4. Natural-language facilities in information systems: asset or liability?

by Prof. Dr. G.A.M. Kempen

4.1 Some questions

When people communicate with one another there is a continuous process of feedback whereby they can verify whether their messages are being understood and correctly interpreted. This convergence mechanism appears to be essential to being 'on speaking terms'. How can feedback be incorporated into an information system? How can structures be built into the system to provide adequate explanation of its reasoning whenever it is required?

Generally speaking an information system will have various users, each with their own specific cognition. For maximum user affinity a system should be adaptable to different cognitive levels. To what extent can users attune systems to their own level? Could a system detect and adapt to its user's cognitive level automatically?

These questions were formulated in the chapters of Part II of the present volume:

Does user-friendly accessibility require the user to have a conceptual model of the system? Or, conversely, should the system be aware of the inaccurate or incomplete conception that the user has of it?

Is it a practical proposition to set up visual display terminals with comprehensive facilities for information retrieval or extended calculations, for example, without at the same time providing a skilled operator?

Can a user always be expected to learn a control language? If so, he will have to perform the translation from intention to processing (turning his 'fuzzy' problem into a sharply defined and processable question) himself. Should this be a social or educational objective?

Or should the system do it for him, so that it will help him even with his fuzzy problem or sloppy thinking?

4.2 Introduction

Computer programs for processing natural language are written with two objects in mind. In the first place, language processing can be an end in itself: one only has to think of software for the automatic hyphenation of words, spelling correction or (much more difficult) summarising or writing reports. Second, it can be a means to some other end: for example, a database that can be accessed with questions and commands formulated in ordinary language. There are naturally hybrids too, such as a spelling corrector that automatically copes with typing errors in questions addressed to a data bank.

It is commonly asserted that language-processing software in this second category will enhance the user-friendliness of information systems. Elsewhere in this book there is more than one attempt to seek advantages in 'language technology'. My intention here is to examine how far such hopes are justified. I shall begin with brief résumé of what language processing software can and cannot do, now and in the near future (Section 4.3), and will then turn to the user of information systems (Section 4.4). Will he or she always want easy access to the complete range of means of communication offered by natural language? Or are there circumstances when the user will find a highly formalised language more efficient and comfortable to use, despite its inherent limitations? In Section 4.5, finally, I shall briefly discuss the economic perspectives: user-friendliness does, after all, imply a certain measure of 'budget-friendliness'. I shall be using the term 'dialogue system' to cover all software systems designed to enable linguistic interaction with information systems. (Other terms are 'natural language interface' and 'language front-end'.)

4.3 Dialogue systems

In June 1981, a conference was held in Stanford on 'Applied Computational Linguistics in Perspective', under the auspices of the Association for Computational Linguistics. There, a company of American experts with a great deal of experience in the field of applied artificial intelligence and computational linguistics tried to arrive at a considered and realistic scenario for applications of their subject. One of the discussion groups was particularly concerned with dialogue systems, and it is the report produced by this group (Hendrix, 1982) that provides the basis for this section of my article, since it discusses all the aspects relevant to user affinity. In assessing the comparative importance of these factors I have not simply accepted the views of the group, both because the builders of dialogue systems may be expected to be well endowed with optimism and because the users themselves were not involved in their evaluation.

The report distinguishes three levels of linguistic skill at which dialogue systems may operate. At the lowest level (level 1) we find all the systems currently suitable for practical use, and the few systems commercially available as software packages. Level 2 systems so far exist solely as laboratory prototypes, and work on level 3 systems has only just begun. The report refers exclusively to systems using English. The technology for handling other languages is significantly further behind, with the possible exceptions of German, French, Swedish and Japanese. There are no Dutch-language dialogue systems either in Holland or Belgium, though a project group at the University of Nijmegen is currently working on the construction of such a system with financial support from the government.

The following overall characterisation of the three levels may serve as a point of departure. Level I systems offer a basic package of facilities essential to any dialogue system, i.e. modules for:

— the analysis and interpretation of recurrent words, word groups and sentences, especially interrogatives;
— generating sentential answers;
— extending and adapting the dictionary of words and idioms on which questioner and answer generator can draw;
— constructing programs in the formal language (command language, query language, control language) used by the information system.

However, the underlying concept of level I systems leads to all sorts of constraints which cannot easily be eliminated, and dialogue systems on levels 2 and 3 therefore had to be constructed according to new principles. Level 2 has powerful methods of solving referential descriptions, level 3 for reasoning about mental states such as goals, plans, and beliefs.

