# A Data-driven method for Adaptive Referring Expression Generation in Automated Dialogue Systems: Maximising Expected Utility

**Srinivasan Janarthanam (s.janarthanam@ed.ac.uk)**
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB

**Oliver Lemon (olemon@inf.ed.ac.uk)**
School of Informatics
University of Edinburgh
Edinburgh EH8 9AB

**Abstract**

Adaptive generation of referring expressions in spoken dialogue systems is beneficial in terms of improving grounding and mutual understanding between dialogue partners: a system and its human user. For example, an automated system may need to change its referring behaviour depending on a continuously updated estimate of the user's lexical knowledge or domain expertise. However, hand-coding such adaptive referring expression generation (REG) policies is difficult, and is not guaranteed to approach an optimal solution. Here we present a data-driven Reinforcement Learning framework to automatically learn adaptive REG policies for spoken dialogue systems. These policies maximise the expected utility of the REG choices made by the system, with respect to statistical models of the behaviour and linguistic knowledge of different users. Here we present preliminary results obtained when training a REG policy using such a model.

**Keywords:** Referring expression generation; reinforcement learning; user simulation

## Introduction

Referring expression generation (REG) is the natural language generation (NLG) problem of choosing the referring expressions for use in utterances to refer to various domain objects. Adaptive REG helps in efficient grounding between dialogue partners (Isaacs & Clark, 1987), improving task success rates and even increasing learning gain. For instance, in a technical support task, the dialogue agent could use technical jargon with experts, descriptive expressions with beginners and a mixture of the two with intermediate users. Similarly, in a city navigation task, the dialogue agent could use proper names for landmarks with locals but descriptive expressions with foreign tourists. Speakers designing their utterances for a particular audience at a particular time is called `audience design` (Clark & Murphy, 1982). This kind of alignment is a relatively simple problem when the system interacts with a known user. However, it is impossible to predict the expertise of an unknown user, so we need adaptive REG policies which can estimate a new user's linguistic expertise based on their utterances so far in a dialogue. Hand-coding such an adaptive REG policy is a difficult task, and there is no guarantee that a hand-coded solution performs optimally.

To address such issues (Lemon, 2008) first presented the case for treating NLG in dialogue as a Reinforcement Learning or statistical planning problem. Recently, (Deemter, 2009) discussed the application of `Game Theory` to NLG decision problems (utility based decision making).

In this paper, we extend the reinforcement learning framework to automatically learn a REG policy that aligns to different kinds of users dynamically during dialogues through `Audience Design` . Such a policy will maximise the expected utility of the REG decisions made by the system. We describe this framework in the context of a technical support dialogue task. In the section "Related Work", we present some of the previous work related in generating referring expressions and reinforcement learning for dialogue management strategies. In the section "The Reinforcement Learning Framework", we present the reinforcement learning framework for learning referring expression generation policies. The section "Preliminary Experiments" presents the results of our preliminary experiments using this framework.

## Related work

Prior work in generating referring expressions include several content determination or attribute selection algorithms to determine which attributes, like the colour, shape or size, be used to refer to the domain objects (Dale, 1989; Reiter & Dale, 1992, 1995; Gatt & Belz, 2008). The selection criteria is based on principles like Gricean quantity maxim (Grice, 1975), etc. However, these referring expressions are only descriptive in nature. In contrast, our aim is to choose between descriptive expressions and technical terms referring to the domain objects based on the user's expertise and business requirements.

Reinforcement Learning (RL) has been successfully used for learning dialogue management policies (Levin, Pieraccini, & Eckert, 1997). The learned policies allow the dialogue manager to optimally choose appropriate dialogue acts such as instructions, confirmation requests, etc. In contrast, we present an RL framework to learn to choose appropriate referring expressions.

(Janarthanam & Lemon, 2009a) present a reinforcement learning framework to learn REG policies using a hand-coded user simulation. In contrast, here we present a framework that learns from a statistical user simulation model.

