Intelligent Information Systems[1]

Michael Lebowitz
Department of Computer Science -- Columbia University
New York, NY 10027

ABSTRACT

Natural language processing techniques developed for Artificial Intelligence programs can aid in constructing powerful information retrieval systems in at least two areas. Automatic construction of new concepts allows a large body of information to be organized compactly and in a manner that allows a wide range of queries to be answered. Also, using natural language processing techniques to conceptually analyze the documents being stored in a system greatly expands the effectiveness of queries about given pieces of text. However, only robust conceptual analysis methods are adequate for such systems. This paper will discuss approaches to both concept learning, in the form of *Generalization-Based Memory*, and powerful, robust text processing achieved by *Memory-Based Understanding*. These techniques have been implemented in the computer systems IPP, a program that reads, remembers and generalizes from news stories about terrorism, and RESEARCHER, currently in the prototype stage, that operates in a very different domain (technical texts, patent abstracts in particular).

1 Introduction

As computer systems that provide information about documents become larger and more complicated, the need to apply techniques from Artificial Intelligence will become greater. The techniques developed in the area of natural language processing can aid in constructing powerful information retrieval systems in at least two general ways. First of all, the automatic construction of new concepts allows a large body of information to be organized compactly and in a manner that allows a wide range of queries to be answered. Secondly, applying natural language processing techniques to analyze conceptually the documents being stored in a system greatly expands the effectiveness of queries about given pieces of text. For conceptual analysis to be useful in an information retrieval system, the system must, of course, be powerful enough to process large numbers of texts it has not been specially prepared for.

This paper will discuss approaches to both concept learning, in the form of *Generalization-Based Memory*, and powerful, robust text processing achieved by *Memory-Based Understanding*. Both of these techniques have been implemented in the computer systems IPP [Lebowitz 80], a program that reads, remembers and generalizes from news stories about terrorism, and RESEARCHER [Lebowitz 82], currently in the prototype stage, that operates in a very different domain (technical texts, patent abstracts in particular).

It is not difficult to see how automatic concept creation and natural language text understanding, if practical, could be useful in information systems. It is generally acknowledged in the field ([Heaps 78, Salton and McGill 83], for example) that selecting concepts by which to index documents is one of the major problems in Information Retrieval. Concept learning addresses this problems by generalizing across multiple texts to determine the important concepts in a domain, and text understanding helps by extracting concepts from a text regardless of the specific language used.

We can see how AI can help in Information Retrieval by looking at texts illustrative of the domains handled by IPP and RESEARCHER. TEXT1 is the beginning of a typical terrorism news story that IPP processed, while TEXT2 is the first part of a patent abstract, processed by RESEARCHER.

TEXT1 - UPI, 18 January 80, Lebanon

> A hijacker gunman seized a Middle East Airlines jetliner enroute to Cyprus today, ordered the plane back to Beirut, then surrendered after two hours of negotiations in which he demanded an investigation into the disappearance of a Lebanese Shiite Moslem leader.

[1]This research was supported in part by the Defense Advanced Research Projects Agency under contract N00039-82-C-0427.

TEXT2 - APPARATUS FOR RECORDING AND READING DATA FROM A MAGNETIC PLATE

An apparatus for recording and reading-out data from a circular, foil-like magnetic plate, which is movably arranged in an interchangeable cassette having slot-like openings on both sides, and has a centrally arranged perforation, said apparatus comprising in combination:

a first drive shaft on which said magnetic plate is capable of being coupled in such a way as to bring about rotation of said magnetic plate;

a first linear motor for driving said first drive shaft of said magnetic plate;

Both of this stories illustrate how natural language processing techniques can help in retrieving useful information. If TEXT1 was in an information system, it should be retrieved if a query was made about terrorism, despite the absence of that word in the text. In addition, hijackings in Lebanon might be an interesting and useful conceptual category, but that can only be determined by generalizing over several instances. Similarly, the patent abstract TEXT2 describes a device similar to a magnetic disc drive, and should be retrieved by queries in that area, which will be possible only if the text is analyzed for meaning and related to other known objects in memory. A further discussion of the various uses AI can play in information systems can be found in [Schank, Kolodner and DeJong 80].

Obviously the domains handled by IPP and RESEARCHER are very different. The fact that memory-based understanding and generalization-based memory prove useful in both domains is a strong indication of their generality and possible widespread applicability.

