Rule-Based Dialogue Management Systems

Article · January 2001

Source: CiteSeer

CITATIONS
3

READS
327

1 author:

Nick John Webb
Union College
55 PUBLICATIONS **556** CITATIONS

SEE PROFILE

Rule-Based Dialogue Management Systems

Nick Webb
n.webb@sheffield.ac.uk
Natural Language Processing Group
University of Sheffield, UK

1 Introduction

Simple question answering systems are no longer enough to answer the complex questions asked by users, and increasingly there is the need to enter into dialogue in order to refine or reiterate user requests. Complex Dialogue Management (DM) components have been developed which allow users to have a more intelligent style of interaction with their information systems. More recently there have been efforts to abstract the principles of DM design, with a view to creating flexible, possibly portable DM systems (Agarwal, 1997; Dybkjær et al., 1998).

Within DM design itself research seems principally focused on two strands of development, either modelling dialogue cognitively (conversation as a collaborative task) or modelling linguistically (capturing individual features of discourse). Both of these remain as important research goals, but for the simplified dialogues which occur in some information retrieval systems, they may represent unrealistic aims. This paper presents an alternative method which fits between these two approaches, by modelling dialogue actions as a series of rules. It is our assertion that these rules give a greater flexibility to dialogue than script-based methods, and as such are a useful tool for describing specific application based dialogues. There is no attempt to claim that we can use these rules to capture all that humans do in dialogue, such as image management or persuasion, but for the majority of current application domains these general abilities are not required. Instead we focus on simple interrogative and command dialogues based on *slot-filler* representations, and attempt to write sets of rules which describe the dialogue process necessary to ensure successful filling and manipulation of these slots.

Many previous implementations of Dialogue Managers have centred around transition based systems. Dialogues are described as increasingly complex state machines, with communicative acts being the actions motivating transitions between states. Whilst this approach is conceptually very clear, it suffers from certain limitations. Description and implementation of state machines becomes much more complex when the information expected from the user can be presented in an arbitrary manner. Each state has to be given sufficient possible transitions to other states to cover each eventuality. There is nothing to stop the user interacting with the Dialogue Management System (perhaps to include new, updated search information) while the system itself is consulting the knowledge base. It is difficult to capture this kind of interaction using a pure state based theory. Furthermore, a greater number of knowledge sources (of which the user can be seen as one) are used by dialogue managers in order to successfully complete a query. The ability to specify and interface to each of these knowledge sources raises a similar problem.

There is a current trend towards modelling dialogue using statistical methods. Taking very large corpora of dialogue information (such as data gained from the Verbmobil project), it is possible to automatically ascertain the *dialogue steps* involved in a set of particular dialogues. However, this seems to be fundamentally modelling the wrong information. Whilst this technique can be very successful at capturing much of the surface regularity within language, it has no bearing on the *motivators* of such a dialogue. Traditionally, in order to find these motivators, we need large plan recognition systems, but it may be possible, within the tightly restricted domain of short interrogative dialogues to model those actions which motivate dialogue within specific domains.

2 Dialogue Managers: A quick overview

2.1 Motivation

Question-Answering systems (e.g. TEAM (Grosz et al., 1987), LADDER (Hendrix et al., 1978)) give answers to correctly phrased natural language questions, but are unable to help the user when required information does not exist exactly in the format required, or the question itself is not understood. At this stage it becomes necessary to conduct a dialogue with the user, in order to further specify or clarify the information they require.

Whatever the system, there have been identified a number of dialogue motivators or reasons for dialogue to be conducted, which can be considered to be universal across domains and applications (Abella, Brown, and Buntschuh, 1996). Some of these are:

1. Queries generated by missing required information
    - Such as missing destination information in a flight booking system
2. Relaxation
    - dropping those constraints preventing system response
3. Augmentation
    - The opposite of relaxation, if the system needs more information to steer it to an accurate response
4. Confirmation
    - Paraphrasing user input for confirmation
5. Disambiguation of user inputs
    - When a user utterance could cover a number of system entries
6. Detection of user confusion/error correction
    - Re-routing the current dialogue to deal with user confusion

Some of these criteria are very general dialogue descriptors however, and may not be valid for specific applications. We are concerned with slot-filling dialogues, where the goal of the interface (the *Frontend*) is to fill some slots from natural language like input, and pass these to some information source (the *Backend*). This Backend could represent a number of different things, for example a microwave, the yellow pages or a train timetable.

In other words interrogative or command dialogues, where the *dialogue* can be seen as repeated attempts to operate over the domain. Each operation may take several steps, updating or revising slots to overcome incompatibility, for example. Once input is in a *correct* form, the slots are executed on the domain, which generally leads to termination and reporting of outcome. This may lead to a repeated attempt (in case of success) or revision and retrying (in the case of failure). Using a DM system in conjunction with a slot-filling mechanism should allow us to explain and explore faults in input then select different responses internally, possibly by automatic variation of slot-fillers.