## The Reinforcement Learning Framework

A basic RL setup consists of a learning agent, its environment (in our case, a human user) and a reward function, all modelled as a Markov Decision Process (Sutton & Barto, 1998). The learning agent explores the worth of taking different possible actions in different states and reinforces the state-action pairs for which the environmental rewards are high. In addition, it learns optimal plans (or policies) that are *sequences* of actions leading to the best overall utility for a particular environment. In our model, the learning agent is the NLG module of the dialogue system, whose objective is to learn an adaptive REG policy. The environment consists of a user who interacts with the dialogue system. Since learning occurs over thousands of interaction cycles, real users are replaced by user simulations that simulate real user's dialogue behaviour. The reinforcement learning framework is shown in the figure 1. In the following sections, we discuss the salient features of the important components of the architecture.

### Dialogue Manager

The dialogue manager is the central component of the dialogue system. Given the dialogue state, it identifies the next dialogue act ($A_{s,t}$) to give the user. A dialogue act is a semantic representation of an utterance, and does not specify *how* the utterance is to be generated. The dialogue management policy here is modelled using a simple hand-coded finite state automaton. It issues step-by-step instructions to complete the task and also issues clarifications on referring expressions when requested by the user. We use a hand-coded dialogue manager here so as to localise the learning to the REG module. In future work, dialogue management and NLG will be jointly optimised (Lemon, 2008).

### NLG module

The task of the NLG module is to translate dialogue acts into utterances. It identifies the which REs to use in the utterance to refer to the domain objects. As a learning agent in our model, it has three choices: jargon or descriptive, for every domain object referred to in the instruction. Jargon expressions are technical terms like 'broadband filter', 'ethernet cable', etc. Descriptive expressions contain attributes like size, shape and color. e.g. 'small white box', 'thick cable with red ends', etc. The decision to choose one expression over the other is taken based on the user's domain knowledge which is represented by a `user model` in the dialogue state of the system. It is updated dynamically during the conversation based on the user's dialogue acts. For instance, the user's request for clarification of a referring expression is recorded in the user model as the user's ignorance of the expression. Descriptive expressions for the domain objects are predefined. Therefore, the content (i.e. attributes like size, shape, etc) of the descriptive referring expressions are not chosen during the conversation in the same sense as (Reiter & Dale, 1992), although the model presented here can be extended in this way. In addition to the dialogue act, the dialogue system also issues a set of referring expressions choices ($REC_{s,t}$) chosen by the NLG module.

### User Simulation

Several user simulation models have been used for dialogue management policy learning (Schatzmann, Weilhammer, Stuttle, & Young, 2006; Schatzmann, Thomson, Weilhammer, Ye, & Young, 2007). However, they cannot be used to learn adaptive referring expression generation policies for two reasons. Firstly, they do not simulate different kinds of users (e.g. experts, novices, intermediates, etc). Secondly, they interact with the dialogue system using only the dialogue acts and are not responsive to the referring expression used by the system. In order to enable the NLG module to evaluate its referring expression choices with respect to different users, we propose a two-tiered user simulation model with these two features. The user's environment is also simulated, which the user observes and manipulates based on the system's instruction. For every dialogue session, a new domain knowledge profile is sampled which defines the simulation's dialogue behaviour during the session. Therefore, for instance, a novice profile will produce novice dialogue behaviour with lots of clarification requests. The simulated user could be a novice, an intermediate or an expert based on the domain knowledge profile being used during the dialogue session. In the first step, every referring expression used by the system ($RE_{s,t}$) is examined based on the domain knowledge profile ($DK_{u,RE}$) and the dialogue history ($H$). This step is more likely to produce a clarification request ($CR_{u,t}$) if the REs are unknown to the user and have not be clarified earlier.

The probability that the user issues a clarification request is then given by:

$$P(CR_{u,t}|RE_{s,t}, DK_{u,RE}, H)$$

In the second step, the model issues an appropriate user action ($A_{u,t}$) and an environment action ($EA_{u,t}$)based on the system's instruction ($A_{s,t}$) and the clarification request ($CR_{u,t}$). The user's actions are `provide _info` , `acknowledge` , `request _clarification` , `request _repeat` , etc.

$$P(A_{u,t}|A_{s,t}, CR_{u,t})$$
$$P(EA_{u,t}|A_{s,t}, CR_{u,t})$$

In our experiments, all the parameters were set empirically using data containing interactions between real users and spoken dialogue systems which were collected using Wizard-of-Oz experiments (Janarthanam & Lemon, 2009b). Our model is built using a dialogue corpus collected from 17 users with different levels of domain knowledge. Using a `dialogue similarity` measure (Cuayahuitl, Renals, Lemon, & Shimodaira, 2005; Cuayahuitl, 2009), based on Kullback-Leibler divergence, we show that this two-tier model simulates real users more closely than more standard n-gram models. Ideally, for two similar models the dialogue similarity measure must be close to zero.