2 Dynamic Concept Creation -- Generalization-Based Memory

For computer systems such as IPP and RESEARCHER to organize information so that a wide variety of queries can be answered, it is useful to create new concepts by detecting similarities in various pieces of texts. This is done using a generalization process. It is clearly necessary in an information system to break down large numbers of similar pieces of texts into categories. The advantages of doing so automatically are apparent. As we will see, using the concepts created to organize information also has efficiency benefits. In this section we will look at the concept creation, or generalization, process used in IPP. RESEARCHER uses a similar method.

The concept creation process used in IPP begins by making tentative generalizations about a situation based on only a small number of examples. It then records specific items in memory in terms of the concepts created. It is also possible to make more specific generalizations and to record these, as well, under the more general cases.

The organization of long-term memory in IPP is shown schematically in Figure 1. We will refer to the objects stored in memory, which are used to build generalizations, as *instances* (e.g. the the information extracted from pieces of text by IPP). An instance is described in terms of a set of *features*. The combinations of generalizations, themselves sets of features, and the events and sub-generalizations they organize will be called *GEN-NODEs*.

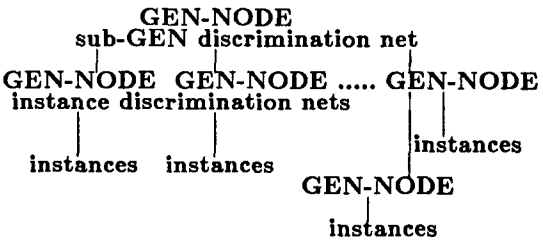

Figure 1: Generalization-based memory

As shown in Figure 1, instances and sub-GEN-NODEs are stored under a GEN-NODE using discrimination networks (D-NETs) [Charniak, et al. 80]. D-NETs provide an efficient way to retrieve any objects stored with a given set of indices. In the IPP memory model, every feature of an instance or sub-GEN-NODE is initially used as an index, resulting in shallow, bushy D-NETs that allow retrieval of an object given any of its features. The resulting plethora of indices is pruned by ceasing to use as indices features that pertain to a large number of objects in a given D-NET.

The use of a hierarchy of GEN-NODEs with D-NETs as a method of memory organization allows efficient storage of information, since information in a generalization does not have to be repeated for each instance that it describes. In addition, it allows relevant generalizations and instances to be efficiently found in memory during processing, allowing further generalizations.

The IPP generalization process itself is relatively simple. Once IPP has extracted the important information from a story (in a manner described in the next section), it searches through memory, using the hierarchy of D-NETs to search efficiently, to find the GEN-NODE that is most similar to the new information. Then it checks to see if any of the instances stored under that GEN-NODE have additional similarities to the new instance. If so, a new GEN-NODE is made, in effect creating a new concept. The details of this process can be found in [Lebowitz 83].

For each generalization made by IPP, an evaluation process continually looks for later instances for which the generalization might be relevant. Importantly, this occurs as a normal part of the memory update process, since the generalizations to be evaluated are exactly those that might be used to store the new instances. Then, IPP checks whether the generalizations are confirmed or contradicted by the new instances. Ultimately, as a result of this confirmation process, generalizations can be validated, refined or discarded. The result of this is that over the long run a generalization-based memory system will be left with a set of the important concepts that best characterize a domain.

3 Memory-Based Understanding

For a generalization-based memory to be effectively built up as described in the previous section, it is necessary to analyze conceptually the text being stored, as using the raw text would not reveal similarities that exist only at a conceptual level. In addition, our system must be able to handle large number of texts it is not specially prepared for. Fortunately, it is possible to do this using an understanding process that operates synergistically with generalization-based memory. In this section we will look at the memory-based understanding techniques used in IPP and RESEARCHER to achieve a high level of performance. Details of the IPP understanding techniques can be found in [Lebowitz 81].

3.1 IPP

IPP, as a text understanding program, integrates top-down and bottom-up processing. Its top-down processing, which is the source of its power, has very much the same predictive flavor as that in parsers such as ELI [Riesbeck and Schank 76], CA [Birnbaum and Selfridge 81], the Word Expert Parser [Small 80] and the understanding system FRUMP [DeJong 79]. Importantly, in IPP the predictions made are generated almost exclusively from the structures used to represent events in memory, and not from specific words.

The kind of syntactic information which plays a major role in ATN parsers [Thorne et al. 68, Woods 70] and other syntactic-oriented systems ([Winograd 72, Marcus 80], for example) is largely implicit in IPP's parsing rules, to be used when needed.

