Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz data: Bootstrapping and Evaluation

Verena Rieser
School of Informatics
University of Edinburgh
Edinburgh, EH8 9LW, GB
vrieser@inf.ed.ac.uk

Oliver Lemon
School of Informatics
University of Edinburgh
Edinburgh, EH8 9LW, GB
olemon@inf.ed.ac.uk

Abstract

We address two problems in the field of automatic optimization of dialogue strategies: learning effective dialogue strategies when no initial data or system exists, and evaluating the result with real users. We use Reinforcement Learning (RL) to learn multimodal dialogue strategies by interaction with a simulated environment which is "bootstrapped" from small amounts of Wizard-of-Oz (WOZ) data. This use of WOZ data allows development of optimal strategies for domains where no working prototype is available. We compare the RL-based strategy against a supervised strategy which mimics the wizards' policies. This comparison allows us to measure relative improvement over the training data. Our results show that RL significantly outperforms Supervised Learning when interacting in simulation as well as for interactions with real users. The RL-based policy gains on average 50-times more reward when tested in simulation, and almost 18-times more reward when interacting with real users. Users also subjectively rate the RL-based policy on average 10% higher.

1 Introduction

Designing a spoken dialogue system is a time-consuming and challenging task. A developer may spend a lot of time and effort anticipating the potential needs of a specific application environment and then deciding on the most appropriate system action (e.g. confirm, present items,...). One of the key advantages of statistical optimisation methods, such as Reinforcement Learning (RL), for dialogue strategy design is that the problem can be formulated as a principled mathematical model which can be automatically trained on real data (Lemon and Pietquin, 2007; Frampton and Lemon, to appear). In cases where a system is designed from scratch, however, there is often no suitable in-domain data. Collecting dialogue data without a working prototype is problematic, leaving the developer with a classic chicken-and-egg problem.

We propose to learn dialogue strategies by simulation-based RL (Sutton and Barto, 1998), where the simulated environment is learned from small amounts of Wizard-of-Oz (WOZ) data. Using WOZ data rather than data from real Human-Computer Interaction (HCI) allows us to learn optimal strategies for domains where no working dialogue system already exists. To date, automatic strategy learning has been applied to dialogue systems which have already been deployed using handcrafted strategies. In such work, strategy learning was performed based on already present extensive online operation experience, e.g. (Singh et al., 2002; Henderson et al., 2005). In contrast to this preceding work, our approach enables strategy learning in domains where no prior system is available. Optimised learned strategies are then available from the first moment of online-operation, and tedious handcrafting of dialogue strategies is omitted. This independence from large amounts of in-domain dialogue data allows researchers to apply RL to new application areas beyond the scope of existing dialogue systems. We call this method 'bootstrapping'.

In a WOZ experiment, a hidden human operator, the so called "wizard", simulates (partly or com-

pletely) the behaviour of the application, while subjects are left in the belief that they are interacting with a real system (Fraser and Gilbert, 1991). That is, WOZ experiments only *simulate* HCI. We therefore need to show that a strategy bootstrapped from WOZ data indeed transfers to real HCI. Furthermore, we also need to introduce methods to learn useful user simulations (for training RL) from such limited data.

The use of WOZ data has earlier been proposed in the context of RL. (Williams and Young, 2004) utilise WOZ data to discover the state and action space for MDP design. (Prommer et al., 2006) use WOZ data to build a simulated user and noise model for simulation-based RL. While both studies show promising first results, their simulated environment still contains many hand-crafted aspects, which makes it hard to evaluate whether the success of the learned strategy indeed originates from the WOZ data. (Schatzmann et al., 2007) propose to 'bootstrap' with a simulated user which is entirely hand-crafted. In the following we propose an entirely data-driven approach, where all components of the simulated learning environment are learned from WOZ data. We also show that the resulting policy performs well for real users.