Ideally, we can say that all information retrieval systems should be able to meet a set of criteria, which could include (but would not be restricted to):

- recognising attribute being set from value
    - e.g. hot $\rightarrow$ temperature
- explaining mismatches of settings with domain and each other
- offering ranked alternative actions in case of failure
- having a degree of *quality consciousness* from a users perspective
    - avoiding pointless repetitions
    - guiding user towards system goal
- spotting missing or contradictory parameters

It is possible, and indeed in the majority of existing systems it is the case that these features could be built into each system interface. However if it can be achieved it would be better to abstract them away from specific domains, creating something more like a widget set seen in the design of graphical user interfaces than anything currently seen in the field of Natural Language Understanding.

2.2 Related Work

There is a great deal of work in the area of natural language dialogue systems. In the main these sophisticated systems (for example OVIS (Nederhof et al., 1997), The Phillips train timetable system (Aust et al., 1995), Sundial (Peckham, 1993; McGlashan et al., 1992), TRAINS (Allen et al., 1995) and Verbmobil (Wahlster, 1993)) are concerned with time and schedule information. These are heavily restricted domains, in terms of size and complexity of data. In addition, there are a number of systems which retrieve addresses from *Yellow Pages*. Most notable of these are *Voyager* (and its sibling, *Galaxy*) (Zue et al., 1990; Zue, 1994) and IDAS[1], an interactive directory assistance project funded by the European Union. IDAS has as a goal the effective disambiguation of user queries, and the narrowing of the search space of the query. This project is still in the development phase. *Voyager* and *Galaxy* have been around for some time, but the implementations are restricted to sample domains or small-scale address databases (e.g. 150 objects in the *Voyager* system (Glass et al., 1995)).

3 Rule-Based Approach

The majority of DM systems at this time are reactive systems which maintain a context of the ongoing dialogue by combining the Information level, the Conversational level and the Intentional level. Most will model the Information level as some slots relating to the domain task, and will dispense with the Intentional level altogether, (although this is not the case for either Verbmobil or TRAINS). The key problem remains the best approach for representing the Conversational level. By restricting the nature and complexity of the domains in this way it allows us to simplify this three layer model. In the example of a yellow pages enquiry system, the Information level is represented by the slots we choose to operate over the domain. Furthermore, there is no need to reason about the Intentional level, as we can assume that a user querying a yellow pages system has the intention of retrieving advertiser information.

Closer examination of the domain specifications and restrictions suggests further requirements. We must be able to handle events from a variety of sources, of which the user is only one. We can expect events to come from our Backend database, such as having no responses to a query or having a command rejected. We may have to deal with specific parser responses or additional information from other independent knowledge sources (such as a thesaurus or world knowledge component).

We see then a model where the user and all internal modules are seen as alike by the DM system, and as such have to deal with arbitrary input and state in response to any of these modules. This suggests a model based on the product of machines or a production rule system. The specification of a product of machines representation could be too complex a description for our requirements, so we have chosen to use production rules to represent the Conversational level of our dialogue, so long as the range of possibilities is fairly limited.

To implement such a model requires that we have a single[2] set of (task dependent) slots surrounded by some modules which produces events, where events are seen as event source, event type and some operands. For example:

User + String + Input String

Yellow Pages DB + List of Addresses + Address List

Box Office + Ticket Not Booked + Reason

The action part of a production rule can be seen as the same sort of thing. Current dialogue state can be calculated at any time from the dialogue history, which is a complete list of past events and states. Input from the user would be parsed, and dependent on the content of the dialogue history, added to existing slots or put into new ones. Operations over those slots would then depend on slot contents and previous operations.

This is not a wholly new approach. (Arakawa and Morimoto, 1997) suggest a dialogue management system built using a production system, and indeed implement one for a hotel reservation database, but are most concerned with the linguistic modelling capability of the system. They highlight the benefits of using production systems, such as ease of development and aspects of modularity. Prior to this, (Monk, 1990) had used action-effect rules to specify a multi-modal interface, claiming that these rules were much easier to generate. At the same time (Olsen Jr,

[1] `http://www.linglink.lu/le/projects/idas`

[2] that is to say, a single set for each dialogue task - a system could have multiple sets around which multiple dialogues could operate

1990) was suggesting the use of Prepositional Production Systems for dialogue description, although specifically handling graphical dialogues.

3.1 An Implementation: The YPA

The YPA is a directory enquiry system which allows a user to access advertiser information in classified directories (De Roeck et al., 2000).

