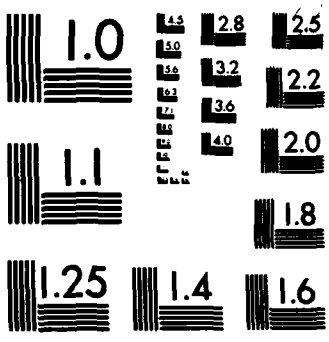
AD-A131 382 ARTIFICIAL INTELLIGENCE IMPLICATIONS FOR INFORMATION RETRIEVAL(U) ILLINOIS UNIV AT URBANA COORDINATED SCIENCE LAB G DEJONG APR 83 AFOSR-TR-83-0658 1/1

UNCLASSIFIED F49620-82-K-0009 F/G 6/4 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER **AFOSR-TR- 83-0658** | 2. GOVT ACCESSION NO. A131382 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) "ARTIFICIAL INTELLIGENCE IMPLICATIONS FOR INFORMATION RETRIEVAL" | | 5. TYPE OF REPORT & PERIOD COVERED TECHNICAL |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) Gerald DeJong | | 8. CONTRACT OR GRANT NUMBER(s) F49620-82-K-0009 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS University of Illinois Urbana, IL 61801 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS PE61102F; 2304/A2 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS AFOSR/NM Bolling AFB DC 20332 | | 12. REPORT DATE April 1983 |
| | | 13. NUMBER OF PAGES 15 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) UNCLASSIFIED |
| | | 15a. DECLASSIFICATION DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

AUG 15 1983

A

18. SUPPLEMENTARY NOTES

SIXTH ANNUAL INTERNATIONAL ACM SIGIR CONFERENCE, JUNE 6-8, 1983, Washington D.C.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Overall, the field of information retrieval is already more aware than many other fields of the relevance of artificial intelligence. (AI) (1-6). NOnetheless there remain exciting applications of artificial intelligence that have been so far overlooked. In this paper we will point out some of the ways artificial intelligence might influence the field of information retrieval. We will then examine one application in more CONT.

DD FORM 1 JAN 73 1473

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

detail to discover the kind of technical problems involved in its fruitful exploitation.

# ARTIFICIAL INTELLIGENCE IMPLICATIONS FOR INFORMATION RETRIEVAL

Gerald DeJong
Coordinated Science Laboratory
University of Illinois
1101 W. Springfield Ave.
Urbana, IL 61801

April 1983

An Invited Talk at the

Sixth Annual International ACM SIGIR Conference
June 6-8, 1983 - Washington, D.C.



---

This work was supported by the Air Force Office of Scientific Research under grant F49620-82-K-0009.

## 1. Introduction

Overall, the field of information retrieval is already more aware than many other fields of the relevance of artificial intelligence (AI) [1-6]. Nonetheless there remain exciting applications of artificial intelligence that have been so far overlooked. In this paper we will point out some of the ways artificial intelligence might influence the field of information retrieval. We will then examine one application in more detail to discover the kind of technical problems involved in its fruitful exploitation.

Before proceeding, it is important to interject a note of caution. While the promise of artificial intelligence is indeed bright, the time of complete fulfillment of its promise is a long way off. Of course, some of the expected contributions are shorter term than others. However, the more difficult problems will fall only after a good deal of basic research is accomplished. Artificial intelligence researchers have, in the past, been culpable of what can most charitably be described as over-optimism [7,8]. This naivete on the part of even the most respected of researchers stemmed from the profound subtleties underlying intelligent behavior. The problem is compounded by the fact that some of the most difficult of intelligent behavior (i.e. common sense) seems intuitively easy.

## 2. Categories of AI Applications

Artificial intelligence applications to information retrieval fall into four broad categories: (1) human-database interfaces, (2) conceptual indexing, (3) automatic data entry, and (4) active memory techniques. In the remainder of this section we will describe each of these areas briefly.

Artificial intelligence applications in these categories are in different stages of realization. There have already been several information retrieval systems written using some of the AI applications to be mentioned. Other AI applications have not yet found their way into the information retrieval community. All of the AI applications, however, must still be regarded as experimental and research oriented. It is yet too early to expect more than the the most meager practical benefits from AI.

