CONVERSATIONAL HYPERTEXT: INFORMATION ACCESS THROUGH NATURAL LANGUAGE DIALOGUES WITH COMPUTERS

Thomas Whalen & Andrew Patrick

Division of Behavioural Research
Communications Research Centre
3701 Carling Ave.
P.O. Box 11490, Station "H"
Ottawa, Ontario, Canada K2H 8S2

ABSTRACT

One need not create a natural language understanding system in order to create a hypertext data base that can be traversed with unconstrained natural language. The task is simplified because the computer creates a constrained context, imposes a non-negotiable topic, and elicits simple questions. Two small hypertext data bases describing the authors' organization and the terms and rules of baseball were implemented on an IBM PC. When ten untrained people were allowed to search through these data bases, 59 per cent of their queries were answered correctly by the first data base and 64 per cent by the second.

KEYWORDS: Hypertext, information retrieval, natural language interface.

INTRODUCTION

Most contemporary electronic data bases conform to a relational model [1]. Casual users have many difficulties accessing information stored in relational data bases. First, the users must be well-trained. Query languages must be complex to be powerful enough for searching complex data bases. As well, the users must know the structure of the data base and have a good idea of its contents.

Second, the users must search for specific items of information because queries always return specific items. It is impossible to query a relational data base for general information. The closest users can come to retrieving general information is to retrieve a long list of specific items and draw general conclusions themselves.

HYPERTEXT DATA BASES

For casual users, hypertext provides a better model of electronic information storage than relational data bases. A hypertext data base consists of a large number of paragraphs of textual information that are linked together. Some of the words and phrases in each paragraph are highlighted. These highlighted words and phrases act like embedded menus. If the user selects one, a new paragraph of information is retrieved which explains or discusses that word or phrase. By repeatedly selecting these highlighted words and phrases, the user is able to navigate through the paragraphs, retrieving information of interest.

Hypertext has a number of advantages. Users require no training because the system is obvious from the beginning. Also, the paragraphs can describe any topic as generally or specifically as necessary. Further, the users do not require knowedge of the topic before beginning.

There are, however, a number of disadvantages to hypertext systems. Users can have trouble actually getting to the specific information required. They may have to navigate through a large number of paragraphs to get to the desired goal. Along the way, users are likely to get sidetracked and lost.

NATURAL LANGUAGE CONTROL FOR A HYPERTEXT SYSTEM

At the Communications Research Centre, we have been studying natural language as a method for navigating through hypertext data bases. This method produces the appearance of having a conversation with the computer, so we have been calling the technique *conversational hypertext*. The user types a question or comment in response to each paragraph displayed on the computer screen. The computer then extracts critical cues from this sentence which determine which paragraph will be presented next.

A natural language, such as English or French, has characteristics which make it useful as a hypertext navigation system. Users do not need any training: virtually everyone already knows at least one natural language. Natural language allows users to phrase their queries without constraining them to a specific query language. Also, if the computer detects critical words, parts of words, or phrases, it can jump directly to the relevant paragraph without making the user read many intervening paragraphs, yet the essential structure of the hypertext network is retained. Finally, if the users do not know what to ask, they can input queries about the information presented in each paragraph in order to navigate between adjacent paragraphs in the network.

In this system, natural language controls the presentation of information by the computer, but the computer does not have to understand natural language. This allows us to develop a conversational hypertext system now, even though the natural language understanding problem has not been solved and will probably not be solved in the near future.

The problem of controlling information presentation with natural language is much simpler than natural language understanding because the computer need only detect critical words or phrases in the user's sentences and respond to them. For example, if a user has asked almost any question about "CD-ROM", the computer does not have to interpret the nuances of the user's query to know that it should present the paragraph of text that describes CD-ROMs.

The fundamental issue in this research is the degree of variability found in people's queries. If every person were to type exactly the same sentence in response to each paragraph of text, the problem would be trivial because the computer would only have to detect a single sentence. Conversely, if every person were to type a completely unique sentence in response to each paragraph of text, the problem would be unsolvable because the computer would have to respond to every possible sentence.

Of course, there is always some variability in users' inputs. Different people do type different sentences into the computer at different times. However, a number of socio-linguistic factors act to limit the variability of the user's inputs [2].

First, the overall context limits the variability. Users realize that they are typing on a computer, so they do not type things that are obviously nonsensical in that context. For example, no one ever asks a computer what it had for lunch. Someone who did ask such a question out of perversity would not be at all surprised or upset when the computer did not return any information.

Second, the computer has the power to set a non-negotiable topic. The computer opens the conversation by presenting the first paragraph to the user. If the first paragraph in a conversational hypertext data base states that the computer knows about the disease AIDS, people will not ask it about the current state of satellite communications research in Canada.

Third, people quickly learn that the computer is looking for critical cues in their sentences. No one is misled into believing that the computer actually understands their queries. As a result, people tend to use the same vocabulary that the computer uses. Further, if they do not have a specific question, they use the most recent paragraph as a prompt by asking about specific topics mentioned in that paragraph.

CONVERSATIONAL HYPERTEXT IMPLEMENTATION

In order to test these principles, we wrote a program for the IBM PC. This program allows us to write a number of paragraphs of text, to connect them in a network, and to navigate through the network using natural language queries.

Each paragraph and transition between paragraphs in the data base is labeled with a string consisting of alphanumeric characters separated by asterisks. The asterisks act as "wild card" characters. This string is

called a *parse template*. An input sentence is said to satisfy a parse template if the input sentence contains the same characters in the same order as the parse template, except for the asterisks which automatically match any number of characters in the input sentence.

