16 Natural language interfaces

David Chan

Introduction

A serious obstacle to the use of database systems, decision-support systems and expert systems is the inability of end-users to communicate to the systems exactly what they want. Instead they have to learn artificial languages to express their requirements. Not every user who wants to access a system is able and willing to learn a special language in order to communicate. Users would prefer to use a natural language like English which they are familiar with. Natural language interfaces (NLI) provide an answer to the man-machine interface problems. This paper is about NLI and their capabilities. The present and future trends in NLI development are also examined.

Intermediate solutions

The retrieval of stored information in the computer has always been a problem for users. There was a time when end-users had to approach programmers in the data processing department to write programs in COBOL, PASCAL, and so on, to retrieve information from their databases. This group of end-users was not able to retrieve all the information it wanted for itself. Then came user-friendly systems which provided facilities for this group to have easy access to the computer as well as to the information in the databases. These user-friendly systems do not require end-users to have any knowledge of programming constructs or the structure of the databases. The development of menu systems and query languages were attempts towards making application systems friendlier and therefore more accessible.

Menu system

Menu systems have the advantage that they require very little training to use. In a typical menu system, the user is given a list of options on the screen and the user just needs to indicate his choice by entering a number of characters associated with the chosen option.

Menus are not reliable or effective for the following reasons:

(a) the user may not understand the options;
(b) the number of options are not adequate and none seems to be appropriate;

(c) the user may want to be more specific in his requirements;
(d) the user may be misled into thinking that a particular option will provide what he wants; and
(e) the user must have a knowledge of the tree-like menu structure.

Query languages

Two good examples of non-procedural query languages are QBE (Question By Example) and SQL (Structure Query Language).

QBE is generally easier to use than command languages such as SQL and DATATRIEVE. A QBE example is described below.

Suppose a list of names of employees working in the library of a certain organisation is required. The right record, in this case EMPLOYEE, must be selected. To list the names of the fields in this record, the table on the screen is filled in accordingly:

EMPLOYEE P.				

The system function operator 'P.' will display all the fields of the EMPLOYEE record as shown below:

EMPLOYEE	EMP-NO	NAME	SALARY	DEPT

It is possible for QBE to create a very large table but only part of it can be shown at any one time. Both horizontal as well as vertical scrolling functions are available.

To request for a list of employees working in the library, the following data and command are entered into the table:

EMPLOYEE	EMP-NO	NAME	SALARY	DEPT
		P.		LIBRARY

After the ENTER key is pressed, the names of the employees will be displayed under the NAME column as shown below:

EMPLOYEE	EMP-NO	NAME	SALARY	DEPT
		ANG LEE CHOO		LIBRARY
		BOEY CHOEK ENG		
		CHAN KAH WAI		

The QBE example shows that programming knowledge is not required. However the user must learn and remember the various commands. He must also have some knowledge of the structure of the database.

SQL has a powerful set of English-like commands. The user must enter a query using the right command and syntax. The basic form of a query is shown below:

SELECT	(specify the fields the user wants to see)
FROM	(specify the files from which the records are to be retrieved)
WHERE	(state the search criteria)

It is apparent that the user will have to learn and remember the right commands, syntax of the language and the structure of the underlying database.

Natural language interfacing

NLI will provide the most convenient and flexible means for man-machine communication. If NLI is available, users will not have to learn and remember the type of artificial languages described earlier. There is also no need for them to understand the technical details of the database or knowledge base. Using a natural language like English, they can ask and answer questions as though they are communicating with another person.

NLI must have language understanding capabilities. A NLI system is required to understand NL inputs supplied by the user regardless of whether the inputs are questions, answers, instructions or comments. Natural language understanding is not easy. It consists of two separate tasks, understanding written text and understanding spoken language. This paper is concerned with the understanding of written text and not the understanding of spoken language. To understand spoken language, all the knowledge for the understanding of written text is required in addition to the knowledge about phonology and ambiguities in speech (Rich, 1983).

Linguistic capabilities

A Natural Language Interface system must have linguistic capabilities.

To understand a sentence alone, a number of analysis tasks are involved. It is necessary to understand every word in the sentence. These words are then combined to form a structure to represent the meaning of a sentence.

Understanding words

This task requires the use of a lexicon or a dictionary. Each word will have to be checked for its presence in the lexicon. However it must be remembered that many words have more than one meaning. It is also possible for words to have synonyms. It is therefore necessary to associate with each word in the lexicon grammatical markers and semantic markers. Grammatical markers identify the possible grammatical uses of the word and semantic markers provide information about the context in which the word is used.