The first category, artificial intelligence interfaces between humans and databases, is involved with making computers generally more accessible to untrained people. Natural language processing work falls into this category. Imagine, for example, a database system that could be conversed with, much the same way as a user might interact with an extremely well informed human. We will examine this category in more detail in the next section.

The second category, conceptual indexing, attempts to organize a database in terms of the meaning of its entries. In such a database the system must, in some sense, "understand" its entries. Queries, specified in the same meaning representation, are then used to index into the database. To the extent that this is successful, false positive database responses are eliminated. The meaning query represents precisely the user's conceptual question. It is not an approximation to the users question as are current query languages. Only those database items which satisfy the meaning query are retrieved. Of course, there is the possibility that the user himself cannot formulate

precisely what we wants. This, however, is a different theoretical problem which ought to be dealt with in the first category, intelligent user interfaces.

Much conceptual indexing work is motivated by psychological considerations. There is an interesting example (due to Don Norman) that typifies human database search powers:

1) Have you ever shaken hands with President Lincoln?

People, of course, respond "No." The interesting aspect is how fast people are able to respond. Not only are subjects fast, they are very certain about the correctness of the answer. Subjects never change their mind if given more time to think it over. One might first hypothesize that people are able to construct an easy proof, by comparing their own birth date with a guess at when Lincoln died, thus showing that shaking Lincoln's hand would be impossible. This would be a difficult enough process to implement on a computer. However, people are performing in a much more subtle and complex way. Consider:

2) Have you ever shaken hands with President Carter?

A negative response is not provable by the same method since Carter is still living. Yet, people are also very fast in answering this query even if they answer in the negative. They are also very sure of the correctness. Clearly, human information is organized in a very interesting way. If this organization could be captured in an information retrieval system that system would have interesting capabilities far surpassing current systems.

Two notable attempts to capture certain aspects of human memory organization are IPP [9] and CYRUS [10]. Both systems build on work of Schank [11], and both view learning as an inseparable component of memory management. IPP reads and remembers news stories about terrorism. The conceptual content of an input article is stored away in memory. The process of storage also stimulates the system to form generalizations based on comparisons with what has been stored in memory previously. In some sense, the memory index of a new entry is dynamically constructed to be the tests of similarities and differences to other items in memory.

CYRUS is a memory model based primarily on the professional life of Cyrus Vance, ex-Secretary of State under Carter. Background information about Secretary Vance was programmed in. Representations of news reports about Vance's activities given to the system are organized and retrievable on conceptual grounds. CYRUS could answer questions like:

3) How many times has Vance met with Gromyko?

CYRUS' knowledge is organized around a time-line-like data structure called an era. Different eras are used to organize events from different, nearly-separable segments of Vance's life. The above system, since it is concerned with Vance qua Secretary of State, would search that era first.

There are two recent and very interesting books on organizing computer

memory in human-like ways. These are by Roger Schank [12] and Don Norman [13].

The third category is automatic data entry. Many databases, especially document retrieval systems, are dynamic. Most often items must be added to an existing database although some applications require deletion of old elements as well. Using conventional database organizations this process, while possibly time consuming, is relatively straight forward. However, if we adopt a conceptually organized database of the previous category, this is not the case. For such databases, a new item must be mapped into a meaning representation before it can be added to the database. The meaning representation can, of course, be coded by humans. There are several obvious and insurmountable obstacles to hand coding all of the new data. First, it is prohibitively time consuming as well as being tedious and, therefore, error-prone. Second, given current meaning languages, consistency is impossible to insure. With inconsistent coding, database integrity is compromised and the advantages of conceptual organization are lost. The ideal solution is to have an artificial intelligence system read and understand text to be added to the database. "Understanding" here is defined as mapping from natural language text to a meaning representation. The meaning representation can then be used to index the original text item. Thus, automatic data entry maintains a dynamic conceptually organized database with no human intervention.