It converts *semi-structured data* in the *Yellow Pages*[3] machine readable classified directories into a set of indices appropriate to the domain and task (Kruschwitz et al., 2000), and converts natural language queries into filled slot and filler structures appropriate for queries in the domain (Webb et al., 1999). The generation of answers requires a domain dependent *query construction* step, connecting the indices and the slot and fillers. The YPA illustrates an unusual but useful intermediate point between information retrieval and logical knowledge representation.

The dialogue component takes input from the user, and tries to match it to the Backend database. If there is a direct match, the Dialogue Manager reports this to the user, as a list of relevance ranked addresses. However, if there is no direct match, the Query Construction Component will perform successive alterations on the query, using some World Model and possible interactions with the User. It is the task of the Dialogue Manager to present and control these interactions.

In the beginning, the DM component of the YPA was modelled on the PURE system of (Agarwal, 1997). However, it soon became clear that the Backend construction process (described in more detail in (De Roeck et al., 1998)) created a variety of index tables which needed to be consulted at different (possibly arbitrary) points in dialogues. Add to this the further external knowledge sources (a thesaurus, domain knowledge), all of which could be beneficial and we saw the need for a dialogue management system which could individually hook in to each of these knowledge sources.

The dialogue manager was implemented as a series of declarative rules, which controlled every process of the dialogue. Once a DM had been built which replicated the functionality of the previous Dialogue Manager, ways were sought to extend its capability. By increasing the granularity of the rules, that is allowing individual rules to hook into individual index tables, we created the possibility of greater flexibility in dialogue construction - specifying under what circumstances each knowledge source was consulted.

In order to demonstrate this, two exactly opposite rule sets were created. one based on the DM component of before and one which performed the actions in counter-intuitive order. There were examples which demonstrated that in some instances, the reversed rule set performed better than the intuitive rule set in retrieving addresses which were more relevant, in a shorter number of dialogue steps.

Furthermore, it was a demonstration of how it was possible to alter the order of dialogue steps by adding and deleting rules. The rules are such that the query will be examined before being submitted to the database to ensure that the location information is filled - indicating that the location slot is considered as mandatory in this application. By removing this rule, the any query without location information will be submitted to the database. If that query produces too many replies (say the user asked for taxi companies) the query construction component will identify missing location information, and ask that the user specify some location. Thus, the results of these two rule sets will be the same, but the dialogue may progress in different directions.

It became apparent during this demonstration that what would be desirable would be the ability to have dynamic rule selection, governed by some form of heuristics, which would allow sets of rules to be selected dynamically based on the current status of the query. This remains a research goal.

3.2 Future Work

This system of rule-based dialogue management grew from a demand to meet the requirements of a specific system, but contained the potential to be abstracted for different domains of the same sort. In order to prove this, once the dialogue manager for the YPA was complete, the same DM system was used to create a set of rules to control a simulated microwave. This was a successful and relatively painless exercise.

Currently we build a deterministic set of rules through a hand-coded approach. Ideally we want to be able to move towards a system where

[3]Yellow Pages® and Talking Pages® are registered trade marks of British Telecommunications plc in the United Kingdom

there is some high level description of a dialogue system (as some sort of state machine, perhaps) which is subsequently compiled down to a set of deterministic rules. It is in this direction that work progresses.

Finally, as mentioned in the previous section, we would like to be able to have some dynamic, 'on-the-fly' rule selection process from a set of possible eventualities, which will allow us to achieve a high level of precision in the retrieved data.

4 Conclusions

We have demonstrated the potential for modelling a restricted class of man-machine dialogues through the use of rule sets. We can demonstrate greater flexibility through the use of these rule sets in the control of dialogue, and believe that there exists the potential for creating tools to allow the description and development of dialogues for new domains.

5 Acknowledgements

Work on the YPA was conducted at the University of Essex, funded by a grant from British Telecoms Intelligent Systems Research group, based at Adastral Park, Martlesham, UK. I would like to thank Sam Steel and Udo Kruschwitz for their useful comments on this paper.

References

Abella, A., M. K. Brown, and B. Buntschuh. 1996. Development Principles for Dialog-Based Interfaces. In *Proceedings of the Workshop on Dialog Processing in Spoken Language Systems, ECAI-96*, Budapest.

Agarwal, R. 1997. Towards a PURE Spoken Dialogue System for Information Access. In *Proceedings of the ACL/EACL Workshop on "Interactive Spoken Dialog Systems: Bringing Speech and NLP Together in Real Applications"*, pages 90–97, Madrid.

