ARNE JÖNSSON

Application-Dependent Discourse Management for Natural Language Interfaces: An Empirical Investigation

Abstract

This paper presents results from a refined analysis of a series of Wizard of Oz-experiments with focus on discourse management. The study is part of a larger project aimed at designing a general natural language interface. It is argued that a natural language interface needs different referent resolution mechanisms for different combinations of background systems and scenarios. I present three mechanisms that can be used: proximity, object hierarchy and goal-directed control. Finally I suggest the use of a Natural Language Interface Management System (NLIMS) for customization of natural language interfaces to different applications.

1 Introduction

The language used by a user when communicating with a computer application (database, expert system etc.) in natural language will differ from the language used in a spoken face-to-face communication between humans as well as the language used in written communication.[1]

The development of natural language interfaces more sophisticated than simple question-answering requires knowledge of the characteristics of such an interaction, not only of the language but also of how the discourse progresses. In a series of experiments (Dahlbäck & Jönsson 1989; Jönsson & Dahlbäck 1988) we have studied these phenomena and presented results on the characteristics of the interaction between a human and a natural language interface.

We have conducted experiments using five different background systems (see section 2) and 21 subjects, all computer novices. After every simulation we have interviewed the subject in order to find out how they liked the 'system'. No

[1] In Jönsson & Dahlbäck (1988) we elaborate more on this.

Proceedings of NODALIDA 1989, pages 297-307

subject understood that it was a simulation and many (too many?) were *not* surprised with the system's capabilities. We have focused on computational aspects of the interpretation of the Subjects/Users utterances. This means that we have not considered the utterances typed by the system (simulator).

There are 1047 utterances which we have divided into four different categories; Initiative, Response, Resp/Init and Clarification, (cf. Linell, Gustavsson, & Juvonen (1988)). What poses problems from a computational point of view are mainly the user Initiatives and especially the indexical initiatives. There are 193 indexical subject utterances, which is 49% of the initiatives. Dahlbäck & Jönsson (op. cit.) further analyze these indexical utterances and also provide some other results. But the results have to be refined in order to see what mechanisms different kinds of NLI's should adopt when interpreting an indexical utterance.

2 Scenarios and Background Systems

In our experiments we have varied the type of background system and the scenario. The analysis of these experiments exposes considerable differences between different combinations of background systems and scenarios. These differences must be considered, as this study is concerned with exploring computational mechanisms for handling indexicals. Consequent our background systems and scenarios require a presentation.

The *scenario* is the characteristics of the task the subject has to solve. We used two different types of scenario, namely information retrieval and configuration. They differ in that a *configuration*, besides data retrieval, must deal with a dynamically updated object, such as an order or specification, during the discourse, while *information retrieval* is less complex and involves only retrieving facts from an information base.

Further, we use two different kinds of *background systems*, simple database systems and advisory systems. A *database system* simply returns facts from the database. The subject must state an information seeking question, he or she cannot ask for an opinion or evaluation from the system. An *advisory system*, on the other hand, can also be used for consultation. The subject can ask for advice i.e. the system's evaluation of certain facts. In both systems we have most answers as pre-stored texts, or text fragments which are completed, and the simulator (wizard) selects the appropriate text.

The different experimental settings as a function of the combination of background system and scenario are shown in Table 1. The name refers to the background system and the number of experiments of each combination is stated inside the parenthesis. Section 4 contains some examples from our simulations.

There are two systems of the most simple combination, namely database background system and information retrieval; one where the subjects ask questions concerning publications at our department (PUB), and one where they can ask questions about one of the education programmes (C-line). These systems allow the subjects to ask questions and also to refer to previous pieces

Scenario		Background system: Data base	Background system: Advisory system
Scenario	Information retrieval	PUB (4) C-line (5)	Wines (4)
Scenario	Configuration	Travel (3)	HiFi (5)

Table 1: Scenarios and background systems.

of discourse, but they do not allow questions like *Which books have I ordered*? (configuration) or *Is that a good course*? (advice).

A bit more sophisticated is the information retrieval advisory system for selecting wines where the subjects can get advice on whether a wine is suitable for some specific meal, dialogue 1. However, it is not possible to select one wine for the first course and then order that wine, i.e. configuration is not possible. This does not prevent the user from referring to previously mentioned wines.

```
.
U:9> What do you recommend with crab-filled avocado?
S:10> Searching ...
       With crab-filled avocado a white wine is suitable
       Do you want more information about any specific wine?
U:11> Yes  in the price range from 30 to 50 SEK
S:12> Please wait
       Do you want rhine-, mosel- or burgundy wines. Or all.
U:13> Rhine wine
S:14> Searching ...
       The following wines exist ...  listing ....
.
```

Dialogue 1. Wines:1 (The corpus is in Swedish and translated for verbatim correctness.)