The only system I am aware of that attempted to combine conceptual organization of memory with automatic data entry is CYFR [14]. While one can easily imagine systems that automatically update a non-conceptually based memory (for example, using key words), these are of little interest from an artificial intelligence point of view. There have been other artificial intelligence systems run on unedited real-world inputs. However, the inputs are screened by a human to eliminate those outside the program's domain of expertise. This significantly changes the spirit of the task.

CYFR resulted from combining the FRUMP system [15] with the CYRUS [10] system. FRUMP reads the UPI newswire inputted directly to the computer. The level of understanding achieved by FRUMP is shallow due to the large amount of domain knowledge required. Its understanding approximates that of a human skimming a newspaper article quickly. Its knowledge is confined to 63 topic areas. It can only understand news articles concerning one of these topics. The topics range from earthquakes and floods to trade agreements and wars. In particular it knows about diplomatic visits, negotiations, etc. These are precisely the situations in which Secretary Vance participated most. The output of the FRUMP system is an "understood" (that is, conceptual) representation of the input. All representations containing a reference to the concept for Vance are given to the CYRUS system which then organized them in its memory. Thus, when functioning properly, a news article reporting a Vance trip to West Germany would be understood and inserted into the database. A user could ask the system "Where is Vance now" and "Why is he there" and receive an up-to-the-minute reply.

It should be stressed that the day to day performance of the system is far from flawless. Many technical problems were discovered and addressed in constructing the system and many more in analyzing why problem inputs were incorrectly processed. In all fairness, however, CYFR is intended as a

research tool, not as a production level computer system. When it fails in interesting ways CYFR is doing exactly what it was intended to do.

The fourth category is made up of a broad range of artificial intelligence research that might be termed "active memory" approaches to information retrieval. By "active memory" I mean a system in which the memory itself plays an active role in the update and retrieval functions. For example, consider a human memory. Human memories are known to be reconstructive [16]. That is, certain facts are not stored. Rather, they are reconstructed on demand from other information. Clearly a human memory with all its frailties and failings is insufficient for many information retrieval applications. However, the notion opens the possibility of, in some more fundamental way, storing rules that characterize data elements rather than storing all of the data elements individually. Carrying this a step further, an active memory might compute other functions as well. In particular, enforcing database consistency or adding true but unmentioned inferences might be done by the database itself.

An obvious way to proceed is to incorporate a theorem prover into the database system. Queries could be answered by first looking for the answer explicitly, and, if that failed, trying to prove the query based on other information in the database. The most straight forward application of this is the notion of deductive data retrieval. A number of such systems have been constructed [17,18,19]. Charniak, Riesbeck and McDermott [20] and Nilsson [21] both give an introductory overview of how this can be implemented using a first-order predicate calculus theorem prover. There are, of course, problems with a normal predicate calculus theorem prover. For example, membership in the set of true theorems is, in general, semi-decidable. That is, in general, one cannot disprove a false theorem. One interesting solution to this dilemma is proposed by Allen, Frisch, and Litman [22]. They weaken the first-order inference engine in just the right way so that their theorem prover is guaranteed to halt on any input.

Another problem is that predicate calculus theorems are monotonic. That is, once a theorem is proved, it cannot later be disproved by the addition of more information. This is tenable only if the system has complete knowledge of its world. No inconsistencies or gaps in knowledge are tolerable. The real world of humans is not like this. People most often are not absolutely sure of information. Rather, they believe something because they find it convenient to do so and it does not conflict in important ways with what they have previously believed. That is, they make assumptions about the world. If these assumptions are later judged to be unwarranted, they can be given up. Notice, however, that giving up an assumption can require giving up other beliefs that were believed because the original assumption was believed. The ability to "take back" a previously proved theorem is beyond first-order predicate logic. This is the basis for adopting a non-monotonic logic [23]. Underlying much of this work is the notion of data dependency [24,25,26]. Informally, the technique involves storing reasons for believing events in memory as well as the events themselves. If the system discovers that a stored event, in fact, did not occur, it can undo all of the structure that was built under the false assumption that the event did occur. An interesting application of this is to maintain database consistency (see, for example, the truth maintenance system of Doyle [25]).