2 Wizard-of-Oz data collection

Our domains of interest are information-seeking dialogues, for example a multimodal in-car interface to a large database of music (MP3) files. The corpus we use for learning was collected in a multimodal study of German task-oriented dialogues for an in-car music player application by (Rieser et al., 2005). This study provides insights into natural methods of information presentation as performed by human wizards. 6 people played the role of an intelligent interface (the "wizards"). The wizards were able to speak freely and display search results on the screen by clicking on pre-computed templates. Wizards' outputs were not restricted, in order to explore the different ways they intuitively chose to present search results. Wizard's utterances were immediately transcribed and played back to the user with Text-To-Speech. 21 subjects (11 female, 10 male) were given a set of predefined tasks to perform, as well as a primary driving task, using a driving simulator. The users were able to speak, as well as make selections on the screen. We also introduced artificial noise in the setup, in order to closer resemble the conditions of real HCI. Please see (Rieser et al., 2005) for further detail.

The corpus gathered with this setup comprises 21 sessions and over 1600 turns. Example 1 shows a typical multimodal presentation sub-dialogue from the corpus (translated from German). Note that the wizard displays quite a long list of possible candidates on an (average sized) computer screen, while the user is driving. This example illustrates that even for humans it is difficult to find an "optimal" solution to the problem we are trying to solve.

(1) **User:** Please search for music by Madonna .
Wizard: I found seventeen hundred and eleven items. The items are displayed on the screen. *[displays list]*
User: Please select 'Secret'.

For each session information was logged, e.g. the transcriptions of the spoken utterances, the wizard's database query and the number of results, the screen option chosen by the wizard, and a rich set of contextual dialogue features was also annotated, see (Rieser et al., 2005).

Of the 793 wizard turns 22.3% were annotated as presentation strategies, resulting in 177 instances for learning, where the six wizards contributed about equal proportions.

Information about user preferences was obtained, using a questionnaire containing similar questions to the PARADISE study (Walker et al., 2000). In general, users report that they get distracted from driving if too much information is presented. On the other hand, users prefer shorter dialogues (most of the user ratings are negatively correlated with dialogue length). These results indicate that we need to find a strategy given the competing trade-offs between the number of results (large lists are difficult for users to process), the length of the dialogue (long dialogues are tiring, but collecting more information can result in more precise results), and the noise in the speech recognition environment (in high noise conditions accurate information is difficult to obtain). In the following we utilise the ratings from the user questionnaires to optimise a presentation strategy using simulation-based RL.

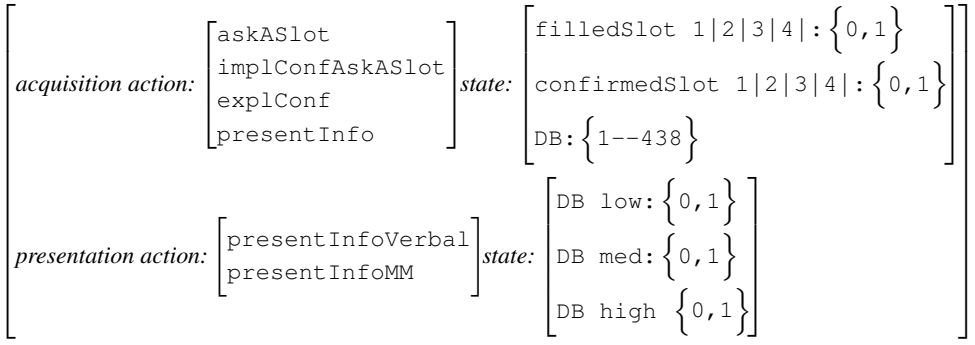


Figure 1: State-Action space for hierarchical Reinforcement Learning

3 Simulated Learning Environment

Simulation-based RL (also know as "model-free" RL) learns by interaction with a simulated environment. We obtain the simulated components from the WOZ corpus using data-driven methods. The employed database contains 438 items and is similar in retrieval ambiguity and structure to the one used in the WOZ experiment. The dialogue system used for learning comprises some obvious constraints reflecting the system logic (e.g. that only filled slots can be confirmed), implemented as Information State Update (ISU) rules. All other actions are left for optimisation.

