CUCS-101-84

Ill-Formed Text and Conceptual Processing

Michael Lebowitz

February, 1984

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

Ill-Formed Text and Conceptual Processing[1]

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

Abstract

In this paper, we discuss the problem of ill-formed (or incorrectly processed) text in the context of conceptual analysis text processing systems. We show that *syntactically* ill-formed text is not a major problem for such systems. *Conceptually* ill-formed text and conceptually ill-formed representations of text do cause interesting problems. We define conceptual ill-formedness and then present ideas for how it can handled in the context of two text processing systems, IPP and RESEARCHER.

1 Introduction

Natural language text can be ill-formed in many different ways. Much of the work on ill-formed text has concentrated on syntactic problems ([Weischedel and Black 80; Kwasney and Sondheimer 81], among others). Such work has looked at syntactically anomalous input and input with syntax for which a given system is not prepared. However, syntactic ill-formedness is not the whole problem. If fact, systems that concentrate on direct *conceptual analysis* of text must approach ill-formedness from a different perspective. This is particularly the case for text processing systems that take large numbers of carefully written texts (news stories and patent abstracts in our case), and analyze them.

In this paper, we will consider the issue of ill-formed input in the context of conceptual analysis. We will discuss two major issues -- 1) the relation of conceptual analysis to *syntactically ill-formed* input, and 2) defining *conceptually ill-formed* input and representations (either in absolute terms, or in relation to the

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

capabilities of a computer system) and how it might be dealt with by a text processing system.

As sources of sample texts and text processing examples, we will make use of two computer systems that perform textual analysis. IPP [Lebowitz 80; Lebowitz 83a; Lebowitz 83b] is a program that was developed at Yale University to read, remember and learn from news stories about international terrorism. It was used primarily to study problems in learning from real-world input, but also involved conceptually-based text understanding. It will provide us here with a corpus of text examples (from newspapers the UPI newswire), along with selected stories that IPP was unable to process. RESEARCHER [Lebowitz 83c], currently under development at Columbia University, also reads, remembers and learns from text, in this case technical text in the form of patent abstracts. RESEARCHER will provide examples of much more complicated input.

2 Conceptual Understanding Methods

To provide a context for our discussion of how conceptual analysis systems relate to ill-formed text, we will describe here the basics of the conceptual understanding techniques used by our systems. In particular, we will describe the methods used in RESEARCHER which are, at least in an abstract sense, similar to those used by IPP (which the RESEARCHER methods were based on). IPP's understanding have been described in detail in [Lebowitz 80; Lebowitz 83b].

Systems using conceptual analysis methods build meaning representations directly from text, constantly making use of predictions about what will come next. Only the minimum amount of syntactic information is used, and its use is usually embedded procedurally in conceptual processing. Research into conceptual understanding has included [Birnbaum and Selfridge 81; Dyer 83; DeJong 79; Hayes and Mouradian 81; Lebowitz 83b; Lebowitz 83c; Rieger 78; Riesbeck 75; Riesbeck and Schank 76; Schank, et al. 80; Small 80; Wilks 73]. Crucial to conceptual analysis is that explicit syntactic processing is not done prior to building a meaning representation, nor is an explicit syntactic representation of the text constructed, as in [Harris 78; Kaplan 77; Kaplan 75; Marcus 80; Winograd 72; Woods 70; Woods and Kaplan 72] and others. The virtues of conceptual understanding for many

tasks and many domains, as well as its cognitive plausibility, has been discussed elsewhere ([Schank and Birnbaum 82; Lebowitz 83b], for example).

RESEARCHER processes texts by using very simple syntactic rules to identify "pieces" of the ultimate representation and then "puts the pieces together" using a combination of syntactic and conceptual heuristics. EX1 shows a patent abstract typical of those 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].

EX1 - P41; U.S. Patent Abstract #4323939

> 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.

For text processing purposes, there are several important points to notice about EX1. 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. (We will return to this point later). 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.

EX2 shows EX1 segmented in a manner that motivates RESEARCHER's text processing techniques.

EX2 - (*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*).

EX2, 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 EX3, 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 take many 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.

The analysis shown in EX2 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".

Processing in RESEARCHER uses a functional classification of words that concentrates on those that refer to physical objects and those 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]).

"Finding the pieces", i.e., identifying the objects described in a text, 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 representation of complex noun phrases is done, although some fairly strong syntactic rules about the construction of simple noun groups is used.

The noun phrase recognition process involves the same "save and skip" strategy described 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.

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 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 text processing strategy for

each is the same (although they are treated differently during generalization) -- 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. Determining which pieces to relate often involves complex semantic tests which we will not discuss here.

We will conclude this brief presentation of RESEARCHER's text interpretation methods by showing some of the processing of EX1. 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.[2] (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". 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.

Figure 2 shows the final representation constructed by RESEARCHER after reading all of EX1. 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.