Another related field is inductive inference. Researchers in this field attempt to construct generalizations that characterize a set of inputs. The primary motivation is to study learning and inference, not information retrieval. However, the area does have implications for information retrieval and one day may well form an important component of most information retrieval systems. If an exact characterization can be constructed, the inputs need not be stored. They can be derived from the rule whenever necessary.

Let us consider the well known system of Winston [27]. Suppose we want to know, for any object, whether or not it is an arch. This is a ludicrous assumption, to be sure, but it makes the problem tractable from an artificial intelligence point of view. An information retrieval solution might be to set up a database indexed by the name or description of the objects. In each entry is stored whether or not that object is an arch. There must be on entry for each object of concern. Thus, we have easily constructed a rather trivial effective procedure to test for "archness." Winston's approach can be viewed as a trainable "active memory" database. It tries to construct rules of archness that can be applied to an object's description to classify it as an arch or not. The rules are constructed through analysis of a set of training examples selected by a human teacher. Each example consists of an object description and a classification as to whether or not it is an arch. If the system succeeds in constructing such rules, the system need not actually store entries for each object. Rather, the classification rules can be applied to object descriptions as they are presented to the system. The system has a kind of reconstructive ability. In at least one way this method is superior to the conventional database approach. It can classify objects never before seen which could not possibly be contained in the database.

Reconstruction is possible only if the inputs exhibit some systematic structures. However, most databases have such systematic properties, and human memory seems to take advantage of it. This example illustrates the trivial domain of "archness." The techniques have been applied to some real-world domains as well. Lebowitz [9] and Kolodner [10] have been discussed earlier. Michalski [28] has constructed a system that learns characteristics of soybean diseases and can be used to diagnose soybean problems.

## 3. Problems with AI Interfaces

The category of AI interfaces has received more attention from the artificial intelligence community than have the other areas. The general problem attacked is to make computers more accessible to untrained users. The following systems have all been concerned, to a greater or lesser extent, with facilitating human interaction with computer systems (possibly databases) through artificial intelligence techniques [2,4,10,17,22,29,31,33,34,35,36]. Much of this work concentrates on how a computer can be programmed to "understand" natural language commands and queries.

The goal, of course, is to construct a database system that can flexibly and effectively be accessed by users with no training in a database query language, and perhaps with no computer training at all. Thus, a user interested in legal precedence might ask a database of legal cases simply:

4) Could you cite all cases in the last five years in which a husband sued and was awarded child custody.

The user can address queries to the computer in much the same manner he would ask a learned colleague. Of course, one would then like the computer to respond in much the same way that a learned colleague might. In this particular example, the system might respond

5) There are over 2000 such cases. Do you really want them listed?

To which the person responds

6) I meant only in California.

The system says

7) There are over 500 such cases. Do you really want them listed?

Finally, the person sharpens the query with

8) What about when the man has no steady income?

and is given three relevant cases.

This illustrates very well that an untrained user might not be able to articulate his query. Indeed, if the user had to deal with a conventional database he might have done better. The process of inputing the query via some formal query language may help crystalize the request in the user's mind. There are several important points to note here. First, the system engages in a conversation-like interaction with the user. This eliminates the need for a precisely specified query; the query gradually forms out of three distinct inputs. Second, each input itself raises typical but very difficult natural language processing questions. In the first query it is not specified who was sued or even for what reason. Clearly, any person would assume that the husband's wife is referred to as the target of the suit and that the suit is for custody of a child which is presumably the offspring of the husband and wife. This information is not literally specified in the text. Literally, this request is a simple conjunction. As such it would match the case of John who three years ago sued his neighbor for backing into John's car and last year was divorced from his wife and was awarded custody of their children by the divorce judge. Clearly, John's case should not be retrieved. There is also the problem of context. The second and third human inputs are quite meaningless alone. They are understandable only in terms of what has gone on before.

These and other questions have been addressed by the above systems. For the most part, these systems map natural language queries onto some more or less conventional query language. That is, the natural language processing system is a kind of front end to a standard database system.

We will now examine some of the stickier problems that arise in constructing natural language processing systems. This will help illustrate the current state of artificial intelligence research and explain why artificial

intelligence is not yet a major component of current practical information retrieval systems.