3.1 MDP and problem representation

The structure of an information seeking dialogue system consists of an information acquisition phase, and an information presentation phase. For information acquisition the task of the dialogue manager is to gather 'enough' search constraints from the user, and then, 'at the right time', to start the information presentation phase, where the presentation task is to present 'the right amount' of information in the right way– either on the screen or listing the items verbally. What 'the right amount' actually means depends on the application, the dialogue context, and the preferences of users. For optimising dialogue strategies information acquisition and presentation are two closely interrelated problems and need to be optimised simultaneously: *when* to present information depends on the available options for *how* to present them, and vice versa. We therefore formulate the problem as a Markov Decision Process (MDP), relating states to actions in a hierarchical manner (see Figure 1): 4 actions are available for the information acquisition phase; once the action `presentInfo` is chosen, the information presentation phase is entered, where 2 different actions for output realisation are available. The state-space comprises 8 binary features representing the task for a 4 slot problem: `filledSlot` indicates whether a slots is filled, `confirmedSlot` indicates whether a slot is confirmed. We also add features that human wizards pay attention to, using the feature selection techniques of (Rieser and Lemon, 2006b). Our results indicate that wizards only pay attention to the number of retrieved items (`DB`). We therefore add the feature `DB` to the state space, which takes integer values between 1 and 438, resulting in $\underline{2^8 \times 438 = 112,128}$ distinct dialogue states. In total there are $\underline{4^{112,128}}$ theoretically possible policies for information acquisition. [1] For the presentation phase the `DB` feature is discretised, as we will further discuss in Section 3.6. For the information presentation phase there are $\underline{2^{2^3} = 256}$ theoretically possible policies.

3.2 Supervised Baseline

We create a baseline by applying Supervised Learning (SL). This baseline mimics the average wizard behaviour and allows us to measure the relative improvements over the training data (cf. (Henderson et al., 2005)). For these experiments we use the WEKA toolkit (Witten and Frank, 2005). We learn with the decision tree J4.8 classifier, WEKA's implementation of the C4.5 system (Quinlan, 1993), and rule induc-

[1] In practise, the policy space is smaller, as some of combinations are not possible, e.g. a slot cannot be confirmed before being filled. Furthermore, some incoherent action choices are excluded by the basic system logic.

	baseline	JRip	J48
timing	52.0(± 2.2)	50.2(± 9.7)	53.5(±11.7)
modality	51.0(± 7.0)	93.5(±11.5)*	94.6(± 10.0)*

Table 1: Predicted accuracy for presentation timing and modality (with standard deviation ±), * denotes statistically significant improvement at $p < .05$

tion JRIP, the WEKA implementation of RIPPER (Cohen, 1995). In particular, we learn models which predict the following wizard actions:

- Presentation timing: *when* the 'average' wizard starts the presentation phase
- Presentation modality: in *which modality* the list is presented.

As input features we use annotated dialogue context features, see (Rieser and Lemon, 2006b). Both models are trained using 10-fold cross validation. Table 1 presents the results for comparing the accuracy of the learned classifiers against the majority baseline. For presentation timing, none of the classifiers produces significantly improved results. Hence, we conclude that there is no distinctive pattern the wizards follow for *when* to present information. For strategy implementation we therefore use a frequency-based approach following the distribution in the WOZ data: in 0.48 of cases the baseline policy decides to present the retrieved items; for the rest of the time the system follows a hand-coded strategy. For learning presentation modality, both classifiers significantly outperform the baseline. The learned models can be rewritten as in Algorithm 1. Note that this rather simple algorithm is meant to represent the average strategy as present in the initial data (which then allows us to measure the relative improvements of the RL-based strategy).