Another combination is configuration and database. This is found in the travel system. Here the subjects can order a trip to a holiday resort by retrieving information from a database and using these facts to configure a trip. However, they can not get advice, only access the database. This is discussed in section 4.2.

Finally, we have the HiFi dialogues, which are both advisory system and configuration. Here the subjects can get advice on HiFi equipment and also select their own outfit. The fact that a system is an advisory system does not prevent the use of canned texts. In fact we try to use pre-stored texts and sentence fragments as much as possible in order to get a uniform setting and amplify the belief that the subject interacts with a computer.

3 Discourse Management

The research reported here is part of a larger project on designing a general Swedish natural language interface. We have a pilot version of the system, called FALIN (Ahrenberg, 1989) which is a constraint-based, object-oriented model for a natural language dialogue system. FALIN provides a content structure which is an exhaustive description of the utterance. The content structure can be seen as an instantiated case frame and can be used for accessing the database or knowledge base. As an example, the content structure for utterance U:104 *ordered turntable* in dialogue 2, HiFi:3, would look like:

CLASS:	$Question;
SPEAKER:	Subject#3;
ADDR:	System;
BASIS:	Subject#3.Order;
ASPECT:	$Turntable

When FALIN builds a content structure it can access information from what is called the `Discourse.State` for finding the referent to an anaphoric expression. The `Discourse.State` contains information about which of the many objects introduced in the discourse is currently active and also pending objects. This information is used to determine the referent, i.e. to select which of the many possible objects in the discourse that a certain anaphoric expression refers to. Thus, one needs some kind of dynamic discourse representation. This representation can be constructed either from the discourse using certain coherence criteria (see below) or like Grosz (1977), and Reichman (1985), from the task structure. In the present case the task structure is reflected by the conceptual structure of the knowledge base, see figure 1. The conceptual model of the knowledge base is a static structure and not a dynamically created description of the discourse.

The problem of managing discourse is normally divided in two parts: segmentation and finding the focused object. **Segmentation** is the process of finding out which parts in the discourse belong together and where one changes topic. The problem of segmentation is connected to the problem of coherence in discourse, where *cue words* (Reichman 1985), *coherence relations* (Hobbs 1985), *recency and temporal progression* (Webber 1988; Merkel 1989) are regarded as major coherence criteria.

The **focusing** problem is to find, given a segment, which of the many different objects that are actually talked about, is in focus. This is often done by

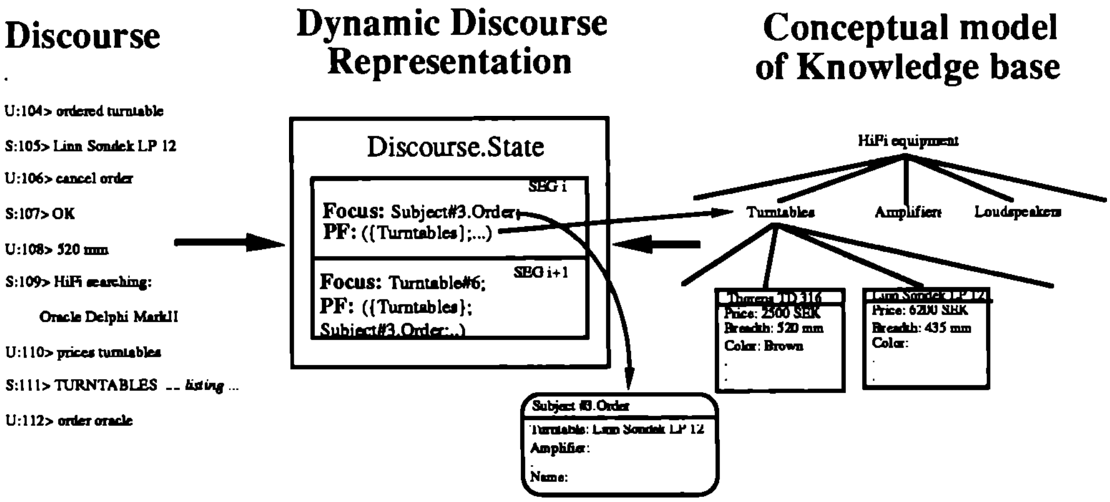

Figure 1: The Dynamic Discourse Representation

maintaining both a reference to the focused object (or objects), if any, and also a, possibly empty, list of objects that might be focused later, potential foci (PF) (Sidner 1983).

