Coding Schemes for Studies of Natural Language Dialogue*

Lars Ahrenberg Nils Dahlbäck Arne Jönsson†
Department of Computer and Information Science
Linköping University, S-581 83 LINKÖPING, SWEDEN
lah@ida.liu.se, nilda@ida.liu.se, arnjo@ida.liu.se

Abstract

This paper presents a coding scheme which has been used for the analysis of NLI-dialogues collected by means of Wizard-of-Oz techniques. The scheme covers both dialogue structure and focus structure. Dialogue structure is coded in terms of segments consisting of moves belonging to general illocutionary types, such as initiative and response, and being further specified as to their topical domains. Focus structure is coded in terms of a number of focal parameters, which may differ from one type of dialogue to another. Relations between values of focal parameters on neighboring discourse segments are determined by a simple model.

The coding scheme is flexible and has a high degree of inter rater reliability. It enables comparisons between different types of dialogues and testing of assumed models for dialogue management. A dialogue manager has been implemented that can be customized on the basis of the analysis of a representative set of dialogues from a given application using the coding scheme.

Introduction

The present paper has a number of theoretical starting points. Before turning to the description of our dialogue model, and the empirical analysis of our corpora, we want to present these underlying assumptions.

Our work is based on a sub-language approach (Grishman & Kittredge 1986), in two respects. The first assumption is that the language used when interacting with a computer will differ from the language used between people, and that therefore the empirical base for computational models of discourse should basically rely on so-called Wizard of Oz-data. But we furthermore believe that different dialogue types within these two categories will differ. Consequently one should be careful when wishing to use empirical results on dialogue structure from one kind of domain to another, for instance from advisory dialogues to information retrieval dialogues.

Another assumption is that for the development of user-friendly software the analysis model should be the minimal model that can accommodate the kind of dialogues occurring in these specific situations, and that computational tractability should be preferred to generality with unknown properties, as well as to general models with known high complexity.

The assumptions above are presumably rather uncontroversial. What perhaps is less so is our belief that computational theories of discourse should only be considered as theories of computers' processing of language and not general theories of discourse for all kinds of agents and situations. The reason for this is that the cognitive architecture of present day computers and people are different. Hence, procedural computational accounts of the process of discourse (or any other cognitive phenomenon, for that mattere) using concepts from present day computer technology cannot be seen as a psychological account. "Two programs can be thought of as strongly equivalent or as different realizations of the same algorithm or the same cognitive process if they can be represented by the same program in some theoretically specified virtual machine" (Pylyshyn 1984, p. 91). A consequence of this is that "any notion of equivalence stronger than weak equivalence[1] must presuppose an underlying functional architecture, or at least some aspects of such an architecture." (ibid., p 92) "Typical, commercial computers, however, are likely to have a far different functional architecture from that of the brain; hence, we would

*This work results from a project on Dynamic Natural-Language Understanding supported by The Swedish Council of Research in the Humanities and Social Sciences (HSFR) and The Swedish National Board for Industrial and Technical Development (NUTEK) in the joint Research Program for Language Technology.

†On leave until July 1995 to Computer Science Department, Monash University, Clayton, VICTORIA 3168, AUSTRALIA, email: arnjo@bruce.cs.monash.edu.au.

[1] i.e. realizing the same input-output function.

expect that, in constructing a computational model, the mental architecture must first be emulated (that is, itself modeled) before the mental algorithm can be implemented" (ibid., p 96).

We believe that the conclusion to be drawn from these arguments is that most, if not all, present day computational theories of discourse are about computers' processing of language, and nothing else. Or, to phrase the same point somewhat differently, since there are no attempts to first emulate a theory of the human cognitive apparatus, it is difficult to regard them as theories about anything but computers. Another conclusion is that since computational theories of discourse are about computers' processing of language, the language samples used for providing the empirical ground of the computational theories should come from relevant application domains for such software technology.

The theoretical position above provides another argument for our view that computational work on dialogue should use Wizard of Oz experiments and other similar corpora as its empirical base. Furthermore that the computational models should be geared towards minimal models for the specific domains, and that questions of computational tractability are important issues in the development of these models.

