ADAPTIVE MULTIMODAL DIALOGUE MANAGEMENT BASED ON THE INFORMATION STATE UPDATE APPROACH

Kallirroi Georgila and Oliver Lemon

University of Edinburgh
School of Informatics
{kgeorgil, olemon}@inf.ed.ac.uk

ABSTRACT

This paper provides an overview of the aims of the TALK project[1] focusing on the issue of integrating Reinforcement Learning (RL) with the Information State Update (ISU) approach to dialogue management, in order to develop adaptive multimodal dialogue systems. The project will build showcases for in-car and in-home information and control, but its main aim is to advance our understanding of generic technologies that will extend the ISU approach to adaptive multimodal and multilingual dialogue.

1. INTRODUCTION

The Information State Update (ISU) approach, as developed in the TRINDI and SIRIDUS projects [1, 2], successfully provided a backbone for naturally interactive, yet practical, spoken dialogue systems. However, within this approach important requirements for multimodality, reconfigurability, learning, and adaptivity have not yet been addressed in an integrated and fundamental way. Thus new technical approaches are required in the following areas:

1. Unifying multimodality and multilinguality,
2. Automatic generation and reconfiguration of multimodal interfaces,
3. Multimodal presentation in the ISU approach,
4. Learning and adaptivity.

Though the field of spoken dialogue systems has developed very quickly in the last decade, rapid design of dialogue strategies remains problematic. Several approaches to the problem of automatic strategy learning have been proposed and the use of the formalism of Markov Decision Processes (MDPs) and Reinforcement Learning (RL) introduced by Levin and Pieraccini [3, 4] is becoming part of the state of the art in this area. Similar approaches have been proposed in [5, 6, 7, 8, 9]. A dialogue strategy would be for example for the system to decide on the type of confirmation (explicit, implicit, none) or on the modality it would use to present the requested information (speech, text, icons). However, to obtain a fully automatic procedure, the learning agent needs either real interactions with a user through an automated speech recognition (ASR) system, a large amount of corpus data, or a sequence of simulated interactions with a virtual user. In TALK we will use a baseline system for collecting data and performing experiments with RL. This baseline system will exploit different strategies so that data suitable for training the dialogue system to handle various dialogue phenomena can be acquired.

In the remainder of the paper the ISU approach, the baseline system, and the main research issues on learning and adaptivity are explained in more detail. Finally, the potential impact of TALK as well as its expected contributions to standards are presented.

2. THE INFORMATION STATE UPDATE APPROACH

The Information State Update (ISU) approach allows a declarative representation of dialogue modelling. “The term *Information State* of a dialogue represents the information necessary to distinguish it from other dialogues, representing the cumulative additions from previous actions in the dialogue, and motivating future action” [1].

The ISU dialogue management we currently use in TALK research concerning learning and adaptivity and for implementing the baseline system is called DIPPER [10] available at http://www.ltg.ed.ac.uk /dipper. The DIPPER architecture is a collection of software agents for prototyping (spoken) dialogue systems implemented on top of the Open Agent Architecture (OAA) [11]. DIPPER is not a dialogue system itself, but it supports building (spoken) dialogue systems, by offering interfaces to speech recognisers, speech synthesisers, parsers and other kinds of agents.

[1] The TALK consortium consists of the Universities of the Saarland, Edinburgh, Gothenburg, Cambridge, and Seville, and the non-academic partners DFKI, BMW F+T, Bosch, and Linguamatics. See http://www.talk-project.org.

Although DIPPER supports many off-the-shelf components useful for spoken dialogue systems, it comes with its own dialogue management component (the DIPPER DME), based on the information state approach to dialogue modelling. The DIPPER DME component borrows many of the core ideas of the TrindiKit, but is stripped down to the essentials, uses a revised update language (independent of Prolog), and is more tightly integrated with OAA. The DIPPER DME is written in Sicstus Prolog. Moreover, a new version has been implemented in Java and is currently being tested.