---

[2]Some relations are simply ambiguous, without real-world knowledge, and *must* be understood by using memory. one such example here is whether the "transducer" is part of the "disc" or the "assembly". The version of RESEARCHER shown here uses a simple heuristic, but [Lebowitz 84] describes a more accurate approach.

```
Running RESEARCHER at 2:56:57 PM, Wed 4 Jan 84
Patent: P41

(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        : NP 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            : NP 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 &MEM0 (UNKNOWN-ASSEMBLY# -- 'ASSEMBLY') to follow
A               : New instance word -- skip
TRANSDUCER      : NP 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         : NP 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            : NP 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: RESEARCHER Processing EX1

The representation in Figure 2 captures all the information from EX1 that is needed for the learning aspects of RESEARCHER. It was acquired using the "putting pieces together" strategy, 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's Representation of EX1

3 Conceptual Analysis and Syntactic Ill-Formedness

3.1 Is syntactic ill-formedness a problem?

The frequency of syntactically ill-formed input surely varies in different settings. It is not coincidence that most of the work dealing with ill-formed input has involved interactive systems, particularly database front ends. In such systems, users normally do not extensively edit their input, and hence the chance of ill-formed input is relatively high. On the other hand, in the construction of systems that read more carefully written text, such as that we have worked with, news stories and patent abstracts, are less likely to encounter syntactically ill-formed input. The introduction of mechanisms to explicitly recognize and correct such problems may, therefore, not be warranted (particularly, since, as we will see below, the fact that our systems do not do explicit syntactic analysis makes the detection of syntactic irregularity difficult).

To test the hypothesis that syntactic ill-formedness is rare in written text, we looked at the corpora of texts compiled for use with IPP and RESEARCHER. These comprise roughly 675 paragraph-long terrorism stories taken directly from newspapers and the UPI newswire and about 100 United States patent abstracts, which were also paragraph length, but rather longer and more complicated than the news stories. The texts in each case fit with the hypothesis. Virtually no *obviously* syntactically ill-formed input was found. Not surprisingly, in some cases the text varied from "grammar book" English, but a human reader would hardly notice any problems. Reasonable syntactic rules would probably handle all the texts, although the rules would have to change from domain to domain.

The patent abstracts used for RESEARCHER did include some texts that were clearly ill-formed in a traditional sense. For example, EX1 lacks certain syntactic niceties (like a main verb). On the other hand the grammatical constructions used seem to be quite regular, and, again, reasonable syntactic rules could probably be devised for these texts.

So, it would seem that syntactic ill-formedness is not a major issue for text processing systems (although, of course, if we do information retrieval using natural language queries, or process less polished input such as school papers or transcripts of conversations, we will run into all the problems that database front-end research has encountered). This does, however raise the question of why we do not use prior syntactic analysis in our understanding systems. After all, if there is little or no non-syntactic input, why not? We will address this issue after looking at how we do deal with syntactically ill-formed input, should it arise.

3.2 How does conceptual analysis deal with syntactic ill-formedness?

One of the advantages that has long been claimed for conceptual analysis of the kind described above, i.e., without doing explicit syntactic analysis, is that it can automatically deal with syntactically ill-formed input. It is instructive to see why this is the case.

Consider the following example taken from [Kwasney and Sondheimer 81]:

EX3 -

Draw a circles.

A conceptual analyzer trying to build a conceptual representation directly from this command, perhaps using a "put the pieces together" strategy, will never notice that anything is wrong with the syntax. It will realize that "draw" indicates a command, "a" introduces a noun group, and "circles" is a head plural nouns. So the command indicates that the system should draw more than one circle.

The situation would be similar for this slightly more subtle example (also from [Kwasney and Sondheimer 81]):

EX4 -

I, along with many other Germans, are concerned about the Russian threat.

Again, a conceptual analyzer would just sweep through the words of the sentence, not noticing the disagreement between subject and verb number. In this case, such behavior seems cognitively correct as well as practical. Many human readers, and virtually all conceptual analyzers, would not notice the ill-formedness. In any case, understanding would not be greatly changed if the verb was corrected. Such lack of perception of ill-formedness likely becomes more pronounced as problems get more subtle and sentences more convoluted.

These examples are typical of the way a conceptual analyzer deals with syntactically ill-formed input -- it doesn't. It just goes about its business and finds the best meaning it can. The prime advantage of such an approach is simply that no extra effort is spent trying to correct syntactic anomalies. Since our methods do not need a separate syntactic representation, the lack of one for a text is not a significant problem for understanding.

On the other hand, there are also disadvantages in this approach. In fact, some disadvantages can be expected by observing that people notice basic syntactic ill-formedness (such as number disagreement), even when they are perfectly able to understand the text. Such phenomena in people almost always carry a corresponding advantage.

It is our feeling that the underlying reason for noticing syntactic ill-formedness is *not* that such recognition indicates that a syntactic processing failure must be corrected before further process can occur. Rather, both in human understanders and AI systems, recognition of syntactic ill-formedness can serve as a heuristic for identifying *understanding* problems. EX3 provides a good example. While our conceptual analysis was perfectly plausible, it is also possible that the human user intended to say "Draw a circle", making a mistake with the noun, not the article. Recognition of the syntactic problem might provide a valuable clue indicating that the understander should probe for further evidence as to the user's meaning.

Notice that there is no need for syntactic analysis to be temporally prior to conceptual analysis to gain this heuristic advantage, nor even really interact very much with the conceptual processing. It would be possible to conduct the syntactic processing in parallel, or augment the conceptual processing to check for at least basic syntactic problems (like agreement). In fact, this later step is taken in many working systems (although they frequently ignore anomalies that they find). In IPP and RESEARCHER, since syntactic ill-formedness was not a major issue, the programs are allowed to simply "fly on by" syntactic irregularities.

As mentioned earlier, there is one more question here -- even though correcting syntactic anomalies may be difficult, we commented above that such problems are rare. So why don't we go ahead and perform syntactic analysis, and bite the bullet when there is a problem. The answer to this question is fourfold: 1) our methods do not gain significantly from having a syntactic parse available, so the added layer of complexity is superfluous; 2) performing conceptual analysis from a syntactic parse can sometimes be more difficult that directly from the text, e.g., extracting the real meaning of a news story from a clause embedded in "sources said", 3) even though natural text may not frequently violate English grammar rules, unless we have very complete syntactic processing rules, our system may often believe there is a syntactic problem, and, most importantly, 4) while text is usually syntactically correct, it is often quite convoluted, and finding the syntactic representation may involve considerable effort. For example, consider EX5.

EX5 - S308; UPI; 11 Nov 79; Iran

> The commander of the paramilitary police in the mountains of Iran's Kurdistan region was killed yesterday by a dissident subordinate trying to hijack a helicopter to Iraq the official Pars news agency reported.