The first problem to consider is one of grammaticality. One would like a complete syntactic grammar of English. This would aid immensely in processing the input queries. It is a problem because, despite researchers best efforts no complete grammar of English exists. Nontheless, syntactic parsing is a popular first step in natural language processing systems. Researchers have produced syntactic grammars that are quite sophisticated and complex (for example, see Hobbs [37]). However, none covers all English constructions. Thus, allowing unconstrained English queries is not possible if we insist that a syntactic parse be constructed in order to derive the semantic component. The semantic component is made up of the routines that deal with meaning. These would actually produce a query in the query language.

Given this obstacle, there are two ways to proceed. The first is to be content with processing only a subset of English. In fact, this is not as limiting as it seems at first. Most practical query systems would not make use of the full flexibility afforded by unconstrained English anyway. The limitations are primarily theoretical. Treating syntax as a first step in processing the input remains a popular technique. Augmented transition network parsing [2,36] follows this paradigm. There are, of course, non-ATN approaches that also adopt this "syntax first" strategy [30,35].

The second method of circumventing the lack of a complete English grammar is to design a system that is primarily driven by semantic considerations. This method does not require that a complete syntactic parse be produced prior to semantic processing. Rather, syntactic processing is done only when a semantic analyzer requests it. This approach has also been popular in the AI community [9,15,33].

A second, but related, problem is that of ungrammatical inputs. People often treat fragments or phrases as sentences. Human listeners usually have little trouble understanding these. Ungrammatical inputs can be very tricky for natural language processing systems, however. Again there are two possible solutions. The first is to anticipate all possible classes of ungrammatical inputs in order to write an expanded syntactic grammar that covers these cases as well as proper English queries. This approach was taken by Brown and Burton [29] in their successful semantic grammar work. The other approach, treating syntactic considerations as secondary, results in a system that is less sensitive to ungrammatical inputs. Since there is no explicit syntactic grammar there is no absolute form requirement that ungrammatical inputs violate. Syntax is still used, to the extent that it is needed, to aid in semantic processing. If this syntactic knowledge is violated the input cannot be processed. However, the required syntax is much less than that needed to completely parse the input. Any input violating these meager syntactic constraints would be meaningless to humans, too. It is unfair to expect the system's understanding to surpass a human's. The system is completely insensitive to alterations in the non-essential syntactic relations. On the negative side, theoretically, these systems often make no distinction at all between grammatical, but incorrect and wildly incorrect syntactic constructs. This would be a disadvantage in an information retrieval application if ungrammaticality was used to signal that the user might not know himself what he wants.

The next problem we will consider is establishing pronoun referents. When a user specifies a natural language query containing a pronoun, the natural language component must resolve the pronoun before a well formed database query can be constructed. Pronoun resolution can be considered as a component of the larger problem of anaphora and referring expressions. However, in the interest of brevity and accessibility we will consider only a few representative pronoun problems.

In the simplest case, a pronoun stands in place of another word or phrase in the input. For example, consider sentence (9).

9) John went to bed after he dined.

The pronoun "he" refers back to the proper noun "John." We interpret this sentence as meaning that the same person who ate also went to bed and that his name is John. An obvious and simple (but unfortunately wrong) way to process this pronoun is to keep a list of other nouns mentioned thus far in processing. When a pronoun is encountered, we search this list for a noun that matches in gender and number. In this case, since "he" is singular and male, the noun "John" is chosen as the referent rather than "bed." There are several immediate problems with this solution. Consider

10) After he dined John went to bed.

Here, again, "he" refers to "John." However, the noun "John" is not encountered until after the pronoun "he." Clearly, we cannot always insist that pronouns be resolved as they are encountered. There is the much more difficult problem of which noun phrase to select when more than one agree in gender and number. It is also not uncommon for the pronoun's referent to occur in a different sentence from the pronoun. See Charniak [30] for a interesting discussion of this. Even without complicating things by introducing many noun phrases or multiple sentences we can see that selecting a referent for a pronoun can be subtle and difficult problem. Consider sentences (11) and (12). Sentence (11) is similar to (9) and (10). Again the pronoun "he" refers to "John." However, in sentence (12) which possesses many similarities to the previous sentences, "he" cannot refer to "John."

