8 Conclusion

8.1 Towards a conclusion

The goal of this chapter is to put the work described in previous chapters into perspective. First, I summarize the virtues of the work, and its potential. Then I compare it with similar work carried out concurrently by others. Finally, I list some of the questions that it leaves unanswered that ought to be answered.

8.2 The work in review

8.2.1 What has been achieved

I have presented a semantic interpreter and two disambiguation systems: one for lexical ambiguity and one for structural ambiguity. The systems have been designed to work closely with one another and with an existing parser and knowledge-representation system.

The semantic interpreter, Absity, is "Montague-inspired", in that it adapts to AI several aspects of Montague's (1973) way of thinking about semantics: it is compositional; it has a strong notion of "semantic object"; it operates in tandem with a parser; its partial results are always well-formed semantic objects; and it imposes a strong typing upon its semantic objects. The semantic objects are objects of the knowledge representation, and the types are the types that the representation permits (which, we saw, correspond to the syntactic categories of English).

The structural disambiguator is the Semantic Enquiry Desk. It tells the parser what to do whenever the parser needs semantic help to decide between two or more alternative structures. The SED makes its decisions by looking at the semantic objects in the partial results of the semantic interpreter, and, if necessary, by calling the knowledge base for information on plausibility and referential success.

Polaroid Words are the individual, one-per-word processes for lexical disambiguation. Each process figures out the meaning of the word for which it is responsible through negotiation with its "friends" (certain nearby Polaroid Words), using the knowledge base to find simple properties of semantic objects when necessary. Even when "undeveloped", Polaroid Words can be regarded by Absity as semantic objects that it can manipulate, and both the SED and other Polaroid Words can obtain well-formed partial information from a PW that has not been able to terminate.

8.2.2 Implicit claims

Implicit in my remarks about semantic objects is the claim that objects in a knowledge representation, although they are just bits in a computer, are somehow qualitatively different from "mere symbols", and I am critical (especially in section 2.3.1 and 8.3.1 below) of systems in which, I allege, the symbols of natural language are merely translated into other symbols instead of being interpreted into semantic objects (in the strict Montague 1973 sense of these terms). This claim needs some defense.

To build the defense, we must consider what a semantic object is, or could consist of. It seems to be generally agreed these days that semantic objects are not merely mental images or "capsules of thought" (Sapir 1921: 13; see Kempson 1977: 16 for discussion; but also see Rapaport 1985 on Meinongian semantics in AI); rather they are thought to be things like formal mathematical entities (Montague 1973) or situations and entities in the world (Barwise and Perry 1983). However, when people comprehend language, they do not "do semantics" with real objects, but rather with PERCEPTS of these objects. When we consider an agent understanding language, we must posit some internal representation of those semantic objects, a representation in which meanings of new sentences can be constructed, facts stored and retrieved, and inferences made—in other words, a knowledge representation in the AI sense.

It is not necessary to discuss here the many outstanding issues concerning the nature of mental representations. Rather, the point is that when we speak of an agent performing semantic computations, we are committing ourselves to an identification of the agent's percepts, or cognitive semantic objects, with the "real" semantic objects (whatever their nature may be) that are external to that agent. Thus I am arguing for a psychologically oriented semantics, in the style of Jackendoff (1983, 1984), and claiming, additionally, that once the identification of internal and external objects is made, the distinction between the two may, for most purposes, be ignored.

We can thus claim that Frail objects are fully fledged, first-class semantic objects; they may be made of silicon bits, but we can nevertheless identify them with things out there. Conversely, "mere symbols" (by definition) don't have the properties to permit this; they are just words. They can be stored and retrieved, even used in formal inference, but no direct correspondence to real semantic objects inheres in them. The difference between semantic objects and mere symbols is the same as the difference between a knowledge base and a database.

Also implicit in this work is the idea that an NLU system should be composed of interacting processes in which everything looks as well-formed as possible to everything else. Thus Absity always keeps the semantic objects that it is building well-formed, even when they are not final, so that the SED and the Frail representation can use them. Similarly, a PW represents a semantic object, and even if it can't decide which particular object it is, it will have a set of well-formed possibilities visible for other processes to make use of, while at the same time being manipulable as a single object by processes that don't care about its eventual meaning.

