PARSING VS. TEXT PROCESSING IN THE ANALYSIS OF DICTIONARY DEFINITIONS

Thomas Ahlswede and Martha Evens
Computer Science Dept.
Illinois Institute of Technology
Chicago, IL 60616
312-567-5153

ABSTRACT

We have analyzed definitions from *Webster's Seventh New Collegiate Dictionary* using Sager's Linguistic String Parser and again using basic UNIX text processing utilities such as *grep* and *awk*. This paper evaluates both procedures, compares their results, and discusses possible future lines of research exploiting and combining their respective strengths.

Introduction

As natural language systems grow more sophisticated, they need larger and more detailed lexicons. Efforts to automate the process of generating lexicons have been going on for years, and have often been combined with the analysis of machine-readable dictionaries.

Since 1979, a group at IIT under the leadership of Martha Evens has been using the machine-readable version of *Webster's Seventh New Collegiate Dictionary* (W7) in text generation, information retrieval, and the theory of lexical-semantic relations. This paper describes some of our recent work in extracting semantic information from W7, primarily in the form of word pairs linked by lexical-semantic relations. We have used two methods: parsing definitions with Sager's Linguistic String Parser (LSP) and text processing with a combination of UNIX utilities and interactive editing.

We will use the terms "parsing" and "text processing" here primarily with reference to our own use of the LSP and UNIX utilities respectively, but will also use them more broadly. "Parsing" in this more general sense will mean a computational technique of text analysis drawing on an extensive database of linguistic knowledge, e.g., the lexicon, syntax and/or semantics of English; "text processing" will refer to any computational technique that involves little or no such knowledge.

This research is supported by National Science Foundation grant IST 87-03580. Our thanks also to the G & C Merriam Company for permission to use the dictionary tapes.

Our model of the lexicon emphasizes lexical and semantic relations between words. Some of these relationships are familiar. Anyone who has used a dictionary or thesaurus has encountered synonymy, and perhaps also antonymy. W7 abounds in synonyms (the capitalized words in the examples below):

(1) funny 1 1a aj affording light mirth and laughter : AMUSING
(2) funny 1 1b aj seeking or intended to amuse : FACETIOUS

Our notation for dictionary definitions consists of: (1) the entry (word or phrase being defined); (2) the homograph number (multiple homographs are given separate entries in W7); (3) the sense number, which may include a subsense letter and even a sub-subsense number (e.g. 2b3); (4) the text of the definition.

We commonly express a relation between words through a triple consisting of Word1, Relation, Word2:

(3) funny SYN amusing
(4) funny SYN facetious

A third relation, particularly important in W7 and in dictionaries generally, is taxonomy, the species-genus relation or (in artificial intelligence) the *IS-A* relation. Consider the entries:

(5) dodecahedron 0 0 n a *solid* having 12 plane faces
(6) build 1 1 vt to *form* by ordering and uniting materials . . .

These definitions yield the taxonomy triples

(7) dodecahedron TAX solid
(8) build TAX form

Taxonomy is not explicit in definitions, as is synonymy, but is implied in their very structure. Some other relations have been frequently observed, e.g.:

(9) driveshaft PART engine
(10) wood COMES-FROM tree

The usefulness of relations in information retrieval is demonstrated in Wang et al. [1985] as well as in Fox [1980]. Relations are also important in giving coherence to text, as shown by Halliday and Hasan [1977]. They are abundant in a typical English language dictionary, as we will see later.

We have recognized, however, that word-relation-word triples are not adequate, or at least not optimal, for expressing all the useful information associated with words. Some information is best expressed as unary attributes or features. We have also recognized that phrases and even larger structures may on one hand be in some ways equivalent to single words, as pointed out by Becker [1975], or may on the other hand express complex facts that cannot be reduced to any combination of word-to-word links.

Parsing

Recognizing the vastness of the task of parsing a whole dictionary, most computational lexicologists have preferred approaches less computationally intensive and more specifically suited to their immediate goals. A partial exception is Amsler [1980], who proposed a simple ATN grammar for some definitions in the *Merriam-Webster Pocket Dictionary*. More recently, Jensen and her coworkers at IBM have also parsed definitions. But the record shows that dictionary researchers have avoided parsing. One of our questions was, how justified is this avoidance? How much harder is parsing, and what rewards, if any, will the effort yield?