11) After John dined he went to bed.
12) He went to bed after John dined.

This phenomenon, while subtle, is understood. The rule is that the pronoun subject of an independent clause cannot find its referent as the subject of one of its own following dependent clauses. Now we will consider briefly some further problems with pronoun resolution.

Consider the following interchange (13 and 15 are user inputs, 14 is a database response):

13) Does every salesman own his own car?
14) Yes
15) How many of them are more than five years old?

The problem is the pronoun "them" in sentence (15). In this context

"them" ought to be taken to refer to the cars that the salesmen own. This reference is complicated by the fact that there is no previous literal phrase that the pronoun "them" stands for. In the previous examples, "he" stood for "John" which was explicitly mentioned elsewhere in the input. Here, there is no such corresponding phrase that means "the group of cars that the salesmen own." Furthermore, there is no easy way to rule out the referent "every salesman" for "them" which in fact does explicitly occur in the sentence. The "every salesman" can be ruled out on semantic (or "meaning") grounds by realizing that salesmen are, in general, much older than five years and that this fact is common knowledge which is probably known by the user. However, getting a computer system to construct such an informal proof or even getting it to ask the right question at the right time without causing unmanageable side-effects is very difficult. In any case this process only rules out "every salesman" for the referent. It says nothing about how to construct the correct referent. Nash-Webber and Reiter [32] call this the problem of implicit sets. Clearly a good deal of semantic processing must be done before all of the input words can be interpretted.

Another example showing how semantics can influence pronoun resolution is the following:

16) Does every salesman who has been with the firm more than two years get at least 18% commission?
17a) Yes
18) List them

In sentence (18) "them" refers to "the salesmen who have been with the firm more than two years and get at least 18% commission." Here the referent comes from a previous input. The problem is more subtle that that, however. Consider the interchange slightly altered:

16) Does every salesman who has been with the firm more than two years get at least 18% commission?
17b) No
18) List them

In this version, the "them" in sentence (18) most likely refers to "the salesmen who have been with the firm more than two years and do NOT get at least 18% commission." This is, in some sense, the complement of what the same pronoun referred to in the previous interchange. Obviously, the choice of pronoun referents depends not only on the previous input which contains the referent but also on intervening database responses.

There are many other very difficult problems besides ungrammatical input and pronoun resolution. A few of the trickier problems are 1) lexical ambiguity, 2) conjunction 3) prepositional phrase attachment and 4) literally incorrect inputs.

Lexical ambiguity has to do with processing words that can have more than one meaning. In English it is the rule rather than the exception. One need only consult a dictionary to be convinced of this. It is the rare entry that has only one word sense. Most words have at least three entries and some have twenty or thirty. Furthermore, not all meanings are the same part of speech.

For example, consider the word "bow." My dictionary gives 17 meanings. Some are for a verb (both transitive and intransitive) and some are for a noun. Clearly, the word senses that the user intended must be selected by the system if it is to correctly determine the meaning of an input.

Conjunction is the use of connectives to join sentence parts. The problem is that syntactic entities distribute over connectives in many different ways. Usually, these different ways yield different sentence meanings. For example:

19) Is it time to re-order the high-voltage diodes and transistors?

This input has two slightly different syntactic parses corresponding to two slightly different interpretations. They differ in whether or not the user has presupposed that the transistors are also high-voltage. This may seem like a small difference but in the right circumstances it can have a large effect on how the query is answered. Suppose, for example, that there are two types of transistors in stock: normal transistors and high-voltage transistors. A different interpretation of the word "transistors" may well change the response of the system.

The solution must involve the use of domain-specific semantic and pragmatic knowledge. For example, suppose that the company uses only one kind of transistor - the low-voltage kind. Clearly, the word "transistors" in (16) ought to be understood as referring to that item. Thus, the adjective "high-voltage" must be understood NOT to distribute to "transistors." Notice, however, that this means that the actual syntactic parse produced can be influenced by knowledge about the contents of the company's stock inventory.

