EFFECTS OF THE USER MODEL ON SIMULATION-BASED LEARNING OF DIALOGUE STRATEGIES

Jost Schatzmann, Matthew N. Stuttle, Karl Weilhammer and Steve Young

Cambridge University Engineering Department
Trumpington Street, Cambridge CB2 1PZ, United Kingdom
{js532, mns25, kw278, sjy}@eng.cam.ac.uk

ABSTRACT

Over the past decade, a variety of user models have been proposed for user simulation-based reinforcement-learning of dialogue strategies. However, the strategies learned with these models are rarely evaluated in actual user trials and it remains unclear how the choice of user model affects the quality of the learned strategy. In particular, the degree to which strategies learned with a user model generalise to real user populations has not be investigated. This paper presents a series of experiments that qualitatively and quantitatively examine the effect of the user model on the learned strategy. Our results show that the performance and characteristics of the strategy are in fact highly dependent on the user model. Furthermore, a policy trained with a poor user model may appear to perform well when tested with the same model, but fail when tested with a more sophisticated user model. This raises significant doubts about the current practice of learning and evaluating strategies with the same user model. The paper further investigates a new technique for testing and comparing strategies directly on real human-machine dialogues, thereby avoiding any evaluation bias introduced by the user model.

1. INTRODUCTION

The application of machine-learning techniques to dialogue management is currently a growing research area [1]. In particularly the use of reinforcement-learning for finding optimal dialogue strategies is attracting interest [2] [3]. Previous research has demonstrated that systems can successfully learn from training data what constitutes a good dialogue strategy. However, it is usually not possible to learn an optimal strategy directly from a corpus of dialogues, since the size of current corpora is rarely sufficient to exhaustively explore the vast space of possible dialogue states and strategies. Moreover, no guarantee can be given that the optimal strategy is indeed present in the given training corpus, regardless of the size of the corpus.

As a solution to this problem, several research groups [4], [5], [6], [7] have investigated the use of a two-phased learning setup involving a simulated user. First, a stochastic model of real user behaviour is trained on a corpus of human-computer dialogues using supervised learning. In the second phase, the dialogue manager uses reinforcement-learning to learn an optimal strategy through interaction with the simulated user. This setup allows any number of training episodes for strategy learning to be generated and further, it allows strategies which are not present in the corpus to be explored.

A variety of different techniques for user simulation have been proposed in the literature, but the evaluation of these techniques is still an open question of research. It has been shown that the currently available techniques fail to reproduce the variety of human behaviour. For example, as shown in [8], simple statistical metrics are often sufficient to discern synthetic dialogues from real ones. However, it remains unclear how much the learned strategy depends on the choice and the quality of the user simulation. It is also unclear how reliable the current practice of learning and evaluating strategies with the same user model is.

This paper attempts to investigate these questions by training three of the most prominent stochastic user models on a COMMUNICATOR corpus of human-machine dialogues and learning strategies with each of these models. We then quantitatively and qualitatively compare the learned strategies with respect to the handcrafted COMMUNICATOR strategies and with respect to each other. We also test each strategy on user models that have not been used to learn the strategy. Finally, we report on initial experiments with a new and potentially more objective technique for evaluating the quality of learned strategies using real dialogue data.

2. BACKGROUND

2.1. Dialogue as a Markov Decision Process

Human-computer dialogue can be modelled as a Markov Decision Process (MDP) [3] with a finite state space S, a finite action set A, a set of transition probabilities T and a reward function R. At each time step, the dialogue manager is in a particular state $s \in S$. It executes the discrete action $a \in A$, transitions into the next state s' according to the transition probability $p(s'|s, a)$ and receives a reward r. The Markov Property ensures that the state and reward at time $t + 1$ only depend on the state and action at time t.

The MDP model of dialogue allows us to view a dialogue management strategy (or policy) π as a mapping from states to actions: For every state s, the policy selects the next system action a based only on s. It also enables us to formalize dialogue management as a mathematical optimization problem. The optimal policy π^* is the policy that maximizes the cumulative reward over time.