Among other complexities in EX5, notice that it is ambiguous whether the phrase "in the mountains of Iran's Kurdistan region" should be attached to the noun phrase "The commander of the paramilitary police" (where he commanded) or the verb phrase "was killed" (where he was killed). Incorrectly attaching the phrase could be viewed as syntactic ill-formedness, but, in reality, this is a conceptual problem. If we must rely on conceptual analysis to resolve problems such as these, we have lost most of the advantage of having a syntactic representation in the first place.

EX6 illustrates a similar point.

EX6 - S396; UPI; 26 Jan 80; South Africa

> A shootout between police and three black nationalist guerrillas who seized white hostages inside a suburban bank raised fears today that the attack could be the beginning of a guerrilla war against white rule in South Africa.

In EX6, "police and three black nationalist guerrillas who seized white hostages inside a suburban bank" forms a single, complex, noun phrase. But this is largely irrelevant for understanding the story, since we are really interested the two sides and location of the shootout. The syntactic structure gives us little help in finding this information, as there are many other forms this information could take.

Since we found examples like EX5 and EX6 to be the rule, not the exception in the kinds of domains we are concerned with, we have elected to ignore most syntactic problems and concentrate directly on conceptual analysis, including conceptual ill-formedness when relevant.

4 Conceptually Ill-Formed Input

The conceptual analysis process we use with creates new issues regarding ill-formed input. While, as we have seen, syntactically ill-formed input is not a major problem for our systems, we may be confronted with *conceptually ill-formed input*. We will first consider whether input can, in a theoretical sense, be conceptually ill-formed, and then look at the practical problems involved.

It is easy to come up with a theoretical definition of syntactically ill-formed input -- a piece of text that is not accepted by the grammar of the language (this, of course, assumes that such a grammar exists). The conceptual analogue of this definition is not clear. Even if we assume that there is a grammar of all possible concepts -- all the ways that the elements of a representation scheme can be combined, perhaps -- this grammar will accept such a large space of concepts, that it will cover many concepts that we would think of as ill-formed. For example, the meaning of Chomsky's famous, "Colorless green ideas sleep furiously", can certainly be *represented*, and yet we might like to consider it conceptually ill-formed.

The alternative to using a grammar of *possible* meanings is to try and formulate rules, possibly a grammar, of *sensible* meanings. Then, any input that failed to correspond to the rules would be considered ill-formed. But this is a horrendously difficult problem. We are not trying to come up with rules about what is *usual*, since if we did we would reject input that is unusual, and often quite interesting. We have to try and define what is *plausible*. But almost anything we can represent is plausible (unless basic rules of the representation are violated), so we are back to rules so vague that they eliminate very few cases.

Even if we abandon the idea of coming up with a theoretical definition of conceptually ill-formed input and just look at the problem practically, the problem is still very hard. While it is easy to come with rules that handle simple cases, e.g., the actor in a terrorist attack must be human, rules like these are not entirely satisfying. First of all, cases like this come up very rarely in text (although, unfortunately, rather more frequently in misanalyzed text) Secondly, many representations can be interpreted metaphorically so that they make sense (see [Russell 75])

So, we are left with the prospect of developing heuristics that identify representations that are *probably* not correct. Generally, this means we are no longer looking for input that is genuinely conceptually ill-formed, but instead, looking for representations that do not reflect the meaning that the writer had in mind. Of course, since the text is all we have to go by, our rules will have to be heuristic in nature.

While we have not implemented any routines that identify conceptually ill-formed representations, we have considered the problem. In general, there seem to be two broad classes of ill-formed representations built by our systems. We came up with these classes by looking at the 15-20% of the terrorism stories that IPP processed incorrectly, and the patent abstracts that we have been collecting. The first class includes those cases where substantial information is simply left out of the representation (including the null representation). Often problems of this kind are simply due to text outside of our representation scheme, but sometimes the reasons for the anomaly can be more subtle. The second class includes those representations that violate some gross semantic rule, of the sort mentioned above. Such rules are very domain-specific. A common example from the terrorism domain would be that the actor and victim of an extortion cannot be the same person. Deciding what constitutes a "gross violation" is obviously not trivial. We imagine that this process must involve relating the new input to what is already known in memory [Lebowitz 83a].

Before going on to consider what we might do when conceptual anomalies are found, as well as some examples of such anomalies, it is worth emphasizing two problems with the approach of considering strange representations as ill-formed. In one direction, we may have the problem that we mentioned before -- we may eliminate some unusual, but correct representations. The worst aspect of these cases is that they probably constitute interesting examples. At the other end of the spectrum, we may accept some representations that are plausible, but not what the writer had in mind. Ultimately, we might want to have rules that involve the course of the *processing*, and not just the final result. It is also worth noting that these two problems can arise in detecting syntactic ill-formedness. The first problem is probably less important, since syntactic ill-formedness is better defined than the conceptual variety, but the second problem, accepting parses that are grammatical, but not what was intended, is quite likely to occur.

4.1 Examples of conceptually ill-formed representations

In this section, we will look at several examples of ill-formed representations from the systems we have worked on. Note that each of these examples is only ill-formed in a "practical" sense, i.e., the system could not understand it. A search of our corpus of examples revealed none that were theoretically conceptually ill-formed. While there were certainly many examples that strike a human reader as odd, it was always possible to determine a plausible meaning. Also, we have selected examples that are on the borderline of anomalous, as these tend to be the most productive to study (although not always the most amusing).

Most of the stories where IPP left a piece of information from the text out of the story representation involved concepts not in our representation scheme (novel terrorist demands were a particular problem). This was also the case for virtually all the cases where IPP came up with no representation at all for a story. However, there were some more interesting cases where IPP just missed a piece of the story. EX7 is typical.

EX7 - S538; UPI; 15 May 80; United States

A heavily armed gunman took a teen-age hostage and attempted to hijack an old flying boat to Capetown South Africa today.

Though it may seem mundane enough, the phrasing "took a teen-age hostage" caused IPP considerable trouble. This is because the word "hostage" is playing a dual semantic role in EX7. It serves both to confirm the action in the story, extortion through the taking of a hostage, and to identify the person taken hostage. This is true since in IPP's understanding scheme, the main verb of the sentence, "took", cannot be used by itself to identify the action, since it is so ambiguous. The conceptual analysis process must look for confirmation of the action, which comes from the word "hostage". This method will work fine for the more common constructions, "took a teen-age boy hostage" or "held captive a teen-age hostage". However, in this case, it causes the program to miss the noun role of "hostage" (Notice that in "took a teen-age boy hostage", "hostage" not playing a noun role, conceptually.)

Figure 3 shows the representation IPP constructed for EX7.