When the program receives a sentence from the user, it compares it to the parse templates attached to transitions and paragraphs in the data base according to the following procedure. First, it compares the sentence with the parse templates for each of the transitions leading from the paragraph currently being displayed. If none of the parse templates for the current paragraph is satisfied, the input sentence is compared to the parse templates for the transitions from the previous five paragraphs which were displayed. If none of these parse templates are satisfied, then the input sentence is compared to the parse templates which label each of the paragraphs in the data base. If none of these are satisfied, the program prints a message which indicates that the desired information was not found.

If any parse template is satisfied by the input sentence, then the paragraph pointed to by the transition or labeled by the template is made the current paragraph. However, this paragraph is not displayed yet. The parse templates for the transitions leading from the new current paragraph are then compared with the input sentence. If any of those parse templates are satisfied, the paragraph that its transition points to is made current. Thus, the program continues to follow transitions and designate new current paragraphs for as long as the transitions are satisfied. Only when the input sentence cannot satisfy any further transitions is the last current paragraph displayed. The program then waits for the user to type another sentence.

The rationale for this search strategy is easier to understand than the details. When the user types a sentence, the program presumes that the user is most likely asking about something in the current paragraph. If not, then the user may be remembering something in one of the other paragraphs presented recently. If that is not the case either, then the user may be trying to change the topic to something else covered in the data base. Once the program finds the next paragraph, it does not stop because the users' question may be asking for more detail than is presented in the first paragraph found. If the input sentence is a valid response to the new paragraph as well, then the search should be continued with the new paragraph as a starting point.

The effect of this procedure is that a query will be compared with more than one parse template in sequence. Each parse template may only look for a single cue in the sentence, but the sequence of templates may examine or parse a large part of the sentence. For example, a person accessing information about Communications Canada may type "How many people work in the Division of Behavioural Research?" This sentence would satisfy the parse template, "*DIVISION OF BEHAVIOURAL RESEARCH*" which is the label on a paragraph which describes the division in general terms. However, there may be a parse template, "*PEOPLE*WORK*" which leads from that paragraph to another paragraph which describes the staff of the division. Furthermore, that paragraph may have a parse template, "*HOW MANY*" which leads to a paragraph which tells that there are six researchers, one research assistant, and one administrative assistant in the division.

This example shows the justification for calling these strings "parse templates". Even though each template only looked for a single cue, the sequence of templates actually parsed the whole sentence. In actual practice, a whole sentence is seldom parsed, but it is common for the conversational hypertext system to select more than one part of speech from a sentence before the search is completed.

TESTING THE CONVERSATIONAL HYPERTEXT SYSTEM

Using this program, we created a small data base of information about the Division of Behavioural Research at the Communications Research Centre. This data base was created completely empirically. A few paragraphs were written and connected in a small network. People were then asked to search for information. More paragraphs and transitions between paragraphs were created to satisfy any querys that the data base could not answer correctly. This testing and creation cycle continued until we had created about 68 paragraphs connected by 108 transitions.

At that point, we tested ten people without further changing the data base [3]. Each person was asked to use the data base for as long as they found interesting in-

formation. On average, people entered 20.3 queries before stopping. The data base correctly responded to 59% of these 203 queries. There are two types of correct responses. The data base can provide the correct information, as happened for 79% of the correct responses. Alternatively, the data base can correctly respond that the requested information is not in the data base, as happened for 21% of the correct responses.

When asked, 9 of the 10 people said that they would use this type of data base again if it had different information in it. As well, 7 people said that they thought the system was searching for key words and phrases in their sentences. No one was deceived into believing that the program actually understood their queries.

In a second experiment, we created a data base which explained the terms and rules of baseball. This data base consisted of 204 paragraphs and 386 transitions. It was written from existing books of rules and terms of baseball, then "tuned up" from empirical testing, unlike the first data base which was created empirically from the beginning. When this data base was tested in the same way as the first, the results were very similar. The data base correctly responded to 61% of a total of 415 queries. However, information was returned for only 54% of the correct responses. For 46% of the correct responses, the program correctly replied that the requested information was not in the data base. The reason for this discrepancy was clear from an examination of peoples' queries. People often asked for information about team standings or about specific baseball players. These queries were outside the topic of the data base, "terms and rules of baseball".

CONCLUSIONS

Based on these results, we concluded that it is feasible to provide natural language access to hypertext data bases. People can obtain information they want from such a data base, and they are generally satisfied with the system.

Creating such a data base requires a considerable amount of work since it is completely "hand-crafted", and must be "tuned-up" using empirical results. However, it is not necessary to use empirical results throughout the data base creation. Rather, the bulk of the data base can be created directly from existing information.

Finally, we have found that one of the most important steps in creating a conversational hypertext system is selecting an appropriate topic. It is critical to explicitly limit the topic in order to reduce the variability of the users' inputs. However, it is not possible to limit the topic arbitrarily. A topic must be selected which the users' will consider reasonable [4].

We are presently continuing this research in order to find ways to decrease the number of errors made and to find more efficient ways to create natural language hypertext systems.

REFERENCES

1. Codd, E. F. A relational model for large shared data banks. *Communications of the ACM, 13*, (1970), 377-387.

2. Brown, G., and Yule, G. *Discourse Analysis*. Cambridge University Press, Cambridge, 1983.

3. Whalen, T. E. *The feasibility of natural language interfaces for electronic database access*. Department of Communications Technical Memorandum, IR0073/87, 1987.

4. Patrick, A. S. *On the role of natural topics in natural language databases*. Department of Communications Technical Memorandum, BT0082/87, 1987.