TOWARDS MULTIMODAL NATURAL LANGUAGE INTERFACES FOR INFORMATION SYSTEMS - the LINLIN approach

Lars Ahrenberg, Nils Dahlbäck, Annika Flycht-Eriksson, Arne Jönsson, Pernilla Qvarfordt, Lena Santamarta & Lena Strömbäck
{lah, nilda, annfl, arnjo, perqv, lensa, lestr}@ida.liu.se

Natural Language Processing Laboratory
Department of Computer and Information Science
Linköping University, SE-581 83, LINKÖPING, SWEDEN

1 Introduction

This paper briefly presents some assumptions and results that have been guiding our research on natural language interfaces for the past ten years[1]. We first review results relating to different modules of the pure natural language system and then proceed to discuss current work on a multimodal application.

Natural language interfaces to information systems constitute specific sub-languages, a set of closely related linguistic or multimodal genres. This entails that these dialogues only contain a subset of all the language features associated with general natural language dialogues. A corollary of this is that we can develop general dialogue models for this class of systems, which can be customized to particular applications. There is hence a middle way between the full-fledged human dialogue capability systems, and the one-shot developed system. On the one hand, many aspects are still basic research issues, on the other hand, there is a lack of generalizability. We are striving for generalizability within the sub-languages of multimodal interaction. The advantage of this approach is that we can seek for, and often find, solutions that are sufficient for the sub-language of information systems, without having to manage the full human linguistic capability. An important aspect of our approach is that empirical investigations are necessary in order to reveal the sub-language of a particular information system. In what follows we will illustrate this approach for a number of different components of a dialogue system, both by describing obtained results and by describing current work in progress within this framework. An overview of our system is presented in figure 1.

2 The Dialogue Manager

The dialogue manager is the kernel of the system [4]. Its design is based on data from Wizard of Oz investigations [2] that lead us to the following conclusions: The basic interaction with

[1] The References section includes references to papers presenting our current work in more detail.

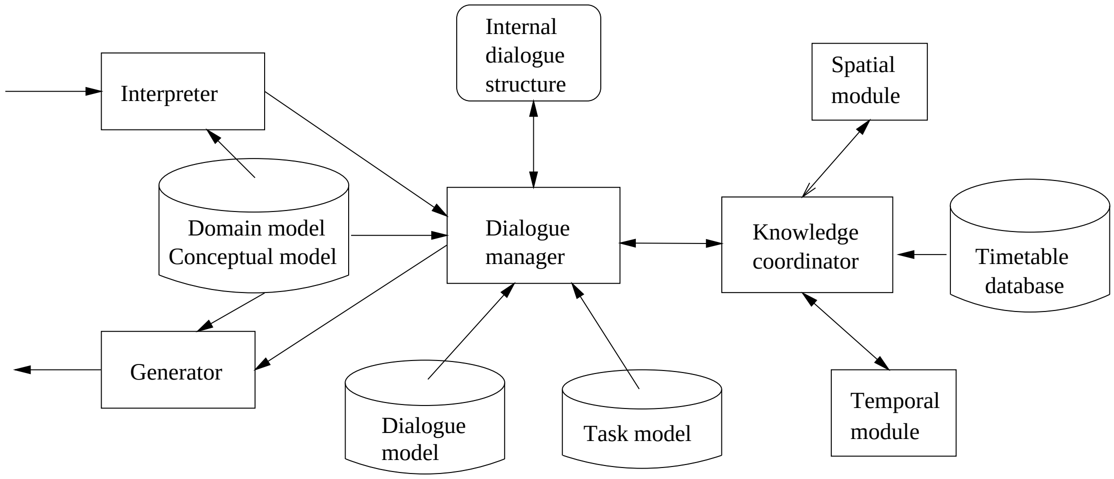

Figure 1: An overview of the LINLIN system. Arrows indicate dataflow.

information systems is simple and can be structured in terms of IR-segments, sequences of utterances starting with an initiative and ending with a response. The most common type of initiatives are questions, which are also fairly simple. The majority can be analyzed as asking about properties of given objects, or conversely, asking for objects satisfying a list of properties. Two types of topics, beside database objects and their properties, occur: the system itself, and previous dialogue contributions, i.e. utterances. Adjacent segments cohere in simple ways, although the complexity may vary from one domain to another. The common cases are: continued querying on a given set of objects, or continued querying on a given set of properties

The dialogue manager controls the interaction by means of a dialogue grammar and holds information needed by the modules in the interface, including the dialogue manager itself. It can be viewed as a controller of resources for interpretation, background system access and generation. Actual interpretation and generation are carried out by other modules of the interface, including the ability to interpret sentence fragments, multi-sentential, extra-grammatical utterances and anaphora resolution.