The arguments behind our positions are spelled out in more detail in (Dahlbäck 1991b; 1991a; Dahlbäck & Jönsson 1992; Dahlbäck, Jönsson, & Ahrenberg 1993; Jönsson 1993a; Dahlbäck forthcoming) and will not be pursued further in this paper. Here we will instead focus on our dialogue model and its coding scheme for dialogue analysis, and present results and observations from its use in empirical studies of man-machine dialogues, as well as describe the development of an implemented system based on this work. Before doing so, we want to point out that while the assumptions presented above motivate why our work has been conducted along the lines described here, we believe that the empirical results obtained are of interest also if these assumptions are not fully accepted.

The Need for Wizard of Oz Studies

It is important that the language samples used for providing the empirical ground come from relevant settings and domains. In other words, the development of NLI-software should be based on an analysis of the language and interaction style used when communicating with NLIs.

This is what motivates data collection by means of Wizard-of-Oz techniques (Dahlbäck, Jönsson, & Ahrenberg 1993; Fraser & Gilbert 1991), i.e. studies where subjects are told that they are interacting with a computer system through a natural language interface, though in fact they are not. The method is not as simple to use as might seem the case on first appearances, but with the use of a well-designed simulation environment and a carefully designed study, it is possible to achieve a close approximation of a computer system's communicative ability.

Of course you cannot expect to gather all the data you need for the design of a given application system by means of Wizard-of-Oz studies, e.g. as regards vocabulary and syntactic constructions related to the domain. But for finding out what the application-specific linguistic characteristics are, or for gathering data as a basis for theories of the specific genre of human-computer interaction in natural language, the Wizard-of-Oz-technique seems to us to be the best available alternative.

Experimental Data

We have run Wizard of Oz-experiments both as part of research on the general characteristics of NLI-dialogues, and as part of the development of a NLI for a specific application. To circumvent the risk of drawing general conclusions that in fact are only a reflection of the specific experimental setting used, we have used six different background systems. We have varied not only the content domain, but also the 'intelligence' of the systems, and the number and types of tasks possible to perform by the user. Our corpus can be sub-divided into two corpora intensively analyzed and used in our empirical studies, called corpus 1 and 2 below.

Corpus 1 contains dialogues with five real or simulated background systems. PUB is a library DB in use at our department. C-line is a simulated DB containing information about the computer science curriculum at Linköping University. In the HiFi-system the user can order HiFi-equipment after having queried a (simulated) DB containing information about the available equipment. The Travel system simulates an automated travel agency offering charter holidays to Greek islands. These systems differs from the first two in two respects; the system is more 'cognitively' advanced, and there are more actions that can be performed by the user, i.e. not only asking for information but also order something. The Wine system is a simulated advisory system, capable of suggesting suitable wines for different dishes, if necessary within a specific price range.

A general overview of this corpus is presented in (Jönsson & Dahlbäck 1988). Dahlbäck (1991b) and Dahlbäck & Jönsson (1992) report on dialogue structure while Dahlbäck (1992) presents an analysis of the distribution and function of pronouns. Dahlbäck (1991b) presents the most detailed analysis of both the dialogue structure and the pronoun patterns and also analyses the use of definite descriptions.

Corpus 2 was collected using a refined Wizard of Oz-simulation environment which also made a limited use of graphics. This corpus consists of two different background systems. Cars, which is an INGRES database of used car models and a considerably revised and enlarged version of the travel system used in corpus 1. In this corpus half of the subjects could only obtain information from the system, whereas the other half of them also could order the trip as was the case in corpus 1. Some results from the analysis of this corpus will be presented below, as well as the dialogue system developed from it. Further results are presented in Jönsson (1993a, 1993b). All systems, with the possible exception of the advisory system in corpus 1, belong to the class of systems Hayes & Reddy (1983) call simple service systems.

Apart from these dialogues, we have in pilot studies collected an additional set of dialogues of approximately the same size, including pilot studies of domains that we found less appropriate for NLI:s or for Wizard of Oz experiments. The latter are described in Dahlbäck, Jönsson, & Ahrenberg (1993).