A third implicit idea is that, while not compromising the previous principle, things should generally happen in their own sweet time but as soon as possible. Thus, a Polaroid Word is not obliged to decide upon its final answer either immediately upon its creation or at any particular time thereafter (except in certain special circumstances; see sections 5.3.4 and 7.2.7). Nevertheless, a PW is expected not to dawdle, but to announce its answer as soon as it possibly can (*cf.* Mellish 1982b, 1985: 27; Just and Carpenter 1980).

A fourth aspect of the approach is that the design of each of the interacting processes must accede to the demands of the design of the other processes, and the design and development of each must therefore be coordinated. Thus, the parser demands of the structural disambiguator that it be able to make semantic decisions halfway through the parse. The disambiguator therefore requires of the semantic interpreter that it be incremental and have well-formed partial results that can be used for inference in and retrieval from the knowledge base. The semantic interpreter thus requires that the knowledge representation be compositional and support the concept of typed semantic objects. It also requires of the parser (thereby completing a circle of demands) that it support tandem processing; this, in turn, requires that lexical disambiguation appear to be immediate. Lexical disambiguation requires that the parser be able to insert pseudo-prepositions where necessary and do categorial disambiguation.

In the present research, the parser, Paragram, and the knowledge representation system, Frail, were the given starting points. Nevertheless, they too were forced to change: the grammar for Paragram had to accommodate pseudo-prepositions and Frail had to be given frame determiners. In addition, many of the requirements for the next version of Frail were discovered.

Nevertheless, many of the main ideas in the system and its components are independent of the other components of the system, and should be adaptable to other systems. Absity, for example, should be able to work with other parsers and with other knowledge representations, provided only that they support its requirements of compositionality and support of semantic objects; most frame-based representations—in particular, Sowa's (1984) conceptual graphs—should meet these requirements.

Where available data permitted it, the design of the system was also influenced by considerations of psychological reality. All else equal, we preferred the approach for which reasonable (but modest) claims of cognitive modeling could be made. Also, we have resisted the influences of "artificial difficulty" upon our design; we have not added any complexities just in order to be able to handle strange ambiguities that would also give people trouble or would be improbable in natural discourse.

8.2.3 Applications of the work

My hope for this research is that it be applicable in the development of general text understanding systems in wide domains of discourse. (To some extent, it is also applicable to the development of narrow-domain system interfaces, but problems such as lexical ambiguity are generally less of a worry in such cases.) There are three applications, all a long way off yet, that interest me in particular:

1. The digestion of large corpora of text. Enormous volumes of information in natural language are now available in machine-readable form. It would be useful to have a system that could really understand and integrate the information, in order to answer questions about it. Present systems simply use key words, or clever tricks based on key words, to find and present potentially relevant slabs of text. This is quite inadequate, for example, in legal information systems, where the search for a precedent case will often require matching at some very abstract level of the argument, and key words are of little or no help.[1]

2. Improving machine translation. High-quality translation without human assistance will never be possible. However, there is still much that could be done to improve the machine's part in the system. Moreover, high-speed, low-quality, unassisted machine translation may yet become practical, and acceptable in many situations.

3. Keeping expert systems up-to-date. How nice it would be if the medical expert systems of the future could keep their knowledge bases up-to-date by reading the medical journals each week. This poses additional problems beyond just digesting text: the relevant parts must be picked out from the rest, and then integrated with the existing knowledge base. Moreover, this knowledge base might not be Frail-like, or, indeed, be in any form suitable as a target for semantic interpretation. (For example, the production rules of present-day diagnosis systems are not a suitable target.) Further interpretation may therefore be necessary.

8.3 Related work of others

In this section, I discuss other recent work whose goals or methods are similar to those of the present project. For most of these, it is not possible to do much more than mention similarities and differences; I cannot as a rule compare merits, as both the present work and those mentioned below are as yet insufficiently mature to permit judgments any less vague than "seems promising" or "seems problematic".

