Putting Pieces Together: Understanding Patent Abstracts

Michael Lebowitz

January, 1984

CUCS-98-84

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

Putting Pieces Together: Understanding Patent Abstracts[1]

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

One aspect of the development of RESEARCHER [Lebowitz 83a], an intelligent information system that reads, remembers and learns from patent abstracts, is the use of strongly semantic-based text understanding methods. We show in this paper how patent abstracts can be processed by using only very simple syntactic rules to identify "pieces" of the ultimate representation and then "putting the pieces together". An example of RESEARCHER processing a sample abstract is shown.

1 Introduction

Natural language text comes in many different forms. It seems likely that different kinds of text are best handled with different kinds of processing, at least for working computer systems, and probably by human understanders. In [Lebowitz 83b] we discussed one experiment in strongly semantic-based processing for understanding news stories. Here we describe another such experiment in the context of a computer system, RESEARCHER [Lebowitz 83a], that reads and learns from patent abstracts. We show how such texts can be processed by using only very simple syntactic rules to identify "pieces" of the ultimate representation and then "putting the pieces together".

TEXT1 shows a patent abstract typical of the sort read by RESEARCHER. We are concerned mostly with abstracts that describe the physical structures of objects. The goal of the text interpretation phase of RESEARCHER is to build up descriptions of objects, including the physical relations between various sub-parts of the objects, using a canonical, frame-based representation scheme [Wasserman and Lebowitz 83].

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

TEXT1 - A hard fixed head disc drive assembly having a rotating record disc with a transducer cooperating with the surface of the disc. The transducer is mounted on a carriage which has three spaced, grooved bearings, two of which are received by a fixed cylindrical track, the third bearing engages a spring-loaded cylindrical track which urges said first two bearings against said fixed track, whereby the carriage is centered on said tracks for movement therealong radially of said disc surface.

There are several important points to notice about TEXT1 for text processing purposes. First of all, in traditional terms, the syntax of the abstract is very strange. For example, the first "sentence" has no main verb. Many traditional grammars could not be easily applied to this domain. Furthermore, frequently, very different syntactic structures function quite similarly in patent abstracts. For example, the phrases "a transducer cooperating with the surface of the disk" and "the third bearing engages a spring-loaded cylindrical track" describe very similar physical relations, but use different linguistic structures. While preliminary identification of the syntactic structure might aid in the building of a conceptual representation, patent abstracts seem like an ideal domain to test strongly semantic-based methods that build a conceptual representation directly from the text.

TEXT2 shows TEXT1 segmented in a manner that motivates RESEARCHER's text processing techniques. We see that this text, and most other patent abstracts that provide physical descriptions, can be broken into segments of two types -- those that describe physical objects (which we refer to as *memettes*), shown in italics in TEXT2, and those that relate various memettes to each other. The memette-describing segments are usually (though not always) simple noun phrases, but the relational segments often take different forms, including verbs and prepositions. The key point is the functionality of the relational segments is largely independent of their syntactic form, so we can process them solely on the function they serve, ignoring structural complexities.

TEXT2 - (*A hard fixed head disc drive assembly*) (having) (*a rotating record disc*) (with) (*a transducer*) (cooperating with) (*the surface*) (of) (*the disc*). (*The transducer*) (is mounted on) (*a carriage*) (which has) (*three spaced, grooved bearings*), (*two*) (of which) (are received by) (*a fixed cylindrical track*), (*the third bearing*) (engages) (*a spring-loaded cylindrical track*) (which urges) (*said first two bearings*) (against) (*said fixed track*), (whereby) (*the carriage*) (is centered on) (*said tracks*) (for movement therealong radially of) (*said disc surface*).

The analysis shown in TEXT2 leads directly to RESEARCHER's text interpretation methods. The RESEARCHER interpretation phase consists largely of two sub-phases -- memette identification and memette relation, or "identifying the pieces" and "putting the pieces together". We will look at each of these sub-phases independently, after looking at the top-level structure of RESEARCHER's text interpreter. We will indicate how each of these sub-phases must ultimately be able to access the system's long term memory of the objects it is reading about. Finally, we will show how RESEARCHER processes TEXT1.

2 Text Processing Overview

The text interpretation methods used in RESEARCHER are based on the memory-based understanding techniques designed for IPP [Lebowitz 83b]. This processing involves a top-down goal of recognizing conceptual structures integrated with simple, bottom-up syntactic techniques. Since patents are not focused on events, as are the news stories IPP processed, the action-based methods of IPP (or other conceptual analyzers, e.g., [Birnbaum and Selfridge 81]) must be extensively modified in a manner consistent with the analysis shown in TEXT2.