A complete dialogue system can be implemented using DIPPER DME, OAA and a collection of agents, which includes: (1) agents for input/output modalities, (2) agents for the dialogue move engine, and (3) supporting agents. The DIPPER DME is the core of the system controlling the flow of information among the agents.

DIPPER follows TrindiKit closely, taking its record structure and datatypes to define information states (see an example information state in Figure 1). Update rules (see examples in Figure 2) specify the information state change potential in a declarative way: applying an update rule to an information state results in a new state. An update rule is a triple `<name, conditions, effects>`. The conditions and effects are defined by an update language, and both are recursively defined over terms. The terms of the update language allow developers to refer to specific values within information states, either for testing a condition or applying an effect. For example, the term `is^lastmoves` refers to the field `lastmoves` in the record `is`.

```
infostate(record([is:record([
            lastspeaker:atomic,
            turn:atomic,
                 gnd:record([dh:stack(atomic),
                        obl:stack(atomic)]),
            input:stack(atomic),
            lastinput:stack(atomic),
            output:stack(atomic),
            nextmoves:stack(Acts),
            lastmoves:stack(Acts),
            filledslotsvalues:stack(atomic),
            filledslots:stack(atomic),
            task:stack(atomic),
            taskstep:atomic,
            deliberation:atomic,
            int:stack(Acts)])])) :-
            Acts = record([pred:atomic,
                       dp:atomic,
                  prop:record([pred:atomic,
                  args:stack(atomic)])]).
```

Figure 1. An example DIPPER information state definition.

Figure 3 depicts the Graphical User Interface of the DIPPER DME, showing the current information state, the last applied update rule, and system messages.

```
urule(generation,
  [
   top(is^int)=[release_turn],
   is^lastspeaker=user,
   prolog(checkfilledslots(top(is^nextmoves),
       is^filledslots,Z)),
   Z=0
  ],
  [
   prolog(reverse_and_utter(is^nextmoves,
        X,Y)),
   push(is^lastmoves,X),
   clear(is^nextmoves),
   clear(is^output),
   push(is^output,Y),
   solve2(callfestival(Y,_X)),
   assign(is^lastspeaker,system),
   clear(is^input),
   assign(is^turn,user),
   assign(is^deliberation,1)
  ]
 ).

urule(queueplan,
  [
   is^turn=system
  ],
  [
   push(is^nextmoves,top(is^int)),
   clear(is^int)
  ]
 ).
```

Figure 2. Example DIPPER update rules.

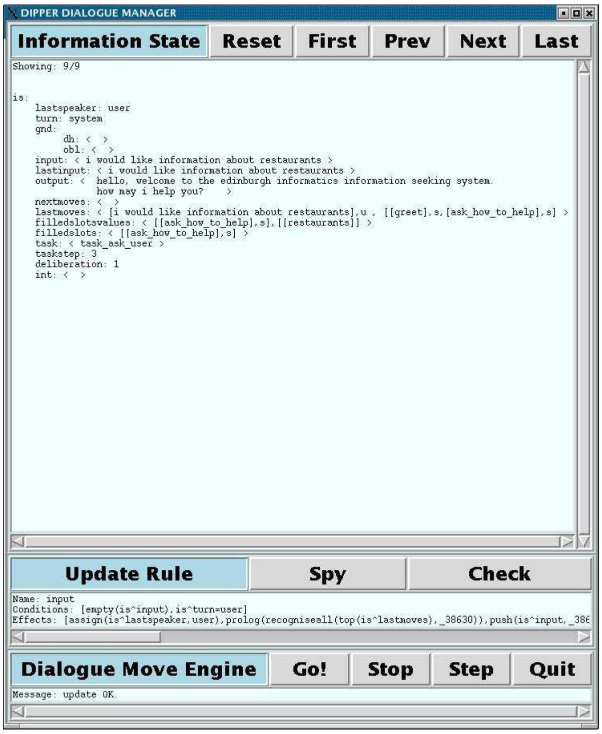

Figure 3. The Graphical User Interface of DIPPER.