We used Sager's Linguistic String Parser, as we have done for several years. It has been continuously developed since the 1970s and by now has a very extensive and powerful user interface as well as a large English grammar and a vocabulary (the LSP Dictionary) of over 10,000 words. It is not exceptionally fast — a fact which should be taken into account in evaluating the performance of parsers generally in dictionary analysis.

Our efforts to parse W7 definitions began with simple LSP grammars for small sets of adjective [Ahlswede, 1985] and adverb [Klick, 1981] definitions. These led eventually to a large grammar of noun, verb and adjective definitions [Ahlswede, 1988], based on the Linguistic String Project's full English grammar [Sager, 1981], and using the LSP's full set of resources, including restrictions, transformations, and special output generation routines. All of these grammars have been used not only to create parse trees but also (and primarily) to generate relational triples linking defined words to the major words used in their definitions.

The large definition grammar is described more fully in Ahlswede [1988]. We are concerned here with its performance: its success in parsing definitions with a minimum of incorrect or improbable parses, its success in identifying relational triples, and its speed.

Input to the parser was a set of 8,832 definition texts from the machine-readable W7, chosen because their vocabulary permitted them to be parsed without enlarging the LSP's vocabulary.

For parsing, the 8,832-definition subset was sorted by part of speech and broken into 100-definition blocks of nouns, transitive verbs, intransitive verbs, and adjectives. Limiting the selection to nouns, verbs and adjectives reduced the subset to 8,211, including 2,949 nouns, 1,451 adjectives, 1,272 intransitive verbs, and 2,549 transitive verbs.

We were able to speed up the parsing process considerably by automatically extracting subvocabularies from the LSP vocabulary. so that for a 100-definition input sample, for instance, the parser would only have to search through about 300 words instead of 10,000.

Parsing the subset eventually required a little under 180 hours of CPU time on two machines, a Vax 8300 and a Vax 750. Total clock time required was very little more than this, however, since almost all the parsing was done at night when the systems were otherwise idle. Table 1 compares the LSP's performance in the four part of speech categories.

Part of speech of defd. word	Pct. of defs. parsed	Avg. no. of parses per success	Time (sec.) per parse	Triples generated per success
nouns	77.63	1.70	11.05	11.46
adjectives	68.15	1.85	10.59	5.45
int. verbs	64.62	1.59	11.96	6.62
tr. verbs	60.29	1.50	43.33	9.15
average	68.65	1.66	18.89	9.06

Table 1. Performance time and parsing efficiency of LSP by part of speech of words defined (adapted from Fox et al., 1988)

In most cases, there is little variation among the parts of speech. The most obvious discrepancy is the slow parsing time for transitive verbs. We are not yet sure why this is, but we suspect it has to do with W7's practice of representing the defined verb's direct object by an empty slot in the definition:

(11) madden 0 2 vt to make intensely angry

(12) magnetize 0 2 vt to communicate magnetic properties to

The total number of triples generated was 51,115 and the number of unique triples was 25,178. The most common triples were 5,086 taxonomies and 7,971 modification relations. (Modification involved any word or phrase in the definition that modified the headword; thus a definition such as "cube: a regular solid . . ." would yield the modification triple (cube MOD regular)).

We also identified 125 other relations, in three categories: (1) "traditional" relations, identified by previous researchers, which we hope to associate with axioms for making inferences; (2) syntactic relations between the defined word and various defining words, such as (in a verb definition) the direct object of the head verb, which we will investigate for possible consistent semantic significance; and (3) syntactic relations within the body of the definition, such as modifier-head, verb-object, etc. The relations in this last category were built into our grammar; we were simply collecting statistics on their occurrence, which we hope eventually to test for the existence of dictionary-specific selectional categories above and beyond the general English selectional categories already present in the LSP grammar.