The dialogue manager can utilize information from various knowledge sources, such as domain knowledge, task knowledge and conceptual knowledge [1], in order to control the interaction. For instance, for task oriented information applications, where information on a variety of parameters is needed in order to access the background system, such as departure and/or arrival time and day, the dialogue grammar does not only utilize information on objects and properties, as discussed above, but also information in the task model to see what information is missing. This is used by the dialogue manager to generate for the user meaningful follow-up questions to underspecified information requests, simply by inspecting the task model and asking for the additional information required to fulfill the task. As we add a separate task model we only need to update the dialogue grammar to also consider information in the task frame.

3 The Interpreter

The interpreter is responsible for the linguistic analysis. The design principles given above also applies to the design of the interpreter. Our previous system relied on a traditional deep and complete analysis of the utterances. This required much effort on building grammars and lexicons and also gave problems with robustness due to the large variations of user input. Instead we have used shallow and partial interpretation, where the grammars and lexicons are derived from our empirical investigations. The interpretation is driven by the information needed by the background system. Partial interpretation is particularly well suited for dialogue systems, as we can utilize information from the dialogue manager to guide the analysis.

The interpreter is based on a variant of PATR-II and a chart parser which allows for both partial and traditional complete analysis. The interpreter provides domain concepts and a set of markers to the dialogue manager. The information needed by the interpretation module, i.e. grammar and lexicon, is derived from the database of the background system and information from dialogues collected in Wizard-of-Oz-experiments.

Results on partial and shallow interpretation is presented in [5, 8].

4 The Generator

For our first systems (assuming written interaction) answers were mostly taken directly from the background system, and if so, in tabular form. Answers often included information from previous questions so that the user had access to all information s/he had asked for about a given set of objects. In the case that the user inquired about system properties, or seemed to need information about the system, pre-stored answers were used, giving the user complete information about some aspect of the system. Thus, global considerations of user needs and efficiency of interaction overruled the Gricean maxim of "Do not make your contribution more informative than is required".

In a multimodal system this seems like a useful design solution, but it does not work well for spoken systems where the limitations of the channel requires more condensed responses. As the dialogue system will be used also for telephone interaction the information presented has to be restricted. This restriction makes the maxim of relevance become very important and the answer has to be presented in a way which is maximally tailored to the user's needs and linguistic preferences. The same information from the background system will thus be presented to the user in many different surface forms depending on the what was the focus of the question of the user. Prosody will have a central role in the structuring and focusing of spoken information.

5 Domain and task knowledge

For many simple information retrieval systems, a background system access is possible from the user's initial request. However, there are some cases where the user's request requires the system to initiate a clarification subdialogue or respond with information on the properties of the background system. For this to work the dialogue manager needs to consult a domain model. For some applications, we also need a model of how domain concepts are to be interpreted in a certain context and their relation to the domain model, for instance, the concept "room" in a travel agency application has properties such as shower associated with it that is not what is normally used to describe a room. The use of domain and concept models are further discussed in [1].

However, there are applications where it is not a straightforward task to map a request by the user to a background system access, for instance, the application that we are currently investigating, timetable information for local buses. From our empirical investigations on the properties of that sub-language we found that we need more sophisticated knowledge models. A user's natural way of expressing a departure or arrival location in a bus traffic information system is not by means of the "official" name of a bus stop. Instead, other expressions are utilized, such as an area or town district, a set of reference points, or a street. Consequently, a spoken dialogue system that provides timetable information for local bus traffic must be able to map such an imprecise description to a set of bus stops that correspond to the description before it can access the timetable database. This can be accomplished by mapping a user's qualitative geographical description into a quantitative and utilize a geographical information system to find the corresponding bus stops [3]. Requests to and information given by the spatial and temporal modules are coordinated by a module called the knowledge coordinator. This module receives requests for information from the dialogue manager and decides what knowledge source to consult.

For the timetable information domain, the dialogue manager also needs task models in order to prompt the user for information on different aspects before we can access the background system. In this domain we have identified several different tasks that the system need to perform, for example providing trip information and route information. For the trip information task, information is needed by the system on departure and/or arrival time and day, as well as a more detailed specification of departure and arrival locations. Knowledge about the tasks is represented in task models which are used by the dialogue manager to ask the user for the information required to fulfill the task.

6 Towards multimodal interaction

Current work aims at extending the interface to handle multimodal interaction in the domain of traffic information. Figure 2 shows an overview of the prototype system's graphical interface including maps, tables, forms, menus and buttons. Input modalities are mouse keyboard and speech.