```
Story: S538 (5 15 80) UNITED-STATES

(A HEAVILY ARMED GUNMAN TOOK A TEEN-AGE HOSTAGE AND ATTEMPTED TO HIJACK AN
OLD FLYING BOAT TO CAPETOWN SOUTH AFRICA TODAY)

Story Representation:

** MAIN EVENT **
EV24 =
 MEM-NAME    S-EXTORT
 ACTOR       HEAVILY ARMED GUNMAN
 METHODS
    EV25 =
     MEM-NAME    $HIJACK
     ACTOR       HEAVILY ARMED GUNMAN
     VEHICLE     OLD BOAT
     TO          CAPETOWN *SOUTH-AFRICA*
     OUTCOME     *FAIL*
 TIME        TODAY

2388 msec CPU (0 msec GC), 3000 msec clock, 3929 conses
```

Figure 3: Dropped Phrase in IPP Processing

IPP gets most of the conceptual representation of EX7 correct. It identifies the main action (a failed extortion by hijacking, the failure inferred from "attempted"), the actor (a "heavily armed gunman", actually represented in more detail internally), and the vehicle hijacked (a boat). However, it misses the teen-age hostage.

The point here is not that it would be difficult to modify IPP so that it processed EX7 correctly. In fact, that would be rather simple within the IPP framework. However, the conceptual variety of text will lead to some cases missed by any understanding system, and so a robust system must be able to deal with such problems.

This example does illustrate part of our strategy for dealing will conceptually incorrect representations (in fact, the main part, so far). It has always been a goal for our systems to represent as much of a text as possible correctly, even when there are problems with other parts. Thus, even when a system like IPP can't understand *all* of a text, it may understand *enough* to be able to carry out all or part of its main task. For example, after processing EX7, IPP, in its learning role, might be able to determine that hijackings of boats usually fail. Obviously, it

cannot learn as much as if it had totally understood the text. The key here is that everything the system puts in its representation should be correct, even if not complete.

We will consider other possible solutions to conceptually incorrect representations in the next section.

EX8 illustrates the more serious case of conceptually ill-formed representations, where information is not left out of a story representation, it is simply incorrect. This is also a classic example of the kind of story that confuses a conceptual analyzer.

EX8 - S519; UPI; 2 May 80; UNITED-STATES

> A man saying he was setting out to free the American hostages tried to hijack an airliner and fly to Iran but was disarmed early Friday, ending a six-hour siege.

EX8 took place during the period after the takeover of the United States embassy in Teheran. It describes one extortion that attempts to end another. A conceptual analyzer is easily confused by the conjunction of the two extortions. As IPP tries to put together the events such as the hijacking, freeing of hostages, a siege, and disarming the hijacker, based on its stereotypical knowledge of extortion, it gets very confused. Figure 4 shows that representation built for EX8.

Figure 4 shows a perfectly plausible representation of a hijacking of a plane with American passengers by a man who released his hostages and was captured after a siege. Plausible, but wrong, as the hostages referred to were an entirely different group of people than the passengers in the plane. (The fact that the airliner passengers were presumably released, if they were ever actually held, is just coincidence.)

IPP was particularly susceptible to this kind of problem, as it was basically a skimmer that used very little information about surface structure, but other conceptual analyzers would have similar difficulties.