---

[1] For this example I am grateful to Judy Dick, who is presently working at the University of Toronto on the development of suitable representations for arguments.

8.3.1 Semantic interpretation and disambiguation

The present approach to semantic interpretation is distinguished from that exemplified by van Bakel (1984). Van Bakel describes a system, AMAZON/CASUS, in which Dutch sentences are converted into a case-structure representation. However, there is no interpretation per se; words remain words, and no knowledge representation or concept of semantic object is introduced, nor is there any attempt at ambiguity resolution. In our terms, this is a translation into other symbols rather than an interpretation into semantic objects *(see section 8.2.2)*.[2]

One project on semantic interpretation whose motives are similar to ours is that of Jones and DS Warren (1982), who attempt a conciliation between Montague's (1973) semantics and a conceptual dependency representation (Schank 1975). Their approach is to modify Montague's translation from English to intensional logic so that the resulting expressions have a canonical interpretation in conceptual dependency. They do not address such issues as extending Montague's syntax nor whether their approach can be extended to deal with more modern Schankian representations (*e.g.*, Schank 1982a, 1982b). Nevertheless, their work, which they describe as a hesitant first step, is similar in spirit to ours and it will be interesting to see how it develops.

Another project that shares some of the spirit of the present work is that of Hendler and Phillips (1981; Phillips and Hendler 1982), who have implemented a control structure for NLU based on message passing. Their goal is to run syntax and semantics as parallel (but not tandem) processes and provide semantic feedback to the parser. The parser (Phillips 1983) follows in parallel all paths that it can find until either a unique one succeeds syntactically or a semantic decision is needed. Constituents are passed to a "moderator", which chooses from the parser's alternatives when necessary, translates the accepted syntactic constructs to their semantic representation in a Phillips network (Phillips 1975), and passes them on for further semantic processing. The system operates in the domain of patent descriptions, and is strongly expectation-driven. The approach is essentially ad hoc, in that the interpretation of syntactic constituents into various kinds of semantic object is not governed by any form of typing—though without tandem processing, the motivation for a strong typing is lessened.

After Polaroid Words were first reported in Hirst and Charniak 1982, Pazzani and Engelman (1983) described an apparently[3] similar system used in a military question-answering program called KNOBS. Like Polaroid Words, KNOBS had words as procedures that tried to find their sense in context, working in conjunction with a parser, a semantic interpreter,[4] and a frame-like representation of knowledge

---

[2] See Hirst (1985) for further discussion of this work.

[3] Unfortunately, the paper is rather hazy about many basic details.

[4] Pazzani and Engelman first seem to suggest that the system takes Small's (1980) word-expert approach to parsing, but the rest of the paper and its oral presentation at the Conference on Applied Natural

(that, in the case of KNOBS, also contained elements of conceptual dependency). Marker passing is not used, however, and when selectional restrictions fail it, the system falls back on looking at the currently active script *(cf. section 4.1)*.

The motivation behind the use of this approach to disambiguation is unclear, since KNOBS seems intended to operate solely in domains so constrained that they have scarcely room for any lexical ambiguity other than pronouns (which the system treats in a manner similar to other lexical ambiguities).[5] The only examples shown of disambiguation in the system are of the abbreviation *A/C*, which can mean either **aircraft** or **air conditioner**; the examples shown are these:

(8-1) What A/C can fly from Hahn?

(8-2) Send 4 A/C to BE70701.

(8-3) What is an A/C?

In sentence (8-1), *A/C* is resolved as **aircraft** by the selectional restrictions of *fly*. In sentence (8-2), selectional restrictions can't be used, so *A/C* is resolved as **aircraft** by noting that sending **aircraft** to coordinates is part of the "offensive counter air" script that defines the system's domain, while sending **air conditioners** isn't. And in sentence (8-3), where the complete lack of disambiguating cues calls for desperate measures, *A/C* is resolved as **aircraft** by noting that **air conditioners** do not participate in the system's script at all and that the system has nothing sensible to say about them. One might be forgiven for thinking that the ambiguity of *A/C* was introduced solely so the system would have something to disambiguate, and that the strategy applied in desperation to (8-3) might better have been the one used first on all of the examples.[6]

Since the publication of the present work as Hirst 1983b, Lytinen (1984) has reported a system, MOPTRANS, with several similar features. Lytinen's system, like the present work, is concerned with determining how syntax and semantics can

---

Language Processing seem to indicate a control structure like that of Polaroid Words and Absity rather than Small's Word Expert Parser.