2.2. Reinforcement-Learning

Let $r_{(t)}$ be the total return received when starting at time t. A discount factor γ, with $0 \leq \gamma \leq 1$, is typically used to discount immediate rewards more strongly than distant future rewards

The research reported in this paper was supported by the EU FP6 TALK Project. We would like to thank J. D. Williams and H. Ye for many helpful suggestions. We would also like to thank O. Lemon, K. Georgila and J. Henderson at Edinburgh University for providing the annotated Communicator data.

$$r_{(t)} = \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + ... = \sum_{k=1}^{\infty} \gamma^k r_{t+k}. \quad (1)$$

The goal is then to find the optimal policy π^* that maximizes the expected value of $r_{(t)}$. Reinforcement-learning offers a variety of algorithms for finding π^* within the MDP framework through trial-and-error interaction between the learning agent and its dynamic environment.

The Q-Learning algorithm [9] is one of the simplest forms of reinforcement-learning. It works by maintaining Q-values for every pair (s, a) of state s and action a. These values estimate the expected return of taking action a in state s and following π thereafter

$$Q^{\pi}(s, a) = E_{\pi}(r_{(t)}|s_t = s, a_t = a). \quad (2)$$

Once the Q-values for each state-action pair have been estimated, the optimal policy can be found using

$$\pi^*(s) = \arg\max_a Q^*(s, a). \quad (3)$$

The learning process is started by arbitrarily initialising the Q-values for all state-action pairs. As the learning dialogue manager interacts with the (simulated) user, the Q-values are iteratively updated to become better estimates of the expected return of the state-action pairs. After each system action a in state s, the user response results in a transition to state s' and a reward r. The corresponding Q-value estimate is then updated using

$$Q(s, a) := (1 - \alpha)Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a')) \quad (4)$$

where α represents a learning rate parameter that decays from 1 to 0.

During the learning process, it is necessary to achieve a compromise between exploration (trying out random actions) and exploitation (taking actions which have already shown to lead to high rewards). The most straightforward solution is an ϵ-greedy control policy. At each step, a random number $0 < \beta < 1$ is generated and the next action is selected randomly if $\beta < \epsilon$. If $\beta \geq \epsilon$, the best action is taken. ϵ is usually selected so as to decrease with the number of training cycles from 1 to 0.1.

2.3. Approaches to User Simulation

The problem of user simulation for Spoken Dialogue Systems has been approached by various research groups over the last decade. With regards to robustness, scalability and portability, the work on stochastic models is of greatest interest to us. Most of this work models dialogue on the abstract level of intentions rather than the word- or acoustic-level since this avoids the problem of natural language generation. In the following paragraph, a very brief overview of some of the most prominent domain-independent techniques is provided. A more detailed comparison can be found in [8].

The Bigram model proposed in early work by Eckert et al. [4] is perhaps the simplest stochastic model for predicting user responses to system actions. It estimates a probability $p(a_u|a_s)$ for every possible pair of system action a_s and user response a_u based on their frequency of occurrence in the training data.

In later work stronger constraints were applied to the user behaviour to correlate better with the conventional structure of human dialogues. The Levin Model [3] limits the Bigram model to selected pairs of system action and user reponse. A system request for a certain piece of information A, for example, is parameterised by the probability of the user actually providing A and the probability of providing n additional pieces of information.

The Bigram and Levin Model both suffer from a lack of goal-consistency in user behaviour. To overcome this problem, Scheffler and Young [5] suggested the use of explicit user goal representations in the form of attribute-value pairs and the use of deterministic rules to ensure that the user acts in accordance with his goal throughout the dialogue. In subsequent work, Pietquin [6] adapted the Levin model and conditioned its parameters on an extended representation of Scheffler and Young's goal model. Pietquin also suggested a number of additional user parameters to account for user memory and satisfaction.

More recently, Georgila, Henderson and Lemon [7] have investigated the use of n-gram models in conjunction with linear feature combination to learn what aspects of the dialogue state are most useful for prediciting user responses.