Algorithm 1 $SupervisedStrategy$

```
1: if DB ≤ 3 then
2:     return presentInfoVerbal
3: else
4:     return presentInfoMM
5: end if
```

3.3 Noise simulation

One of the fundamental characteristics of HCI is an error prone communication channel. Therefore, the simulation of channel noise is an important aspect of the learning environment. Previous work uses data-intensive simulations of ASR errors, e.g. (Pietquin and Dutoit, 2006). We use a simple model simulating the effects of non- and misunderstanding on the interaction, rather than the noise itself. This method is especially suited to learning from small data sets. From our data we estimate a 30% chance of user utterances to be misunderstood, and 4% to be complete non-understandings. We simulate the effects noise has on the user behaviour, as well as for the task accuracy. For the user side, the noise model defines the likelihood of the user accepting or rejecting the system's hypothesis (for example when the system utters a confirmation), i.e. in 30% of the cases the user rejects, in 70% the user agrees. These probabilities are combined with the probabilities for user actions from the user simulation, as described in the next section. For non-understandings we have the user simulation generating Out-of-Vocabulary utterances with a chance of 4%. Furthermore, the noise model determines the likelihood of task accuracy as calculated in the reward function for learning. A filled slot which is not confirmed by the user has a 30% chance of having been mis-recognised.

3.4 User simulation

A user simulation is a predictive model of real user behaviour used for automatic dialogue strategy development and testing. For our domain, the user can either add information (`add`), repeat or paraphrase information which was already provided at an earlier stage (`repeat`), give a simple yes-no answer (`y/n`), or change to a different topic by providing a different slot value than the one asked for (`change`). These actions are annotated manually ($\kappa = .7$). We build two different types of user simulations, one is used for strategy training, and one for testing. Both are simple bi-gram models which predict the next user action based on the previous system action ($P(a_{user}|a_{system})$). We face the problem of learning such models when training data is sparse. For training, we therefore use a cluster-based user simulation method, see (Rieser

and Lemon, 2006a). For testing, we apply smoothing to the bi-gram model. The simulations are evaluated using the SUPER metric proposed earlier (Rieser and Lemon, 2006a), which measures variance and consistency of the simulated behaviour with respect to the observed behaviour in the original data set. This technique is used because for training we need more variance to facilitate the exploration of large state-action spaces, whereas for testing we need simulations which are more realistic. Both user simulations significantly outperform random and majority class baselines. See (Rieser, 2008) for further details.

3.5 Reward modelling

The reward function defines the goal of the overall dialogue. For example, if it is most important for the dialogue to be efficient, the reward penalises dialogue length, while rewarding task success. In most previous work the reward function is manually set, which makes it "the most hand-crafted aspect" of RL (Paek, 2006). In contrast, we learn the reward model from data, using a modified version of the PARADISE framework (Walker et al., 2000), following pioneering work by (Walker et al., 1998). In PARADISE multiple linear regression is used to build a predictive model of subjective user ratings (from questionnaires) from objective dialogue performance measures (such as dialogue length). We use PARADISE to predict Task Ease (a variable obtained by taking the average of two questions in the questionnaire) [2] from various input variables, via stepwise regression. The chosen model comprises dialogue length in turns, task completion (as manually annotated in the WOZ data), and the multimodal user score from the user questionnaire, as shown in Equation 2.

$$TaskEase = -\mathbf{20.2} * dialogueLength + \mathbf{11.8} * taskCompletion + \mathbf{8.7} * multimodalScore; \quad (2)$$

This equation is used to calculate the overall reward for the information acquisition phase. During learning, Task Completion is calculated online according to the noise model, penalising all slots which are filled but not confirmed.

[2] "The task was easy to solve.", "I had *no* problems finding the information I wanted."

For the information presentation phase, we compute a local reward. We relate the multimodal score (a variable obtained by taking the average of 4 questions) [3] to the number of items presented (DB) for each modality, using curve fitting. In contrast to linear regression, curve fitting does not assume a linear inductive bias, but it selects the most likely model (given the data points) by function interpolation. The resulting models are shown in Figure 3.5. The reward for multimodal presentation is a quadratic function that assigns a maximal score to a strategy displaying 14.8 items (curve inflection point). The reward for verbal presentation is a linear function assigning negative scores to all presented items ≤ 4. The reward functions for information presentation intersect at no. items=3. A comprehensive evaluation of this reward function can be found in (Rieser and Lemon, 2008a).