Analysis of Corpus 1

The dialogue structure of corpus 1 is analysed using a simple dialogue tree model called LINDA (For Linköping DiAlogue, see Dahlbäck (1991b, 1991a) for a detailed description). We use only two basic types of moves, initiatives (I) and responses (R). The definition of the categories is solely based on local information. If the move is seen as introducing a goal it is scored as an initiative, if it is a goal-satisfying move, it is scored as a response. One important reason for this is that the categories are domain independent. We can therefore compare dialogues from different domains. Another advantage is that the categories are (fairly) simple to define and identify, making it possible to code the dialogues with high inter-rater reliability, where two independent coders agreed on the coding for 97% of the moves. Another indication of the ease of use of the coding system is that we successfully have used it in student projects in the undergraduate computer science curriculum in Linköping.

Discourse management moves such as *Welcome to WingHolidays. What can we do for you?, Can I help you with anything more? Bye* etc. are all scored as initiatives. We subcategorize them as DO (discourse opening), DC (discourse continuation), and DE (discourse ending), to make it possible to exclude them

Statistics on corpus 1		
System	LINDA-model fit	Adjacency-pairs
PUB	100%	75%
C-line	98%	96%
HiFi	99%	98%
Travel	99%	88%
Wines	92%	78%

Table 1: Results from analysis of corpus 1

from some of the analysis presented below. (Responses to these kinds of initiatives are optional in the model).

Since we only used local information when ascribing a category to a move, we can get a measure of the structural complexity of the dialogues by analyzing them using LINDA. The model only accepts units consisting of an initiative followed by a response or embedding of such units in higher IR-units, e.g. (I R), or successive and recursive embedding such as (I (I R) R), (I (I R) (I R) R), or (I (I (I R) R) R) etc. All moves must belong to some discourse segment, and no segments with the structure (I I R) or (I R R) are allowed. The model fit for the dialogues is presented in Table 1.

We thus find an almost 100% fit to all dialogues but the advisory dialogues, and even this worst case shows a model fit of more than 90%. Furthermore, the use of recursive embedding is limited, as seen in the high percentage of segments consisting of simple adjacency pairs.

The 'low' figure for the PUB dialogue occurs because of a large number of clarification requests from the 'system', asking the user if he wants to see all the titles found in a search, even though they will not fit into one screen page. Apart from this, we once again find the advisory system's pattern to deviate somewhat from the rest.

These results do not mean that the dialogues consists of a sequence of isolated questions and answers. There is frequent use of anaphoric expressions. In fact 49% of the initiatives contain some kind of anaphoric expression (Dahlbäck & Jönsson 1989). What the figures show is rather that in spite of being clear cases of connected discourse, these dialogues have a much simpler structural complexity than most other genres (cf. Guindon, 1988). It thus seems as if most man-machine dialogues in natural language, even when no restrictions on the users' way of expressing themselves, lack most of the complexity found in other types of discourse. Our corpus is admittedly of a limited size, but it covers some of the most typical possible applications for NLI technology, and, apart from the advisory type of system, is not tied to one particular topic domain. Taken together, this gives us confidence in believing that the results have some generalizability

It is also important to point out that the LINDA-structure can be used to direct the search of antecedents to pronouns and other anaphors. This is not as trivial a result as one might believe on first thoughts, given the rather simple structure of the dialogues. But the dialogue structure is not only used to direct the search for the pronouns' or definite NP's antecedents. It is also used to help identifying those cases where the anaphors lack an explicit antecedent. In corpus 1 as much as one third of the user's personal third person pronouns lack an explicit antecedent. For further description of the analysis of anaphors in these dialogues and the possible use of an IR-structure to resolve them, see Dahlbäck (1991b, 1992). A comparison between the patterns of pronoun usage in these dialogues and the ones in technical documentation described in Lappinen & Leass (1994) illustrates the advantage of a sub-language approach to discourse phenomena.