3. RESEARCH OBJECTIVE

A major criticism of simulation-based learning of dialogue strategies is the current lack of a reliable and objective evaluation method. Indeed, the current standard technique for testing strategies learned with a simulated user is to try the strategy on the same simulated user. Using this technique, many research groups have been able to show that learned strategies can outperform competing handcrafted strategies. However, whilst this illustrates the feasibility of learning strategies using a simulated user, it raises the question as to what effect the choice of user model has on the quality of the learned strategy. It also raises doubts as to whether a learned strategy is truly optimal or just optimized for a particular user model since, in effect, the model is being tested on the training data.

The goal of this paper is to investigate this problem by training several user models on the same corpus of dialogues and learning dialogue strategies with each of these user models. We then qualitatively and quantitatively compare the learned strategies with respect to handcrafted strategies and with respect to each other. We also test the learned strategies on user models that have not been used for training.

Ideally, we would prefer to test learned strategies on real users. However, given the high cost of running user trials, unbiassed evaluation methods using existing dialogue corpora would clearly be valuable. The paper concludes with a proposed evaluation approach which attempts to achieve this.

4. EXPERIMENTAL SETUP

4.1. Dataset

The corpus used for the experiments described in this paper is a COMMUNICATOR corpus of 697 real human-computer dialogues recorded with 97 different users. The data was collected with 4 different dialogue managers developed by ATT, BBN, CMU and SRI and covers flight-, hotel- and rental car-bookings. The nature of the dialogues is task-oriented rather than conversational and may be described as "slot-filling" with a high degree of system initiative. The data has been previously annotated using the DATE scheme [10] and tagged with further semantic information as described in [11]. The annotation allows us to view the dialogues at

the abstract level of intentions and we have converted the data such that each dialogue is a sequence of turns containing one or more tuples of the form <speech act, slot name, slot value>. The user utterance "I want to go to London", for instance, is compressed to <provide_info, dest_city, london>.

4.2. User Models

We selected three of the most prominent domain-independent stochastic user models: a) the simple Bigram model [4], b) the more sophisticated Levin model [3] which accounts for some degree of conventional dialogue structure in user behaviour and c) the Pietquin model [6] which extends the Levin model with simple representations of user goal, memory and satisfaction. All of these models simulate dialogue on the abstract level of intentions, as described above and all were trained on the actual recognised speech rather than the reference transcriptions. The simulation thus effectively models both the user and the communication channel, and hence, no separate error modelling is required. Further details of how these user models were adapted for use with the COMMUNICATOR dataset can be found in [8].

4.3. Action Set, State Space and Reward Function

The slot-filling dialogues in the COMMUNICATOR corpus can be suitably represented as Markov Decision Processes (MDPs) with a system action set A and a state space S (as described in Section 2.1). The size of the state space is determined by the number of slots and the number of states each slot can be in. The learning problem is limited to the four slots entailed in a single-leg flight-booking: the origin city, the destination city, the departure date and the departure time. Each of these can be either *unknown*, *known* or *confirmed*, resulting in a total number of $3^4 = 81$ states.

The size of the system action set is determined by the number of system speech acts and the number of slots that each of these acts can be combined with. We have three speech acts: *request_info*, *implicit_confirm* and *explicit_confirm*, representing a request for a slot, an implict confirmation of a slot value or an explicit confirmation. We are interested in dialogues where each dialogue turn can have more than one speech act, so that the system can, for example, implicitly confirm slot A and request slot B in a single turn. In every turn, each of the 4 slots is either requested, implicitly confirmed, explicitly confirmed or not mentioned at all. This results in $4^4 - 1 = 255$ combinations. (The empty turn where no slot is mentioned is discarded.) We further add a $hangup$ action to the system action set, thus increasing the total number of system actions to 256.

