Learning Interaction Patterns for Adaptive User Interfaces

Kristiina Jokinen, Jyrki Rissanen, Heikki Keränen, and Kari Kanto

Media Lab
University of Art and Design Helsinki,
Hämeentie 135 C
FIN-00560 Helsinki Finland
{kjokinen|jyrki|hkeranen|kanto}@uiah.fi

In this paper we discuss possibilities for building adaptive spoken dialogue systems and how learning in such systems could be modelled. Especially, we are interested in the dialogue strategies that different users exploit in their interactions with speech applications, and how the systems could learn differences in the strategies so as to allow personalisation of the system's interaction capabilities according to user characteristics. The paper reports our experiments on a small corpus of email-reading dialogues and presents preliminary results on applying reinforcement learning to explore strategies and interaction patterns for user models that would realize the *design for all* principles, and advance technology towards accessibility and acceptability for the widest possible end-user population.

Introduction

In recent years, the notion of adaptivity has become more important when building speech interfaces that take various users into account. Adaptivity is often realised as personalised user interfaces where user preferences take the form of colour or sound choices, and characteristics are listed in personal profiles. On-line adaptation may be realised in the system's ability to classify users into predefined categories, e.g. on the basis of their navigation choices so as to provide better answers to user queries. However, this kind of adaptation is often based on a static and mechanical view of the users and their preferences, and it does not extend itself to user actions in communicative dialogue situations. The problem in adaptive interfaces seems to be the notion of adaptivity itself: adaptation involves learning, learning involves interaction and interaction is the means through which adaptation takes place.

In this paper we explore possibilities for building adaptive speech interfaces using reinforcement learning inspired methods. The point of departure is to learn the user's interaction patterns by experience: the model is gradually built and modified on the basis of the observed user behaviour. By comparing this model with a general model, the user's individual strategies can be distinguished once the differences between the two models become statistically significant. It is also possible to form group user models by grouping together those users who have similar interaction patterns. It is important to notice that the whole approach is *dynamic*: both the general and the individual user models change through the time and via different interaction patterns.

The paper is structured as follows. We first discuss the data collection task and the starting point for our research. We continue by describing the learning method and the dialogue model used in our experiments, and then report on the results of the experiments. We conclude with future research.

Data collection

The data collection was designed to gather information about the basic interaction between the user and the speech-based email system in a Wizard of Oz setup. Due to privacy restrictions, we could not use the users' real mailboxes in the corpus collection task, and thus a novel multi-user scenario-based approach was developed. The subjects were given a role in a scenario which dealt with problems at work, and they were supposed to read and send messages to each other according to their respective roles. The users were asked to check their mailboxes with the experimental system that incorporates "normal email client functionality". They were requested to phone to the system at least twice a day for a week so as to get enough email activity and to allow development of the topic. The scenario used roles for six people.

We collected 65 dialogues from 6 people who were students and staff. One of our testees was visually impaired. The corpus amounts to 550 different utterances, which were hand-tagged with action labels representing the action states of the future system. The 15 action tags were determined by a group of specialists, and they are as follows: `greet farewell, prompt_for_action, list_msgs, listen_msg, delete_msg, move_msg, dictate_msg, listen_own_msg, send_msg, create_folder, save_to_folder, open_folder, search` and `cancel`. The set is augmented with the tag `end`, representing the system's final state, and together these states were then used as the training material for our RL system.

User modeling

Our approach is to design a general model of the user's interaction strategies on the basis of the corpus and then exploit and update this model online according to the user's characteristic interaction patterns. The learnt patterns can be used in a full dialogue system to predict the system's next actions.

We define the *individual user model (IUM)* as the model constructed on the basis of the interaction patterns of each individual user. We also define the *general group model (GGUM)* as the model consisting of the states and transitions calculated on the basis of the data from all the users so far. Finally, we assume that the user may be a member in a group which has its own *group user model (GUM)*, defined as the average of some IUMs grouped together because of the users have some meaningful common characteristics.