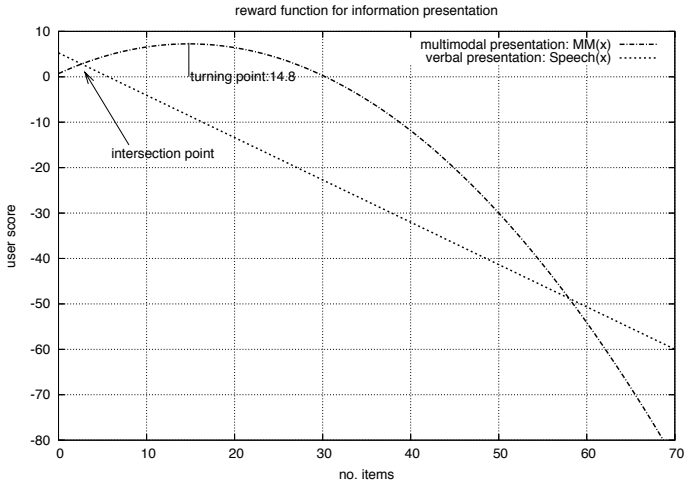


Figure 2: Evaluation functions relating number of items presented in different modalities to multimodal score

3.6 State space discretisation

We use linear function approximation in order to learn with large state-action spaces. Linear function approximation learns linear estimates for expected reward values of actions in states represented as feature vectors. This is inconsistent with the idea

[3] "I liked the combination of information being displayed on the screen and presented verbally.", "Switching between modes did *not* distract me.", "The displayed lists and tables contained on average the right amount of information.", "The information presented verbally was easy to remember."

of non-linear reward functions (as introduced in the previous section). We therefore quantise the state space for information presentation. We partition the database feature into 3 bins, taking the first intersection point between verbal and multimodal reward and the turning point of the multimodal function as discretisation boundaries. Previous work on learning with large databases commonly quantises the database feature in order to learn with large state spaces using manual heuristics, e.g. (Levin et al., 2000; Heeman, 2007). Our quantisation technique is more principled as it reflects user preferences for multi-modal output. Furthermore, in previous work database items were not only quantised in the state-space, but also in the reward function, resulting in a direct mapping between quantised retrieved items and discrete reward values, whereas our reward function still operates on the continuous values. In addition, the decision *when* to present a list (information acquisition phase) is still based on continuous DB values. In future work we plan to engineer new state features in order to learn with non-linear rewards while the state space is still continuous. A continuous representation of the state space allows learning of more fine-grained local trade-offs between the parameters, as demonstrated by (Rieser and Lemon, 2008b).

3.7 Testing the Learned Policies in Simulation

We now train and test the multimodal presentation strategies by interacting with the simulated learning environment. For the following RL experiments we used the REALL-DUDE toolkit of (Lemon et al., 2006b). The SHARSHA algorithm is employed for training, which adds hierarchical structure to the well known SARSA algorithm (Shapiro and Langley, 2002). The policy is trained with the cluster-based user simulation over 180k system cycles, which results in about 20k simulated dialogues. In total, the learned strategy has 371 distinct state-action pairs (see (Rieser, 2008) for details).

We test the RL-based and supervised baseline policies by running 500 test dialogues with a smoothed user simulation (so that we are not training and testing on the same simulation). We then compare quantitative dialogue measures performing a paired t-test. In particular, we compare mean values of the final rewards, number of filled and confirmed slots, dialog length, and items presented multimodally (`MM items`) and items presented verbally (`verbal items`). RL performs significantly better ($p < .001$) than the baseline strategy. The only non-significant difference is the number of items presented verbally, where both RL and SL strategy settled on a threshold of less than 4 items. The mean performance measures for simulation-based testing are shown in Table 2 and Figure 3.

The major strength of the learned policy is that it learns to keep the dialogues reasonably short (on average 5.9 system turns for RL versus 8.4 turns for SL) by presenting lists as soon as the number of retrieved items is within tolerance range for the respective modality (as reflected in the reward function). The SL strategy in contrast has not learned the right timing nor an upper bound for displaying items on the screen. The results show that simulation-based RL with an environment bootstrapped from WOZ data allows learning of robust strategies which significantly outperform the strategies contained in the initial data set.