[5] Polaroid Words, on the other hand, are intended for use in general text understanding systems.

[6] Notice further that if the system DID have something to say about **air conditioners**, then trying to disambiguate (8-3) would be wrong. Presumably the user does NOT want a reply like (i):

(i) An A/C is a winged vehicle that can fly to places, carry passengers and cargo, drop bombs, and intercept other aircraft.

but rather one like (ii) or (iii):

(ii) The abbreviation "A/C" means "aircraft" or "air conditioner", depending upon the context.

(iii) When I used it just now, I meant "aircraft"; in other contexts, it can mean "air conditioner".

I'm afraid I worry about artificial intelligence and artificial stupidity in military control systems. I have visions of systems like KNOBS ordering air conditioners to be sent to strategic locations to intercept the bad guys. Let's hope that the systems provide suitably unambiguous feedback to their users (*e.g.*, *air conditioners sent* rather than *A/C sent*).

work well together to create an unambiguous language-independent representation (which he uses as an interlingua for simple machine translation). Although he describes it in rather different terms, working in the Yale conceptual dependency paradigm (Schank 1975), the principles of MOPTRANS are quite similar to Absity's *(see section 1.1.1)*. However, Lytinen does not attempt a compositional semantics, and it is not always clear why the resulting representation is "correct". For instance, he gives this (not very well formed) example (1984: 112–113):

(8-4) Les ambulances de la Croix Rouge ont transporté d'urgence deux jeunes filles, dont les mains avaient été blessées par suite d'une bombe, à l'hôpital Manolo Morales.
(My translation: *Red Cross ambulances rushed two girls, whose hands had been injured by a bomb, to Manolo Morales Hospital.)*

The representation of this input (shown in figure 8.1) carefully includes the facts about the ambulance, but not the name of the hospital or the hands as the injured body part. Strangely, hands are nevertheless mentioned in the system's English translation, (8-5), though not its (not very well formed) German translation, (8-6):

(8-5) 2 young women who were injured by a bomb in the hands [sic; *see exercise 6.2 in section 9.6*] were rushed by an ambulance owned by the Red Cross to the hospital.

(8-6) 2 junge Frauen wurden nach das Spital mit einem Krankenwagenen von dem Rotkreutz, gehastet. Sie wurden mit einem Bombe verwundet.

The approach to lexical ambiguity in MOPTRANS includes rules similar to those employed by Polaroid Words, and uses simple world knowledge, especially ISA relationships. However, there is nothing in the system like marker passing for disambiguation by association. Lytinen, rather, was more concerned with deciding upon the exact meaning of a vague word in context. Thus,

(8-7) Gunmen seized control of the American embassy.

is recognized as an instance of the frame `take-over-building`, whereas

(8-8) Gunmen seized control of a Boeing 727.

is an instance of `hijack`. By contrast, Absity and PWs (if *seize control* could be recognized as a canned phrase; *cf. section 5.3.3*) would simply invoke the frame `seize-control`, and leave it to other processes (such as Bruin; *see section 1.3*) to make any further inferences or script instantiations that may be warranted. Thus, Absity takes an approach to vagueness similar to that of situation semantics (Barwise and Perry 1983; *cf.* Winograd 1984).