The total number of system state/action combinations that the learning dialogue manager has to explore is $81 * 256 = 20736$. This number is large, and it grows exponentially as the number of slots or speech acts is increased. To mitigate this, it is reasonable to hardcode a limited amount of prior knowledge into the learning process by blocking subsets of the action set for certain states. We use the following simple rules: 1) For all states where slot x is unknown, all actions that request a confirmation for x are blocked. 2) For all states where x is known, the *request_info* action for this slot is blocked. 3) For all states where x is confirmed, the request and confirmation actions are blocked, thus forcing the system to hangup once all slots are filled and confirmed. This reduces the learning process to the true design issues: When do we confirm, do we confirm explicitly or implicitly, and how many slots do we attempt to simultaneously fill or confirm?

The reward function is selected so that it penalises long dialogues with -1 point for every system action and awards +20 points for the successful completion of the flightbooking dialogue (all 4 slots filled and confirmed). We do not reward partial completion of the dialogue to avoid providing clues on how to complete the full task.

4.4. Learning Setup

In choosing the learning algorithm, our aim is to keep the setup as generic and simple as possible. The goal is to avoid any features that might bias the result and distract from the effects of the user model on the learned strategy. The Q-Learning algorithm is selected with a simple ϵ-Greedy control policy as described in Section 2.2 of this paper. The learning rate parameter α is set to decay from 1 to 0 using

$$\alpha(s,a) = \frac{\tau}{\tau + n(s,a)} \tag{5}$$

where $n(s,a) = 1, 2, 3, ...$ is the number of times $Q(s,a)$ has been visited and τ is set to some positive constant. (Good results were obtained with $\tau = 100$.) This definition of α ensures convergence. The discounting factor γ is set to 0.9.

For each of the three user models, 100,000 cycles were used for learning the strategy. We found that the Q-values had converged well by the end of the training process. Additional experiments were run with 1,000,000 cycles, but no change in policy was observed.

5. EXPERIMENTAL RESULTS

5.1. Strategy Evaluation Metric

The "quality" of a dialogue is not precisely defined and this makes dialogue evaluation problematic. As shown in [12], user satisfaction depends on a variety of factors including the task success rate, the dialogue efficiency and qualitative factors such as the system's ability to handle misunderstandings.

For the sake of simplicity and comparability, we chose a straightforward performance measure that has also been previously used by [13] for the evaluation of simulation-based strategy learning. The metric rewards the filling and grounding of slots while penalising long dialogues. For each of the four slots, 25 points are awarded once the slot is filled and another 25 points are awarded once it is confirmed. One point is subtracted from the total score for every system action. [1]

Whereas the learned strategies are trained to complete only the single-leg flight booking, the COMMUNICATOR dialogues often include hotel- or car-rental reservations in addition to the flight booking. If the computation of the length penalty is based on the full dialogue this naturally favours the learned strategies. To guarantee a fair comparison we stop the COMMUNICATOR dialogues when the first flight offer is made. This ensures that only the system actions are counted that contribute to the filling and grounding of the four slots covered by our evaluation metric. We also exclude the small percentage of COMMUNICATOR dialogues covering multiple-leg flight bookings, leaving a total of 643 dialogues for evaluation.

[1] Note that the metric is similar to the reward function used during strategy learning but not identical: During learning we want to reward the successful completion without specifying clues as to how the task can be completed. During evaluation we also want to award points for the partial completion of the task.

5.2. Traditional Strategy Evaluation

The first experiment follows the typical form of evaluation currently used by research groups working on simulation-based strategy learning. The learned strategy is tested on the same user model that was used for training. More precisely, a set of dialogues is simulated between the user model and the DM following the learned strategy. The quality of these dialogues is then compared to the quality of the dialogues in the original COMMUNICATOR training corpus. The results are used to compare the performance of the learned strategy to the handcrafted strategy used when collecting the training corpus.

Figure 1 shows the results obtained with this form of evaluation. ATT, BBN, CMU and SRI denote the scores for the individual COMMUNICATOR systems, ALL denotes the combined corpus and BIG, LEV and PTQ denote the scores obtained with each of the three user models. The sample size for the user model tests is 100 (simulated) dialogues and the sample size for the individual COMMUNICATOR systems varies between 79 and 258 dialogues.