Level I systems provide the following facilities. They can:

1. answer factual questions relating to data in common types of data base system;
2. if desired, coordinate data files (e.g. 'What is Smith's location' is translated into 'What is the location of the department of Smith?');
3. resolve simple cases of pronominalisation (i.e. find out what pronouns refer back to; difficult cases are still beyond the systems' reach);
4. handle simple cases of ellipsis (elliptical questions are incomplete and refer back to earlier sentences in the dialogue: 'Where is John?' ... 'And Peter?');

5. give cooperative answers to 'null-questions' (e.g. to the question 'How many copies of book X do you have in stock?' the system might answer not with 'None' but with 'That book is not known to me' or 'That book is sold out');
6. enrich their linguistic knowledge through interaction with the user (e.g. Define ' JD' as 'Jefferson Davis Jones'. Let 'Q1 Smith salary' = 'What is the salary of employee Smith'. 'Q1 JD AGE?');
7. paraphrase input sentences (questions, commands etc.) so that the user has some check on whether they have been correctly interpreted;
8. correct spelling errors.

Some level 1 systems also offer limited capacity for extension, e.g.:

9. updating data according to commands in natural language (e.g. 'Change Bob Day's location to Building 7.');
10. reacting meaningfully to ungrammatical inputs;
11. answering 'meta-questions' such as:
    — 'What information is in the database?'
    — ‘What are the permissible values for employee job titles?'
    — ‘How up-to-date are the sales data?'
    — 'Can you handle relative clauses?'

The design of level 2 dialogue systems incorporates a system of new principles, the most important ones being the following:

— the linguistic modules can to a high degree be guided by 'world knowledge' (i.e. a representation of the domain of contents to which the information system relates);
— the discourse context (i.e. preceding questions, commands, answers, etc., and their meanings) is retained and exploited to the full;
— input sentences are not translated straight into the formal language of the information system but into a logical interlingua, often a form of first-order predicate calculus (all sorts of linguistic phenomena can then be dealt with in a more general and linguistically more satisfactory manner, so that many ad hoc solutions can be abandoned; at the same time, the step of translating from the logical interlingua to formal command language is relatively simple).

These new basic principles enable dialogue systems to perform much better in interpreting referential descriptions, i.e. various types of noun phrases referring to objects, states and events in the domain with which the dialogue is concerned. Dialogues systems at level 2 are considerably better equipped to determine the correct reference of noun phrases than their predecessors, as the following examples may show. Suppose a car repair information system is asked: 'If I want to top up the water, where do I find the filler cap?' The decision that it is the cap on the radiator and not the petrol tank is based not on linguistic rules but on a knowledge of cars. Nor can the discourse context be disregarded when it comes to resolving referential descriptions. Take the stock control system of a bookshop when it is commanded to: 'Give titles in stock and prices'. If it is functioning at level 1, the system will produce a complete list of all the books in stock. A level 2 system, by contrast, takes accounts of the content of what has gone before. For example, on being asked 'Do you know any of Graham Greene's books?', the listing produced will be confined to works by that author, and the noun phrase 'the same' in ‘And now the same for Ernest Hemingway' will also be interpreted correctly. (This is also an example of pronominalisation which would stump a level I system: 'the same' refers not to an object mentioned earlier but to an earlier sentence in the dialogue.)

A dialogue system at level 3 will be able to draw on an important extra source of information: knowledge of goal-directed behaviour. Starting from needs and goals that it attributes to the user, to itself and possibly to third parties, it will be able to evolve plans and proposals which will as far as possible accord with the interests of those parties. The answers of the dialogue system are based on such plans. Wilensky (1982) is one of the first to have started on the construction of such a system. He is working on a 'consultant' which users of the UNIX operating system will be able to call on when they have problems. The chief purpose of the consultant is to help inexperienced users on their way. Here are some question-and-answer examples to illustrate this intention.

— How do I delete a file?
  Typing 'rm filename' will remove the file with filename from your directory.
— How do I change a file?
  Typing 'emacs filename' will get you into the emacs editor.
— What is the best way of making a file?
  The best way to create a file is to use the editor, i.e. 'emacs filename'.
— A directory?
  Typing 'mkdir directoryname' will create a directory with name directoryname in your current directory.

The two central modules in the consultant are responsible for 'goal analysis' and 'plan formation', respectively. When the meaning of the input sentence has been established, the first module tries to recover the user's intention, after which the second module works out a way of achieving this goal. This plan is finally presented to the user in English. For both modules to function satisfactorily they must be able to cope with multiple goals, which may sometimes be conflicting, since the goal foregrounded by a question may contradict objectives which play a part in the background. For example, if the user says 'I'm trying to get some more disk space' the consultant must not come up with 'Type “rm *” ', since that advice would conflict with one of the user's presumable background goals by leading him to erase all the data files he has created.