When a new user is encountered, the default user model is GGUM, unless the user has given some preferences on the basis of which she can be classified into a user group with a user model GUM. The user's own IUM will be constructed simultaneously. Once the user has used a particular action state transition for a statistically significant number of times, this transition can be considered characteristic of the user and will be used instead of the default transitions from the (G)GUM. It must be noted that the strategies are learnt online, and also the GGUM transitions are updated by the user's actions. The transition possibilities in the action state space can be represented as a tree with particular Q-values associated with each transition (see below), and the optimal path through the state space marked by transitions with the highest Q-values. The IUM can be considered as a sub-tree of the GGUM in the same state space, with different Q-values associated with the transitions. The point when the GGUM will be biased towards the IUM marks the transition from GGUM to IUM, i.e. the user's characteristic interaction patterns have occurred often enough so that their statistical probability exceeds that of the general model.

The algorithm

Our goal is to examine the usefulness of reinforcement learning (RL) [3] in adaptive interactive systems. Recently the Markov Decision Process (MDP) combined with the reinforcement learning has been successfully applied to learning dialogue strategies [6], and we have also adopted this technique. However, the task for learning dialogue strategies in realistic interactive situations requires that the system also models uncertainty concerning the system's own internal state and the observed user actions. The partially observable MDP (POMDP) would thus be a more appropriate decision algorithm, but it is impractical due to its computational complexity. One possibility is to augment the traditional MDP model with the system's beliefs about the state it is in, as is done in [7].

Figure 1 : Reinforcement learning framework

In the RL framework as depicted in Figure 1, the agent takes an action *a*, finds itself in a state *s*, and receives a reward *r*. The task in reinforcement learning is to find a policy that maximizes the agent's reward in an environment. The agent's environment is represented by

- a discrete and finite set *S* of states
- a set *A* of actions that take the agent from one state to another, described by a transition function $d : S \times A \rightarrow S$
- a reward function $r : S \times A \rightarrow R$, which describes the reward an agent gets when it performs an action in a certain state.

The policy $P : S \rightarrow A$ describes the agent's behaviour in a given state.

Q-learning [9] does not need an explicit model of the agent's environment and can be used on-line. The algorithm estimates the values of state-action pairs (Q-values), *Q(s,a)*, which are calculated as the expected discounted sum of future payoffs when an action *a* is taken from state *s* and an optimal policy is followed thereafter. Once the Q-values have been learned, the optimal action from any state is the one with the highest Q-value.

The Q-values are estimated on the basis of experience as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r + \gamma \max_{a'} Q(s',a') - Q(s,a)\right]$$

where α is the learning rate, and $0 < \gamma < 1$ is the discount factor.

Since we want to experiment with the learning of frequent dialogue strategies, the reward function *r()* is simply the transition probability moving from a particular state *s* to state *s'* in the corpus.

Analyses and Experiments

The data analysis was done by identifying action state chains which occurred most often in the corpus. The corpus is not large enough for extensive testing and comparison of each individual's interaction patterns (an average ten dialogues per person). However, there was one subject who produced one third of the dialogues, and thus enough data to justify building of her own model. Therefore a separate IUM for that particular subject was built (we call her Auli, according to her role name in the scenario), and a general user model (GGUM) was build from the data from all other users together. Interestingly enough, Auli was the visually impaired user of the system.

There are a few examples of interaction patterns which can be considered as indications of such patterns that can vary across individual users. Because of the small number of dialogues it is not possible to draw conclusions of any special patterns that vary across the users, but it is encouraging that despite a small corpus, our method is capable of visualizing these distinctions, i.e. they are supported by our experiments to learn interaction strategies from experience.

User groups and Probable Action Paths

Different action paths were learnt for Auli and for all other users. The beginning of the most typical action paths is visualised in Figure. 2.

