

# Simulation-based Learning of Optimal Multimodal Presentation Strategies from Wizard-of-Oz data

Verena Rieser and Oliver Lemon<sup>1</sup>

**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 optimising dialogue management (DM) and Natural Language Generation (NLG) decisions in an integrated fashion. We use Reinforcement Learning (RL) to learn multimodal information presentation strategies through 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. For information seeking dialogues, Dialogue Management and NLG are two closely interrelated problems: the decision of *when* to present information depends on the available options for *how* to present them, and vice versa. We therefore formulate the problem as a hierarchy of joint learning decisions which are optimised together. To evaluate, we compare the RL-based strategy against a supervised learning (SL) 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: the RL-based policy gains on average 50-times more reward when tested in simulation. In related work we evaluate the strategies with real users [16].

## 1 Introduction

One of the key advantages of statistical optimisation methods (e.g. Reinforcement Learning (RL)) for dialogue strategy design is that the problem can be formulated as a precise mathematical model which can be trained on real data [5]. 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.

Here, we learn dialogue strategies by simulation-based RL [20], 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 exists. To date, automatic strategy learning has been applied to dialogue systems which have already been deployed in the real world using hand-crafted strategies. In such work, strategy learning was performed based on already present extensive online-operation experience, e.g. [19, 3]. 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 avoided. 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’.

The use of WOZ data has earlier been proposed in the context of RL. [23] utilise WOZ data to discover the state and action space for MDP design. [9] use WOZ data to build a simulated user and noise model for simulation-based RL. While both studies show promising first results, their simulated environments still contain many hand-crafted aspects, which makes it hard to evaluate whether the success of the learned strategy indeed originates from the WOZ data. In addition, [17] 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.

## 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 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 [12]. 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. Please see [12] for further detail.

The corpus gathered with this setup comprises 21 sessions and over 16K turns. Example 1 shows a typical multimodal presentation sub-dialogue (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 [12]. Of the 793 wizard turns 22.3% were annotated as presentation strategies, result-

<sup>1</sup> School of Informatics, University of Edinburgh, UK, email: {vrieser,olemon}@inf.ed.ac.uk

ing 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 [22]. 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 related 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.

### 3 Simulated Learning Environment

Simulation-based RL (aka 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 low level constraints reflecting the system logic (e.g. that only filled slots can be confirmed), implemented as Information State Update (ISU) rules. The higher level 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 Natural Language Generation task is to present ‘the right amount’ of information – either on the screen or listing the items verbally. What this 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 human wizards pay attention to, using the feature selection techniques of [14]. 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  $2^8 \times 438 = 112,128$  distinct dialogue states. In total there are  $4^{112,128}$  theoretically possible policies for information acquisition.<sup>2</sup> For the presentation phase the DB feature is discretised, as we will further discuss in Section 3.6.

<sup>2</sup> 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 action choices are excluded by the basic system logic.

For the information presentation phase there are  $2^{2^3} = 256$  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. [3]). For our experiments we use the WEKA toolkit [24]. We learn with the decision tree J4.8 classifier, WEKA’s implementation of the C4.5 system [10], and rule induction JRIP, the WEKA implementation of RIPPER [1]. 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.

	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  $\pm$ ), \* statistically significant improvement,  $p < .05$

As input features we use annotated dialogue context features, see [14]. 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 scale back to a frequency-based approach following the distribution in the WOZ data: in 0.48 of the times 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