The size and the relevance of the vocabulary for the application domain are important considerations. Since it is never possible for a system to have a complete vocabulary, it is essential that the vocabulary can be extended easily.

Syntax, semantics and pragmatics

The understanding process continues with syntactic analysis, semantic analysis and pragmatic analysis. 'Syntax is concerned solely with relations between linguistic expressions; semantics with relations between expressions and the objects to which they refer; and pragmatics with relations among expressions, the objects to which they refer, and the users or contexts of use of the expressions,' to quote Montague (1974).

The distinction does not imply that the three tasks are unrelated. In fact the boundaries between the syntax, semantics and pragmatics cannot be clearly defined.

Syntactic analysis, which determines 'the relations between linguistic expressions', requires a grammar of the language so that only grammatically correct sentences are accepted. However in some cases the relations between words, phrases, clauses and so on, in a sentence cannot be accurately determined without reference to the context. The situation (pragmatics) must be known before a sentence can be interpreted correctly. For example the following sentence has different meanings depending on who 'he' and 'his' are referring to:

> Lim told Tan that he had lost his book which he borrowed from the Main Library.

In other cases the syntax cannot be determined without referring to the meaning of the words (semantics). Some sentences might even be

dismissed as ungrammatical if no reference to the meanings of the words are made. The following sentence provides an illustration:

> Linda borrowed Jane Eyre.

Jane Eyre in this sentence is the name of a book and not a person. If Jane Eyre is a person then the sentence is not quite correct.

The study of linguistics is not just the study of sentence grammar. The meaning of a sentence cannot be determined from that sentence alone. The context of the sentence within a discourse must be considered. Thus looking at only one sentence will not reveal all the meaning contained in the sentence. A text grammar in addition to a sentence grammar is therefore required to deal with natural language. The various techniques to deal with natural language understanding are discussed extensively in the literature.

Habitability

Besides having powerful linguistic features a NL Interface system should also be 'habitable' (Wallace, 1984) and this can be accomplished by making the system more intelligent.

Ability to infer

The ability to infer or deduce has always been associated with expert systems. It is important for a NL Interface system to be able to infer from a question what a user actually wants. When a user asks a high level question like 'which category of books are most popular?' the system should be smart enough to compare the previous year/months circulation statistics for all categories of books and come up with an answer. If the system does not have such capabilities, a user will be forced to ask a number of simple questions and deduce the conclusion for himself.

Ability to obtain correct information

Sometimes a NL Interface system has to obtain information from the user first before it can provide answers/advice. There are situations when a user, especially a naive one, is unwilling or unable to provide an answer to a question posed by the system.

Some users may find the questions asked by the system too technical or incomprehensible. The system should therefore be able to rephrase the questions using less technical words or even simpler English. Also, the system should explain or clarify terms and concepts before posing the questions again.

Some users can also be puzzled by the questions asked. They should therefore be given the opportunity to ask the system why the questions

were asked. The system of course should have the capability to provide the reasons.

If the user is reluctant to answer a question for whatever reason, the system should be intelligent enough to ask indirect questions to solicit a reply.

Ability to make anaphoric and other references
Sometimes users make reference to objects previously mentioned in a dialogue by using pronouns or definite noun phrases. For example, a user may ask a question like 'what is the gross salary of the Chief Librarian?' at the beginning of a session. The user should be able to ask later 'what are her responsibilities?' and to expect the system to know that 'her' refers to the Chief Librarian.

Things which have not been mentioned explicitly should also be understood by the system if it is able to infer from the knowledge the system has of the domain of discourse, and the goals and objectives of the user.

Ability to handle ellipsis, telegraphic input and ungrammatical input
In an ordinary discussion a person sometimes leaves out parts of sentences and expects the other person to be able to fill in the missing parts since both parties share the context of discussion. For example, an intelligent NL Interface should be able to answer the following series of questions without asking for clarifications:

> How many books did the library buy last month?
> The month before?
> All delivered?

Sometimes obvious words are left out by the user and the system should know what they are. Ungrammatical but sensible input, abbreviations and other shorthand devices should be accepted by the system.

Ability to accept indirect answers
For various reasons a user may provide indirect answers to questions asked by the system. He may want to provide more than is asked for thinking that the additional information might help the system. He would be annoyed if the additional information he provided is ignored. Sometimes a user may not want to answer a question because it involves mental effort on his part. He would rather provide another piece of information for the system to deduce the answer. For example, the system may ask a question like 'How many years have you been doing research in Information Science?' The system should accept an answer like 'I started way back in 1962.'