4 Empirical Findings

In this section I will illustrate the various strategies found by examples from our corpus. However, as can be seen from Table 1 there are few experiments of each combination of scenario and background system. Therefore I will not give any quantitative data on the occurrence of the different phenomena reported here.

4.1 Some General Findings

First of all there are **individual differences**, as can be seen from the examples in dialogues 2 and 3, both taken from HiFi, the most complex combination of scenario and background system. Dialogue 2 is a good example of what has been described as 'telegraphic' (Guindon, Shuldberg & Connor 1987), or 'computerese' (Reilly 1987).

This user started with a less fragmented language, but when he found that it was possible to interact in this 'cryptic' fashion he started to explore the database—not the language component as it may seem. Compare this dialogue fragment with dialogue 3.

The subject in dialogue 3 is a secretary and she is the subject which uses the most sophisticated language constructions. I have not selected a special part of the dialogue, although utterance 15 is our longest. This subject writes more

```
.
U:104> ordered turntable
S:105> Linn Sondek LP 12
U:106> cancel order
S:107> OK
U:108> 520 mm
S:109> HiFi searching:
         Oracle Delphi MarkII
U:110> prices turntables
S:111> TURNTABLES  .... listing...
U:112> order oracle
.
```

Dialogue 2. HiFi:3

```
.
U:11> What is the price if I instead take a cheaper pair of
        loudspeakers
S:12> What loudspeakers?
U:13> The cheapest there are
S:14> 70 600 SEK
U:15> Now I want to change to a better pair of loudspeakers which
        are at the most 29 400 SEK more than the ones I have now.
        The new items I'm getting should be as appropriate as possible
        for a small room.
```

Dialogue 3. HiFi:1

than one sentence in many of her turn-takings and illustrates the fact that certain non-linguistic factors, like type-writing skills, affect the language.

As noted above, **cue words**, **coherence relations**, and **temporal progression** are regarded as major coherence criteria. In our corpus I found no cue words used by the subject to signal the beginning of a new segment or termination of a segment and temporal progression is not significant for identifying segment shifts.

Although these findings seem pessimistic concerning the future of NLIs there are other results which are more optimistic. Especially the use of what I call Resp/Init, i.e. responses from the system which also include a **system initiative**. If the system does not only respond to the user's input, but also puts forward a follow up question, the dialogue becomes simpler. This is seen in dialogues 1 and 4. In dialogue 1 the system takes an initiative by hinting on a probable new piece of information to the user. In dialogue 4 the system follows a plan to help the user fulfil his goal.

Note that I do not consider the interpretation of the indexical utterances U:8 or U:10 in dialogue 4 problematic as they are responses to questions posed by the

```
.
U:6> We would like to go to Corfu
S:7> What hotel class do you want?
U:8> The best
S:9> How long will you be gone?
U:10> For two weeks
S:11> Vingfritid searching
        Hotel Hilton +++++ costs 5575 SEK/person
U:12> How much does the cheapest class cost?
S:13> Vingfritid searching
        Hotel Royal +++ costs 2245 SEK/person
.
```

Dialogue 4. Travel:2

system and, as such computationally easier to handle than the initiative taken in for instance U:12. This is also seen in dialogue 1 where for instance utterance U:11 is a response that could be interpreted using case-frame expectations like those described in Carbonell & Brown (1988).

4.2 Techniques for Discourse Management

The motivation for my study was to explore which computational mechanisms to use for a certain combination of background system and scenario. I have identified three different mechanisms for finding the referent to an indexical, that are applicable to different dialogue categories. I call them *Proximity*, *Object hierarchy* and *Goal directed control.*

Object **proximity** is found as the major criterion in the database information retrieval dialogues. In these dialogues the subject discusses one task and when that task is finished the object being discussed is dropped and a new object is introduced. This means that finding the referent is no problem; it is the closest object at hand, i.e. the object in focus. If the proximity heuristic fails to find the referent, some simple **recency** criteria can be used, like searching the segment stack for the referent. Carbonell & Brown (1988) present a Multi-Strategy approach to anaphora resolution using case frame information. They restrict themselves to a single previous sentence but suggest the use of recency if context information is necessary. Dialogue 5 from the PUB system gives an example.

This dialogue is very representative for all the PUB dialogues and many of the C-line dialogues. In this dialogue the signals to manipulate the segment stack and creating new segments is found from the syntactic/semantic structure, for example the segment shift in U:15 in dialogue 5. Further, the next utterance, utterance PUB:2:21, is: *What books are there on artificial intelligence?* This utterance cannot be seen as a continuation of any object on the segment stack, and thus the segment stack is popped and a new segment is created.