There are two memory structures used in IPP that are involved in parsing rules. Action Units (AUs) are used to describe stereotypical events (such as hijackings or shootings) in memory and Simple Memory Organization Packets (S-MOPs) (which are related to Schank's MOPs [Schank 80]) describe more abstract situations, such as extortion or an attack on a person. The S-MOPs are the initial versions of the GEN-NODEs described earlier. As such, they can be improved by generalization as more information is acquired. Instances of AUs are organized inside S-MOPs. S-MOPs and AUs are discussed in detail in [Lebowitz 80]. For our purposes here, it is enough to know that AUs and S-MOPs are both frame-like structures [Minsky 75] with roles, and S-MOPs record how AUs are interrelated.

When IPP identifies any Action Units or S-MOPs in a story (which is usually not difficult, as some AUs will be referred to directly in the text and an S-MOP is usually inferable from an AU), IPP uses two general rules to guide processing, one concerning Action Units and the other S-MOPs.

When an Action Unit is used to represent a story, it is important to identify how the various roles of the AU are filled (by characters mentioned or implicit). Instead of using separate predictions based on the words that instantiate an Action Unit, IPP uses a a rule known as the *AU Role Filling Rule* that allows the determination of the fillers of each role of the AU. Action Units contain information describing stereotypical role fillers. This is sufficient to determine how the various characters mentioned in a story fill the roles of Action Units, using top-down expectations largely independent of the specific method used to instantiate the AU.

The prediction from S-MOPs, known as the *S-MOP/AU Rule* simplifies the recognition and explanation of Action Units. S-MOP definitions describe the Action Units that are likely to be found as part of the stereotypical situation they describe -- methods and results, for example. This information is used to identify Action Units that appear later in the text, including those that may be described using ambiguous words.

The two rules used by IPP are summarized in Figure 2.

> AU Role Filling Rule -- Whenever an AU is instantiated from a piece of text, assume its role fillers will fit the stereotypes in memory. Then check each character, place or object mentioned in the text to see if it can be a role filler of the AU.
>
> S-MOP/AU Rule -- Whenever a situation represented by an S-MOP is identified, predict that its associated Action Units will be mentioned in the text. Use this prediction to specify the appropriate meanings for ambiguous words and to determine the relations between Action Units and instantiated S-MOPs.

Figure 2: IPP's memory-based understanding rules

These two understanding rules were crucial in allowing IPP to successfully process about 70-80% of the 600+ stories from newspapers and newswires that it processed. Since the predictions based on these rules come from memory structures rather than individual words, word definitions in IPP are realtively simple, and it was possible to give the program a fairly large vocabulary, 3000+ words. This played an important role in achieving a high level of robustness.

It is also interesting to note that since both the AU Role Filling Rule and S-MOP/AU rule depend on structures in memory, and since these structures change over time (through the concept creation process described above), IPP's understanding ability changes, and in fact improves, over time. This occurs as IPP learns more about stereotypical role fillers (e.g. the IRA are terrorists in Northern Ireland) and Action Units (e.g. extortion in Latin America usually takes the form of embassy takeovers).

3.2 RESEARCHER

A quick glance at the patent abstract shown earlier, TEXT2, indicates clearly that the rules used in IPP will not apply directly to technical texts. In particular, we notice that such texts are not focused on actions as are news stories, and involve descriptions of complex physical objects rather than events. In addition, they make considerable use of rather special purpose language.

These differences do not mean, however, that we must abandon the idea of memory-based understanding. It simply means that the exact memory-based rules that we use to help in processing must be different.

Specifically, the predictions used for understanding in RESEARCHER are based on the physical descriptions it builds up, in much the same way IPP made predictions from events. The representation used in RESEARCHER is again frame-like (with the memory frames referred to as *memettes*), but emphasis is given to the physical properties of the components of an object, and structural relations among the parts. The goal of RESEARCHER's understanding process is to record in memory how a new object being described differs from stereotypical objects already known (and ultimately to generalize new stereotypes).

Processing in RESEARCHER must concentrate on words that refer to physical objects in memory. Such words are known as Memory Pointers (MPs). These words guide RESEARCHER's processing, and make use of any information gathered bottom-up, in much the same way as IPP used S-MOPs and Action Units. This steers RESEARCHER to the needed determination of how the objects described by MPs differ from known stereotypical objects, and how the objects relate to each other (including how parts make up the main object).