Processing in RESEARCHER uses a functional classification of words that concentrates on those that refer to physical objects and that describe physical relations between such objects. Such words are known as Memory Pointers (MPs) and Relation Words (RWs) (including words that indicate assembly/component relations). RESEARCHER does careful processing of MP phrases (usually noun phrases) to identify memettes, modifications to memettes, and reference to previous mentions of memettes. This processing is interspersed with the application of RWs to create relations among memettes.

In broad terms, the structure of our processing is similar to the cascaded ATN methodology [Woods 80; Bobrow and Webber 80], where syntactic grammars frequently hand off syntactic components to a semantic analyzer that builds semantic structures and eliminates impossible constructs. However, we use only a small number of different syntactic constructs, eliminating the need for a formal syntactic grammar by focusing on the role of words in the conceptual representation. Furthermore, while the cascaded ATN methodology views the understanding process as a syntactic processor giving what it finds to the semantic analyzer, we look on the process as being primarily a conceptual analysis that requests linguistic structures when needed (much as in [DeJong 79]).

3 Finding the Pieces

Since the descriptions read by RESEARCHER focus on how objects relate to each other, the identification of objects is obviously crucial. "Finding the pieces" consists primarily of bottom-up recognition of simple noun phrases followed by a reference component that determines whether the object being mentioned has a previous reference in the text. No explicit syntactic analysis of complex noun phrases is done.

The noun phrase recognition process involves the same "save and skip" strategy used in [Lebowitz 83b]. Using a one-word look-ahead process, RESEARCHER saves noun phrase words in a stack until the head MP (usually head noun) is found. Then the words in the stack are popped off and used to modify the memette indicated by the head noun.

In the current version of RESEARCHER, we concentrate more on delimiting the noun phrases accurately and carry out the internal analysis of these phrases using a few simple heuristics. Doing such analysis is one place where information from memory will ultimately be needed. For example, in the first noun phrase of TEXT1, "A hard fixed head disc drive assembly", there is no way of knowing whether "hard" modifies "fixed head", "disc", "disc drive" or "assembly" without using information about the structure of disc drives. We expect to have RESEARCHER automatically learn this information [Lebowitz 83a] and have it available for text interpretation.

The final aspect to "finding the pieces" involves checking for previous reference in the text. Here we are able to take advantage of some of the arcane nature of patent abstracts. A very strict formalism is used to identify previous references, involving the word "said" and repetition of identifying modifiers. Without such formalism, the process would be very complicated, as abstracts frequently refer to many very similar objects. As it is, we can use a fairly simple, procedural reference process that avoids many techniques needed for other sorts of text. The process is complicated somewhat by the introduction of phrases referring to subgroups of objects mentioned earlier (e.g., "three bearings, two of which ... the third bearing ..." in TEXT1).