As can be seen, all of the three user models appear to achieve very good results. Even the simple Bigram model receives a score of 129.9 ± 6.0 (mean reward $\pm$ standard error), a relative improvement of 20.1% over the best competing handcrafted strategy (BBN, 108.2 ± 4.3). [2]

Viewed individually, each of the user models seems to produce an optimal strategy, superior to the competing handcrafted ones. However, the graph also shows a clear dependency between user model quality and dialogue strategy performance. The Pietquin and the Levin model clearly outperform the simple Bigram model, indicating that an increase in user model sophistication leads to a better strategy.

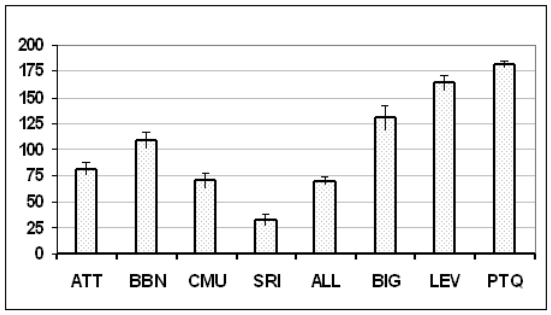

Fig. 1. Strategies learned and evaluated on the same user model appear to outperform handcrafted strategies. The bars show the mean reward obtained and the thin lines indicate 95% confidence intervals.

We also measured the Q-values of the best action in the initial state of each of the learned policies. The Pietquin policy achieved the best value of 9.01, followed by the Levin model with 7.43 and the Bigram model with 4.64. These numbers confirm that the Pietquin model and the Levin model are comparable in performance and that both lead to a more efficient strategy than the Bigram model.

[2]The improvements for all three user models relative to the best handcrafted strategy are significant with a confidence of 99.9% according to the standard t-test.

5.3. Cross-model Evaluation

The availability of several user models allows us to investigate how a learned strategy performs when evaluated with a user model that was not used for learning. The results, pictured in Figure 2 are revealing: When tested on the Bigram model, all strategies perform very similarly. The difference in reward is in fact statistically insignificant. But when tested on a better user model, such as the Levin or Pietquin model, the performance of the Bigram strategy deteriorates drastically. (62.90 ± 14.3 on the Pietquin model, sample size = 100). This is clearly lower than good handcrafted strategies. [3]

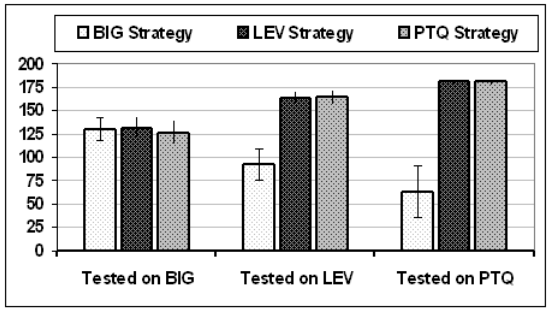

Fig. 2. Cross-Evaluation shows that strategies learned with a poor user model may appear to perform well when tested on the same user model but fail when tested on a better user model.

Interestingly, the figure also shows that a strategy learned with a good user model such as the Pietquin or the Levin user model still performs well when tested on a poor user model such as the Bigram. This result is important and promising, because it suggests that a strategy learned with a good user model can generalise well to other user models and is not necessarily overfitted to the particular model used to train it.

5.4. Qualitative Evaluation of Learned Strategies

To gain a better understanding of the effects of the user model on simulation-based learning, it is helpful to analyse not only the performance of the learned strategies, but also their characteristics.

Due to the restrictions placed on the dialogue manager, all of the learned strategies follow the logical constraints described in Section 4.3. However, the strategies differ in when and how they confirm, and how many slots they request at a time. As the state space is large, it is not possible to visualise the strategies in the form of flow-charts, nevertheless a few general observations can be made.