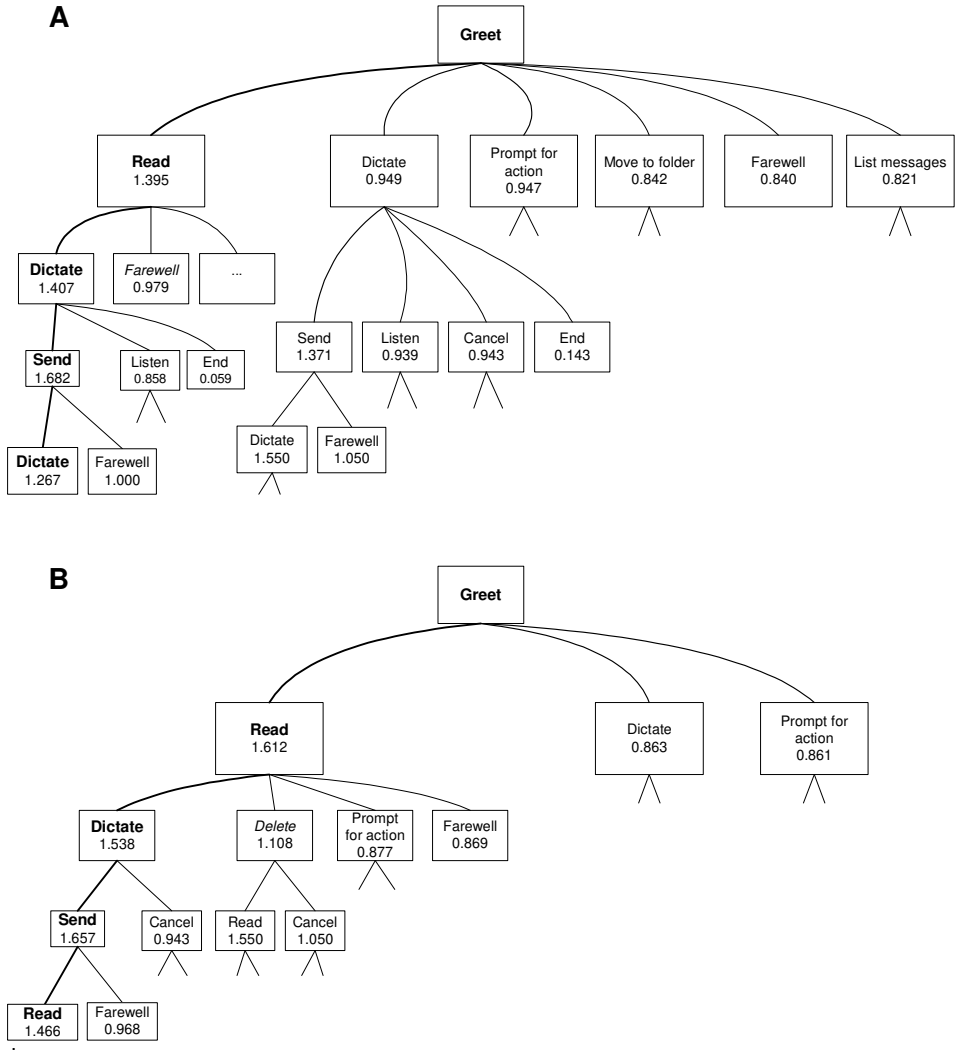

Figure 2: Most probable interaction paths for all users (A) and for Auli (B).

Notice that compared to the average users, Auli has never taken some actions in a particular state. For instance, after the initial *greet*, Auli preferred to read or dictate a message but did not venture on folder commands (list messages, move a message to a folder).

Users with Different Interaction Patterns

Figure 2 also shows differences in the possible state transitions from the state *read* occurring after the initial *greet*. Although the small size of the corpus and the small number of users prevent us from making any definite conclusions, the results suggest that there is a clear difference between Auli and the other subjects: whereas the general pattern seems to be to repeat the *dictate-send* cycle (or finish the cycle with *farewell*), Auli prefers a longer chain in which *dictate* and *send* are followed by *read* (or finished by *farewell*). In other words, after having dictated and sent a message, Auli tends to read a new message, while the others tend to dictate a new message in the same situation. Auli was also more clear-cut after dictating a message: she either sent it or cancelled the action, whereas the others used the *listen*-option to confirm that the dictated message was as intended.

Users with Fixed Habits

Some users exploit a very limited part of the functionality available, although they use the system quite a lot. For instance, the users Kata and Risto called to the system almost the same number of times during experiment, but in his dialogues Risto used only 13 different three-state interaction chains, whereas Kata used 50 different chains. It is exactly this kind of variation in the individual users' interaction habits that the system is to learn and store in the user's IUM, and which can then be exploited by the system in determining the user's individual habits and preferences on-line.