We have now looked at a rising curve of linguistic skills available to dialogue systems now or in the near future. This does not mean that within a few years we shall be able to communicate with computers just as we do with people. Far from it. The concepts underlying the design of dialogue systems have in recent years been improved considerably, and this 'vertical' trend of deepening knowledge will continue in the future. But this does not automatically lead to a broadening of knowledge and skills within particular levels. Modern dialogue systems always operate on a very narrow knowledge base, and extension sideways is no trivial matter. It is labour-intensive because large volumes of knowledge have to be inventoried, analysed and encoded without error. Moreover, it makes heavy demands on external memory, and may lead to forbiddingly long response times.

4.4 Are dialogue systems user-friendly?

The final sentences of the previous section reveal an important disadvantage of dialogue systems: they prevent the user from seeing their limitations. It does not take long for the user to come up against words, constructions and types of question which the system turns out to understand wrongly or not at all. A dialogue system only becomes user-friendly, therefore, when this 'wall of misunderstanding' is lower than the 'mountain of work' which the user has to get through in order to have an understanding of the information system's own formal language. This disadvantage is aggravated by the fact that users tend to overestimate the system when it has run for some time without trouble. Evidently they apply all sorts of rules like 'If he knows A, he will probably know B too', and then go on to apply them to machines, often erroneously. Verbosity is another disadvantage. Questions and commands in natural language are normally much longer than in a formal language specially attuned to the domain of application. Think of the ultra-short commands in text editors. Dialogue systems are pointless in this area unless they function as consultants providing solutions to difficult cases, or when rare commands are to be used, e.g. 'How do I change the margins?' (Hendrix, 1982).

The great advantage of dialogue systems is that they can bring about a considerable improvement in the accessibility of information systems. For one thing, the circle of potential users wanting access to the information system may become much wider due to the lowering of the language barrier. For another, accessing the system will take up far less time—a particularly attractive point to novices and occasional users unfamiliar with the formal command language, but of less importance to regular and experienced users who may continue to prefer the formal language. The attraction of a dialogue system also depends on the complexity of operating the information system. Another possibility is menu selection, where the user chooses what he wants to do by using a keyboard (of traditional or specialised design), a light pen, a touch-sensitive screen, or a 'mouse'. This alternative is probably more attractive if there are no more than a few dozen different instructions which can be given to the information system, and the user can get to where s/he wants to be by a limited number of choices (menus and/or keystrokes). Dialogue systems capable of fulfilling a consultant role are likely to be more widely applicable, because they can offer help even to experienced users when they come to areas with which they are less familiar.

4.5 Are dialogue systems budget-friendly?

Market research has recently been carried out in Germany to establish the possible demand for dialogue systems (Morik, 1983), and there seems to be a great deal of interest in this new technology. Sums of DM 200,000 are mentioned as the maximum investments in purchases or development work. In the United States, a level 1 dialogue system bas recently been marketed for around $ 70,000, including installation charge and adaptation to the databank system used by the purchaser. In the longer term, industry in Germany sees the benefits overtaking the costs, and the market survey shows that two types of use hold particularly good prospects. The first of these concerns users without a technical grounding—managers, etc.—who need to use an information system for their work. A dialogue system enables them to converse with the information system direct, without the need for intervention by the computer department. For example, a building contractor can carry out and present detailed price calculations even while he is talking to his customer. The second type of use is consulting sources of information, such as literature references on a specific topic, or departure times on an airline's route system, or sources of advice (e.g. to decide which of a particular range of products to buy). I recently saw the following example in the United States. If you want references to literature, you are not often allowed to sit down at the terminal yourself, but have to be helped by a special person who is familiar with the formal language and has the relevant keywords in his/her head. A rudimentary dialogue system—even below level 1—would be able to take over this task. Systems of this kind are in fact operational in some large American libraries.

The investments required for the acquisition or development of a dialogue system are undoubtedly extremely high, even if one is willing to make do with level 1. Even so, the experience gained abroad and described above still justifies the hope that user affinity and ' budget-friendliness' in dialogue systems will eventually be reconciled.

References

Hendrix, G. (1982). Natural-language interfaces. *American Journal of Computational Linguistics, 8,* 56-61.

Morik, K. (1983). *Marktstudie zu natürlichsprachlichen Zugangssysteme*. Bericht ANS-14, Forschungsstelle für Informationswissenschaft und Künstliche Intelligenz., Universität Hamburg.

Wilensky, R. (1982). *Talking to UNIX in English: an overview of an on-line consultant.* Report No. 82/104, Computer Science Division, University of California, Berkeley.