Analysis of Corpus 2

As is well known, computational dialogue models for natural language interfaces need to account for two concurrent tasks: *dialogue structure* and *focus structure* (cf. Grosz & Sidner 1986). Dialogue structure involves managing relationships between segments in the dialogues. Focus structure is concerned with the recording and structuring of entities mentioned in the discourse to allow a user to refer to them in the course of the interaction.

In the analysis of corpus 2 we used an extended version of the LINDA-model which accounts for both tasks. For the dialogue structure, an initiative-response (IR) structure similar to the one used for corpus 1 is assumed. A dialogue is divided into three main classes on the basis of structural complexity. There is one class corresponding to the size of a dialogue (D), another class corresponding to the size of a discourse segment and a third class corresponding to the size of a single speech act, or dialogue move.

To specify the functional role of a move we use the parameters Type and Topic. **Type** corresponds to the illocutionary type of the move. In simple service systems two sub-goals can be identified: 1) "specify a parameter to the system" and 2) "obtain the specification of a parameter" (Hayes & Reddy 1983, p. 266). Initiatives are categorized accordingly as being of two different types 1) update, U, where users provide information to the system and 2) question, Q, where users obtain information from the system. Responses are categorized as answer, A, for database answers from the system or answers to clarification requests. Other Type categories are Greeting, Farewell and Discourse Continuation (DC).

Topic describes which knowledge source to consult. In information retrieval applications three different topics are used: the database for solving a task (T), system-related, i.e. acquiring information about the database, (S) or, finally, the ongoing dialogue (D).

The attributes Objects and Properties, account for the focal information structure of a move (query). Users specify an object, or a set of objects, and ask for some concept information, e.g. the value of a property of that object or set of objects. **Objects** denote a set of primary referents, and **Properties** a complex predicate ascribed to this set (Ahrenberg 1987). These are focal parameters in the sense that they can be in focus over a sequence of IR-units.

Coding the focus structure depends on how the information in the user initiative and the answer provided from the database specify the values to the focal parameters Objects and Properties. A move can fully specify both Objects and Properties. However, many utterances provide only a partial specification to the focal parameters, which means that contextual information is needed to make them fully specified. Our hypothesis is that the values of Objects and Properties for a previous segment provide an initial local context for the next segment. These values are changed (or made more specific, as the case may be) with values from the initiative and the response from the background system, when provided.

The coding scheme needs to account for three types of specifications, termed FS (FullySpecified), LC (LocalContext) and GC (GlobalContext). FS denotes utterances which are fully specified or can not be further specified using context information. Thus, not only correct utterances, are considered FS but also vague or erroneous utterances. LC is used for user utterances which can be specified as regards Properties or Objects from the local context. The information should be found in local focus, i.e. the current segment. This means that focal information from the previous IR-node provides the correct information. Finally, GC (for Global Context) denote utterances that cannot be specified from local focus.

To illustrate the coding scheme consider the utterance *How rust prone is Volvo 244?*. This is coded $Q_T FS$[2] as it is an initiative of type Question querying information from the background system, i.e. Topic T and both Objects and Properties are specified. Let

[2]For brevity, when presenting the dialogue grammar, the Topic of a move is indicated as a subscript to the Type.

Statistics on focusing heuristics			
	CARS	TRAVEL1	TRAVEL2
FS	52%	44%	59%
LC	43%	50%	39%
GC	5%	6%	2%

Table 2: User-initiatives classified according to context-dependence.

Statistics on dialogue structure			
	CARS	TRAVEL1	TRAVEL2
No of rules	15	12	14
Q_T/A_T	60%	83%	70%
Q_D/A_D	12%	2%	2%
Q_S/A_S	9%	2%	2%
Q_T/A_D	7%	2%	3%
Q_T/A_S	5%	6%	4%
Others	7%	5%	19%

Table 3: Types of dialogue segments and their relative frequency in three different applications. (Others denote for instance Ordering rules, Greetings, and Farewells)

us assume that the next user utterance after the system's answer is *Mercedes 200*. This is coded $Q_T LC$ as information on Properties, i.e. rust prone, is needed but found in local focus. Another example is *which 10 car models are most spacious*, where the provided aspect of the Property, i.e. spacious, is ambiguous. However, this is also coded $Q_T FS$ as context information would not further disambiguate the utterance. Jönsson (1991) gives a detailed description of the use of the coding scheme in cases of underspecified and ambiguous user utterances, as well as descriptions of the coding of clarification sub-dialogues, questions about the system's properties etc. The same source provides detailed information on the management of topic structure in the different domains analyzed.