One major advantage of RL is that it allows us to provide additional information about user preferences in the reward function, whereas SL simply mimics the data. In addition, RL is based on delayed rewards, i.e. the optimisation of a final goal. For dialogue systems we often have measures indicating how successful and/or satisfying the overall performance of a strategy was, but it is hard to tell how things should have been exactly done in a specific situation. This is what makes RL specifically attractive for dialogue strategy learning. In the next section we test the learned strategy with real users.

4 User Tests

4.1 Experimental design

For the user tests the RL policy is ported to a working ISU-based dialogue system via table look-up, which indicates the action with the highest expected reward for each state (cf. (Singh et al., 2002)). The supervised baseline is implemented using standard threshold-based update rules. The experimental conditions are similar to the WOZ study, i.e. we ask the users to solve similar tasks, and use similar questionnaires. Furthermore, we decided to use typed user input rather than ASR. The use of text input

Measure	SL baseline		RL Strategy	
	SIM	REAL	SIM	REAL
av. turns	8.42(±3.04)	5.86(±3.2)	5.9(±2.4)***	5.07(±2.9)***
av. speech items	1.04(±.2)	1.29(±.4)	1.1(±.3)	1.2(±.4)
av. MM items	61.37(±82.5)	52.2(±68.5)	11.2(±2.4)***	8.73(±4.4)***
av. reward	-1741.3(±566.2)	-628.2(±178.6)	44.06(±51.5)***	37.62(±60.7)***

Table 2: Comparison of results obtained in simulation (SIM) and with real users (REAL) for SL and RL-based strategies; *** denotes significant difference between SL and RL at $p < .001$

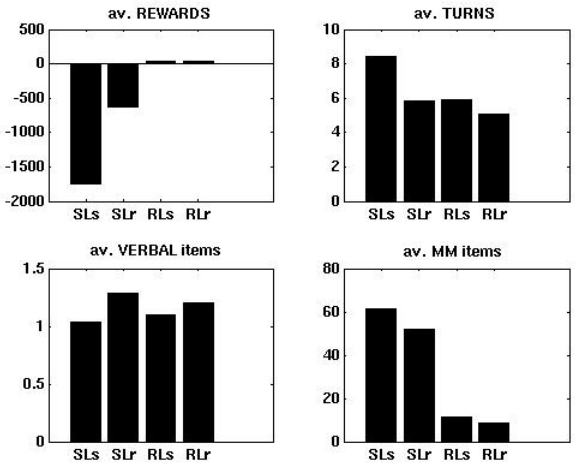


Figure 3: Graph comparison of objective measures: SLs = SL policy in simulation; SLr = SL policy with real users; RLs = RL policy in simulation; RLr = RL policy with real users.

allows us to target the experiments to the dialogue management decisions, and block ASR quality from interfering with the experimental results (Hajdinjak and Mihelic, 2006). 17 subjects (8 female, 9 male) are given a set of 6×2 predefined tasks, which they solve by interaction with the RL-based and the SL-based system in controlled order. As a secondary task users are asked to count certain objects in a driving simulation. In total, 204 dialogues with 1,115 turns are gathered in this setup.

4.2 Results

In general, the users rate the RL-based significantly higher ($p < .001$) than the SL-based policy. The results from a paired t-test on the user questionnaire data show significantly improved Task Ease, better presentation timing, more agreeable verbal and multimodal presentation, and that more users would use the RL-based system in the future (Future Use). All the observed differences have a medium effects size ($r \geq |.3|$).

We also observe that female participants clearly favour the RL-based strategy, whereas the ratings by male participants are more indifferent. Similar gender effects are also reported by other studies on multimodal output presentation, e.g. (Foster and Oberlander, 2006).

Furthermore, we compare objective dialogue performance measures. The dialogues of the RL strategy are significantly shorter ($p < .005$), while fewer items are displayed ($p < .001$), and the help function is used significantly less ($p < .003$). The mean performance measures for testing with real users are shown in Table 2 and Figure 3. However, there is no significant difference for the performance of the secondary driving task.