The MOPTRANS approach to structural ambiguity is to look for the attachment that is best semantically, and see if syntax will permit it. The system is not sensitive to lexical preferences.

```
Exp0 =
  Concept  Explode-Bomb
  Inst     Obj6 =
             Concept Bomb
             Inst-of Exp0
  Object   Hum21 =
             Concept Person
             Gender  Female
             B-Part  Obj5 =
                         Concept Bodypart
             Age     Young
             Number  2
  Result   Inj2 =
             Concept Injured
             R1 Hum21
             Result-Of Exp0
Ptr99 =
  Concept  Ptrans-by-Ambulance
  Object   Hum21
  To       Loc7 =
             Concept Hospital
  Inst     Obj4 =
             Concept   Ambulance
             Owned-By  Org5 =
                         Concept Medical-Org
                         Owns    Obj4
                         #Name   Red Cross
             Inst-Of   Ptr99
  Precond  Inj2
```

Figure 8.1. Lytinen's (1984) conceptual dependency representation of sentence (8-4).

The well-known BORIS system by Dyer (1983) also uses non-compositional semantics; the emphasis of this research is on "deep" understanding of texts and the knowledge structures required for this task. To resolve lexical ambiguity, BORIS looks at the surrounding context, using both top-down expectations created by preceding text and bottom-up processes for particular ambiguous words that look at subsequent text (Dyer 1983: 180–182). These are both implemented as demons, and the latter kind resemble Polaroid Words in some ways. The two methods cooperate in the system, with the bottom-up demons deferring to top-down expectations; this deference is in contrast to the results of Marslen-Wilson and Tyler (1980; Tyler and Marslen-Wilson 1982), which suggested the priority of bottom-up processes *(see section 2.4)*. For example, the word *gin* is resolved by BORIS as an **alcoholic drink** and a **card game**, respectively, by the expectations of `ingest` (from *drinks*) and `competitive-activity` (from *plays*) in the following sentences:

(8-9) John drinks gin.

(8-10) John plays gin.

Bottom-up processes would also be created for *gin*, but would terminate after noticing that top-down expectations already have the matter in hand. The bottom-up demons would, however, get to do the work in sentences like the following, where no expectations are yet created and the demons must look at the interpretation of what follows:

(8-11) The gin spilled.

(8-12) The gin players ...

It is not clear what would happen with a sentence like:

(8-13) John played with his gin.

Important recent work that extends the syntactic complexity of Montague's work is that on generalized phrase structure grammars (GPSGs) (Gazdar 1982). Such grammars combine a complex transformation-free syntax with Montague's semantics, the rules again operating in tandem. Gawron *et al* (1982) have implemented a database interface based on a GPSG. In their system, the intensional logic of the semantic component is replaced by a simplified extensional logic, which in turn is translated into a query for database access. Schubert and Pelletier (1982) have also sought to simplify the semantic output of a GPSG to a more "conventional" logical form, and Rosenschein and Shieber (1982) describe a similar translation process into extensional logical forms, using a context-free grammar intended to be similar to a GPSG;[7] Popowich (1984) has done similar work in logic grammars.

The GPSG approaches differ from mine in that their output is a logical form rather than an immediate representation of a semantic object; that is, the output is not tied to any representation of knowledge, and the process is therefore, as with van Bakel's work above, only translation rather than interpretation. In Gawron *et al*'s system, the database provides an interpretation of the logical form, but only in a weak sense, as the form must first pass through another (apparently somewhat ad hoc) translation and disambiguation process. Nor do these approaches provide any semantic feedback to the parser. (Gawron *et al* produce all possible parse trees for the input sentence and their translations and then throw away any that don't make sense to the database; Church and Patil (1982) have shown that this is not a good idea.) These differences, however, are independent of the choice of GPSG; it should be easy, at least in principle, to modify these approaches to

---

[7] Rosenschein and Shieber's semantic translation follows parsing rather than running in parallel with it, but it is strongly syntax-directed and is, it seems, isomorphic to an in-tandem translation that provides no feedback to the parser.

give Frail output, or, conversely, to replace Paragram in my system with a GPSG parser.[8]

The work of Mellish (1985) on the semantic interpretation of noun phrases addresses issues largely orthogonal to those treated in this book. However, Mellish's approach, the interpretation being refined incrementally as more information becomes available, is very like that of Polaroid Words, which, I think, meet his stated requirements for semantic analysis (1985: 27).

