Learning Adaptive Referring Expression Generation Policies for Spoken Dialogue Systems using Reinforcement Learning

Srinivasan Janarthanam
School of Informatics
University of Edinburgh
s.janarthanam@ed.ac.uk

Oliver Lemon
School of Informatics
University of Edinburgh
olemon@inf.ed.ac.uk

Abstract

Adaptive generation of referring expressions in dialogues is beneficial in terms of grounding between the dialogue partners. However, handcoding adaptive REG policies is hard. We present a reinforcement learning framework to automatically learn an adaptive referring expression generation policy for spoken dialogue systems.

1 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 could help in efficient grounding between dialogue partners (Issacs and Clark, 1987), improve task success rates or even increase 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. Although adapting to users seems beneficial, adapting to an unknown user is tricky and hand coding such adaptive REG policies is a cumbersome work. (Lemon, 2008) first presented the case for treating NLG as a reinforcement learning problem. In this paper, we extend the framework to automatically learn an adaptive REG policy for spoken dialogue systems.

2 Related work

Reinforcement Learning (RL) has been successfully used for learning dialogue management policies (Levin et al., 1997). The learned policies allow the dialogue manager to optimally choose appropriate instructions, confirmation requests, etc. In contrast, we present an RL framework to learn REG policies.

3 Reinforcement Learning Framework

A basic RL setup consists of a learning agent, its environment and a reward model (Sutton and Barto, 1998). The learning agent explores by taking different possible actions in different states and exploits the actions for which the environmental rewards are high. In our model, the learning agent is the NLG module of the dialogue system, whose objective is to learn an 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. In the following sections, we discuss the salient features of the important components of the architecture in the context of a technical support task (Janarthanam and Lemon, 2009a).

3.1 Dialogue Manager

The dialogue manager is the central component of the dialogue system. Given the dialogue state, it identifies the next dialogue act to give to the user. The dialogue management policy is modelled on a simple handcoded finite state automaton. It issues step by step instructions to complete the task and also issues clarifications on REs used when requested by the user.

3.2 NLG module

The task of the NLG module is to translate the dialogue act into a system utterance. It identifies the REs to use in the utterance to refer to the domain objects. As a learning agent in our model, it has three choices - jargon, descriptive and tutorial. 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. Tutorial expressions are a combination of the two. The decision to choose one expression over the other is taken based on the user's domain knowledge, which is updated progressively in a user model (state) during the conversation.

3.3 User Simulation

In order to enable the NLG module to evaluate the REG choices, our user simulation model is responsive to the system's choice of REs. For every dialogue session, a new domain knowledge profile is sampled. Therefore, for instance, a novice profile will produce novice dialogue behaviour with lots of clarification requests. For user action selection, we propose a two-tiered model. First, the system's choice of referring expressions ($REC_{s,t}$) is examined based on the domain knowledge profile (DK_u) 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.

$$P(CR_{u,t}|REC_{s,t}, DK_u, H)$$

If there are no clarification requests, then issue an appropriate user action ($A_{u,t}$) based on the system's instruction ($A_{s,t}$) and if there is one, the user action will be the clarification request itself.

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

These parameters are set empirically by collecting real user dialogue data using wizard-of-Oz experiments (Janarthanam and Lemon, 2009b).

4 Learning REG policies

REG policies are learned by the NLG module by interacting with the user simulation in the learning mode. The module explores different possible state-action combinations by choosing different REs in different states. At the end of each dialogue session, the learning agent is rewarded based on parameters like dialogue length, number of clarification requests, etc. The magnitude of the reward allows the agent to reinforce the optimal moves in different states. Ideally, the agent gets less reward if it chooses the inappropriate REs, which in turn results in clarfication requests from the user. The reward model parameters can be set empirically using wizard-of-Oz data (Janarthanam and Lemon, 2009b). The learned policies predict optimal REs based on the patterns in knowledge. For instance, a user who knows 'broadband cable' will most likely know 'ethernet cable'.

5 Evaluation

Learned policies can be evaluated using the user simulation and real users. Policies are tested to see if they produce optimal moves for the given knowledge profiles. Learned policies can be compared to hand-coded baseline policies based on parameters like dialogue length, learning gain, etc. Real users are asked to rate the system based in its adaptive features after their interaction with the dialogue system.

6 Conclusion

A framework to automatically learn adaptive REG policies in spoken dialogue systems using reinforcement learning has been presented. Essential features to learn an adaptive REG policy have been highlighted. Although the framework is presented in the context of a technical support task, the same is suitable for many other domains.

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

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

S. Janarthanam and O. Lemon. 2009a. *Learning Lexical Alignment Policies for Generating Referring Expressions for Spoken Dialogue Systems*. Proc. ENLG'09.

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

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

E. Levin, R. Pieraccini and W. Eckert. 1997. *Learning Dialogue Strategies within the Markov Decision Process Framework*. Proc. ASRU97.

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