We will now mention a similar problem, prepositional phrase attachment. Consider sentence (20).

20) Is it legal, in Arkansas, to hang a man with a moustache?

This is an old joke. The answer is "Yes, but it works better if you use a rope." The prepositional phrase "with a moustache" is initially interpreted as adjectival (modifying "man"). However, syntax allows the prepositional phrase to be attached to the verb "hang" instead. This is the reading required by the answer. Syntactically, both readings are acceptable. Semantically, however, one is far better. The idea of strangling a man on the gallows with his own moustache is ludicrous. Again, high level knowledge, in this case about stereotypic executions, is necessary before a low level syntactic parse can be decided upon.

Finally, people seldom say what they mean and occasionally say nearly the opposite. Consider again sentence (19). The query asks whether it is time to reorder "diodes and transistors." In fact, the user probably meant a disjunction rather than a conjunction. If there were no more transistors but sufficient diodes remained, a literal interpretation of the question requires that the database respond negatively; it is not yet time to reorder them both. Clearly, a system should respond to what is meant and not what is said. Consider another example (patterned after one of Schank [34]):

21) Are the cans of tuna received September 12 still palatable?

Literally interpreted, the answer would be "no." Cans are not a suitable food no matter what they contain or when they were received. Clearly, however, the literal meaning is not what was intended. The question of palatability is to be answered concerning the tuna inside the cans and not the cans themselves.

Examples like these crop up frighteningly often in the real world. One possible response to all this might be "well, if people are going to talk to computerized databases, they at least ought to say what they mean." This misses the point. The purpose of introducing an artificial intelligence natural language front end to a database is to permit the user to operate in his own native mode. We want to eliminate the need for specialized training by allowing users to communicate in English, a language in which they are already fluent. To put constraints on what English is acceptable and what is not violates the spirit of the task.

## 4. Conclusion

I would hope that the reader is left with some notion of the vast and promising possibilities of incorporating artificial intelligence in information retrieval systems. At least equally important, however, is the need to develop a realistic appreciation of the current state of the field. Reviewing the previous sections I am inclined to believe I have painted a slightly too pessimistic picture. But perhaps this is worthwhile in order to counteract the flashy, science-fictiony side of the field that has been over-exploited by some.

There is little doubt that most information retrieval systems will ultimately contain a significant AI component. Indeed, a number of current systems already incorporate artificial intelligence techniques. Some of these systems are even commercial products and so are more production oriented. However, they lack the full generality and flexibility that one would like in such systems. The systems are characterized by rather narrow solutions that degenerate if extended too broadly. The major problem remains lack of world knowledge. The representation and organization of which makes up a significant portion of current artificial intelligence research.

## 5. References

[1] Salton, G., (1978) Automatic content analysis in information retrieval. Technical report 68-5. Department of Computer Science. Cornell University, Ithica, New York.

[2] Waltz, D., (1977) Writing a natural language data base system. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, August, Cambridge, MA.

[3] Kellogg, C., (1982) Knowledge management: a practical amalgam of knowledge and data base technology, Proceedings of the National

Conference on Artificial Intelligence, Pittsburgh, PA.

[4] Williams, M., Tou, F., Fikes, R., Henderson, A., and Malone, T., (1982), RABBIT: cognitive science in interface design, Proceedings of the Fourth Annual Conference on Cognitive Science, Ann Arbor, MI.

[5] Montgomery, C., (1981) Where do we go from here, in Information Retrieval Research, Butterworths, London.

[6] Hafner, C., (1981) Representation of knowledge in a legal information retrieval system, in Information Retrieval Research, Butterworths, London.

[7] Simon, H., (1962) in Management and the Computer of the Future, M. Greenberger (ed.), MIT Press, Cambridge, MA.

[8] Simon, H., (1970) The shape of automarion, in Perspectives on the Computer Revolution, Z. Pylyshyn (ed.), Prentice-Hall, Englewood Cliffs, NJ.

[9] Lebowitz, M., (1980) Generalization and memory in an integrated understanding system. Computer science research report 186, Ph.D. dissertation, Yale University, New Haven, CT.