3. THE TALK BASELINE SYSTEM

We have made some progress in constructing a baseline system that will be used to collect data and perform experiments with RL. The baseline system is implemented on the DIPPER architecture. The major components are:

- ATK for speech recognition [12],
- Festival for speech synthesis [13],
- O-Plan for dialogue planning and content planning and structuring [14],
- DIPPER DME for dialogue management [10].

The semantic content of the user's input is incorporated into the recognition network used by ATK and in some cases additional semantic interpretation may be performed by a pattern matching based parser. The idea is to match streams of user's input with semantic patterns and thus extract the semantic content of the user's utterance. The parts of the user's spoken utterance that do not match with any of the existing patterns are ignored. A number of other supporting agents are available (e.g. various parsers and theorem provers [10], the Open CCG realizer [15]), and may be incorporated at a later point.

The baseline system will support information-seeking and map tasks to match the project's requirements for in-car and in-home information and control. A number of different strategies are being investigated based on studying the SACTI-1 corpus collected by Cambridge University. SACTI stands for Simulated ASR-Channel: Tourist Information [16, 17]. The corpus consists of human-human dialogues. However, simulated speech recognition errors have been included to reflect the ASR channel.

The baseline system will be used for collecting data for learning via RL algorithms. The drawback of using human-computer dialogues as a source of dialogue data is that existing spoken dialogue systems typically use a fixed policy, thus making the data unsuitable for training a machine learning approach. To deal with this problem and thus deviate from fixed policies we plan to create a mapping of "reasonable" actions for each state [9].

The sequence of information states derived from the system after the human-computer dialogues have taken place will feed the RL algorithms. Moreover, in order to take advantage of already existing corpora such as SACTI-1 [16, 17], Darpa Communicator [18], etc., we aim to explore semi-automatic annotation of dialogues.

By studying human-human and human-computer data, and taking into account various degrees of recognition error rates, we anticipate being able to learn dialogue policies that will cover a variety of dialogue phenomena and therefore lead to more robust spoken dialogue systems.

4. INTEGRATING RL AND ISU

As has been explained in the previous section, a sequence of information states will form the vectors to feed RL algorithms. System actions will be determined by various update rules. Following this idea, in combining the ISU approach and RL we are presented with several questions:

- How can we reduce the number of states in the ISU approach? Here we are considering using equivalence relations or "state-tying". Can we use hierarchical RL (e.g. the SHARSHA algorithm [19]) for this purpose?
- Considering the task complexity and the huge number of possible states, should we try to learn strategies for the whole dialogue or just focus on specific dialogue phenomena e.g. confirmation?
- Can we discover features of the state history which are useful for dialogue management? This could reduce the burden of the system designer to think of everything relevant to include in a state.
- Most attempts to use RL in dialogue are based on simulating the user. Can models of human wizards' behaviour be used effectively in combination with RL?
- Can we exploit the structured nature of some ISU state features in addressing the above questions?

In future work we also plan to investigate the use of learning techniques to perform speech recognition which is sensitive to detailed dialogue context information [20, 21].

5. POTENTIAL IMPACT AND CONTRIBUTIONS TO STANDARDS

Over the next ten years, people will need to be able to control an increasing number of intelligent devices, and interact with a wide variety of services in their homes and cars, and many other contexts. This will require a new generation of multimodal interfaces which adapt to the user and environment, provide information at the appropriate level of detail for a variety of modalities (including different languages), and are dynamically reconfigurable to new tasks. TALK attempts to deal with all these goals.

To achieve the full aims of TALK will require innovative work on architecture, integration and reuse of components, and not only specific technological improvements. From our experience, we anticipate that standards and sharing of modules will emerge during the course of the project. Wherever appropriate TALK will be based on existing and/or developing standards, and will thereby also influence decisions on standardization for multimodal and multilingual representation. Particular emphasis will be on W3C standards when appropriate. Balancing adherence to existing annotation standards with the (necessary) further development of these standards will be an important task in the process of designing annotation formats for the data collection efforts.