Discussion and related work

A common problem in interactive systems is that they are static and usually provide only one type of dialogue strategy for the users to follow. Also the system's knowledge of the user is often restricted to a few levels of user expertise: this undervalues the user's versatile competence which varies depending on the task at hand. In practical systems the user models are often encoded into a number of possible certainty levels dealing with the system's beliefs of whether certain task concepts are clear and mutually known or whether they must be confirmed [1,2,5]. Some systems can switch between system-initiative and mixed-initiative strategies and learn the differences automatically (Elvis, TOOT [6,8]), but more subtle changes concerning on-line learning of the user's *interaction patterns* require further investigations.

The static nature of the systems is a problem for applications that are intended to be used in mobile and versatile environments, by various users with different abilities and requirements. There is thus a huge need for adaptive systems that can adapt to various users automatically. The ideal system would learn user characteristics and usability patterns from the interaction that the user has with the system, cf. [4].

Future research

The goal of our research is to study fast and efficient methods for learning appropriate interaction strategies for various users, and to provide a practical system with a dynamic user model component. This paper reports on-going research which centers on the learning method and its applicability in interactions. The next step is to collect more data for testing and developing the method further. Besides user simulations, the prototype will be integrated into a working system, and user studies will be conducted on evaluating such a system's usability in real situations.

We will also focus on resolving some problems concerning adaptation studies in general. The assumption has been that the user adapts her actions when using an intelligent computer application as a tool. However, our studies show that the users have clear preconceptions of the tool and how to use it, and they do not venture exploring the limits of the system, even though they are encouraged to do so by being given minimal instructions for the use of the system and what to expect from it. In our experiments, the particular tool was a speech-controlled mobile email reading application which the users could use to read and send emails, and to organise their mailbox. The users tacitly assumed certain system limits, and their exploration of, and adaptation to wider system capabilities seemed to be prevented by their knowledge of the computers and by their familiarity with IT technology in general. It is interesting to notice that one subject requested help-facility but he never actually tested whether the system had such a facility or not (the setup indeed contained a help). Furthermore, some common facilities, like

the use of folders to organise the mailbox, were not commonly used although this was explicitly mentioned in the instructions and folders are available also in conventional mail systems.

Finally, the initial transition probabilities P are calculated from the dialogues collected and tagged in the pilot study. Although the data is sparse, it is assumed that enough useful strategies can be found to bootstrap the basic system. The hypothesis is that it is possible to initialise a generic system by bootstrapping it with only a few strategies, possibly learnt from a small and noisy dataset, and then letting the system to find optimal policy by dynamically modifying the transition probabilities according to user actions. Future research will investigate this hypothesis further.

Acknowledgements

The research is carried out in EU 5th Framework research project DUMAS (IST-2000-29452). We gratefully acknowledge the financial support by the European Union.

References

1. Alexandersson, J., Engel, R., Kipp, M., Koch, S., Kussner, U., Reithinger, N., and Stede, M. (2000). Modeling Negotiation Dialogs. In W. Wahlster (Ed.), *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer Verlag, pp. 441-451.

2. Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A. (2000). An architecture for a generic dialog shell. *Natural Language Engineering* 6(3), 1-16.

3. Barto, A. and Sutton, R. (1993). *Reinforcement* Learning: An Introduction. MIT Press, Cambridge, Mass.

4. Jokinen, K. (2000). Learning Dialogue Systems. In *Proceedings of the LREC workshop From Spoken Dialogue to Full Natural Interactive Dialogue*, pp. 13-17.

5. Jönsson, A. (1997). A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering* 3(2/3), 103-122.

6. Litman, D., Kearns, M., Singh, S., and Walker, M. (2000). Automatic Optimization of Dialogue Management. *Proceedings of the 18th COLING*, pp. 502-508.

7. Roy, N., Pineau, J., and Thrun, S. (2000). Spoken Dialog Management for Robots. *Proceedings of the ACL 2000*.

8. Singh, S., Kearns, M., Litman, D.J. and Walker, M. (1999). Reinforcement Learning for Spoken Dialogue Systems. *Advances in Neural Information Processing Systems* 12*, MIT Press*.

9. Watkins, C. J. C. H and Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3), 279-292.