The graph in Figure 3 shows how actively the different systems pursue the filling and grounding of slots. As in Figure 1 above, the first five columns denote the COMMUNICATOR datasets and the last three columns denote the dialogues generated using the simulated users. The bars indicate what percentage - averaged over all dialogue states - of unknown slots is requested and what percentage of known slots is explicitly/implicitly confirmed. The thin lines indicate 95% confidence intervals.

This analysis shows that the learned strategies request and confirm much more actively than the handcrafted strategies. On aver-

[3]Again the difference in means is significant with a confidence of more than 99.9%.

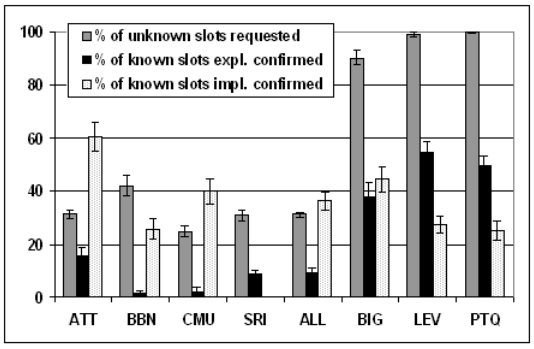

Fig. 3. Qualitative analysis reveals that the learned strategies overload the user with requests.

age, the learned strategies request more than 90% of the unknown slots in each state. In fact, manual analysis shows that the opening system action for all the learned strategies is to request all four slots at once. The COMMUNICATOR systems in contrast only request between 24% and 43% of the unknown slots in each state.

Figure 3 is interesting because it shows that the learned strategies are different from any strategy present in the corpus. It is only due to the simulation-based approach that such strategies can be learnt. At the same time, the figure shows that the quality of the user model has an immediate effect on the characteristics of the learned strategy. In reality, a large number of simultaneous questions is likely to lead to lower recognition accuracy on the user response, a larger number of misunderstandings and a higher likelyhood of the user hanging up. But since none of the user models acknowledge that users are unlikely to respond to many questions at once, the learned strategies overload users with requests. We can also assume that the good evaluation results described in Section 5.2 are at least partly due to the fact that the learned strategies can exploit the over-optimistic level of user cooperativeness inherent in the three models.

A second observation we can make is that the handcrafted strategies have a higher ratio of implicit to explicit confirmations than the learned strategies. This is explained by the fact that we have constrained the learned strategies not to reconfirm already confirmed slots. The COMMUNICATOR systems use implicit confirmations throughout the dialogue to remind the user of what has been established as "common ground" so far.

More interestingly however, the Bigram strategy has a much higher ratio of implicit to explicit confirmations than the other two learned strategies. This may seem surprising, given the poor results obtained with the Bigram strategy when tested on the Levin or Pietquin model. One might expect that the use of implicit confirmations improves the dialogue efficiency. Manual analysis reveals that the Bigram strategy for many states is to *only* implicitly confirm slots, without requesting or explicitly confirming any other slots. When tested on the Bigram user model, this works because the model sees instances during the training where the system turn contains an implicit confirmation and the user response is not empty. But the Bigram model cannot learn that an implicit confirmation request on its own does not necessarily trigger a response from the user. The Pietquin and the Levin model account for a higher degree of conventional dialogue structure. They have a much lower probability of replying to an implicit confirmation request, especially if the slot value that is being confirmed is correct. Hence, a DM following the Bigram strategy interacting with the Pietquin model may often spend several turns trying to implicitly confirm slots with no response coming from the user, hence causing high length penalties and poor evaluation results as shown in Figure 2.

5.5. Strategy Evaluation on Real Dialogue Data

In view of the results presented above, it is apparent that user model-based forms of evaluation can produce misleading results. Studies with real users are clearly the most preferable form of evaluation, but given the costs involved in such studies any technique not involving human users may be seen as a helpful tool during the development phase of a spoken dialogue system.