Figure 1 shows a sample definition and the triples the parser found in it.

```
ABDOMEN 0 1 N THE PART OF THE BODY
          BETWEEN THE, THORAX AND THE
          PELVIS
 (THE) pmod (PART)
 (ABDOMEN 0 1 N) lm (THE)
 (ABDOMEN 0 1 N) t (PART)
 (ABDOMEN 0 1 N) rm (OF THE BODY BETWEEN
          THE THORAX AND THE PELVIS)
 (THE) pmod (BODY)
 (THE) pmod (PELVIS)
 (THE) pmod (THORAX)
 (BETWEEN) pobj (THORAX)
 (BETWEEN) pobj (PELVIS)
 (ABDOMEN 0 1 N) part (BODY)
```

Figure 1. A definition and its relational triples

In this definition, "part" is a typical category 1 relation, recognized by virtually all students of relations, though they may disagree about its exact nature. "lm" and "rm" are left and right modification. As can be seen, "rm" does not involve analysis of the long postnominal modifier phrase. "pmod" and "pobj" are permissible modifier and permissible object, respectively; these are among the most common category 3 relations.

We began with a list of about fifty relations, intending to generate plain parse trees and then examine them for relational triples in a separate step. It soon became clear, however, that the LSP itself was the best tool available for extracting information from parse trees, especially its own parse trees. Therefore we added a section to the grammar consisting of routines for identifying relations and printing out triples. The LSP's Restriction Language permitted us to keep this section physically separate from the rest of the grammar and thus to treat it as an independent piece of code. Having done this, we were able to add new relations in the course of developing the grammar.

Approximately a third of the definitions in the sample could not be parsed with this grammar. During development of the grammar, we uncovered a great many reasons why definitions failed to parse; there remains no one fix which will add more than a few definitions to the success list. However, some general problem areas can be identified.

One common cause of failure is the inability of the grammar to deal with all the nuances of adjective comparison:

(13) accelerate 0 1 vt to bring about at an earlier point of time

Idiomatic usages of common words are a frequent source of failure:

(14) accommodate 0 3c vt to make room for

There are some errors in the input, for example an intransitive verb definition labeled as transitive:

(15) ache 1 2 vt to become filled with painful yearning

As column 3 of Table 1 indicates, many definitions yielded multiple parses. Multiple parses were responsible for most of the duplicate relational triples.

Finding relational triples by text processing

As the performance statistics above show, parsing is painfully slow. For the simple business of finding and writing relational triples, it turns out to be much less efficient than a combination of text processing with interactive editing.

We first used straight text processing to identify synonym references in definitions and reduce them to triples. Our next essay in the text processing/editing method began as a casual experiment. We extracted the set of intransitive verb definitions, suspecting that these would be the easiest to work with. This is the smallest of the four major

W7 part of speech categories (the others being nouns, adjectives, and transitive verbs) with 8,883 texts.

Virtually all verb definition texts begin with *to* followed by a head verb, or a set of conjoined head verbs. The most common words in the second position in intransitive verb definitions, along with their typical complements, were:

become + noun or adj. phrase
(774 occurrences in 8,482 definitions)
make + noun phrase [+ adj. phrase]
(526 occurrences)
be + various
(408 occurrences)
move + adverbial phrase
(388 occurrences)

Definitions in *become*, *make* and *move* had such consistent forms that the core word or words in the object or complement phrase were easy to identify. Occasional prepositional phrases or other postnominal constructions were easy to edit out by hand. From these, and from some definitions in *serve as*, we were able to generate triples representing five relations.

(16) age 2 2b vi to become mellow or mature
(17) (age 2 2b vi) va-incep (mature)
(18) (age 2 2b vi) va-incep (mellow)

(19) add 0 2b vi to make an addition
(20) (add 0 2b vi) vn-cause (addition)

(21) accelerate 0 1 vi to move faster
(22) (accelerate 0 1 vi) move (faster)

(23) add 0 2a vi to serve as an addition
(24) (add 0 2a vi) vn-be (addition)

(25) annotate 0 0 vi to make or furnish critical or explanatory notes
(26) (annotate 0 0 vi) va-cause (critical)
(27) (annotate 0 0 vi) va-cause (explanatory)

We also attempted to generate taxonomic triples for intransitive verbs. In verb definitions, we identified conjoined headwords, and otherwise deleted everything to the right of the last headword. This was straightforward and gave us almost 10,000 triples.

These triples are of mixed quality, however. Those representing very common headwords such as *be* or *become* are vacuous; worse, our lexically dumb algorithm could not recognize phrasal verbs, so that a phrasal head term such as *take place* appears as as *take*, with misleading results.