Allen, J., L. Schubert, G. Ferguson, P. Heeman, C. Hwang, T. Kato, M. Light, N. Martin, B. Miller, M. Posesio, and D. Traum. 1995. The TRAINS project: a case study in building a conversational planning agent. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:7–48.

Arakawa, N. and T. Morimoto. 1997. Semantic and Discourse Processing with A Feature Structure-based Production System. In *Proceedings of the Natural Language Processing Pacific Rim Symposium 1997*, Phuket, Thailand.

Aust, H., M. Oerder, F. Seide, and V. Steinbiss. 1995. The Philips automatic train timetable information system. *Speech Communication*, 17:249–262.

De Roeck, A., U. Kruschwitz, P. Neal, P. Scott, S. Steel, R. Turner, and N. Webb. 1998. YPA - an intelligent directory enquiry assistant. *BT Technology Journal*, 16(3):145–155.

De Roeck, A., U. Kruschwitz, P. Scott, S. Steel, R. Turner, and N. Webb. 2000. The YPA - An Assistant for Classified Directory Enquiry. In *Intelligent Systems and Soft Computing: Prospects, Tools and Applications*, volume 1804 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer Verlag.

Dybkjær, L., N. O. Bernsen, R. Carlson, L. Chase, N. Dahlbäck, K. Failenschmid, U. Heid, P. Heisterkamp, A. Jönsson, H. Kamp, I. Karlsson, J. v. Kuppevelt, L. Lamel, P. Paroubek, and D. Williams. 1998. The DISC approach to spoken language systems development and evaluation. In *Proceedings of the 1^{st} International Conference on Language Resources and Evaluation*, pages 185 – 189, Granada, Spain.

Glass, J., G. Flammia, D. Goodine, M. Phillips, J. Polifroni, S. Sakai, S. Seneff, and V. Zue. 1995. Multilingual Spoken-Language Understanding in the MIT VOYAGER System. *Speech Communication*, 17:1–18.

Grosz, B. J., D. E. Appelt, P. A. Martin, and F. C. N. Pereira. 1987. TEAM: An experiment in the design of transportable natural language interfaces. *Artificial Intelligence*, 32:173–243.

Hendrix, G. G., E. D. Sacerdoti, D. Sagalowicz, and J. Slocum. 1978. Developing a natural language interface to complex data. *ACM Transactions on Database Systems*, June:105–147.

Kruschwitz, U., A. De Roeck, P. Scott, S. Steel, R. Turner, and N. Webb. 2000. Extracting Semi-Structured Data - Lessons Learnt. In *Proceedings of the 2^{nd} International Conference on Natural Language Processing (NLP2000)*, Patras, Greece. Springer-Verlag.

McGlashan, S., N. Fraser, N. Gilbert, E. Bilange, P. Heisterkamp, and N. Youd. 1992. Dialogue Management for Telephone Information Systems. In *Proceedings of the International Conference on Applied Language Processing*, Trento, Italy.

Monk, A. 1990. Action-effect rules: a technique for evaluating an informal specification against principles. *Behaviour and Information Technology*, 9(2):147–155.

Nederhof, M.-J., G. Bouma, R. Koeling, and G. van Noord. 1997. Grammatical analysis in the OVIS spoken-dialogue system. In *Proceedings of the ACL/EACL Workshop on "Interactive Spoken Dialog Systems: Bringing Speech and NLP Together in Real Applications"*, Madrid.

Olsen Jr, D. R. 1990. Prepositional Production Systems for Dialog Description. In *Proceedings of Computer Human Interaction: Human Factors in Computing Systems (CHI'90)*, pages 57–63.

Peckham, J. 1993. A new generation of spoken dialogue systems: results and lessons from the SUNDIAL project. In *Proceedings of the 3^{rd} European Conference on Speech Communication and Technology*, pages 33 – 40, Berlin, Germany.

Wahlster, Wolfgang. 1993. Verbmobil: Translation of Face-to-Face Dialogues. In *Proceedings of the 3^{rd} European Conference on Speech Communication and Technology*, pages 29–38, Berlin, Germany.

Webb, N., A. De Roeck, U. Kruschwitz, P. Scott, S. Steel, and R. Turner. 1999. Natural Language Engineering: Slot-Filling in the YPA. In *Proceedings of the Workshop on Natural Language Interfaces, Dialogue and Partner Modelling (at the Fachtagung für Künstliche Intelligenz KI'99)*, Bonn, Germany.

Zue, V. 1994. Toward Systems that Understand Spoken Language. *IEEE Expert Magazine*, February:51–59.

Zue, V., J. Glass, D. Goodine, H. Leung, M. Phillips, J. Polifroni, and S. Seneff. 1990. The VOYAGER Speech Understanding System: Preliminary Development and Evaluation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing.*