To avoid any bias that may be introduced through the user model, evaluation of the learned strategy should be based directly on a real dialogue corpus. However, doing this is not trivial, since the learned strategy is not necessarily a strategy that has been used in the corpus, and even if it does appear in the corpus, the number of instances is likely to be too small to derive any conclusion from the outcome of the corresponding dialogues.

Our suggestion to overcome these problems is to analyse whether the dialogues which appear to follow a strategy similar to the learned one are indeed successful. To do this, we first calculate a similarity-score $Sim(\pi_d, \pi^*)$ for each dialogue d based on how similar the strategy π_d followed in this (real) dialogue is to the learned policy π^*. Secondly, we compute the reward $Rew(d)$ obtained in dialogue d using the evaluation metric described in 5.1. If the learned policy has a high quality, we expect to see a positive correlation between $Sim(\pi_d, \pi^*)$ and $Rew(d)$, indicating that dialogues with a high similarity to the learned policy tend to achive higher rewards than other dialogues.

The similarity between the learned strategy π^* and the *observed* strategy π_d can be expressed as a function of the system actions in d. Let

$$Sim(\pi_d, \pi^*) = \frac{1}{n} \sum_{i=1}^{n} \theta_\pi(a_i) \quad (6)$$

where n is the number of system actions in d and $\theta_\pi(a_i)$ expresses how well the selected system action a_i agrees with the learned policy π^*. Of course, many definitions for $\theta_\pi(a_i)$ are possible. One option is the reciprocal rank of a_i according to the ordering of Q-values in the learned policy for the state s in which a_i was executed

$$\theta_1^\pi(a_i) = \frac{1}{1 + |\{k \mid Q^\pi(s, a_k) > Q^\pi(s, a_i), k \neq i\}|}. \quad (7)$$

Figure 4 visualises the similarity-reward-correlation using the above definition for the CMU dataset and the Pietquin strategy. Every dot represents one dialogue, with its y-axis value indicating the reward obtained in the dialogue and its x-axis value indicating its similarity to the strategy learned with the Pietquin model.

When tested on the full COMMUNICATOR corpus (sample size = 643), we obtain a correlation coefficient ρ of 0.28 for the Pietquin strategy, 0.21 for the Levin strategy and 0.16 for the Bigram strategy. The ranking hence corresponds to the performance of the three user models. In fact, the correlation coefficient found with the Bigram strategy is not statistically significant with a confidence of more than 95% ($\rho < 0.195$) - an indicator of the low quality of this strategy.

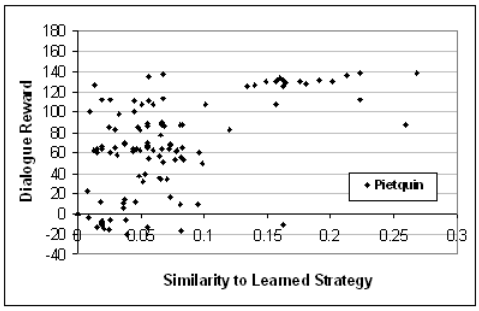

Fig. 4. The scatterplot shows that dialogues with a high similarity to the Pietquin strategy tend to achieve higher rewards. (Correlation coefficient $\rho = 0.58$, statistically significant with a confidence of more than 99.9%, sample size = 132 dialogues)

To assess how dependent our results are on the choice of similarity criterion, two further metrics were tested. First, the value of action a_i according to π^* as the ratio of its Q-value to the sum of all Q-values for the state s in which a_i is executed

$$\theta_2^{\pi}(a_i) = \frac{Q^{\pi}(s, a_i)}{\sum_k Q^{\pi}(s, a_k)}. \quad (8)$$

Secondly, the similarity between a_i and the best action a_{π^*} according to π^* for the state s in which a_i is executed. This may be expressed as the ratio of <speech act, slot name> pairs present in both a_i and a_{π^*} to pairs present in a_i or a_{π^*}

$$\theta_3^{\pi}(a_i) = \frac{|\{a \in a_i\} \cap \{a \in a_{\pi^*}\}|}{|\{a \in a_i\} \cup \{a \in a_{\pi^*}\}|}. \quad (9)$$

Using any of the three similarity metrics presented above, we obtained positive values for the correlation coefficient ρ for all of the three learned strategies when evaluating on the full COMMUNICATOR corpus. However, the Pietquin strategy consistently outranked the two competing strategies. It was also the only strategy that always achieved a score which was statistically significant with a confidence of more than 95%.