The vacuous triples can easily be removed from the total, however, and the incorrect triples resulting from broken phrasal head terms are relatively few. We therefore felt we had been highly successful, and were inspired to proceed with nouns. As with verbs, we are primarily interested in relations other than taxonomy, and these are most commonly found in the often lengthy post-headword part of the definitions.

The problems we encountered with nouns were generally the same as with intransitive verbs, but accentuated by the much larger number (80,022) of noun definition texts. Also, as Chodorow et al. [1985] have noted, the boundary between the headword and the postnominal part of the definition is much harder to identify in noun definitions than in verb definitions. Our first algorithm, which had no lexical knowledge except of prepositions, was about 88% correct in finding the boundary.

In order to get better results, we needed an algorithm comparable to Chodorow's Head Finder, which uses part of speech information. Our strategy is first to tag each word in each definition with all its possible parts of speech, then to step through the definitions, using Chodorow's heuristics (plus any others we can find or invent) to mark prenoun-noun and noun-postnoun boundaries.

The first step in tagging is to generate a tagged vocabulary. We used an *awk* program to step through the entries and run-ons, appending to each one its part or parts of speech. (A run-on is a subentry, giving information about a word or phrase derived from the entry word or phrase; for instance, the verb *run* has the run-ons *run across*, *run after*, and *run a temperature* among others; the noun *rune* has the run-on adjective *runic*.) Archaic, obsolete, or dialect forms were marked as such by W7 and could be excluded.

Turning to W7's defining vocabulary, the words (and/or phrases) actually employed in definitions, we used Mayer's morphological analyzer [1988] to identify regular noun plurals, adjective comparatives and superlatives, and verb tense forms. Following suggestions by Peterson [1982], we assumed that words ending in *-ia* and *-ae* (virtually all appearing in scientific names) were nouns.

We then added to our tagged vocabulary those irregular noun plurals and verb tense forms expressly given in W7. Unfortunately, neither W7 nor Mayer's program provides for derived compounds with irregular plurals; for instance, W7 indicates *men* as the plural of *man* but there are over 300 nouns ending in *-man* for which no plural is shown. Most of these (e.g., *salesman*, *trencherman*) take plurals in *-men* but others (*German*, *shaman*) do not. These had to be identified by hand. Another

group of nouns, whose plurals we found convenient rather than absolutely necessary to treat by hand, is the 200 or so ending in *-ch*. (Those with a hard *-ch* (*patriarch, loch*) take plurals in *-chs*; the rest take plurals in *-ches*.) We could have exploited W7's pronunciation information to distinguish these, but the work would have been well out of proportion to the scale of the task.

After some more of this kind of work, we had a tagged vocabulary of 46,566 words used in W7 definitions. For the next step, we chose to generate tagged blocks of definitions (rather than perform tagging on the fly). We wrote a C program to read a text file and replace each word with its tagged counterpart. (We are not yet attempting to deal with phrases.)

Head finding on noun definitions was done with an *awk* program which examines consecutive pairs of words (working from right to left) and marks prenoun-noun and noun-postnoun boundaries. It recognizes certain kinds of word sequences as beyond its ability to disambiguate, e.g.:

(28) alarm 1 2a n a { signal }? warning } of danger

(29) aflatus 0 0 n a { divine }? imparting } of knowledge or power

The result of all this effort is a rudimentary parsing system, in which the tagged vocabulary is the lexicon, the tagging program is the lexical analyzer, and the head finder is a syntax analyzer using a very simple finite state grammar of about ten rules. Despite its lack of linguistic sophistication, this is a clear step in the direction of parsing.

And the effort seems to be justified. Development took about four weeks, most of it spent on the lexicon. (And, to be sure, more work is still needed.) This is more than we expected, but considerably less than the eight man-months spent developing and testing the LSP definition grammar.

Tagging and head finding were performed on a sample of 2157 noun definition texts, covering the nouns from *a* through *anode*. 170 were flagged as ambiguous; of the remaining 1987, all but 58 were correct for a success rate of 97.1 percent.

In 37 of the 58 failures, the head finder mistakenly identified a noun (or polysemous adjective/noun) modifying the head as an independent noun:

(30) agiotage 0 1 n { exchange } business

(31) alpha 1 3 n the { chief } or brightest star of a constellation

There were 5 cases of misidentification of a following adjective (parsable as a noun) as the head noun:

(32) air mile 0 0 n a unit { equal } to 6076.1154 feet

The remaining failures resulted from errors in the creation of the tagged vocabulary (5), non-definition dictionary lines incorrectly labeled as definition texts (5), and non-noun definitions incorrectly labeled as noun definitions (6). The last two categories arose from errors in our original W7 tape.