5 Comparison of Results

We finally test whether the results obtained in simulation transfer to tests with real users, following (Lemon et al., 2006a). We evaluate the quality of the simulated learning environment by directly comparing the dialogue performance measures between simulated and real interaction. This comparison enables us to make claims regarding whether a policy which is 'bootstrapped' from WOZ data is transferable to real HCI. We first evaluate whether objective dialogue measures are transferable, using a paired t-test. For the RL policy there is no statistical difference in overall performance (reward), dialogue length (turns), and the number of presented items (verbal and multimodal items) between simulated

Measure	**WOZ**	**SL**	**RL**
av. Task Ease	.53±.14	.63±.26	.79±.21***
av. Future Use	.56±.16	.55±.21	.67±.20***

Table 3: Improved user ratings over the WOZ study where *** denotes $p < .001$

and real interaction (see Table 2, Figure 3). This indicates that the learned strategy transfers well to real settings. For the SL policy the dialogue length for real users is significantly shorter than in simulation. From an error analysis we conclude that real users intelligently adapt to poor policies, e.g. by changing topic, whereas the simulated users do not react in this way.

Furthermore, we want to know whether the subjective user ratings for the RL strategy improved over the WOZ study. We therefore compare the user ratings from the WOZ questionnaire to the user ratings of the final user tests using a independent t-test and a Wilcoxon Signed Ranks Test. Users rate the RL-policy on average 10% higher. We are especially interested in the ratings for Task Ease (as this was the ultimate measure optimised with PARADISE) and Future Use, as we believe this measure to be an important indicator of acceptance of the technology. The results show that only the RL strategy leads to significantly improved user ratings (increasing average Task Ease by 49% and Future Use by 19%), whereas the ratings for the SL policy are not significantly better than those for the WOZ data, see Table 3. [4] This indicates that the observed difference is indeed due to the improved strategy (and not to other factors like the different user population or the embedded dialogue system).

6 Conclusion

We addressed two problems in the field of automatic optimization of dialogue strategies: learning effective dialogue strategies when no initial data or system exists, and evaluating the result with real users. We learned optimal strategies by interaction with a simulated environment which is bootstrapped from a small amount of Wizard-of-Oz data, and we evaluated the result with real users. The use of WOZ data allows us to develop optimal strategies for domains where no working prototype is available. The developed simulations are entirely data driven and the reward function reflects real user preferences. We compare the Reinforcement Learning-based strategy against a supervised strategy which mimics the (human) wizards' policies from the original data. This comparison allows us to measure relative improvement over the training data. Our results show that RL significantly outperforms SL in simulation as well as in interactions with real users. The RL-based policy gains on average 50-times more reward when tested in simulation, and almost 18-times more reward when interacting with real users. The human users also subjectively rate the RL-based policy on average 10% higher, and 49% higher for Task Ease. We also show that results obtained in simulation are comparable to results for real users. We conclude that a strategy trained from WOZ data via bootstrapping is transferable to real Human-Computer-Interaction.

In future work will apply similar techniques to statistical planning for Natural Language Generation in spoken dialogue (Lemon, 2008; Janarthanam and Lemon, 2008), (see the EC FP7 CLASSiC project: `www.classic-project.org`).

Acknowledgements

The research leading to these results has received funding from the European Community's 7th Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSiC project `www.classic-project.org`), the EC FP6 project "TALK: Talk and Look, Tools for Ambient Linguistic Knowledge (IST 507802, `www.talk-project.org`), from the EPSRC, project no. EP/E019501/1, and from the IRTG Saarland University.

[4] The ratings are normalised as some of the questions were on different scales.

References

W. W. Cohen. 1995. Fast effective rule induction. In *Proc. of the 12th ICML-95*.

M. E. Foster and J. Oberlander. 2006. Data-driven generation of emphatic facial displays. In *Proc. of EACL*.

M. Frampton and O. Lemon. (to appear). Recent research advances in Reinforcement Learning in Spoken Dialogue Systems. *Knowledge Engineering Review*.