A NL Interface system must possess enough knowledge to be intelligent and to be able to communicate effectively with its users. The types of knowledge the system should have include:

(a) knowledge of the user's goals and objectives;
(b) knowledge of the user's level of understanding of the system;
(c) general world knowledge;
(d) knowledge of the domain of discussion; and
(e) knowlege of the context of discourse.

The system should also have an inferencing mechanism in order to provide the facilities discussed.

Current states of NLI developments
NL Interface systems are already making their appearance. They are either general purpose systems or specially tailored ones. General purpose systems are normally front-end software which are separated from the database or knowledge-based systems. One advantage of general-purpose systems is that they are portable. They can be used with different database or knowledge-based systems.

Two very successful commercial NL Interface systems are RAMIS II (produced by Mathematics Products Group) and INTELLECT (produced by Artificial Intelligence Corporation).

RAMIS II
The RAMIS II ENGLISH component of RAMIS II allows a user to express queries in free-form conversational English. The facility includes a general dictionary of commonly used English words and phrases as well as a File Specific Dictionary (FSD). The FSD allows the user to add words and phrases that are specific to a particular application. There is also a dynamic dictionary containing information acquired by the system through dialogue with the user.

RAMIS II ENGLISH sessions are very simple and require very little knowledge of the database organisation. A query can be formulated in a number of ways. For example, to ask for the number of books in each branch library, any of the following questions will be able to generate the required output:

> Show me the number of books in each branch library.
>
> I need to know the number of books each branch has. Please give me the answer.
>
> How many books are there in each branch library.

INTELLECT
INTELLECT runs on a large number of computer systems and is

known by other names like OnLine English, ELI – English Language Interface and GRS.

INTELLECT is only a query system which is used for retrieval of information from databases. Database management systems which can use INTELLECT as a front-end include ADABAS and FOCUS. It has a knowledge base which contains all the facts and rules for English grammar and for the processing of database queries.

One unique feature of this system is that when a query is read it is echoed back in a standard form. This allows the user to check whether the request has been interpreted correctly.

The system can process telegraphic and ungrammatical English. INTELLECT also automatically assumes a relation between each new query and the preceding queries in a session. Since spelling error is a very common source of confusion, INTELLECT will ask for a clarification of a query if it fails to recognise any word.

More powerful features can be provided by special-purpose interfaces which combine the domain and linguistic knowledge with the database or knowledge base. However, because of their complexities, special-purpose NL Interface systems require a substantial amount of time and effort to build.

Conclusion

There will be extensive use of NL interfaces to database, decision-support and expert systems in the near future. More general-purpose NL interface systems will be made available even for microcomputer systems. Some organisations may attempt to develop their own special-purpose NL interface systems in-house.

It is anticipated many developers will claim that they have developed NL interfaces. Interfaces which merely allow the use of English-like commands, which still have restrictive vocabularies and syntax, which use canned text, and which are not at all intelligent should not be classified as NL interfaces. New products will be severely tested and unfairly in some cases. Users will expect the new systems to have capabilities beyond what can be achieved in the foreseeable future and will be tempted to ask questions outside the domain of the database or knowledge base.

Research activities in the areas of natural language understanding and knowledge representation will continue. Systems supporting all aspects of database or knowledge-base manipulation, instead of just information retrieval, will also be developed.

References

Bates, M. and Bobrow, R. J., 'Natural Language Interfaces: What's Here, What's Coming, and Who Needs It', in *Artificial Intelligence Applications for Business.* New Jersey, Ablex 1984, pp. 179–94.

Charniak, E. and McDermott, D., *Introduction to Artificial Intelligence.* Addison-Wesley, 1985.

Harris, M. D., *Introduction to Natural Language Processing.* Reston Publishing, 1985.

Jones, D. S. and Wilks, Y. (eds), *Automatic Natural Language Parsing.* Ellis Horwood, 1983.

Jones, K. S., 'Natural Language Interface for Expert Systems', in *Research and Development in Expert Systems.* Cambridge University Press, 1985, pp. 85–94.

Martin, J., *Fourth Generation Languages.* Savant, 1985.

Montague, R., *Formal Philosophy.* New Haven, Yale University Press, 1974.

Rich, Elaine, *Artificial Intelligence.* International Student Edition. Tokyo, McGraw-Hill, 1983.

Salton, G. and McGill, M. J., *Introduction to Modern Information Retrieval.* McGraw-Hill International, 1983.

Wallace, M., *Communicating with Databases in Natural Languages.* Ellis Horwood, 1984.

Webber, B. L. and Finin, T., 'In Response: Next Steps in Natural Language Interaction', in *Artificial Intelligence Applications for Business.* New Jersey, Ablex, 1984, pp. 211–34.