[10] Kolodner, J., (1980) Retrieval and organizational strategies in conceptual memory: a computer model. Computer science research report 187, Ph.D. dissertation, Yale University, New Haven, CT.

[11] Schank, R., (1980) Language and memory. Cognitive Science 4, 243-283.

[12] Schank, R., (1982) Dynamic Memory, Cambridge University Press, New York, NY.

[13] Norman, D., (1982), Learning and Memory, W. H. Freeman, San Francisco

[14] Schank, R., Kolodner, J. and DeJong, G., (1980) Conceptual information retrieval. Yale computer science research report 190. December, New Haven, CT.

[15] DeJong, G., (1979) Prediction and substantiation: a new approach to natural language processing. Cognitive Science 3, 251-273.

[16] Spiro, R., (1979) Prior knowledge and story processing: integration, selection, and variation, Technical Report 138, Center for the Study of Reading, University of Illinois, Urbana, IL.

[17] Joshi, A., Kaplan S.J., Lee, R., (1977) Approximate responses from a data base query system: an application of inferencing in natural language, Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA.

[18] Furukawa, K., (1977) A deductive question answering system on relational data bases, Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA.

[19] McSkimin, J. and Minker, J., (1977), The use of a semantic network in a deductive question-answering system, Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA.

[20] Charniak, E., Riesbeck, C., and McDermott, D., (1980) Artificial Intelligence Programming, Lawrence Erlbaum, Hillsdale, NJ.

[21] Nilsson, N., (1980), Principles of Artificial Intelligence, Tioga Press, Palo Alto, CA.

[22] Allen, J., Frisch, A., and Litman, D., (1982), ARGOT: The Rochester dialogue system, Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, PA.

[23] McDermott, D., and Doyle, J., (1980) Non-monotonic logic I, Artificial Intelligence, vol. 13, no. 1,2, pp. 41-72.

[24] Fikes, R., (1975) Deductive retrieval mechanisms for state description models. Proceedings of the Fourth International Joint Conference on Artificial Intelligence. pp. 99-106.

[25] Doyle, J., (1978) Truth maintenance systems for problem solving. MIT AI Technical report TR-419, MIT, Cambridge, MA

[26] Stallman, R., and Sussman, G., (1979) Problem solving about electrical circuits, in Artificial Intelligence: An MIT Perspective, vol. 2, P. Winston and R. Brown (eds.), MIT Press, Cambridge, MA.

[27] Winston, P., (1970) Learning structural descriptions from examples. MIT Project MAC report 76, Cambridge, MA.

[28] Michalski, R., and Chilausky, R., (1980) Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis, Policy Analysis and Information Systems, vol. 4, no. 2.

[29] Brown, J. and Burton, R., (1975) Multiple representations of knowledge for tutorial reasoning. In Representation and Understanding D. Bobrow and A. Collins (eds.), Academic Press, New York.

[30] Charniak, E., (1972) Toward a model of childrens story comprehension. AITR-266, Artificial Intelligence Laboratory, MIT, Cambridge, MA.

[31] Marcus, M., (1978) A theory of syntactic recognition for natural language. MIT Ph.D. dissertation, Cambridge, MA.

[32] Nash-Webber, B. and Reiter, R., (1977), Anaphora and logical form: on formal meaning representations for natural language, Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA.

[33] Riesbeck, C., (1975) Conceptual analysis, in Conceptual Information Processing, R. Schank (ed.), North Holland, Amsterdam.

[34] Schank, R., (1973) Identification of conceptualizations underlying natural language, in Computer Models of Thought and Language, R. Schank and K. Colby (eds.), W. H. Freeman, San Francisco.

[35] Winograd, T., (1972) Underatanding Natural Language. Academic Press, New York.

[36] Woods, W., (1970) Transition network grammars for natural language analysis, Communications of the Association for Computing Machinery, vol. 13, no. 10.

[37] Hobbs, J., (1974) A metalanguage for expressing grammatical restrictions in nodal spans parsing of natural language, Courant computer science report NSO-2, New York University, New York.

END

FILMED

9-83

DTIC