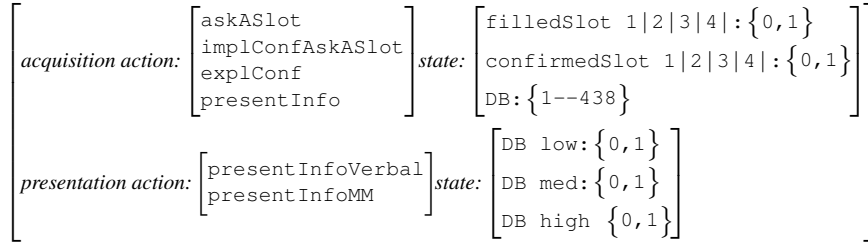
```

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. [8]. 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, 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%.



**Figure 1.** State-Action space for hierarchical Reinforcement Learning

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. 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, 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 [13]. For testing, we apply smoothing to the bi-gram model. The simulations are evaluated using the SUPER metric proposed earlier [13], 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.

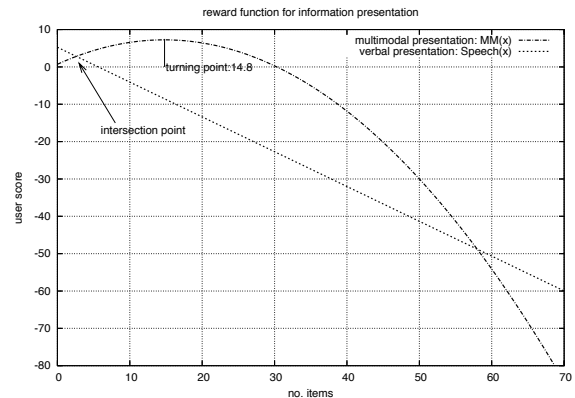
### 3.5 Reward modelling

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 [7]. In contrast, we learn the reward model from data, using a modified version of the PARADISE framework [22], following pioneering work by [21]. 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 user ratings from the questionnaire) 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.

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

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.

For the information presentation phase, we compute a local reward. We relate the multimodal score (a variable obtained by taking the average of 4 user ratings from the questionnaire) 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  $\leq 4$ . The reward functions for information presentation intersect at no. items=3.



**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 of non-linear reward functions (as introduced in the last 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. [6, 2]. 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 [15].

### 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 [4]. The SHARSHA algorithm is employed for training, which adds hierarchical structure to the well known SARSA algorithm [18]. 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 [11] for details).

We test the RL-based and supervised baseline policies by running 500 test dialogues with the smoothed user simulation. We then compare quantitative dialogue measures performing a paired t-test (with pair-wise exclusion of missing values). 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. The major strength of the learned policy is that it learns to keep the dialogues reasonably short 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. In contrast to SL, programming by reward allows us to provide additional information in the reward function, and therefore enables learning a policy which reflects the user preferences. In related work we evaluate the learned strategy with real users (see [11, 16]).

	SL baseline	RL strategy
reward	-1747.3 ( $\pm 527.6$ )	37.3 ( $\pm 54.5$ )***
dialog length	8.7 ( $\pm 3.7$ )	6.3 ( $\pm 3.1$ )***
verbal items	1.1 ( $\pm .28$ )	1.0 ( $\pm .31$ )
MM items	59.78 ( $\pm 74.2$ )	11.5 ( $\pm 2.2$ )***

**Table 2.** Comparison of mean performance for SL and RL policies (with standard deviation  $\pm$ ); \*\*\* denotes statistical significance at  $p < .001$

## 4 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 optimising DM and NLG decisions in an integrated fashion. We used a simulated environment which is “bootstrapped” from small amounts of WOZ data, thus allowing strategy optimization for domains where no working prototype is available. We compared the RL-based strategy against a supervised strategy which mimics the human wizards’ policy from the original data. Our results show that RL significantly outperforms Supervised Learning: the RL-based policy gains on average 50-times more reward when tested in simulation. In related work we evaluate the learned strategy with real users [16].

## ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 216594 (CLASSIC project [www.classic-project.org](http://www.classic-project.org)) and from the EPSRC, project no. EP/E019501/1 and from the ITRG Saarland University.

## REFERENCES

- [1] W. W. Cohen, ‘Fast effective rule induction’, in *Proc. of the 12th ICML-95*, (1995).