RESEARCHER's memory-based parsing rules reflect some relatively simple knowledge we have about the way complex objects are usually described. The primary MP parsing rules are shown in Figure 3.

Figure 4 shows the output from RESEARCHER's processing of the title of TEXT2, illustrating the kind of processing involved. (RESEARCHER can process all of TEXT2 accurately. However, the output from such process is too extensive for our purposes here.)

In the output trace in Figure 4, we can see how

Unless otherwise specified, assume an MP:

1) refers to an object mentioned previously

2) refers to a part of an object mentioned previously

3) refers to an object known in memory

4) refers to a part of an object known in memory

disambiguating, if needed, to fulfill these rules.

Figure 3: Memory-based parsing rules

```
Running RESEARCHER at 10:19:10 AM
Patent: TEXT2

(APPARATUS FOR RECORDING AND READING DATA FROM
 A MAGNETIC PLATE)

Processing:

APPARATUS     : MP word -- memette UNKNOWN-STRUCTURE#
FOR (FOR1)    : Purpose indicator -- skip
RECORDING     : Purpose word -- save and skip
AND           : Conjunction word -- skip
READING       : Purpose word -- save and skip
DATA          : MP word -- memette DATA#
New DATA# instance (&MEMO)
New UNKNOWN-STRUCTURE# instance (&MEM1)
Relating memettes &MEM1 (UNKNOWN-STRUCTURE#) (SUBJECT)
  &MEMO (DATA#) (OBJECT)  [P-READS]
Relating memettes &MEM1 (UNKNOWN-STRUCTURE#) (SUBJECT)
  &MEMO (DATA#) (OBJECT)  [P-WRITES]
FROM          : Relation word -- save and skip
A             : New instance word -- skip
MAGNETIC      : Token refiner - save and skip
PLATE         : MP word -- memette DISC#
New DISC# instance (&MEM2) Feature: DEV-PURPOSE/MAGNETISM
Relating memettes &MEMO (DATA#) (SUBJECT)
  &MEM2 (DISC#) (OBJECT)  [R-UPON]

Text Representation:

** MEMETTE IN FOCUS **
&MEM1 (UNKNOWN-STRUCTURE#)

A list of relations:

 Subject:              Relation:     Object:

UNKNOWN-STRUCTURE#   {P-READS}     DATA#
UNKNOWN-STRUCTURE#   {P-WRITES}    DATA#
DATA#                {R-UPON}      DISC#
```

Figure 4: RESEARCHER processing TEXT2

RESEARCHER identifies the various objects mentioned in the text as instances of general structures described in memory (such as DISC# and UNKNOWN-STRUCTURE#). RESEARCHER creates new memettes to represent this structures, &MEM1 for the "apparatus", for example, and records how these instances differ from the abstract stereotypes. Notice how by using an instance of the memory structure DISC# to represent the object described in the text as a "magnetic plate", RESEARCHER will be able to associate this text with others about discs. RESEARCHER also assumes, using rules 2 and 4 from Figure 3, that a piece of text is describing a single object, in this case describing parts of the "apparatus". The final part of the output in Figure 4 indicates that RESEARCHER has identified the

important physical relations mentioned in TEXT2. Obviously in the complete text of TEXT2, many more relations are described.

RESEARCHER's memory-based understanding can be graphically seen when it has a great deal of information about the text it is reading. As a simple example, consider RESEARCHER's behavior when it reads TEXT2 for a second time, so that a description of the newly described device is in memory prior to the second reading. This output is shown in Figure 5.

```
Running RESEARCHER at 10:22:25 AM
Patent: TEXT2

(APPARATUS FOR RECORDING AND READING DATA FROM
 A MAGNETIC PLATE)

Processing:

APPARATUS     : MP word -- memette UNKNOWN-STRUCTURE# [&MEM1]
FOR (FOR1)    : Purpose indicator -- skip
RECORDING     : Purpose word -- save and skip
AND           : Conjuction word -- skip
READING       : Purpose word -- save and skip
DATA          : MP word -- memette DATA# [&MEM1 (DATA#)]
Relation already established [P-READS]
Relation already established [P-WRITES]
FROM          : Relation word -- save and skip
A             : New instance word -- skip
MAGNETIC      : Token refiner - save and skip
PLATE         : MP word -- memette DISC# [&MEM1 (DISC#)]
Recognized instance of &MEM2 (DISC#)
Relation already established [P-UPON]

Text Representation:

** MEMETTE IN FOCUS **
&MEM1 (UNKNOWN-STRUCTURE#)

A list of relations:

 Subject:            Relation:     Object:

UNKNOWN-STRUCTURE#   {P-READS}     DATA#
UNKNOWN-STRUCTURE#   {P-WRITES}    DATA#
DATA#                {R-UPON}      DISC#
```