Among the 170 definitions flagged as ambiguous, there were two mislabeled definitions and one vocabulary error. There were 128 cases of noun followed by an *-ing* form; in 116 of these the *-ing* form was a participle, otherwise it was the head noun. (The other case flagged as ambiguous was of a possible head followed by a preposition also parsable as an adjective. This flag turned out to be unnecessary.) There were also seven instances of miscellaneous misidentification of a modifying noun as the head. Thus the "success rate" among these definitions was 148/170 or 87.1 percent.

We are still working on improving the head finder, as well as developing similar "grammars" for postnominal phrases and for the major phrase structures of other definition types. In the course of this work we expect to solve the major problem in this particular grammar, that of prenominal modifiers identified as heads.

Parsing, again

Simple text processing, even without such lexical knowledge as parts of speech, is about as accurate as parsing in terms of correct vs. incorrect relational triples identified. (It should be noted that both methods require hand checking of the output, and it seems unlikely that we will ever completely eliminate this step.) The text processing strategy can be applied to the entire corpus of definitions, without the labor of enlarging a parser lexicon such as the LSP Dictionary. And it is much faster.

This way of looking at our results may make it appear that parsing was a waste of time and effort, of value only as a lesson in how not to go about dictionary analysis. Before coming to any such conclusion, however, we should consider some other factors.

It has been suggested that a more "modern" parser than the LSP could give much faster parsing times. At least part of the slowness of the LSP is due to the completeness of its associated English grammar, perhaps the most detailed grammar associated with any natural language parser. Thus a

probable tradeoff for greater speed would be a lower percentage of definitions successfully parsed.

Nonetheless, it appears that the immediate future of parsing in the analysis of dictionary definitions or of any other large text corpus lies in a simpler, less computationally intensive parsing technique. In addition, a parser for definition analysis needs to be able to return partial parses of difficult definitions. As we have seen, even the LSP's detailed grammar failed to parse about a third of the definitions it was given. A partial parse capability would facilitate the use of simpler grammars.

For further work with the machine-readable W7, another valuable feature would be the ability to handle ill-formed input. This is perhaps startling, since a dictionary is supposed to be the epitome of wellformedness, by definition as it were. However, Peterson [1982] counted 903 typographical and spelling errors in the machine-readable W7 (including ten errors carried over from the printed W7), and my experience suggests that his count was conservative. Such errors are probably little or no problem in more recent MRDs, which are used as typesetter input and are therefore exactly as correct as the printed dictionary; errors creep into these dictionaries in other places, as Boguraev [1988] discovered in his study of the grammar codes in the *Longman Dictionary of Contemporary English.*

Before choosing or designing the best parser for the task, it is worthwhile to define an appropriate task: to determine what sort of information one can get from parsing that is impossible or impractical to get by easier means.

One obvious approach is to use parsing as a backup. For instance, one category of definitions that has steadfastly resisted our text processing analysis is that of verb definitions whose headword is a verb plus separable particle, e.g. *give up*. A text processing program using part-of-speech tagged input can, however, flag these and other troublesome definitions for further analysis.

It still seems, though, that we should be able to use parsing more ambitiously than this. It is intrinsically more powerful; the techniques we refer to here as "text processing" mostly only extract single, stereotyped fragments of information. The most powerful of them, the head finder, still performs only one simple grammatical operation: finding the nuclei of noun phrases. In contrast, a "real" parser generates a parse tree containing a wealth of structural and relational information that cannot be adequately represented by a formalism such as word-relation-word triples, feature lists, etc.

Only in the simplest definitions does our present set of relations give us a complete analysis. In most definitions, we are forced to throw away essential information. The definition

(33) dodecahedron 0 0 n a solid having 12 plane faces

gives us two relational triples:

(34) (dodecahedron 0 0 n) t (solid)
(35) (dodecahedron 0 0 n) nn-attr (face)

The first triple is straightforward. The second triple tells us that the noun *dodecahedron* has the (noun) attribute *face*, i.e. that a dodecahedron has faces. But the relational triple structure, by itself, cannot capture the information that the dodecahedron has specifically 12 faces. We could add another triple

(36) (face) nn-attr (12)

i.e., saying that faces have the attribute of (a cardinality of) 12, but this triple is correct only in the context of the definition of a dodecahedron. It is not permanently or generically true, as are (28) and (29).