Proximity can also be viewed as a simple variety of goal-directed control (see

```
.
U:13> Who has written a book on linguistics?
S:14> PUB Searching. There are three authors of books about
        linguistics. Do you want to see them all?
U:15> No only one.
S:16> PUB Searching. Brown and Yule has written a book on
        linguistics.
U:17> What is the title
S:18> PUB Searching. Discourse analysis
U:19> Where is it.
S:20> PUB Searching. Owner: Mats Wir'en
.
```

Dialogue 5. PUB:2

below). However, I regard this mechanism separately since my goal is to find the simplest possible mechanism for each combination of background system and scenario. The reason is that one should not use a more sophisticated mechanism than necessary for a certain application.

In section three I discussed whether the referent is to be found using information from the discourse or from a conceptual model of the database. Object proximity is one example of using the discourse as the source for finding the referent. If a conceptual model of the database is used we get what I call **object hierarchy**. Grosz (1977) used partitioned semantic networks—focus spaces—which reflected the task structure for interpreting discourse. Bobrow & Webber (1980) use a general object-centred knowledge representation for syntactic and semantic processing. Figure 1 gives an example of a conceptual model of the database for the HiFi dialogues. Dialogue 6 taken from the travel system dialogues is a good example of when it is more or less impossible to find the referent using only information about the discourse.

Here the subject discusses excursions in utterances S:33 to U:40. Thereafter the topic is dropped and the subject orders his travel. Twenty eight utterances later he once more raises the question of excursions in U:68. However, this question does not refer to excursions in general (as before), but to excursions on Corfu. Information about where the user wants to go is found in the subject's order and using the hierarchically organized database the information on excursions is found by searching this hierarchy.

Finally we have **Goal directed control**. I use the phrase goal directed control to describe the mechanism where the system uses knowledge about what goals the user has to fulfil. Grosz & Sidner (1986) identify a discourse segment purpose which provides information on what purpose a certain segment has in the discourse. Carberry (1989) presents a number of discourse goals and rules for utterances in information-seeking dialogues. She also identifies a number of plans for nonprimitive tasks. Litman (1985) provides not only plans but also

```
.
S:33> Do you want VingFritid to book any excursions?
U:34> Which ones exist?
S:35> VingFritid searching. Corfu excursions: Town sight-seeing,
        Island tour, Buss to Glyfada, Boat tour to Kassiopi,
        Afra by night.
U:36> What is Island tour and Afra by night
S:37> Island tour: .. listing...
        Afra by night: .. listing...
U:38> Can't  I book these excursions when I arrive?
S:39> Yes you can.
U:40> Then I will not book any excursion now
.
.
U:68> Can you send us a brochure on the excursions?
.
```

Dialogue 6. Travel:2

meta plans. These studies all report the importance to model and use the users expectations and communication goals when interpreting an utterance.

Dialogue 2 is an example where knowledge about the user's goal can help. Utterance U:108 signals a segment shift from one turntable model to another. The user wants to find a turntable in the database with breadth 520 mm.[2] However, 520 mm could also be interpreted as a question whether the canceled turntable was 520 mm or if there is any item in the database that is 520 mm.

To handle this we assume that the discourse state consists of both focused object and potential foci (PF) objects. After utterance S:107, Turntables is in PF as turntables are currently discussed, so 520 mm could be interpreted as either the breadth of the canceled turntable or of any turntable. Now, the user's goal is to complete an order and after canceling one item it is probable that he will order another of the same type. So a search for another turntable occurs.

To summarize:

- Proximity is the major criteria in the database information retrieval dialogues
- Object hierarchy is important for database configuration dialogues
- Goal directed control is necessary for advisory system configuration dialogues.

[2]In this particular dialogue, the user has discussed the breadth of the items before and therefore breadth is the aspect being discussed. I will avoid too many different details and therefore omit the mechanisms to handle this.

5 Discussion

This paper is somewhat pessimistic concerning the building of a general NLI, as the mechanisms for handling indexicals differ depending on both task structure and type of background system. However, developing the various mechanisms is one problem and knowing which mechanism to use another. This second problem, I believe, can be solved by the use of Natural Language Interface Management System (NLIMS). Kelly (1983) created a natural language interface using Wizard of Oz experiments for customizing the lexicon and grammar for a calender system and Good, Whiteside, Wixon & Jones (1984) report similar ideas for developing a command language. My idea is to use a NLIMS for deciding the priority of the different discourse strategies. Such a system should have a component for morphological and syntactic analysis, a collection of different semantic components and a collection of mechanisms for discourse management. The language engineer first builds a prototypical interface using knowledge about the type of background system and also maybe some information concerning the future use of the system. Then he runs a series of simulations. These simulations are used for updating the grammar and lexicon automatically (Jönsson & Ahrenberg, 1989) and also for selecting the appropriate mechanisms for discourse management. The discourse handler can consist of more than one mechanism with different priorities, i.e. if the referent cannot be found using one mechanism another is tried. Of course the system must be updated after a number of runs which also could imply that new simulations need to be performed.