6. CONCLUSION

This paper has presented a series of experiments that investigate the effect of the user model on simulation-based reinforcement-learning of dialogue strategies. Our motivation for work in this area has been to examine how good a user model needs to be in order to learn truly optimal strategies. The results show that it is possible to fit a strategy to any particular user model, but that this strategy is not necessarily a good one. In particular, a strategy learned with a poor user model may appear to perform well when tested on the same user model but it fails when tested on a better user model. The converse, however, appears not to be true: A strategy learned with a good user model will still perform well when tested on a poor user model.

The results indicate that the choice of user model has a significant impact on the learned strategy. They also raise serious doubts about the current practice of learning and evaluating strategies with the same user model. On the positive side, they also demonstrate that a strategy learned with a high quality user model generalises well to other types of user model. One may conclude that the development of realistic user models is a high priority for future research on simulation-based learning of dialogue strategies.

Finally, the paper has investigated a new technique for testing learned strategies directly on existing corpora of real human-machine dialogues with the aim of avoiding the inevitable bias that is incurred when testing with a simulation model. The proposed approach measures the correlation between actual reward and the similarity between the observed and learnt policies. Although this does not give an absolute measure of "goodness", it does provide a good cross-check that the learnt policy is rational and is not merely exploiting the short-comings of the user simulation model. It remains to be seen whether the approach can go beyond this and provide a more reliable predictor of subsequent performance in the field.

7. REFERENCES

[1] S. Young, "Talking to machines (statistically speaking)," in *Proc. of ICSLP*, 2002, Denver, Colorado.

[2] S. Singh, M. S. Kearns, D. J. Litman, and M. A. Walker, "Reinforcement learning for spoken dialogue systems," in *Proc. of NIPS*, 1999, Denver, Colorado.

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

[4] W. Eckert, E. Levin, and R. Pieraccini, "User modelling for spoken dialogue system evaluation," in *Proc. of ASRU*, 1997, pp. 80–87.

[5] K. Scheffler and S. J. Young, "Corpus-based dialogue simulation for automatic strategy learning and evaluation," in *Proc. NAACL Workshop on Adaptation in Dialogue Systems*, 2001, pp. 64–70.

[6] O. Pietquin, *A Framework for Unsupervised Learning of Dialogue Strategies*, Ph.D. thesis, Faculte Polytechnique de Mons, 2004.

[7] K. Georgila, J. Henderson, and O. Lemon, "Learning user simulations for information state update dialogue systems," *Proc. of Eurospeech (to appear)*, 2005.

[8] J. Schatzmann, K. Georgila, and S. Young, "Quantitative evaluation of user simulation techniques for spoken dialogue systems," *Proc. of SIGdial (to appear)*, 2005.

[9] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, 1992.

[10] M. A. Walker, R. J. Passonneau, and J. E. Boland, "Quantitative and qualitative evaluation of DARPA COMMUNICATOR spoken dialogue systems," in *Meeting of the Association of Computational Linguists*, 2001, pp. 515–522.

[11] K. Georgila, O. Lemon, and J. Henderson, "Automatic annotation of COMMUNICATOR dialogue data for learning dialogue strategies and user simulations," in *Proc. of DIALOR*, 2005.

[12] 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 ACL*, 1997, pp. 271–280, Madrid, Spain.

[13] J. Henderson, O. Lemon, and K. Georgila, "Hybrid reinforcement/supervised learning for dialogue policies from communicator data," in *Proc. of IJCAI*, 2005, Edinburgh, Scotland.