The information is present, however, in the parse tree we get from the LSP. It can be made somewhat more accessible by putting it into a dependency form such as

(37) (solid (a) (having (face (plural) (12) (plane))))

which indicates not only that *face* is an attribute of that solid which is a dodecahedron, but that the cardinality *12* is an attribute of *face* in this particular case, as is also *plane*.

In order to be really useful, a structure such as this must have conjunction phrases expanded, passives inverted, inflected forms analyzed, and other modifications of the kind often brought under the rubric of "transformations." The LSP can do this sort of thing very well. The defining words also need to be disambiguated. We do not hope for any fully automatic way to do this, but co-occurrence of defining words, perhaps weighted according to their position in the dependency structure, would reduce the human disambiguator's task to one of post-editing. This might perhaps be further simplified by a customized interactive editing facility.

We do not need to set up an elaborate network data structure, though; the Lisp-like tree structure, once it is transformed and its elements disambiguated, constitutes a set of implicit pointers to the definitions of the various words.

Even with all this work done, however, a big gap remains between words and ideal semantic

concepts. Let us consider the ways in which W7 has defined all five basic polyhedrons:

(38) dodecahedron 0 0 n a solid having 12 plane faces

(39) cube 1 1 n the regular solid of six equal square sides

(40) icosahedron 0 0 n a polyhedron having 20 faces

(41) octahedron 0 0 n a solid bounded by eight plane faces

(42) tetrahedron 0 0 n a polyhedron of four faces

(43) polyhedron 0 0 n a solid formed by plane faces

The five polyhedrons differ only in their number of faces, apart from the cube's additional attribute of being regular. There is no reason why a single syntactic/semantic structure could not be used to define all five polyhedrons. Despite this, no two of the definitions have the same structure. These definitions illustrate that, even though W7 is fairly stereotyped in its language, it is not nearly as stereotyped as it needs to be for large scale, automatic semantic analysis. We are going to need a great deal of sophistication in synonymy and moving around the taxonomic hierarchy. (It is worth repeating, however, that in building our lexicon, we have no intention of relying exclusively on the information contained in W7).

Figure 2 shows a small part of a possible network. In this sample, the definitions have been parsed into a Lisp-like dependency structure, with some transformations such as inversion of passives, but no attempt to fit the polyhedron definitions into a single semantic format.

```
(cube 1 1) T (solid 3 1 (the) (regular)
        (of (side 1 6b (PL) (six)
        (equal) (square))))
(dodecahedron 0 0) T (solid 3 1 (a)
        (have (OBJ (face 1 5a5 (PL)
        (12) (plane)))))
(icosahedron 0 0) T (polyhedron (a)
        (have (OBJ (face 1 5a5 (PL)
        (20)))))
(octahedron 0 0) T (solid 3 1 (a)
        (bound (SUBJ (face 1 5a5 (PL)
        (eight) (plane)))))
(tetrahedron 0 0) T (polyhedron (a) (of
        (face 1 5a5 (PL) (four))))
(polyhedron 0 0) T (solid 3 1 (a) (form
        (SUBJ (face 1 5a5 (PL)
        (plane)))))
(solid 3 1) T (figure (a) (geometrical)
        (have (OBJ (dimension (PL)
        (three)))))
(face 1 5a5) T (surface 1 2 (plane)
        (bound (OBJ (solid 3 1 (a)
        (geometric)))))
(side 1 6a) T (line (a) (bound (OBJ
        (NULL))) (of (figure (a)
        (geometrical))))
(side 1 6b) T (surface 1 2 (delimit
        (OBJ (solid (a)))))
(surface 1 2) T (locus (a) (or (plane)
        (curved)) (two-dimensional)
        (of (point (PL)). . .))
```

Figure 2. Part of a "network" of parsed definitions

If this formalism does not look much like a network, imagine each word in each definition (the part of the node to the right of the taxonomy marker "T") serving as a pointer to its own defining node. The resulting network is quite dense. We simplify by leaving out other parts of the lexical entry, and by including only a few disambiguations, just to give the flavor of their presence. Disambiguation of a word is indicated by the inclusion of its homograph and sense numbers (see examples 1 and 2, above).