```
Story: S519 (5 2 80) UNITED-STATES

(A MAN SAYING HE WAS SETTING OUT TO FREE THE AMERICAN HOSTAGES TRIED TO
HIJACK AN AIRLINER AND FLY TO IRAN BUT WAS DISARMED EARLY FRIDAY ENDING A
SIX-HOUR SIEGE)

Story Representation:

** MAIN EVENT **
EV20 =
 MEM-NAME     S-EXTORT
 ACTOR        MAN
 HOSTAGES     *USA* HOSTAGES
 SCENES
    EV19 =
     MEM-NAME     SS-RELEASE-HOSTAGES
     ACTOR        MAN
     OBJECT       *USA* HOSTAGES
    EV22 =
     MEM-NAME     SS-CAPTURE
     OBJECT       MAN
    EV23 =
     MEM-NAME     SS-SIEGE
     OBJECT       MAN
     BEFORE       EV22
 METHODS
    EV21 =
     MEM-NAME     $HIJACK
     ACTOR        MAN
     VEHICLE      AIRLINER
     CARRYING     *USA* HOSTAGES
     MODE         *HYPOTHETICAL*
 TIME         FRIDAY

3514 msec CPU (0 msec GC), 5000 msec clock, 5184 conses
```

Figure 4: Confused IPP Output

It is difficult to see any solution for EX8, if we are restricted to examining the final representation for ill-formedness, as there is nothing to indicate that the representation is anomalous. We present this example just to show the worst that can happen. Fortunately, the situation is rarely this bad. It takes a rare confluence of events for mistaken analysis to lead to a plausible representation. Often, as mentioned earlier, there will some sort of gross anomaly that we can hope to detect (with all the caveats mentioned earlier). Even with this story, it is possible that if we read a longer version of the events in question, a detectable anomaly would arise.

The patent abstracts that RESEARCHER deals with are considerable more complex that news stories, and hence the proportion of results that are at least somewhat anomalous is currently greater than for IPP. EX9 shows a typical abstract, which we will use to illustrate several ill-formed concepts in the

RESEARCHER representation. (Notice also that the first "sentence" is non-grammatical, but would probably cause a "practical" grammar no syntactic problems.)

EX9 - P22; U.S. Patent Abstract #3815150

> A disc drive for flexible disc cartridge magnetic recording. The disc cartridge is placed in a holder without touching either the recording head or the disc drive spindle. Then the holder is pivoted generally parallel to the spindle axis to move the disc into engagement with the head and spindle. A special clamp on the holder clamps the disc to the spindle with a floating clamp member which can adapt itself to the axis of rotation of the spindle.

EX9, like most patent abstracts, is both complex and difficult for people to understand. Figure 5, shows the representation that RESEARCHER came up with for EX9. The representation consists of a list of objects, including objects specified as parts of others, along with a list of relations, physical and purposive, between objects. There are, not surprisingly, a number of problems in this representation, as RESEARCHER was not specially prepared for this story, and the example was selected for this paper as one likely to confuse the system. We will look at three of the problems, though the reader can no doubt find more.

The first problem will will look at is that in representing "flexible disc cartridge magnetic recording", RESEARCHER loses the information from "magnetic recording", as the system has never been prepared for modifiers that follow an object. ("Flexible magnetic recording disc cartridge" would work fine.) This problem is similar to the "teen-age hostage" IPP example in that it involves missing information. As with that example, the problem is due to a surface construction that the system is not prepared for, and, as an isolated example, would be easy to correct. It also suggests, however, that simple missing information will be a general class of conceptually ill-formed input we will have to detect.

A second problem with the representation in Figure 5 is that it has the "axis" (of the spindle) impelling the "disc". It cannot tell whether the purposive relation designated by "to move" relates the disc with the "holder", the "axis" or the

```
Text Representation:

** ACTIVE INSTANCES **
&MEM0 (DISC-DRIVE#)
   Components: &MEM5
&MEM1 (CASSETTE#) [Mods: RIGIDITY/2]
   Components: &MEM2
&MEM2 (DISC#)
&MEM3 (UNKNOWN-THING# -- 'HOLDER')
&MEM4 (TRANSDUCER#)
&MEM5 (DRIVE-SHAFT#)
   Components: &MEM6
&MEM6 (AXIS#)
&MEM7 (DRIVE-SHAFT#)
   Components: &MEM9 &MEM6
&MEM8 (MOUNTING-MEANS#)
&MEM9 (UNKNOWN-THING# -- 'MEMBER')
   Components: &MEM8

A list of relations:

         Subject:              Relation:          Object:

[&REL5]  &MEM0 (DISC-DRIVE#)   {P-USED-FOR}      &MEM1 (CASSETTE#)
[&REL6]  &MEM0 (DISC-DRIVE#)   {P-WRITES}
[&REL7]  &MEM3 ('HOLDER')      {P-CONTAINS}      &MEM1 (CASSETTE#)
[&REL8]  &MEM4 (TRANSDUCER#)   {P-WRITES}
[&REL9]  &MEM4 (TRANSDUCER#)   {R-CONNECTED-TO}  &MEM3 ('HOLDER')
                               AMOUNT/0
[&REL10] &MEM3 ('HOLDER')      {R-PARALLEL-TO}   &MEM8 (AXIS#)
[&REL11] &MEM6 (AXIS#)         {P-IMPELS}        &MEM2 (DISC#)
[&REL12] &MEM4 (TRANSDUCER#)   {P-ENGAGES}       &MEM2 (DISC#)
[&REL13] &MEM8 (MOUNTING-MEANS#) {R-ON-TOP-OF}   &MEM8 (MOUNTING-MEANS#)
[&REL14]                       {P-ROTATES}       &MEM7 (DRIVE-SHAFT#)
```

Figure 5: RESEARCHER Representation of EX9

"pivoted parallel" relation between them. In this case, there is a subtle clue from the use of the word "to" that indicates that "to move" relates the "pivoted parallel" relation to the disc. However, in many similar cases, there is no such surface indication. As with the "hijacking and Iran hostages" IPP example, there is no gross anomaly in this part of the representation to indicate that something is wrong.

The main point here is that systems that can detect this sort of anomaly will have to have considerable knowledge of its domain, disc drives in this case. Furthermore, while it is possible to come up with static semantic information that will handle any one specific case, general detection of this sort of error will require a broad dynamic memory built up from the texts processed, of the sort discussed in our work [Lebowitz 82a; Lebowitz 83a; Lebowitz 83c] as well as [Schank 82; Kolodner 83; Reiser et al. 83]. Once again, though, in applying this kind of information we will have to take great care not to classify merely non-stereotypical

input as anomalous. (This is, of course, particularly important when dealing with patents, which are supposed to be unique in some way.)

The final example of conceptually ill-formed input we will use from Figure 5 involves the next-to-last relation that shows the "mounting means" on top of itself. The reasons why RESEARCHER came up with this representation are rather arcane. This is the sort of example we can profitably hope to be able to detect algorithmically. It is a violation of a very basic rule of the representation scheme (or rule of the domain, perhaps), that an object cannot be on top of itself. This sort of error we might, in the short term, hope to detect and deal with using some of the methods described in the next section.

4.2 Proposals for dealing with conceptually incorrect representations

Having seen that text does get incorrectly represented conceptually, we need to consider how to handle these cases. While we have not implemented any measures specifically aimed at this problem for our understanding systems, we have considered the issue. ([Hayes and Carbonell 83; Webber and Mays 83] have also looked at how to handle certain classes of conceptually incorrect text). Before making our proposals, there are two preliminaries to deal with -- just when do we assume that we have conceptually incorrect input, and why whatever correction techniques we use should not be used all the time.

We have mentioned throughout this paper several kinds of anomalous representations that we can hope to detect, as well as the problems involved. To review, the major classes of ill-formed representations are those that leave out part or all of the text and those with that grossly violate conceptual rules. The danger of the latter approach is that we might classify simply unusual representations as anomalous. We estimate that about 75% of the anomalous IPP representations either failed to represent a major portion of the text or violated very basic (and easy to detect algorithmically) conceptual rules (such as that people cannot kidnap themselves). Many of the remaining 25% were quite subtle, and would require significant analysis to find general detection rules.

Since we intend to propose rules for dealing with anomalous representations, one might wonder why these methods are not employed all the time. The answer is

twofold. First, the "careful" processing we will propose requires significant extra processing resources, and hence should only be used when absolutely needed. Furthermore, our processing of ill-formed representations might actually cause simpler examples to fail, particularly in practical systems. We might expect that the special-purpose rules will be considerably less robust that the pure conceptual analysis we use most of the time. It will probably be worth our while to do a separate (and hopefully fairly simple) check for conceptually ill-formed representations, and then process them further, rather than using detailed techniques on all texts.

There are two basic possibilities for dealing with conceptually ill-formed representations. As has been pointed out in the research on syntactically ill-formed input, we can either try and fix the anomalous representation or we can reprocess the input (obviously with some changes in method). While the "fix up" method has much appeal for syntactically ill-formed input, for conceptual anomalies, reprocessing will be required. There are many reasons for this, including the fact that reprocessing allows us to make best use of our basic understanding techniques, but the overriding reason is a very simple one: most conceptual anomalies lack some information from the text (possibly along with other problems). Since information needed for a correct representation is missing, we clearly must go back to the text in at least some cases. This contrasts with syntactic anomalies where the whole text is generally accounted for in the syntactic representation, just incorrectly.