Figure 5: RESEARCHER processing TEXT2 again

Notice how RESEARCHER's reprocessing of TEXT2 differs from its initial processing. It has recognized that this "new" text is describing something already known in memory (&MEM1). This makes the entire description built up previously available for use in later processing. Thus RESEARCHER is able to recognize that it already knows about the physical relations described in the second reading. This is in clear contrast with non-memory-based systems that would, by necessity, process TEXT2 identically if it was presented one, two or twenty times. The ability to avoid such detailed processing when it is not necessary is quite important when we are dealing with natural language information sources that are heavily redundant, including newswires and patent abstracts, for example.

4 Conclusion

We have seen in this paper how generalization-based and memory-based processing are useful in two widely disparate, information retrieval-type domains. While the precise details of the implementation of these memory organization and text processing techniques will depend heavily on the specific domain of interest, they provide good starting points for developing memory-based systems. As the scope of intelligent information systems becomes greater and greater, such techniques will be crucial for achieving the needed power and generality to complement the quantity of information we are able to store.

REFERENCES

[Birnbaum and Selfridge 81] Birnbaum, L. and Selfridge, M. Conceptual analysis of natural language. In R. C. Schank and C. K. Riesbeck, Ed., *Inside Computer Understanding*, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1981, pp. 318 - 353.

[Charniak, et al. 80] Charniak E., Riesbeck, C. K., and McDermott, D. V. *Artificial Intelligence Programming.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980.

[DeJong 79] DeJong, G. F. Skimming stories in real time: An experiment in integrated understanding. Tech. Rept. 158, Yale University Department of Computer Science, 1979.

[Heaps 78] Heaps, H. S. *Information Retrieval: Computational and Theoretical Aspects.* Academic Press, New York, 1978.

[Lebowitz 80] Lebowitz, M. Generalization and memory in an integrated understanding system. Tech. Rept. 186, Yale University Department of Computer Science, 1980. PhD Thesis

[Lebowitz 81] Lebowitz, M. Memory-based parsing. Columbia University Department of Computer Science, 1981.

[Lebowitz 82] Lebowitz, M. Intelligent information systems. Columbia University Department of Computer Science, 1982.

[Lebowitz 83] Lebowitz, M. "Generalization from natural language text." *Cognitive Science 7*, 1 (1983), 1 - 40.

[Marcus 80] Marcus, M. *A Theory of Syntactic Recognition for Natural Language.* MIT Press, Cambridge, MA, 1980.

[Minsky 75] Minsky, M. A framework for representing knowledge. In P. H. Winston, Ed., *The Psychology of Computer Vision*, McGraw-Hill, New York, 1975.

[Riesbeck and Schank 76] Riesbeck, C. K. and Schank, R. C. Comprehension by computer: Expectation-based analysis of sentences in context. In W. J. M. Levelt and G. B. Flores d'Arcais, Ed., *Studies in the Perception of Language*, John Wiley and Sons, Chichester, England, 1976. also Yale Computer Science Technical Report #78

[Salton and McGill 83] Salton, G. and McGill, M. J. *Introduction to Modern Information Retrieval*. McGraw Hill, New York, 1983.

[Schank 80] Schank, R. C. "Language and memory." *Cognitive Science 4*, 3 (1980), 243 - 284.

[Schank, Kolodner and DeJong 80] Schank, R. C., Kolodner, J. L. and DeJong, G. F. Conceptual information retrieval. Tech. Rept. 190, Yale University Department of Computer Science, 1980.

[Small 80] Small, S. Word expert parsing: A theory of distributed word-based natural language understanding. Tech. Rept. TR-954, University of Maryland, Department of Computer Science, 1980.

[Thorne et al. 68] Thorne, J., Bratley, P. and Dewar, H. The syntactic analysis of English by machine. In D. Michie, Ed., *Machine Intelligence 3*, American Elsevier Publishing Company, New York, 1968.

[Winograd 72] Winograd, T. *Understanding Natural Language*. Academic Press, New York, 1972.

[Woods 70] Woods, W. A. "Transition network grammars for natural language analysis." *Communications of the ACM 13* (1970), 591 - 606.