Table 1 shows that the two-tier model is much closer to the real user data than the other models. This is because the two-tier model takes into account all the context variables, such as

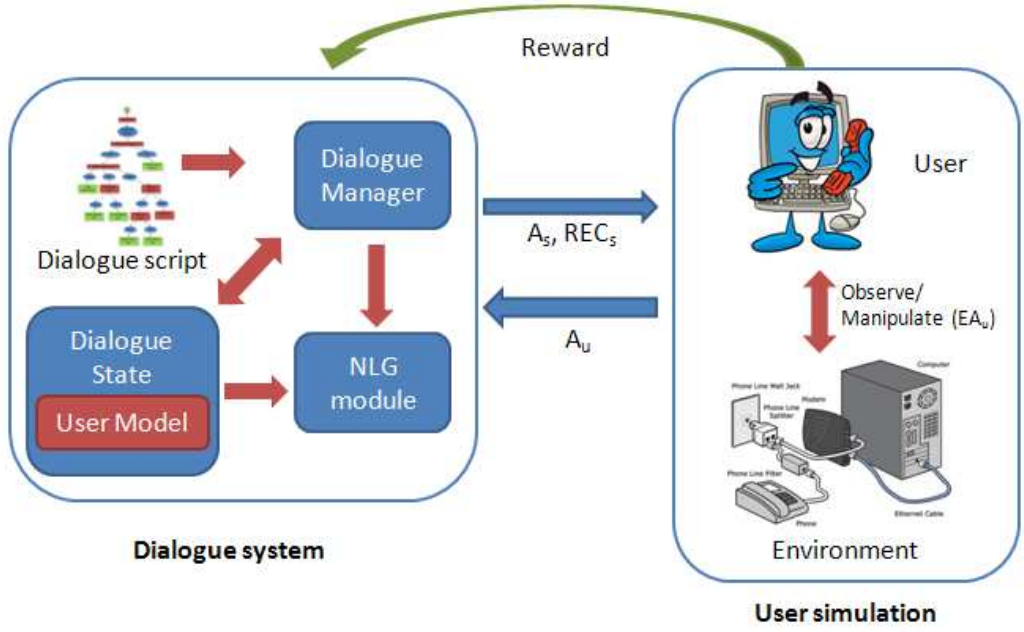


Figure 1: The Reinforcement Learning Framework

Table 1: Dialogue Similarity

| **Model** | $A_{u,t}$ | $EA_{u,t}$ |
|---|---|---|
| Two-tier | 0.078 | 0.018 |
| Bigram | 0.150 | 0.139 |
| Trigram | 0.145 | 0.158 |
| Equal Probability | 0.445 | 0.047 |

the user's domain knowledge and the history of the session, while the other models do not.

### Learning REG policies

REG policies are learned by the NLG module by interacting with the user simulation in learning mode. The module explores different possible state-action combinations by choosing different REs in different states. The user simulation simulates different kinds of users by choosing different knowledge profiles during the dialogue sessions. At the end of each dialogue session, the learning agent is rewarded based on parameters like dialogue length, the number of clarification requests, etc. The magnitude of the reward allows the agent to reinforce its choice of moves in different states. Ideally, the agent gets less reward if it chooses the REs inappropriate to the users, because this results in more clarification requests from the user. Ideally, the reward model parameters are also set empirically using Wizard-of-Oz data (Janarthanam & Lemon, 2009b). The learned policies predict good REs based on the discovered patterns in linguistic knowledge. For instance, a user who knows 'broadband cable' will most likely know 'ethernet cable' and therefore the system should use jargon (for `ethernet cable` ) – and for a user who does not recognise 'broadband cable', the descriptive expression ('the thick cable with red ends') should be used. At the end of the learning cycle, the system learns a policy that adapts to different kinds of users based on their knowledge profiles.