For on-going project info visit the TALK website http://www.talk-project.org.

6. REFERENCES

[1] S. Larsson and D. Traum, "Information state and dialogue management in the TRINDI Dialogue Move Engine Toolkit", *Natural Language Engineering*, vol. 6, no. 3-4, pp. 323-340, 2000.

[2] I. Lewin, C. J. Rupp, J. Hieronymous, D. Milward, S. Larsson, and A. Berman, "Siridus system architecture and interface report", *Technical Report D6.1*, Siridus, 2000.

[3] E. Levin and R. Pieraccini, "A stochastic model of computer-human interaction for learning dialogue strategies", *in Proceedings of Eurospeech*, Rhodes, Greece, 1997.

[4] 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*, vol. 8, no. 1, pp. 11-23, 2000.

[5] K. Scheffler and S. Young, "Probabilistic simulation of human-machine dialogues", *in Proceedings of ICASSP*, 2000.

[6] K. Scheffler and S. Young, "Automatic learning of dialogue strategy using dialogue simulation and reinforcement learning", *in Proceedings of HLT*, 2002.

[7] O. Pietquin and S. Renals, "ASR system modeling for automatic evaluation and optimization of dialogue systems", *in Proceedings of ICASSP*, 2002.

[8] D. Litman, M. Kearns, S. Singh, and M. Walker, "Automatic optimization of dialogue management", *in Proceedings of COLING*, 2000.

[9] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system", *Journal of Artificial Intelligence Research (JAIR)*, 2002.

[10] J. Bos, E. Klein, O. Lemon, and T. Oka, "DIPPER: Description and Formalisation of an Information-State Update Dialogue System Architecture", *in 4th SIGdial Workshop on Discource and Dialogue*, Sapporo, 2003.

[11] D. Martin, A. Cheyer, and D. Moran, "The Open Agent Architecture: a framework for building distributed software systems", *Applied Artificial Intelligence: An International Journal*, vol. 13, no. 1-2, 1999.

[12] S. Young, "ATK: An application Toolkit for HTK, version 1.3", Cambridge University, January 2004.

[13] P. Taylor, A. Black, and R. Caley, "The architecture of the Festival speech synthesis system", *Third International Workshop on Speech Synthesis*, Sydney, Australia, 1998.

[14] K. Currie and A. Tate, "O-Plan: the open planning architecture", *Artficial Intelligence*, vol. 52, pp. 49-86, 1991.

[15] J. Moore, M. E. Foster, O. Lemon, and M. White, "Generating tailored, comparative descriptions in spoken dialogue", *in Proceedings of the 17th International FLAIRS Conference (Florida Artificial Intelligence Research Society)*, 2004.

[16] M. Stuttle, J. Williams, and S. Young, "A framework for dialogue data collection with a simulated ASR channel", *in Proceedings of ICSLP*, 2004 (to appear).

[17] J. Williams and S. Young, "Characterizing task-oriented dialog using a simulated ASR channel", *in Proceedings of ICSLP*, 2004 (to appear).

[18] M. Walker, R. Passonneau, and J. Boland, "Quantitative and qualitative evaluation of Darpa Communicator spoken dialogue systems", *in Proceedings of ACL*, 2001.

[19] D. Shapiro and P. Langley, "Separating skills from preference: Using learning to program by reward", *In Proceedings of ICML*, 2002.

[20] O. Lemon and A. Gruenstein, "Multi-threaded context for robust conversational interfaces: context-sensitive speech recognition and interpretation of corrective fragments", *ACM Transactions on Computer-Human Interaction (ACM TOCHI)*, 2004 (to appear).

[21] M. Gabsdil and O. Lemon, "Combining acoustic and pragmatic features to predict recognition performance in spoken dialogue systems", *in Proceedings of ACL*, 2004 (to appear).