LANGUAGE AND MEMORY: GENERALIZATION AS A PART OF UNDERSTANDING

Michael Lebowitz

Department of Computer Science
Yale University, P.O. Box 2158
New Haven, Connecticut 06520

ABSTRACT

This paper presents the Integrated Partial Parser (IPP), a computer model that combines text understanding and memory of events. An extended example of the program's ability to understand newspaper stories and make generalizations that are useful for memory organization is presented.

INTRODUCTION

Memory of specific events has not been a serious part of previous Artificial Intelligence language understanding systems. Most understanding systems have been designed to simply parse natural language into an internal representation, use the representation for output tasks, and then discard it. This has been true even for systems primarily concerned with creating high-level story representations in terms of structures such as scripts [2] [3] [6], plans and goals [6] [7], frames [1] and other such structures, as well as more syntactically oriented systems.

The Integrated Partial Parser (IPP) is an understanding program that addresses the problems of integrating parsing and memory updating. By making use of memory, IPP is able to achieve a high level of performance when understanding texts that it has not been specially prepared for. IPP has been designed to read and remember news stories about international terrorism taken directly from newspapers and the UPI news wire.

IPP performs six different understanding tasks, in a single, integrated process. These tasks include the addition of new events to long-term memory, being reminded of previous stories, noticing the interesting aspects of stories, making generalizations to improve the quality of its world knowledge, predicting likely future events, and, of course, parsing stories from natural language into an internal representation.

In this paper, I will mention IPP's parsing abilities only in passing. The interested reader is refered to [4] for more details of the parsing process. Here I will present the program's ability to remember events and make generalizations about the stories it reads.

This work was supported in part by the Advanced Research Projects Agency of the Department of Defense and monitored under the Office of Naval Research under contract N00014-75-C-1111.

MEMORY IN UNDERSTANDING

The addition of information to long-term memory is an integral part of IPP's operation. Furthermore, the memory update process actively changes the structure of memory by noticing similarities among events, creating new memory structures based upon generalizations about these events, and using these new structures in storing events.

In order to illustrate IPP's memory, I will present three stories taken directly from newspapers, and show how IPP incorporates them into memory. The stories all describe kidnappings that took place in Italy.

(S1) Boston Globe, 5 February 79, Italy
Three gunmen kidnapped a 67 year-old retired industrialist yesterday outside his house near this north Italian town, police said.

(S2) New York Times, 15 April 79, Italy
A building contractor kidnapped here on Jan. 17 was released last night after payment of an undisclosed ransom, the police said.

(S3) New York Times, 25 June 79, Italy
Kidnappers released an Italian shoe manufacturer here today after payment of an undisclosed ransom, the police said.

After a person has read these three stories he would have undoubtedly drawn some conclusions about kidnapping in Italy. The similar nature of the victims - all businessmen of one sort or another - is immediately apparent. In what follows I will show the actual generalizations IPP made upon reading this sequence of stories. The output that follows is from IPP's operation on the three stories above.

```
Story: S1 (2 5 79) ITALY

(THREE GUNMEN KIDNAPPED A 67 YEAR-OLD RETIRED
INDUSTRIALIST YESTERDAY OUTSIDE HIS HOUSE NEAR THIS
NORTH ITALIAN TOWN POLICE SAID)

*** Parsing and incremental memory access ***
```

```
>>> Beginning final memory incorporation ...

Story analysis: EV1 (S-MOP = S-EXTORT)
  HOSTAGES        AGE          OLD
                  OCCUPATION-TYPE
                               RETIRED
                  ROLE         BUSINESSMAN
                  POL-POS      ESTAB
                  GENDER       MALE
  ACTOR           NUMBER       3
  METHODS         SCRIPT       $KIDNAP
  NATION          LOCATION     ITALY

Indexing EV1 as variant of S-EXTORT

>>> Memory incorporation complete
```

Stories are represented in IPP by MOPs (Memory Organization Packets) [5]. MOPs were designed to represent events in memory, and are thus particularly well suited for the purpose of maintaining a permanent episodic memory. S1 is represented by S-EXTORT, a simple MOP (S-MOP) that is intended to capture the common aspects of extortion and $KIDNAP, a script that represents the low-level events of a kidnapping.