4 Connecting the Pieces

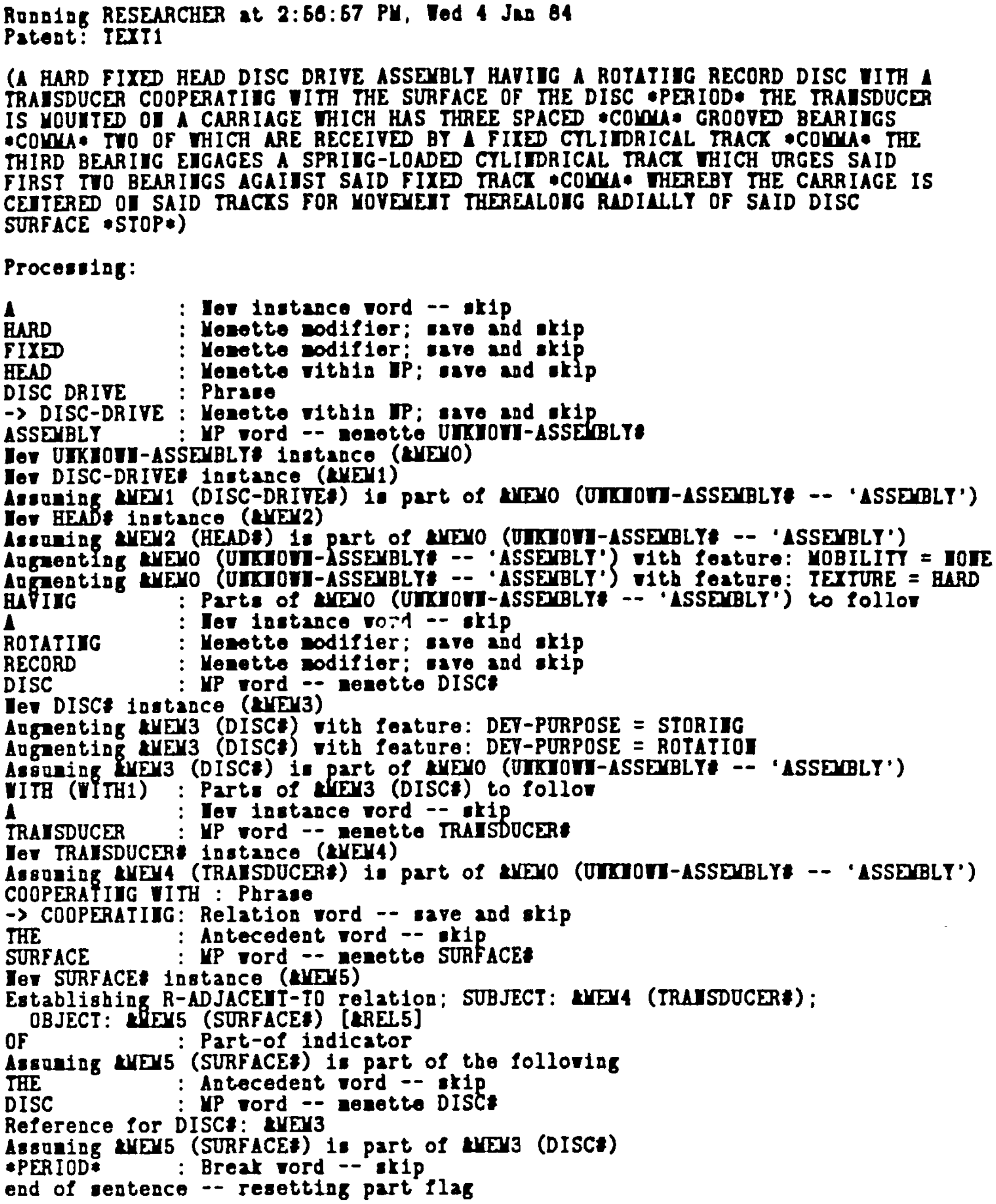
The second major sub-phase to RESEARCHER text processing involves putting together the pieces identified. This process occurs as soon as the objects involved are found. By and large, there are two different kinds of relations found that tie objects together -- assembly/component relations and physical (or functional) relations between memettes. The basic RESEARCHER strategy for each is the same -- maintain information from the relational segments of the text in short term memory and then, when the following memette is identified, determine how the appropriate pieces relate to each other. This process, which is largely independent of the form of the relational text segments, immediately builds up a conceptual representation for later use.

Particular care in this domain has to be given to phrases of the sort "X relation1 Y relation2 Z". It is frequently hard to tell if relation2 relates Z to Y or X. So, in "A hard fixed head disc drive assembly having a rotating record disc with a transducer cooperating with the surface of the disc", it is not apparent from the text whether the transducer is "with" the "rotating record disc" or the "hard fixed head disc drive assembly." This problem is especially crucial in the patent domain. We currently use a set of focus heuristics including some related to [Grosz 77; Sidner 79] and others based on the various relations involved. However, we believe that this is only part of the solution (perhaps a small part), and must be extensively augmented with memory access, in a manner that we are currently implementing. So, in this example, the system should check its knowledge of disc drives (learned from previous examples) and see whether there are indications as to where the transducer belongs.

5 A RESEARCHER Example

We will conclude this brief presentation of RESEARCHER's text interpretation methods by showing some of the processing of TEXT1. Figure 1 shows the processing of the first sentence.