### Evaluation

Learned policies can be evaluated using both user simulations and real users (Rieser & Lemon, 2008). Policies are tested to see if they produce good moves for different lexical knowledge profiles. Baseline policies are hand-coded for comparison, and the learned and baseline policies are tested under the same conditions. In this paper the policies are tested using user simulations with different knowledge profiles and their average rewards are compared.

## Preliminary experiments

Using the setup described in the previous sections, REG policies were learned with a user simulation. Figure 2 shows a training run over 350 dialogues. One should note that these are preliminary experiments – future work remains in exploring better reward functions and learning/training parameters.

The user simulation was calibrated to produce two kinds of users - novices and experts. Expert users knew all the technical terms, whereas the novices knew only a few terms like 'livebox', 'power adaptor' and 'phone socket'. Either these terms were ubiquitous or clearly labelled on top of the domain object. The other terms that only the experts know are 'broadband filter', 'broadband cable' and 'ethernet cable'. The task

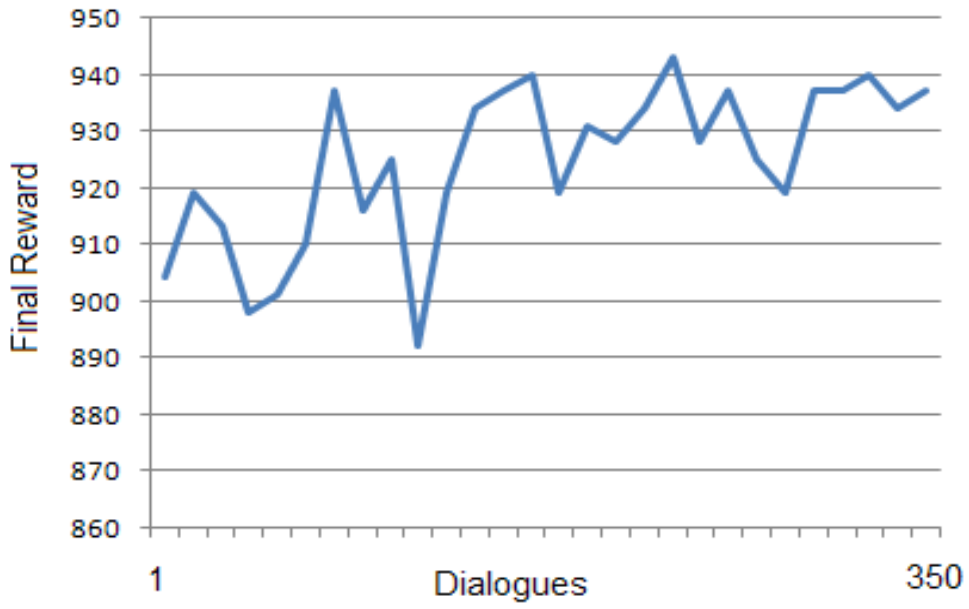


Figure 2: Training a REG policy

of the NLG module is then to learn which expression (descriptive or technical) to use to refer to the domain objects in the instructions during the course of the dialogue, so as to avoid unnecessary clarifications. The user model, which is a part of the dialogue state, is made up of variables representing the user's knowledge of the technical terms. Initially, they are set to the unknown state and then are updated to the `knows` or `doesnt-know` state based on the user's responses.

The initial training sessions were run for 30,000 cycles producing around 350 dialogues using the SARSA reinforcement algorithm. A simple reward function based on task completion and number of clarification requests was used. Clarification requests by the user were punished severely to make the learning agent avoid them. It can do this by using appropriate referring expressions for particular users. The final reward is calculated as follows:

$$TaskCompletionReward(TCR) = 1000$$
$$CostofCR(CCR) = -n(CR) * 30$$
$$FinalReward = TCR + CCR$$

The initial learned policy (LP) has learned to use descriptive expressions most of the time, for both kinds of users. This was possible because both kinds of users understood the descriptive expressions and the clarification requests were very infrequent. We compared the learned policy to baseline handcoded policies and a random baseline (see table 2). We found that the learned policy (LP) is significantly better than the `Jargon-only` policy and the `Random` policy. It was as good as the `Descriptive-only` policy (not statistically significant different) with only 1.8 clarification requests per dialogue.