IPP's long term memory is organized around S-MOPs, structures that represent generalities among scripts, and the ways in which different stories vary from the S-MOPs. The first step in adding a new story to memory is to extract the various features of each instantiated S-MOP. These features consist of the scripts that are instantiated subordinate to the S-MOP (methods and results, typically), and features from each of the role fillers (such as the actor and victim). The output above shows the features that IPP accumulated about the S-EXTORT S-MOP as it read S1. Since there were no similar events in memory when IPP read this story, its addition to memory simply consisted of indexing it under S-EXTORT using each of the features as an index.

Now consider they way IPP reads S2. The collection of the features of the S-EXTORT occurs during parsing of the story.

```
Story: S2 (4 15 79) ITALY

(A BUILDING CONTRACTOR KIDNAPPED HERE ON JAN 17
WAS RELEASED LAST NIGHT AFTER PAYMENT OF AN
UNDISCLOSED RANSOM THE POLICE SAID)

*** Parsing and incremental memory access ***

>>> Beginning final memory incorporation ...

Story analysis: EV3 (S-MOP = S-EXTORT)
  RESULTS         SCRIPT       SS-GET-RANSOM
                  SCRIPT       SS-RELEASE-HOSTAGES
  HOSTAGES        ROLE         BUSINESSMAN
                  POL-POS      ESTAB
                  GENDER       MALE
  METHODS         SCRIPT       $KIDNAP
  NATION          LOCATION     ITALY

Creating more specific S-EXTORT (SpM1)
 from events EV3 EV1 with features:
  HOSTAGES        ROLE         BUSINESSMAN
                  POL-POS      ESTAB
                  GENDER       MALE
  METHODS         SCRIPT       $KIDNAP
  NATION          LOCATION     ITALY

Reminded of: EV1 (during spec-MOP creation)
[EV1 is from S1]

>>> Memory incorporation complete
```

While reading S2, IPP adds it to long-term memory. However, as it uses the indexing procedure described above, it naturally finds S1, which is already indexed under S-EXTORT in many of the same ways as S2 would be. Thus IPP is reminded of S1, and what is more, since there are such a large number of features in common, IPP makes a tentative generalization that these features normally occur together. Informally speaking, IPP has concluded that the targets of kidnappings in Italy are frequently businessmen.

The new generalization causes the creation of a new node in memory, known as a spec-MOP, to be used to remember events that are examples of the generalization. A spec-MOP is simply a new MOP, equivalent in kind to the S-MOPs that IPP starts with, that is used to embody the generalizations that have been made. The events that went into the making of the generalization are then indexed off of the spec-MOP by any additional features they may have.