Table 2 presents the results on focus structure from applying the coding scheme to the CARS and TRAVEL applications. It shows that the large majority of user inputs in these dialogues can be handled by a fairly simple context model.

The dialogues were analyzed using basic dialogue grammar segment rules. A summary of the statistics on dialogue structure as emerged from applying the coding scheme to the corpus is presented in Table 3. It shows that the complexity of the resulting grammars from the customizations of all systems are quite simple. The most common segment consists of a task-related initiative followed by an answer from the database, Q_T/A_T[3], sometimes with an embedded clarification sequence, Q_D/A_D.

The IR-sequences found in the analysis of dialogue structure have a natural explanation if we consider the purpose of the system. Although the segments do not represent information on user's goals, it turns out that the user utterances can be classified into a few classes in goal-related terms. The segments can basically be divided into four classes, taking the user's initiative as the basis for the classification: (i) "proper" information requests that are satisfied by an answer with information from the database, (ii) successful queries about system properties, (iii) successful moves satisfying subordinate goals, such as greetings or discourse continuations; (iv) initiatives that transgress the system's knowledge and which require robust error handling.

[3]Labels of IR-segments have the form of a pair of move labels separated by a slash (/).

Implementation

Based on the results from the analysis of Corpus 1, a natural language interface, LINLIN, was designed (Ahrenberg, Jönsson, & Dahlbäck 1990) allowing customization to the sublanguage utilized in various applications. The kernel of the interface is the Dialogue Manager (Jönsson 1991) which controls the interaction and holds information needed by other modules in the interface, including the Dialogue Manager itself. The information is modeled in dialogue objects. The dialogue objects represent the constituents of the dialogue and involve parameters for focus and dialogue structure as discussed above. The managing of local focus was implemented by a few basic heuristics for copying information from one segment to the next and updating the focal parameters with information from the database.

Based on Corpus 2, dialogue objects have been customized to meet the demands of these systems: CARS and TRAVEL with and without ordering. This requires customizing the focal parameters to reflect the organization of the background system. It also involves modifying the basic heuristic principles on copying and updating from the background system slightly to account for the different ways in which users access the background system (Jönsson 1993b). However, the general principles are still valid and the modifications are more a reflection of the demands from the background system.

The customized dialogue objects for the CARS system has also been integrated with an INGRES database manager and interpreting modules using a grammar and lexicon covering a subset of the utterances found in the corpus (Ahrenberg, Jönsson, & Thurée 1993).

Final Comments

Space limitations prohibits a detailed discussion of the generalizability of the results obtained here. It should be pointed out that the extention of the model from a two-move (IR) to a three-move (IRC) model is possible without invalidating the basic approach. Such a model seems necessary in at least some cases of spoken man-computer dialogues (cf. Bilange 1991; Novick & Sutton 1994). But a crucial assumption in the model is that each move can be assigned one single category. It has been argued that this is not true for all kinds human dialogues, and it is conceivable that there are kinds of human-computer dialogues where this applies as well, and for which other dialogue models are required. On the other hand, the model presented here can probably be used when analyzing human dialogues of the same kind, for example information retrieval and other simple service dialogues. But more theoretical and empirical work is required to make it possible to clarify the application domains for our and other empirical dialogue models.

Summary

We described a dialogue model and a coding scheme for human-computer dialogues. It was shown to have high inter-rater reliability, and the assigned dialogue structure could be used in the management of anaphors. We also described the coding of the topic structure within the same context. The model has been used in the implementation of a customizable NLI.

Acknowledgments

This work has been carried out with the members of the Natural Language Processing Laboratory at Linköping University, Sweden, and we are especially indebted to Åke Thurée.