Given the decision to deal with conceptually ill-formed representations by reprocessing the text, we must consider how the reprocessing should differ from the original. The obvious plan is to reprocess using more resources and "being more careful", which seems to match the plan that people use in similar circumstances. Of course, this leaves the major question of defining just what "being more careful" entails.

By examining the IPP and RESEARCHER texts, we have come to the conclusion that two basic forms of "being more careful" will handle most of the problems encountered. Unfortunately, these two methods involve diametrically opposite kinds of processing. This presumably means that, unless we can find rules for determining which method to use, we will have to try both for each case.

The first method for "being more careful" involves making more use of syntactic rules. Such processing might be as simple as using methods similar to those of syntactic-based processors. On the other hand, in systems such as ours which are constantly making both conceptual and structural predictions, this usually means using the structural predictions instead of the conceptual ones (which are normally given priority).

If we look back at the "hijacking and Iran hostages" example, we will see that such reprocessing would work. The story, which starts out, "A man saying he was setting out to free the American hostages tried to hijack ...", leads to conflicting expectations for "free". The conceptual expectation is that "freeing" should be a "scene" of the hijacking. Structurally, however, the embedded clause introduced by "saying" indicates that the "freeing" involves the hijacker's goals or demands. IPP normally uses the conceptual expectation, and hence its problems with this story. While we would not want to use the more complex structural predictions all the time (for example, news stories constantly involve "police saying" and "sources said" which we want to ignore), they do help in cases like this.

An alternative method for "being more careful" is exactly the converse of the first one -- ignore all structural rules and simply use the conceptual rules. In effect, this involves taking all the pieces of a story and seeing how they most sensibly fit together, ignoring how they appeared in the text. Obviously, this method will only work for the most conceptually normal of cases. There are a surprising number of such examples whose conceptually simplicity is obscured by structural complexity. This is the kind of processing that Charniak discussed in [Charniak 83] when he observed that examples like "Fire match arson hotel" can be understood.

Conceptual-only processing will work for the "teen-age hostage" example we looked at earlier. We can take the pieces, "heavily armed gunman", "teen-age hostage", "attempted to hijack", "old flying boat" and "to Capetown" and put them together in the conceptually obvious way to get the correct representation. Using this method, "teen-age hostage" does not have to play the double role it did in the original, since "attempting hijacking" confirms the taking of hostages. It is our feeling that this method will work for a large class of ill-formed representations.

Conceptual-only processing fits well with models of subjective understanding [Abelson 73; Carbonell 81], in that new text is molded to fit with existing beliefs. On the other hand, it has the obvious problem that unusual text will be misinterpreted to match stereotypical knowledge. It is interesting to look at the first example ever done by IPP in this light [Schank, et al. 80].

EX10 - S1; New York Times; 8 Oct 78; France

> An Arabic speaking gunman shot his way into the Iraqi embassy here this morning, held hostages throughout most of the day, before surrendering to French policemen and then was shot by Iraqi security officials as he was led away by French officers.

In this example, a totally conceptually-based system will misprocess the final shooting. Examples of this sort led to the integration of structural and conceptual expectations in IPP's understanding methods. On the other hand, as we have seen, at times it is necessary to give one method preference over the other.

5 Conclusions

In this paper we have made a number of observations about the role of ill-formed or anomalously represented input in text processing systems that use conceptually-based understanding methods. We will summarize these points here.

- Syntactically ill-formed text is not common in input to text processing systems, at least if we consider syntax in a practical sense.
- When syntactically ill-formed text is encountered, conceptual techniques usually deal with it successfully, normally not noticing it is ill-formed, but sometimes miss inferences that can be drawn by the fact that the input is ill-formed.
- Recognizing conceptually incorrect representations is a more subtle problem, since we want to recognize such problems without classifying texts that are merely unusual as anomalous.
- The general classes of conceptually incorrect representations that we can hope to detect easily are those that omit information from the text and those with gross conceptual anomalies.

- The best way to deal with conceptually incorrect representations is to reprocess the text, "being more careful", which can either consist of paying more intention to structural (syntactic) clues, or conversely, ignoring structural clues and only paying attention to conceptual expectations.