The main point illustrated by Figure 1 is how RESEARCHER text processing consists of memettes being identified and then related together as indicated by the relation words. For example, "a hard fixed head disc drive assembly" and "a rotating record disc" are each identified using a save and skip strategy and then related together based on the relation word "having", making the disc a part of the assembly. (Actually, *instantiations* of the abstract memettes are related, &MEM0 and &MEM3 in this case.) Also worth noting is RESEARCHER's use of a phrasal lexicon for phrases such as "disc drive" and "cooperating with". This simple technique eliminates considerable unneeded processing for phrases that have a meaning not quite equal to the sum of their components. Figure 1 also shows an example of RESEARCHER performing a reference (if not a difficult one), noting that the final disc mentioned is that same as the one mentioned earlier, &MEM3.



```
Running RESEARCHER at 2:56:57 PM, Wed 4 Jan 84
Patent: TEXT1

(A HARD FIXED HEAD DISC DRIVE ASSEMBLY HAVING A ROTATING RECORD DISC WITH A
TRANSDUCER COOPERATING WITH THE SURFACE OF THE DISC *PERIOD* THE TRANSDUCER
IS MOUNTED ON A CARRIAGE WHICH HAS THREE SPACED *COMMA* GROOVED BEARINGS
*COMMA* TWO OF WHICH ARE RECEIVED BY A FIXED CYLINDRICAL TRACK *COMMA* THE
THIRD BEARING ENGAGES A SPRING-LOADED CYLINDRICAL TRACK WHICH URGES SAID
FIRST TWO BEARINGS AGAINST SAID FIXED TRACK *COMMA* WHEREBY THE CARRIAGE IS
CENTERED ON SAID TRACKS FOR MOVEMENT THEREALONG RADIALLY OF SAID DISC
SURFACE *STOP*)

Processing:

A              : New instance word -- skip
HARD           : Memette modifier; save and skip
FIXED          : Memette modifier; save and skip
HEAD           : Memette within NP; save and skip
DISC DRIVE     : Phrase
-> DISC-DRIVE : Memette within NP; save and skip
ASSEMBLY       : MP word -- memette UNKNOWN-ASSEMBLY#
New UNKNOWN-ASSEMBLY# instance (&MEM0)
New DISC-DRIVE# instance (&MEM1)
Assuming &MEM1 (DISC-DRIVE#) is part of &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY')
New HEAD# instance (&MEM2)
Assuming &MEM2 (HEAD#) is part of &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY')
Augmenting &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY') with feature: MOBILITY = NONE
Augmenting &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY') with feature: TEXTURE = HARD
HAVING         : Parts of &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY') to follow
A              : New instance word -- skip
ROTATING       : Memette modifier; save and skip
RECORD         : Memette modifier; save and skip
DISC           : MP word -- memette DISC#
New DISC# instance (&MEM3)
Augmenting &MEM3 (DISC#) with feature: DEV-PURPOSE = STORING
Augmenting &MEM3 (DISC#) with feature: DEV-PURPOSE = ROTATION
Assuming &MEM3 (DISC#) is part of &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY')
WITH (WITH1)   : Parts of &MEM3 (DISC#) to follow
A              : New instance word -- skip
TRANSDUCER     : MP word -- memette TRANSDUCER#
New TRANSDUCER# instance (&MEM4)
Assuming &MEM4 (TRANSDUCER#) is part of &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY')
COOPERATING WITH : Phrase
-> COOPERATING: Relation word -- save and skip
THE            : Antecedent word -- skip
SURFACE        : MP word -- memette SURFACE#
New SURFACE# instance (&MEM5)
Establishing R-ADJACENT-TO relation; SUBJECT: &MEM4 (TRANSDUCER#);
  OBJECT: &MEM5 (SURFACE#) [&REL5]
OF             : Part-of indicator
Assuming &MEM5 (SURFACE#) is part of the following
THE            : Antecedent word -- skip
DISC           : MP word -- memette DISC#
Reference for DISC#: &MEM3
Assuming &MEM5 (SURFACE#) is part of &MEM3 (DISC#)
*PERIOD*       : Break word -- skip
end of sentence -- resetting part flag
```

Figure 1: Processing TEXT1

Figure 2 shows the final representation constructed by RESEARCHER after reading all of TEXT1. It consists of a set of memettes identified, indications of which memettes are parts of others, and a list of relations between memettes. The relations prefixed with R- are physical and those beginning with P- are functional (purposive). There is also a single "meta-relation" that indicates a causal relation between its component relations.