N. M. Fraser and G. N. Gilbert. 1991. Simulating speech systems. *Computer Speech and Language*, 5:81–99.

M. Hajdinjak and F. Mihelic. 2006. The PARADISE evaluation framework: Issues and findings. *Computational Linguistics*, 32(2):263–272.

P. Heeman. 2007. Combining reinforcement learning with information-state update rules. In *Proc. of NAACL*.

J. Henderson, O. Lemon, and K. Georgila. 2005. Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR data. In *Proc. of IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 68–75.

S. Janarthanam and O. Lemon. 2008. User simulations for online adaptation and knowledge-alignment in Troubleshooting dialogue systems. In *Proc. of the 12th SEMDIAL Workshop (LONdial)*.

O. Lemon and O. Pietquin. 2007. Machine learning for spoken dialogue systems. In *Proc. of Interspeech*.

O. Lemon, K. Georgila, and J. Henderson. 2006a. Evaluating Effectiveness and Portability of Reinforcement Learned Dialogue Strategies with real users: the TALK TownInfo Evaluation. In *Proc. of IEEE/ACL workshop on Spoken Language Technology (SLT)*.

O. Lemon, X. Liu, D. Shapiro, and C. Tollander. 2006b. Hierarchical reinforcement learning of dialogue policies in a development environment for dialogue systems: REALL-DUDE. In *Proc. of the 10th SEMDIAL Workshop (BRANdial)*.

O. Lemon. 2008. Adaptive Natural Language Generation in Dialogue using Reinforcement Learning. In *Proc. of the 12th SEMDIAL Workshop (LONdial)*.

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).

T. Paek. 2006. Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical deployment. In *Proc. Dialog-on-Dialog Workshop, Interspeech*.

O. Pietquin and T. Dutoit. 2006. A probabilistic framework for dialog simulation and optimal strategy learnin. *IEEE Transactions on Audio, Speech and Language Processing*, 14(2):589–599.

T. Prommer, H. Holzapfel, and A. Waibel. 2006. Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human-robot interaction. In *Proc. of Interspeech/ICSLP*.

R. Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

V. Rieser and O. Lemon. 2006a. Cluster-based user simulations for learning dialogue strategies. In *Proc. of Interspeech/ICSLP*.

V. Rieser and O. Lemon. 2006b. Using machine learning to explore human multimodal clarification strategies. In *Proc. of ACL*.

V. Rieser and O. Lemon. 2008a. Automatic learning and evaluation of user-centered objective functions for dialogue system optimisation. In *LREC*.

V. Rieser and O. Lemon. 2008b. Does this list contain what you were searching for? Learning adaptive dialogue strategies for interactive question answering. *Journal of Natural Language Engineering (special issue on Interactive Question answering, to appear)*.

V. Rieser, I. Kruijff-Korbayová, and O. Lemon. 2005. A corpus collection and annotation framework for learning multimodal clarification strategies. In *Proc. of the 6th SIGdial Workshop*.

V. Rieser. 2008. *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data (to appear)*. Ph.D. thesis, Saarland University.

J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young. 2007. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *Proc. of HLT/NAACL*.

D. Shapiro and P. Langley. 2002. Separating skills from preference: Using learning to program by reward. In *Proc. of the 19th ICML*.

S. Singh, D. Litman, M. Kearns, and M. Walker. 2002. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *JAIR*, 16.

R. Sutton and A. Barto. 1998. *Reinforcement Learning*. MIT Press.

M. Walker, J. Fromer, and S. Narayanan. 1998. Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email. In *Proceedings of ACL/COLING*.

M. Walker, C. Kamm, and D. Litman. 2000. Towards developing general models of usability with PARADISE. *Journal of Natural Language Engineering*, 6(3).

J. Williams and S. Young. 2004. Using Wizard-of-Oz simulations to bootstrap reinforcement-learning-based dialog management systems. In *Proc. of the 4th SIGdial Workshop*.

I. Witten and E. Frank. 2005. *Data Mining: Practical Machine Learning Tools and Techniques (2nd Edition)*. Morgan Kaufmann.