The power of creating new memory nodes based on generalizations, and of using episodic memory during parsing is shown below in the processing of S3. Notice that as soon as IPP has instantiated S-EXTORT, after it has read "kidnappers," it collects the features available and checks episodic memory to see if there are any spec-MOPs that might be relevant. Since the story is about a kidnapping in Italy, the spec-MOP just created is appropriate. Furthermore, IPP is able to predict from the spec-MOP that the victim might well be a businessman of some sort. This is exactly the sort of prediction that can be useful in disambiguating words and overcoming unclear syntax during parsing. And it cannot be made without an integrated approach involving the access of memory during parsing.

```
Story: S3 (6 25 79) ITALY

(KIDNAPPERS RELEASED AN ITALIAN SHOE MANUFACTURER
HERE TODAY AFTER PAYMENT OF AN UNDISCLOSED RANSOM
THE POLICE SAID)

Processing:

KIDNAPPERS     : Interesting token - KIDNAPPERS

Instantiated $KIDNAP -- S-EXTORT
```

```
>>> Beginning memory update ...

New features: EV5 (S-EXTORT)
METHODS         SCRIPT      $KIDNAP
NATION          LOCATION    ITALY

Best existing S-MOP(s) -- SpM1
[the spec-MOP just created]

Predicted features (from SpM1)
HOSTAGES        GENDER      MALE
                POL-POS     ESTAB
                ROLE        BUSINESSMAN

>>> Memory update complete

.. [rest of the parsing process]

>>> Beginning final memory incorporation ...

Story analysis: EV5 (S-MOP = S-EXTORT)
  RESULTS       SCRIPT      SS-GET-RANSOM
  HOSTAGES      NATIONALITY ITALY
                ROLE        BUSINESSMAN
                POL-POS     ESTAB
                GENDER      MALE
  RESULTS       SCRIPT      SS-RELEASE-HOSTAGES
  METHODS       SCRIPT      $KIDNAP
  NATION        LOCATION    ITALY

Creating more specific S-EXTORT (SpM2) than SpM1
 from events EV5 EV3 with features:
  RESULTS       SCRIPT      SS-GET-RANSOM
                SCRIPT      SS-RELEASE-HOSTAGES
  HOSTAGES      ROLE        BUSINESSMAN
                POL-POS     ESTAB
                GENDER      MALE
  METHODS       SCRIPT      $KIDNAP
  NATION        LOCATION    ITALY

Reminded of: EV3 (during spec-MOP creation)
[EV3 is from S2]

>>> Memory incorporation complete
```

As it finishes reading S3, IPP completes its addition of the new event to memory. The processing is basically the same as that we saw for S2, but instead of beginning the updating process with the basic S-EXTORT S-MOP, IPP has already decided that the new story should be considered as a variant of the first spec-MOP created. IPP is able to create another new spec-MOP from S3, this one including all the features of the first spec-MOP, plus the RELEASE-HOSTAGES and GET-RANSOM results of S-EXTORT. IPP has noticed that these are frequent results of kidnappings of businessmen in Italy.

So after having read these three stories, IPP has begun to create a model of kidnapping in Italy. This is the sort of behavior displayed by interested human readers. It is also the kind of behavior that cannot be captured by an understanding system that does not involve long-term memory.

CONCLUSION

The inclusion of long-term memory as a part of IPP has been a key factor in allowing the program to be a powerful, robust understanding system. (To date it has a vocabulary of about 3200 words and has processed over 500 stories from newspapers and the UPI news wire, producing an accurate representation for 60 - 70%.) Furthermore, the addition of memory makes possible the inclusion of many familiar language-related phenomena, such as reminding and generalization, into a model of the understanding.

As shown in the examples in this paper, memory updating can be implemented as a natural part of language processing. In fact without it, the ability of a program to fully understand a story is doubtful, since it cannot improve its knowledge of the world, and adapt its processing to fit the knowledge it has obtained - important measures of the human understanding ability. A program without episodic memory can never conclude that Italian kidnappings are often against businessmen, or any of a myriad of other generalizations that people make all the time. And that is a crucial part of understanding.

REFERENCES

[1] Charniak, E. On the use of framed knowledge in language comprehension. Research Report #137, Department of Computer Science, Yale University, 1978.

[2] Cullingford, R. Script application: computer understanding of newspaper stories. Research Report #116, Department of Computer Science, Yale University, 1978

[3] DeJong, G. F. (1979) Skimming stories in real time: An experiment in integrated understanding. Research Report #158, Department of Computer Science, Yale University.

[4] Lebowitz, M. Reading with a purpose. In Proceedings of 17th Annual Meeting of the Association of Computational Linguistics, San Diego, CA, 1979.

[5] Schank, R. C. Reminding and memory organization: An introduction to MOPs. Research Report #170, Department of Computer Science, Yale University, 1979.

[6] Schank, R. C and Abelson, R. P. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1977.

[7] Wilensky, R. Understanding Goal-Based Stories. Research Report #140, Department of Computer Science, Yale University, 1978.