The PSI-KLONE system of RJ Bobrow and Webber (1980a, 1980b; Sondheimer, Weischedel, and RJ Bobrow 1984) also has a close coupling of syntax and semantics. Rather than operating in tandem, though, the two are "cascaded", with an ATN parser handing constituents to a semantic interpreter, which is allowed to return them (causing the ATN to back up) if the parser's choice is found to be semantically untenable. Otherwise, a process of INCREMENTAL DESCRIPTION REFINEMENT *(cf. previous paragraph)* is used to interpret the constituent; this relies on the fact that the syntactic constituents are represented in the same formalism, KL-ONE (Brachman 1978), as the system's knowledge base. The semantic interpreter uses PROJECTION RULES to form an interpretation in a language called JARGON, which is then translated into KL-ONE. Bobrow and Webber are particularly concerned with using this framework to determine the combinatoric relationship between quantifiers in a sentence.

Bobrow and Webber address several of the issues that I do, in particular the relationship between syntax and semantics. The information feedback to the parser is similar to the Semantic Enquiry Desk, though because the parser is deterministic in my system, semantic feedback cannot be conflated with syntactic success or failure. Both approaches rely on the fact that the objects manipulated are objects of a knowledge representation that permits appropriate judgments to be made, though in rather a different manner in each case.

Other than Absity, the main piece of work in AI that has been inspired by Montague semantics is Nishida's (1983, 1985) prototype English-to-Japanese translation system. English sentences are translated into a formal language called EFR ("English-oriented Formal Representation"), which is a typed, higher-order logic with a lambda operator; this plays the same role as Montague's intensional logic, and may be, in turn, either interpreted by a semantic network or translated into Japanese. For translation, the English words in the representation are replaced with their Japanese counterparts (which may be structures of arbitrary complexity), and the resulting logical expression is evaluated to give a case-structure representation for the output sentence. Heuristic rewriting rules for topicalizing, eliminating re-

---

[8] The choice of Paragram was largely pragmatic—it was available—and represents no particular commitment to transformational grammars. An Absity-inspired semantic interpreter for an ATN parser is presented in Charniak and McDermott 1985. Except for the type of grammar that it uses, the DIALOGIC system of Grosz *et al* (1982) is similar to that of Rosenschein and Shieber with regard to the properties discussed here.

dundancies, and the like then create the tree for the final Japanese output. For interpretation into a network structure, the English words in the EFR are replaced by pieces of network structure before the expression is evaluated.

Like Absity, Nishida's system runs syntax and semantics in tandem and is compositional. The syntax is extended from that of Montague's fragment. Parses are constrained by simple selectional restrictions, but the system relies on a human operator to decide between alternative parses when necessary.

Of the work mentioned above, only that of Dyer, Lytinen, and Pazzani and Engelman addresses issues of lexical ambiguity as ours does, though Bobrow and Webber's incremental description refinement could possibly be extended to cover it. Also, Gawron *et al* have a process to disambiguate case roles in the logical form after it is complete; this operates in a manner not dissimilar to the case-slot part of Polaroid Words.

8.3.2 Knowledge representation for language understanding

Sowa has recently described a representation of knowledge that is based on conceptual graphs (Sowa 1984). This system combines many of the advantages of frames *(see section 1.2)* with those of decompositional representations such as conceptual dependency (Schank 1973, 1975; *cf.section 2.2.1*). Sowa shows (1985) how his conceptual graphs support an approach to semantics and word and case slot disambiguation that is quite similar in spirit to that of Absity.

8.3.3 Spreading activation in natural language systems

If a spreading activation mechanism is to be included in an NLU system, perhaps it can do more than just lexical disambiguation. Pollack and Waltz (1982; Waltz and Pollack 1985) have described a system in which spreading activation simultaneously chooses the meaning of ambiguous words and a structure from among the alternatives presented to it by a chart parser. The system operates by building a network in which both activation and inhibition may spread, starting from nodes representing the input words. As the system proceeds, nodes representing the constituents of the correct parse rise in their activation level, while incorrect nodes become more inhibited, and eventually the system becomes consistent. (The initial activation levels of the nodes vary, in order to account for syntactic, lexical, and case preferences.) The result is a network that represents the parse of the input. Unlike people and Polaroid Words, the system is not fooled by garden-path sentences such as *The astronomer married the star.* Pollack and Waltz apparently regard this as a feature, because the system initially entertains the incorrect interpretation and then rejects it, doing a sort of "double take".