References

Ahrenberg, L.; Jönsson, A.; and Dahlbäck, N. 1990. Discourse representation and discourse management for natural language interfaces. In *Proceedings of the Second Nordic Conference on Text Comprehension in Man and Machine, Täby.*

Ahrenberg, L.; Jönsson, A.; and Thurée, Å. 1993. Customizing interaction for natural language interfaces. In *Workshop on Pragmatics in Dialogue, The XIV:th Scandinavian Conference of Linguistics and the VIII:th Conference of Nordic and General Linguistics, Göteborg, Sweden, August 16-17, 1993.*

Ahrenberg, L. 1987. *Interrogative Structures of Swedish. Aspects of the Relation between grammar and speech acts.* Ph.D. Dissertation, Uppsala University.

Bilange, E. 1991. A task independent oral dialogue model. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin.*

Dahlbäck, N., and Jönsson, A. 1989. Empirical studies of discourse representations for natural language interfaces. In *Proceedings from the Fourth Conference of the European Chapter of the assosciation for Computational Linguistics, Manchester.*

Dahlbäck, N., and Jönsson, A. 1992. An empirically based computationally tractable dialogue model. In *Proceedings of the Fourteenth Annual Meeting of The Cognitive Science Society, Bloomington, Indiana.*

Dahlbäck, N.; Jönsson, A.; and Ahrenberg, L. 1993. Wizard of oz studies – why and how. *Knowledge-Based Systems* 6(4):258–266.

Dahlbäck, N. 1991a. Empirical analysis of a discourse model for natural language interfaces. In *Proceedings of the Thirteenth Annual Meeting of The Cognitive Science Society, Chicago, Illinois*, 1–6.

Dahlbäck, N. 1991b. *Representations of Discourse, Cognitive and Computational Aspects.* Ph.D. Dissertation, Linköping University.

Dahlbäck, N. 1992. Pronoun usage in NLI-dialogues. A wizard of Oz-study. In *Proceedings of the Second Third Nordic Conference on Text Comprehension in Man and machine, Linköping, Sweden.*

Dahlbäck, N. forthcoming. Kinds of agents and types of dialogues. Manuscript in preparation, Department of Computer and Information Science, Linköping University.

Fraser, N., and Gilbert, N. S. 1991. Simulating speech systems. *Computer Speech and Language* 5:81–99.

Grishman, R., and Kittredge, R. I. 1986. *Analysing language in restricted domains.* Lawrence Erlbaum.

Grosz, B. J., and Sidner, C. L. 1986. Attention, intention and the structure of discourse. *Computational Linguistics* 12(3):175–204.

Guindon, R. 1988. A multidisciplinary perspective on dialogue structure in user-advisory dialogues. In Guindon, R., ed., *Cognitive Science and Its Applications For Human-Computer Interaction.* Lawrence Erlbaum.

Hayes, P. J., and Reddy, D. R. 1983. Steps toward graceful interaction in spoken and written man-machine communication. *International Journal of Man-Machine Studies* 19:231–284.

Jönsson, A., and Dahlbäck, N. 1988. Talking to a computer is not like talking to your best friend. In *Proceedings of the First Scandinavian Conference on Artificial Interlligence, Tromsø.*

Jönsson, A. 1991. A dialogue manager using initiative-response units and distributed control. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics, Berlin.*

Jönsson, A. 1993a. *Dialogue Management for Natural Language Interfaces – An Empirical Approach.* Ph.D. Dissertation, Linköping University.

Jönsson, A. 1993b. A method for development of dialogue managers for natural language interfaces. In *Proceedings of the Eleventh National Conference of Artificial Intelligence*, 190–195.

Lappinen, S., and Leass, H. J. 1994. An algorithm for pronominal anaphora resolution. *Computational Linguistics* 20(4):535–562.

Novick, D. G., and Sutton, S. 1994. An empirical model of acknowledgement for spoken-language systems. In *Proceedings of the 32nd Conference of the Association for Computational Linguistics, New Mexico.*

Pylyshyn, Z. 1984. *Computation and Cognition.* The MIT Press.