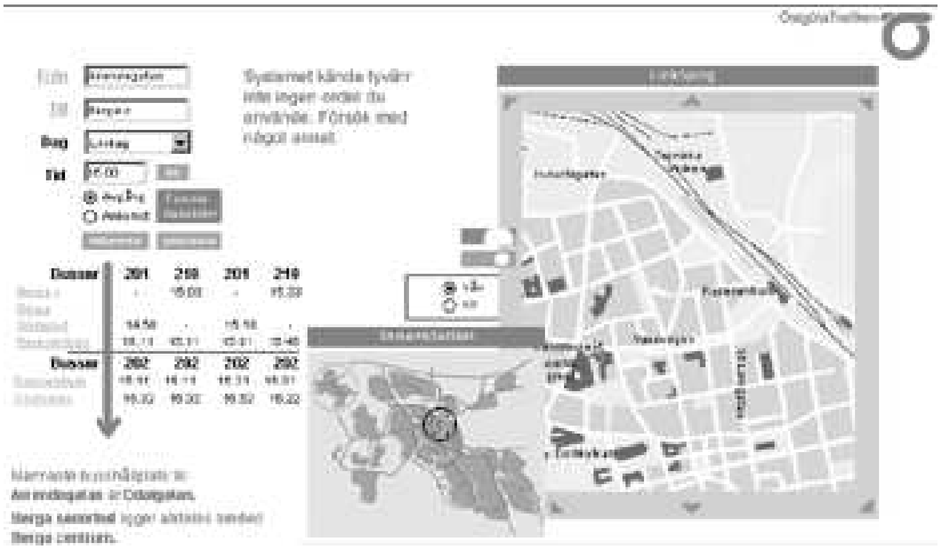

Figure 2: The multimodal interface

Multimodal interaction requires knowledge on the use of various modalities for different communicative acts. Another factor that can influence the communicative acts is the user's domain knowledge. We found, for instance, that users with weak domain knowledge performed better using a multimodal system allowing spoken interaction. They also preferred to use it, whereas users with good domain knowledge preferred to interact by using the mouse, and also performed better when doing so [7, 6].

For multimodal interaction the various modules described above need to be modified. The main reason is that when introducing more modalities the users incorporate other sublanguages in the dialogue. In a multimodal system the users do not just talk about the task, but also how to manipulate the system. This communication also allows for non-verbal signs such as gesture. How this affects the language used, must be determined by empirical investigations. A dialogue act can be performed by the user by presenting information to the system either by speaking, writing or clicking at some point other than the navigation buttons. A currently open research issue is how to distinguish between communicative and non-communicative actions in a multimodal system allowing for this range of communicative acts. The interpretation module needs to be able to integrate input from different modalities such as maps and forms and discriminate those not resulting in a dialogue act.

We plan to make a demonstrational system publicly available at a bus station. For such a system, other input devices than keyboard and mouse are probably more effective. For that reason we intend to experiment with other input devices such as a pen. The step from keyboard, mouse and speech input to speech and pen will not require any changes to the system's basic architecture. Using speech and mouse in combination results in similar gestures as a pen. Writing with a pen is easier than a keyboard, as this is a writing tool used by most people. Finally, selections can be as easy using a pen as a mouse. However, the expressiveness in the gestures using a pen could be higher than using a mouse, since a pen can leave a trace. A trace lets the user see what s/he has done, and s/he can therefore use more symbolic gestures.

We also intend to improve generation by studying what knowledge is required in different types of situations of information mis-matches, and how multimodality can be utilized to remedy these situations. For instance, with multimodal systems there is the possibility to supply system and domain information through a side channel, e.g. a map. Furthermore, in multimodal interaction the spoken and graphical information have to co-operate to facilitate the user's perception of the system's answer. Different kinds and amounts of feedback can influence the user's perception of the system. Examples of different kinds of feedback can be, only visual feedback, only verbal, or a combination of visual and verbal feedback.

7 Summary

In this paper we presented an overview of our work towards a multimodal natural language information system. The system was originally developed for written language interaction, but is now being adapted for spoken and multimodal interaction. While these extensions raise challenges for the interpretation, generation and dialogue modules, we believe that our empirically oriented sub-language approach is feasible and allows for interesting and useful generalizations to be made.

References

[1] Nils Dahlbäck and Arne Jönsson. Integrating domain specific focusing in dialogue models. In *Proceedings of Eurospeech'97, Rhodes, Greece*. European Speech Communication Association, 1997.

[2] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of oz studies – why and how. *Knowledge-Based Systems*, 6(4):258–266, 1993. Also in: *Readings in Intelligent User Interfaces*, Mark Maybury & Wolfgang Wahlster (eds), Morgan Kaufmann, 1998.

[3] Annika Flycht-Eriksson and Arne Jönsson. A spoken dialogue system utilizing spatial information. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.

[4] Arne Jönsson. A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering*, 3(2/3):103–122, 1997.

[5] Arne Jönsson and Lena Strömbäck. Robust interaction through partial interpretation and dialogue management. In *Proceedings of Coling/ACL'98, Montréal*, 1998.

[6] Pernilla Qvarfordt. Usability of multimodal timetables: Effects of different levels of domain knowledge on usability. Master's thesis, Linköping University, 1998.

[7] Pernilla Qvarfordt and Arne Jönsson. Effects of using speech in timetable information systems for www. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.

[8] Lena Strömbäck and Arne Jönsson. Robust interpretation for spoken dialogue systems. In *Proceedings of ICSLP'98, Sydney, Australia*, 1998.