In a similar vein, Small, Cottrell, and Shastri (1982; Cottrell 1985a) developed a system that uses spreading activation and inhibition for both semantic interpretation and lexical disambiguation. The system is based on the connectionist processing and representation models of Feldman and Ballard (1982). As in Small's

Word Expert Parser *(see section 4.2.4)*, syntactic knowledge is not distinct, but rather is integrated with the system's other knowledge sources. Again, the goal is to spread activation and inhibition until a stable network is formed; this network is the interpretation, rather than just the parse, of the sentence. McClelland and Kawamoto (1986) have created a similar system that uses a distributed rather than localist representation.[9]

Charniak (1985) has taken the idea of spreading activation even further by suggesting that almost all aspects of language comprehension can be performed by a marker passer, along with a checker that discards anomalous marker-passing chains *(see section 5.2.3)*. Included are determining case from selectional restrictions, attaching PPs, disambiguating words (of course), analyzing the structure of nominal compounds, and determining the referents of NPs. The basic ideas are that much of conventional parsing may be eliminated, that marker passing can deal with such remaining structural analysis as is necessary, and that vague, imprecise representations will often suffice in the analysis.

There has also been some preliminary work on parsing by spreading activation in a network. The idea is that given a sentence, the network would settle into an equilibrium state representing the parse tree, which could then be used by a system such as Pollack and Waltz's *(above)*. Selman (1985; Selman and Hirst 1985, 1987) and Fanty (1986) have developed algorithms for constructing such a network from a context-free grammar. For related work, see Cottrell 1985b.

These systems are still in an early stage of development, and it remains to be seen whether parsing by spreading activation will be able to deal with the complexities of English syntax that more traditional parsers such as ATNs and Marcus parsers can (but see Jones 1983, Jones and Driscoll 1985), or the various semantic complexities that Absity can (or can't) interpret. Claims of psychological reality for this approach must be treated with caution, however; Auble and Franks (1983) have presented results suggesting that spreading activation plays only a limited role in sentence comprehension in people.

8.3.4 Lexical disambiguation in text databases

The use of on-line dictionaries for lexical disambiguation is being studied by Amsler (1982a) and Lesk (1986). Amsler's goal is analyzing the meaning of large on-line text corpora, with a view to information science applications. Amsler's previous research (*e.g.*, Amsler 1980, 1981) has shown that the structure of the texts of dictionary definitions makes the computational establishment of ISA and PART-OF relations among word senses quite easy. These can then be used to create suitable knowledge structures. For example (Amsler 1982a), a computational

---

[9] The other systems described above, like Polaroid Words, use LOCALIST representations in which each concept is represented by a single node in their formalism. In contrast, DISTRIBUTED representations use a pattern of activation in a set of nodes.

analysis of descriptors used in word senses that describe various vehicles leads to a description of what the vehicle frame is like, that is, what slots it should have and what the slots' typical and default values should be. The structure of definition texts may also be used in lexical disambiguation by following the ISA and PART-OF links. The text of the on-line *Merriam-Webster pocket dictionary* was itself disambiguated (by hand), so that it can be used in automatic disambiguation (Amsler 1980, 1981). In Lesk's version (1986), textual similarities between the definitions are sought directly.

This disambiguation technique is not unlike the marker passing used by Polaroid Words, except that the domain includes both words (possibly ambiguous, if the dictionary is not disambiguated) and concepts (*i.e.*, definitions). Clearly, dictionary processing has the potential for being extremely useful in NLU (*cf.* Amsler 1982b). In particular, the automatic generation of frame structures would greatly assist the development of large-scale knowledge bases. (Dictionary processing could not produce knowledge bases containing all that is necessary for, say, problem solving, but it could eliminate a lot of tedious work by producing a foundation for such bases.)