Table 2: Rewards and number of CRs ($* = p < 0.05$).

| Policy | Avg. Reward | Avg. CRs |
|---|---|---|
| LP | 944.09* | 1.8 |
| Random | 919.31* | 2.69 |
| Desc only | 948.19 | 1.75 |
| Jargon only | 885.35* | 3.82 |

This initial result shows that our framework is functioning as planned: REG policies can be learned based on statistical user simulations, and they perform at least as well as handcoded baselines. Future work will explore much longer training runs and different learning parameters, to produce better REG policies.

## Summary

We argue that the data-driven optimisation framework offered by Reinforcement Learning allows a powerful combination of empirical and computational approaches to the REG problem.

We presented a data-driven approach to learning to produce adaptive referring expressions in spoken dialogue systems using reinforcement learning (a type of statistical planning). We highlighted essential features of the model, which will allow us to learn adaptive REG policies that maximise the expected utility of referring expression choices with respect to the users' domain and linguistic knowledge (Janarthanam & Lemon, 2009a).

We also presented results from preliminary experiments,

which show that the framework allows us to learn good REG policies. In our future work we will further explore reward functions and learning/training parameters, as well as improvements to our user simulation.

## Acknowledgments

The research leading to these results has received funding from the European Community's Seventh Framework (FP7) under grant agreement no. 216594 (CLASSiC Project www.classic-project.org), EPSRC project no. EP/E019501/1, and the British Council (UKIERI PhD Scholarships 2007-08).

## References

Clark, H. H., & Murphy, G. L. (1982). Audience design in meaning and reference. In J. F. LeNy & W. Kintsch (Eds.), *Language and comprehension.* Amsterdam: North-Holland.

Cuayahuitl, H. (2009). *Hierarchical reinforcement learning for spoken dialogue systems*. Ph.D. thesis, University of Edinburgh, UK.

Cuayahuitl, H., Renals, S., Lemon, O., & Shimodaira, H. (2005). Human-Computer Dialogue Simulation Using Hidden Markov Models. In *Proc. of ASRU 2005.*

Dale, R. (1989). Cooking up referring expressions. In *Proc. ACL-1989.*

Deemter, K. van. (2009). What Game Theory can do for NLG: the case of vague language. In *Proc. ENLG'09.*

Gatt, A., & Belz, A. (2008). Attribute Selection for Referring Expression Generation: New Algorithms and Evaluation Methods. In *Proc. INLG-2008.*

Grice, H. P. (1975). Logic and Conversation. *Syntax and Semantics: Vol 3, Speech Acts*, 43-58.

Isaacs, E. A., & Clark, H. H. (1987). References in conversations between experts and novices. *Journal of Experimental Psychology: General*, *116*, 26-37.

Janarthanam, S., & Lemon, O. (2009a). Learning Lexical Alignment Policies for Generating Referring Expressions for Spoken Dialogue Systems. In *Proc. ENLG'09.*

Janarthanam, S., & Lemon, O. (2009b). A Wizard-of-Oz environment to study Referring Expression Generation in a Situated Spoken Dialogue Task. In *Proc. ENLG'09.*

Lemon, O. (2008). Adaptive Natural Language Generation in Dialogue using Reinforcement Learning. In *Proc. SEMdial'08.*

Levin, E., Pieraccini, R., & Eckert, W. (1997). Learning Dialogue Strategies within the Markov Decision Process Framework. In *Proc. of ASRU97.*

Reiter, E., & Dale, R. (1992). A Fast Algorithm for the Generation of Referring Expressions. In *Proc. COLING-1992.*

Reiter, E., & Dale, R. (1995). Computational Interpretations of the Gricean Maxims in the Generation of Referring Expressions. *Cognitive Science*, *18*, 233-263.

Rieser, V., & Lemon, O. (2008). Learning Effective Multimodal Dialogue Strategies from Wizard-of-Oz data: Bootstrapping and Evaluation. In *Proceedings of ACL.*

Schatzmann, J., Thomson, B., Weilhammer, K., Ye, H., & Young, S. J. (2007). Agenda-based User Simulation for Bootstrapping a POMDP Dialogue System. In *Proc of HLT/NAACL 2007.*

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

Sutton, R., & Barto, A. (1998). *Reinforcement learning*. MIT Press.