Summary

In the process of developing techniques of dictionary analysis, we have learned a variety of lessons. In particular, we have learned (as many dictionary researchers had suspected but none had attempted to establish) that full natural-language parsing is not an efficient procedure for gathering lexical information in a simple form such as relational triples. This realization stimulated us to do two things.

First, we needed to develop faster and more reliable techniques for extracting triples. We found that many triples could be found using UNIX text processing utilities combined with the recognition of a few structural patterns in definitions. These procedures are subject to further development and refinement, but have already yielded thousands of triples.

Second, we were inspired to look for a form of data representation that would allow our lexical database to exploit the power of full natural-language parsing more effectively than it can through triples. We are now in the early stages of investigating such a representation.

REFERENCES

Ahlswede, Thomas E., 1985. "A Linguistic String Grammar for Adjective Definitions." In S. Williams, ed., *Humans and Machines: the Interface through Language*. Ablex, Norwood, NJ, pp. 101-127.

Ahlswede, Thomas E., 1988. "Syntactic and

Semantic Analysis of Definitions in a Machine-Readable Dictionary." Ph.D. Thesis, Illinois Institute of Technology.

Amsler, Robert A., 1980. "The Structure of The Merriam-Webster Pocket Dictionary." Ph.D. Dissertation, Computer Science, University of Texas, Austin.

Amsler, Robert A., 1981. "A Taxonomy for English Nouns and Verbs." *Proceedings of the 19th Annual Meeting of the ACL*, pp. 133-138.

Apresyan, Yu. D., I. A. Mel'čuk and A. K. Žolkovsky, 1970. "Semantics and Lexicography: Towards a New Type of Unilingual Dictionary." In Kiefer, F., ed. *Studies in Syntax*. Reidel, Dordrecht, Holland, pp. 1-33.

Becker, Joseph D., 1975. "The Phrasal Lexicon." In Schank, R. C. and B. Nash-Webber, eds., *Theoretical Issues in Natural Language Processing*, ACL Annual Meeting, Cambridge, MA, June, 1975, pp. 38-41.

Boguraev, Branimir, 1987. "Experiences with a Machine-Readable Dictionary." *Proceedings of the Third Annual Conference of the UW Centre for the New OED*, University of Waterloo, Waterloo, Ontario, November 1987, pp. 37-50.

Chodorow, Martin S., Roy J. Byrd, and George E. Heidorn, 1985. "Extracting Semantic Hierarchies from a Large On-line Dictionary." *Proceedings of the 23rd Annual Meeting of the ACL*, pp. 299-304.

Evens, Martha W., Bonnie C. Litowitz, Judith A. Markowitz, Raoul N. Smith, and Oswald Werner, 1980. *Lexical-Semantic Relations: A Comparative Survey*. Linguistic Research, Inc., Edmonton, Alberta.

Fox, Edward A., 1980. "Lexical Relations: Enhancing Effectiveness of Information Retrieval Systems." *ACM SIGIR Forum*, Vol. 15, No. 3, pp. 5-36.

Fox, Edward A., J. Terry Nutter, Thomas Ahlswede, Martha Evens, and Judith Markowitz, forthcoming. "Building a Large Thesaurus for Information Retrieval." To be presented at the ACL Conference on Applied Natural Language Processing, February, 1988.

Mayer, Glenn, 1988. Program for morphological analysis. IIT, unpublished.

Halliday, Michael A. K. and Ruqaiya Hasan, 1976. *Cohesion in English*. Longman, London.

Klick, Vicki, 1981. LSP grammar of adverb definitions. Illinois Institute of Technology, unpublished.

Peterson, James L., 1982. Webster's Seventh New Collegiate Dictionary: A Computer-Readable File Format. Technical Report TR-196, University of Texas, Austin, TX, May, 1982.

Sager, Naomi, 1981. *Natural Language Information Processing*. Addison-Wesley, New York.

Wang, Yih-Chen, James Vandendorpe, and Martha Evens, 1985. "Relational Thesauri in Information Retrieval." *Journal of the American Society for Information Science*, vol. 36, no. 1, pp. 15-27.