- [2] P. Heeman, ‘Combining reinforcement learning with information-state update rules.’, in *Proc. of NAACL*, pp. 268–275, (2007).
- [3] J. Henderson, O. Lemon, and K. Georgila, ‘Hybrid Reinforcement/Supervised Learning for Dialogue Policies from COMMUNICATOR data’, in *IJCAI workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pp. 68–75, (2005).
- [4] O. Lemon, X. Liu, D. Shapiro, and C. Tollander, ‘Hierarchical reinforcement learning of dialogue policies in a development environment for dialogue systems: REALL-DUDE’, in *BRANDIAL*, (2006).
- [5] O. Lemon and O. Pietquin, ‘Machine learning for spoken dialogue systems’, in *Proc. of Interspeech*, (2007).
- [6] E. Levin, R. Pieraccini, and W. Eckert, ‘A stochastic model of human-machine interaction for learning dialog strategies’, *IEEE Transactions on Speech and Audio Processing*, 8(1), (2000).
- [7] T. Paek, ‘Reinforcement learning for spoken dialogue systems: Comparing strengths and weaknesses for practical deployment’, in *Proc. Dialog-on-Dialog Workshop, Interspeech*, (2006).
- [8] O. Pietquin and T. Dutoit, ‘A probabilistic framework for dialog simulation and optimal strategy learning’, *IEEE Transactions on Audio, Speech and Language Processing*, 14(2), 589–599, (2006).
- [9] T. Prommer, H. Holzapfel, and A. Waibel, ‘Rapid simulation-driven reinforcement learning of multimodal dialog strategies in human-robot interaction’, in *Proc. of Interspeech/ICSLP*, (2006).
- [10] R. Quinlan, *C4.5: Programs for Machine Learning.*, Morgan Kaufmann, 1993.
- [11] V. Rieser, *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data (to appear)*, Ph.D. dissertation, Saarland University, 2008.
- [12] V. Rieser, I. Kruijff-Korbayová, and O. Lemon, ‘A corpus collection and annotation framework for learning multimodal clarification strategies’, in *Proc. of the 6th SIGdial Workshop*, (2005).
- [13] V. Rieser and O. Lemon, ‘Cluster-based user simulations for learning dialogue strategies’, in *Proc. of Interspeech/ICSLP*, (2006).
- [14] V. Rieser and O. Lemon, ‘Using machine learning to explore human multimodal clarification strategies’, in *Proc. of ACL*, (2006).
- [15] V. Rieser and O. Lemon, ‘Does this list contain what you were searching for? learning adaptive dialogue strategies for interactive question answering’, *JNLE (special issue on Interactive Question answering, to appear)*, (2008).
- [16] V. Rieser and O. Lemon, ‘Learning effective multimodal dialogue strategies from wizard-of-oz data: Bootstrapping and evaluation (to appear)’, in *Proc. of ACL*, (2008).
- [17] J. Schatzmann, B. Thomson, K. Weilhammer, H. Ye, and S. Young, ‘Agenda-based user simulation for bootstrapping a POMDP dialogue system’, in *Proc. of HLT/NAACL*, (2007).
- [18] D. Shapiro and P. Langley, ‘Separating skills from preference: Using learning to program by reward’, in *Proc. of the 19th ICML*, (2002).
- [19] S. Singh, D. Litman, M. Kearns, and M. Walker, ‘Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system’, *JAIR*, 16, (2002).
- [20] R. Sutton and A. Barto, *Reinforcement Learning*, MIT Press, 1998.
- [21] M. Walker, J. Fromer, and S. Narayanan, ‘Learning optimal dialogue strategies: A case study of a spoken dialogue agent for email’, in *Proceedings of ACL/COLING*, (1998).
- [22] M. Walker, C. Kamm, and D. Litman, ‘Towards developing general models of usability with PARADISE’, *JNLE*, 6(3), (2000).
- [23] J. Williams and S. Young, ‘Using Wizard-of-Oz simulations to bootstrap reinforcement-learning-based dialog management systems’, in *Proc. of the 4th SIGdial Workshop*, (2004).
- [24] I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition)*, Morgan Kaufmann, 2005.