The representation in Figure 2 captures all the information from TEXT1 that is needed for the learning aspects of RESEARCHER. It was acquired using the "putting pieces together" strategy described in this paper, without any further pure linguistic processing.

```
Text Representation:

** ACTIVE INSTANCES **
&MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY') [Mods: TEXTURE/HARD MOBILITY/NONE]
   Components: &MEM1 &MEM2 &MEM3 &MEM4
&MEM1 (DISC-DRIVE#)
&MEM2 (HEAD#)
&MEM3 (DISC#) [Mods: DEV-PURPOSE/ROTATION DEV-PURPOSE/STORING]
   Components: &MEM5
&MEM4 (TRANSDUCER#)
&MEM5 (SURFACE#)
&MEM6 (CARRIAGE#)
   Components: &MEM7
&MEM7 (BEARING#) [Mods: NUMBER/3 DISTANCE/SEPARATE TEXTURE/INCISED]
   Components: &MEM8 &MEM10
&MEM8 (BEARING#) [Mods: NUMBER/2 ORDINAL/1]
&MEM9 (TRACK#) [Mods: MOBILITY/NONE SHAPE/CYLINDRICAL]
&MEM10 (BEARING#) [Mods: ORDINAL/3]
&MEM11 (TRACK#) [Mods: TENSION/SPRING SHAPE/CYLINDRICAL]

A list of relations:

         Subject:                 Relation:           Object:
[&REL5]  &MEM4  (TRANSDUCER#)    {R-ADJACENT-TO}     &MEM5  (SURFACE#)
[&REL6]  &MEM6  (CARRIAGE#)      {P-SUPPORTS}        &MEM4  (TRANSDUCER#)
[&REL7]  &MEM9  (TRACK#)         {P-RECEIVES}        &MEM8  (BEARING#)
[&REL8]  &MEM10 (BEARING#)       {P-ENGAGES}         &MEM11 (TRACK#)
[&REL9]  &MEM11 (TRACK#)         {P-IMPELS}          &MEM8  (BEARING#)
[&REL10] &MEM8  (BEARING#)       {R-ADJACENT-TO}     &MEM9  (TRACK#)
[&REL11] &MEM11 (TRACK#)         {R-SURROUNDED-BY}   &MEM6  (CARRIAGE#)
[&REL12] &MEM11 (TRACK#)         {R-ALONG}           &MEM5  (SURFACE#)
                                 ORIENTATION/RADIAL

A list of meta-relations:

Subject:            Meta-rel:           Object:
&REL10              {M-CAUSES}          &REL11
```

Figure 2: RESEARCHER Representation of TEXT1

6 Conclusions

Language takes many forms. It seems inappropriate to use a single text processing methodology for every form of language. We have shown here a strongly semantic-oriented strategy for processing a class of rather complex texts. It depends on having considerable knowledge of the domain, but such knowledge is clearly needed in any case for full understanding. To date, we have run RESEARCHER on a number of patent abstracts as complex as TEXT1, with about 20 being fully processed with good accuracy. Most of the preparation of new texts involves only the addition of very simple word definitions. (This domain has a rather large vocabulary). With the addition of the memory access methods mentioned here, which are currently being implemented, we believe the understanding power of RESEARCHER will increase further.

While we certainly feel that the semantic-based "putting pieces together" strategy of RESEARCHER will have applicability far beyond patent abstracts, the success of this method illustrates the importance of selecting a processing method appropriate to the domain in building systems to handle large numbers of complex texts. We feel this is an important lesson to learn.

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.

[Bobrow and Webber 80] Bobrow, R. J. and Webber, B. L. PSI-KLONE - Parsing and semantic interpretation in the BBN Natural Language Understanding System. Proceedings of the CSCSI/CSEIO Annual Conference, 1980.

[DeJong 79] DeJong, G. F. "Prediction and substantiation: A new approach to natural language processing." *Cognitive Science 3*, 1979, pp. 251 - 273.

[Grosz 77] Grosz, B. J. Representation and use of focus in a system for understanding dialogs. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, International Joint Conference on Artificial Intelligence, Cambridge, MA, 1977.

[Lebowitz 83a] Lebowitz, M. RESEARCHER: An overview. Proceedings of the Third National Conference on Artificial Intelligence, Washington, DC, 1983.

[Lebowitz 83b] Lebowitz, M. "Memory-based parsing." *Artificial Intelligence 21*, 4, 1983, pp. 363 - 404.

[Sidner 79] Sidner, C. L. *A Computational model of co-reference comprehension in English*. Ph.D. Thesis, Massachussets Institute of Technology, Cambridge, MA, 1979.

[Wasserman and Lebowitz 83] Wasserman, K. and Lebowitz, M. "Representing complex physical objects." *Cognition and Brain Theory 6*, 3, 1983, pp. 333-352.

[Woods 80] Woods, W. A. "Cascaded ATN grammars." *American Journal of Computational Linguistics 6*, 1, 1980.