6 Acknowledgements

This research is very much related to the experiments I conducted with Nils Dahlbäck and as always I am indebted to him. Lars Ahrenberg read previous versions of the paper and forced me to sharpen my terminology. I am also obliged to Magnus Merkel and Mats Wirén for valuable discussions. But of course all remaining mistakes are my own.

References

Ahrenberg, L. 1989. A Constraint-Based Model for Natural Language Understanding and a Pilot Implementation, Research Report, LiTH-IDA-R-89-22, Linköping University.

Bobrow, R. J. & Webber, B. L. 1980. Knowledge Representation for Syntactic/Semantic Processing. *Proceedings of AAAI-80.*

Carberry, S. 1989. A Pragmatics-Based Approach To Ellipsis Resolution. *Computational Linguistics*, 15(2):75–96.

Carbonell, J. G. & Brown, R. D. 1988. Anaphora Resolution: A Multi-Strategy Approach. *Proceedings of the 12th COLING Conference.* Budapest.

Dahlbäck, N. & Jönsson, A. 1989. Empirical Studies of Discourse Representations for Natural Language Interfaces. *Proceedings of the Fourth Conference of the European Chapter of the ACL.* Manchester.

Good, M. D., Whiteside, J. A., Wixon, D. R. & Jones, S.J. 1984. Building a User-Derived Interface. *Communications of the ACM*, 27(10):1032–1043.

Grosz, B. J. 1977. The Representation and Use of Focus in Dialogue Understanding. Unpublished Ph.D. Thesis. University of California, Berkely.

Grosz, B. J. & Sidner, C. L. 1986. Attentions, Intentions, and the Structure of Discourse. *Computational Linguistics*, 12(3):175–204.

Guindon, R., Shuldberg, K. & Connor, J. 1987. Grammatical and Ungrammatical structures in User-Adviser Dialogues: Evidence for Sufficiency of Restricted Languages in Natural Language Interfaces to Advisory Systems. *Proceedings, 25th ACL*. Stanford, CA.

Hobbs, J. R. 1985. On the Coherence and Structure of Discourse. Report No. CSLI-85-37.

Jönsson, A. & Ahrenberg, L. 1989. Extensions of a Descriptor-Based Tagging System into a Tool for the Generation of Unification-Based Grammars. I. Lancashire [Ed.] *Research in Humanities Computing*. Oxford University Press, Oxford.

Jönsson, A. & Dahlbäck, N. 1988. Talking to a Computer Is not Like Talking to Your Best Friend. *Proceedings of the First Scandinavian Conference on Artificial Intelligence*. Tromsø, Norway. IOS, Amsterdam.

Kelly, J. F. 1983. An empirical methodology for writing User-Friendly Natural Language computer applications. *Proceedings of the CHI '83*.

Linell, P., Gustavsson, L. & Juvonen, P. 1988. Interactional Dominance in Dyadic Communication. A Presentation of the Initiative-Response Analysis. *Linguistics*, 26(3).

Litman, D.J. 1985. Plan Recognition and Discourse Analysis: An Integrated approach for Understanding Dialogues. Ph.D. thesis. TR 170, The University of Rochester.

Merkel, M. 1989. Temporal Structure in Discourse. *Proceedings of the Seventh International Conference of Nordic and General Linguistics*. Faroe Islands.

Reichman, R. 1985. *Getting Computers to Talk Like You and Me*. Cambridge, Mass., MIT Press

Reilly, R. 1987. Ill-formedness and miscommunication in person-machine dialogue. *Information and software technology*, 29(2):69–74.

Sidner, C. 1983. Focusing in the Comprehension of Definite Anaphora. M. Brady and R. C. Berwick [Eds.] *Computational Aspects of Discourse*:267–330. Cambridge, Mass., MIT Press.

Webber, B. L. 1988. Tense as Discourse Anaphor. *Computational Linguistics*, 14(2):61–73.

Natural Language Processing Laboratory
Department of Computer and Information Science
Linköping University,
S-581 83 LINKÖPING, SWEDEN
Internet: **ARJ@LIUIDA.SE**