From our study of the problem, it would seem that the problem area most in need of further study to successfully handle conceptually ill-formed input is determination of what makes something appear anomalous enough that it cannot be correct.

A final observation is that analysis of the sort in this paper inevitably leads one to the belief that problems of ill-formed input of all kinds will only be dealt with fully when we have parallel, integrated systems of the sort discussed in [Erman et al. 80; Lebowitz 82b; Charniak 83]. Language is usually redundant enough that text ill-formed in one respect can be interpreted correctly using other information (hence the reprocessing heuristics mentioned above). So, ultimately, we will want parallel systems that make use of the *best* information currently available and tune out ill-formed channels, rather than using one source of information at a time, as is done by most of today's systems.

References

[Abelson 73] Abelson, R. P. The structure of belief systems. In R. C. Schank and K. Colby, Ed., *Computer Models of Thought and Language*, W. H. Freeman Co., San Francisco, 1973.

[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.

[Carbonell 81] Carbonell, J. G. *Subjective Understanding: Computer Models of Belief Systems.* UMI Research Press, Ann Arbor, Michigan, 1981.

[Charniak 83] Charniak, E. "Passing markers: A theory of contextual influence in language comprehension." *Cognitive Science 7*, 3, 1983, pp. 171 - 190.

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

[Dyer 83] Dyer, M. G. *In-depth understanding: A computer model of integrated processing for narrative comprehension.* MIT Press, Cambridge, MA, 1983.

[Erman et al. 80] Erman, L. D., Hayes-Roth, F., Lesser, V. R. and Reddy, D. R. "The HEARSAY-II speech-understanding system: Integrating knowledge to resolve uncertainty." *Computing Surveys 12*, 2, 1980, pp. 213 - 253.

[Harris 78] Harris, L. R. Natural language processing applied to data base query. Proceedings of the 1978 ACM Annual Conference, Association for Computer Machinery, Washington, D. C., 1978.

[Hayes and Carbonell 83] Hayes, P. J. and Carbonell, J. G. A framework for processing corrections in task-oriented dialogues. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 1983.

[Hayes and Mouradian 81] Hayes, P.J. and Mouradian, G. V. "Flexible parsing." *American Journal of Computational Linguistics 7*, 4, 1981, pp. 232 - 242.

[Kaplan 75] Kaplan, R. M. In process models for sentence analysis. In D. A. Norman and D. E. Rumelhart, Ed., *Explorations in Cognition*, W. H. Freeman and Company, San Francisco, CA, 1975.

[Kaplan 77] Kaplan, S. J. Cooperative responses from a natural language data base query system. Moore School of Engineering, University of Pennsylvania, 1977.

[Kolodner 83] Kolodner, J. L. "Maintaining organization in a dynamic long-term memory." *Cognitive Science 7*, 4, 1983, pp. 243 - 280.

[Kwasney and Sondheimer 81] Kwasney, S. C. and Sondheimer, N. K. "Relaxation techniques for parsing ill-formed input." *American Journal of Computational Linguistics 7*, 2, 1981, pp. 99- 108.

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

[Lebowitz 82a] Lebowitz, M. "Correcting erroneous generalizations." *Cognition and Brain Theory 5*, 4, 1982, pp. 367 - 381.

[Lebowitz 82b] Lebowitz, M. Limited parallel parsing. Columbia University Department of Computer Science, 1982.

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

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

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

[Lebowitz 84] Lebowitz, M. Using memory in text understanding. Columbia University Department of Computer Science, 1984.

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

[Reiser et al. 83] Reiser, B. J., Black, J. B. and Abelson, R. P. Knowledge structures in the organization and retrieval of autobiographical memories. Technical Report 22, Yale University Cognitive Science Program, 1983.

[Rieger 78] Rieger, C. "GRIND-1: First report on the Magic Grinder story comprehension project." *Discourse Processing 1*, 3, 1978.

[Riesbeck 75] Riesbeck, C. K. Conceptual analysis. In R. C Schank, Ed., *Conceptual Information Processing*, North Holland, Amsterdam, 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.

[Russell 75] Russell, S. *Disambiguation and understanding of metaphor using a conceptual feature system.* Ph.D. Thesis, Stanford University, 1975.

[Schank 82] Schank, R. C. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, New York, 1982.

[Schank and Birnbaum 82] Schank, R. C. and Birnbaum L. Memory, meaning and syntax. In T. Bever, J. Carroll and L. Miller, Ed., *Talking Minds: The Study of Language in Cognitive Sciences*, MIT Press, Cambridge, MA, 1982. Also Yale Computer Science Technical Report 189

[Schank, et al. 80] Schank, R. C., Lebowitz, M., and Birnbaum, L. "An integrated understander." *American Journal of Computational Linguistics 6*, 1, 1980, pp. 13 - 30.

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

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

[Webber and Mays 83] Webber, B. L. and Mays, E. Varieties of user misconceptions: Detection and correction. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 1983.

[Weischedel and Black 80] Weischedel, R. M. and Black, J. E. "Responding intelligently to unparsable inputs." *American Journal of Computational Linguistics 6*, 2, 1980, pp. 97 - 109.

[Wilks 73] Wilks, Y. An artificial intelligence approach to machine translation. In R. C. Schank and K. Colby, Ed., *Computer Models of Thought and Language*, W. H. Freeman Co., San Francisco, 1973

[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, pp. 591 - 606

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

[Woods and Kaplan 72] Woods, W. A. and Kaplan, R. M. The lunar sciences natural language information system. Final report. Technical Report BBN Report 2265, Bolt Beranek and Newman, Inc., Cambridge, MA, 1972.