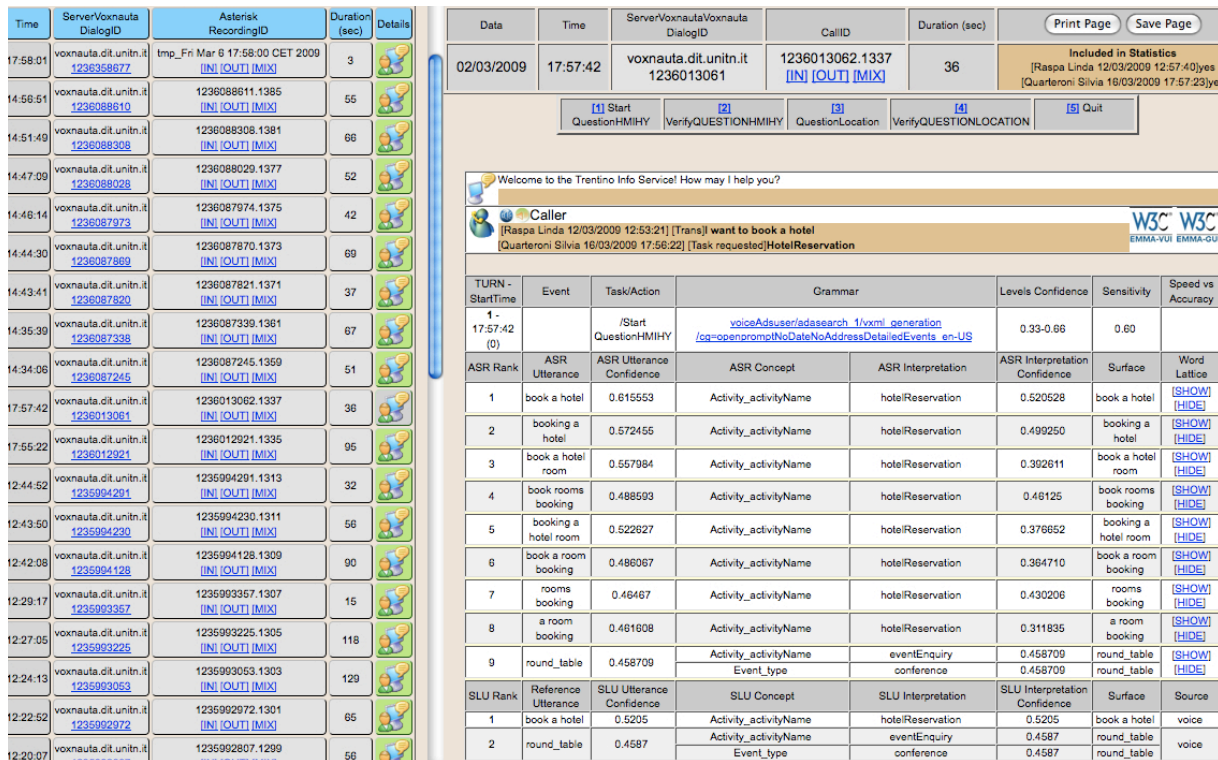
Figure 3: Left Pane: overview of a selection of dialogues in our visualization tool. Right Pane: visualization of a system opening prompt followed by the user's activity request. All *distinct* SLU hypotheses (concept-value combinations) deriving from ASR are ranked based on concept-level confidence (2 in this turn).

rules determine the next parameter to be acquired, resulting in the generation of an appropriate request. See (Varges et al., 2008) for more details.

5 Visualization Tool

In addition to the POMDP-related visualization tool (figure 2), we developed another web-based dialogue tool for both rule-based and POMDP system that displays ongoing and past dialogue utterances, semantic interpretation confidences and distributions of confidences for incoming user acts (see dialogue logs in figure 3).

Users are able to talk with several systems (via SIP phone connection to the dialogue system server) and see their dialogues in the visualization tool. They are able to compare the rule-based system, a randomly exploring learner that has not been trained yet, and several systems that use various pre-trained policies. The web tool is available at `http://cicerone.dit.unitn.it/DialogStatistics/`.

References

E. Levin, R. Pieraccini, and W. Eckert. 2000. A stochastic model of human-machine interaction for learning dialog strategies. *IEEE Transactions on Speech and Audio Processing*, 8(1).

J. Schatzmann, K. Weilhammer, M. Stuttle, and S. Young. 2006. A Survey of Statistical User Simulation Techniques for Reinforcement-Learning of Dialogue Management Strategies. *Knowledge Engineering Review*, 21(2):97–126.

Sebastian Varges, Giuseppe Riccardi, and Silvia Quarteroni. 2008. Persistent information state in a data-centric architecture. In *Proc. 9th SIGdial Workshop on Discourse and Dialogue*, Columbus, Ohio.

J. D. Williams and S. Young. 2006. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech and Language*, 21(2):393–422.