# CHAPTER V LEXICAL DISAMBIGUATION

Disambiguation, as it shall be used in this dissertation, shall refer to a process in which words are assigned sense numbers taken from a standard reference dictionary according to the sense in which they are being used in a text. Specifically, the dictionary is the MPD and the usages are within textual definitions of that dictionary.

This chapter will describe the tasks involved in the manual recognition and disambiguation of the kernels of the definitions in the dictionary and then the computational assembly of the resultant taxonomic structure.

## 5.1 Generation of Coding Form Output

The first task was to use the existing databases to generate the equivalent of a concordance entry for those words which could serve as possible kernels of noun and verb definitions. This was done in three steps.

### 5.1.1 Frequencies of Defining Vocabulary for Nouns, Verbs, and Adjectives

First using the databases, separate lists of all the occurring words (i.e. words used within definitions texts excluding vocabulary examples and usage notes) and main entries were obtained. These lists provided additional frequency information on the number of occurring words in the MPD. The original Olney concordance had included such data for the top 64 overall most frequent words, but the occurrence lists now produced were restricted to text definitions of one part of speech. Thus the most frequent words used to define the nouns of the first noun database (A-NAME) (table 5-1), all the verbs (table 5-2), and all the adjectives (table 5-3) were made available. Data from the nouns of the second noun database (NAME-ZYMASE) would be similar to that for the first database because of the independence of genus terms from the words they define.

Table 5-1 Most Frequent Words in Noun Definitions for Nouns in range A through NAME (Noun Database 1)

(123486 OCCURRENCES; 13612 UNIQUE VALUES)

| FREQUENCY ELEMENT- | VALUE WORD | FREQUENCY ELEMENT- | VALUE WORD |
|---|---|---|---|
| 11483 | A | 256 | BODY |
| 7005 | OF | 239 | MADE |
| 5080 | THE | 235 | ANY |
| 4885 | OR | 234 | STATE |
| 2098 | IN | 220 | BEING |
| 2046 | AN | 218 | GROUP |
| 1974 | AND | 209 | LARGE,PLACE |
| 1674 | , | 192 | ITS |
| 1592 | TO | 170 | RELATED |
| 1574 | ; | 157 | BE |
| 1523 | AS | 145 | LIGHT |
| 1400 | FOR | 144 | ANOTHER |
| 1062 | ALSO | 142 | TWO |
| 1047 | ( | 141 | BETWEEN |
| 1047 | ) | 137 | INTO |
| 1000 | THAT | 133 | ARE |
| 957 | BY | 130 | SUCH |
| 873 | WITH | 129 | PLANT,WATER |
| 855 | ONE | 128 | HAVING |
| 614 | SOMETHING | 126 | ACTION |
| 566 | USED | 122 | FORM,USE |
| 554 | ESP. | 121 | FOOD,MEMBER |
| 546 | FROM | 120 | OFTEN |
| 540 | IS | 115 | VARIOUS |
| 517 | ESP | 114 | ANIMAL |
| 505 | ON | 113 | MATERIAL |
| 445 | PERSON | 109 | IT,QUALITY |
| 434 | USU. | 108 | DEVICE |
| 414 | WHICH | 107 | SUBSTANCE |
| 398 | WHO | 105 | MAN |
| 351 | ACT,SMALL | 102 | TIME |
| 300 | PART | 101 | BUILDING |

Table 5-2 Most Frequent Words in Verb Definitions
( 70979 OCCURRENCES; 7784 UNIQUE VALUES)

| FREQUENCY ELEMENT- | VALUE WORD | FREQUENCY ELEMENT- | VALUE WORD |
|---|---|---|---|
| 10789 | TO | 267 | GIVE |
| 4454 | OR | 243 | ESP. |
| 2751 | A | 241 | OUT |
| 1416 | IN | 240 | MOVE |
| 1359 | OF | 215 | TAKE |
| 1230 | AS | 214 | UP |
| 1219 | WITH | 194 | PUT |
| 1153 | THE | 173 | FORM |
| 1094 | BY | 165 | PLACE |
| 1016 | , | 162 | SOMETHING |
| 816 | ( | 144 | BRING |
| 816 | ) | 144 | ESP |
| 797 | MAKE | 143 | AT |
| 686 | ; | 141 | HAVE |
| 543 | ALSO | 140 | GO |
| 484 | FROM | 129 | OVER |
| 441 | FOR | 124 | USE |
| 434 | AND | 117 | SET |
| 426 | AN | 114 | COME |
| 391 | ON | 114 | THROUGH |
| 376 | BE | 111 | OFF |
| 341 | INTO | 105 | FORCE |
| 310 | IF | 102 | ONE'S |
| 303 | BECOME | 102 | PASS |
| 285 | CAUSE | | |

Table 5-3 Most Frequent Words in Adjective Definitions
(46310 OCCURRENCES; 7561 UNIQUE VALUES)

| FREQUENCY OF ELEMENT- | VALUE WORD | FREQUENCY OF ELEMENT- | VALUE WORD |
|---|---|---|---|
| 3553 | OR | 273 | WITH |
| 2113 | OF | 268 | FROM |
| 1704 | TO | 232 | AS |
| 1583 | , | 189 | MARKED |
| 1198 | THE | 187 | FOR |
| 1041 | A | 183 | AN |
| 902 | IN | 156 | ON |
| 809 | NOT | 144 | LACKING |
| 782 | RELATING | 136 | ESP |
| 588 | BY | 120 | CAPABLE |
| 578 | ; | 117 | AT |
| 483 | HAVING | 113 | ESP.,THAT |
| 445 | ALSO | 112 | ONE |
| 430 | BEING | 108 | MADE |
| 416 | AND | 101 | NO |

### 5.1.2 A Lexical Measure of Ambiguity for Nouns, Verbs, and Adjectives

Additionally, because of the nature of the database design, each main entry definition sense was counted as a separate database entry under that spelling. This produced separate "frequency" counts of the number of senses each noun, verb or adjective had. The number of a main entry's definition senses is interpretable as a measure of the ambiguity of that main entry for that part of speech. Hence the databases provided information indicating the most ambiguous nouns (table 5-4), verbs (table 5-5), and adjectives (table 5-6) in the MPD.

Table 5-4 Most frequent ("ambiguous") Noun Main Entry Senses including no. Subsenses and Run-On's

| No. Senses | Main Entry(s) |
|---|---|
| 31 | WAY |
| 24 | FORM |
| 21 | LINE |
| 20 | WING |
| 19 | DRAFT,TURN |
| 18 | CASE,PLAY |
| 17 | HAND,HEAD |
| 16 | MARK,ORDER |
| 15 | FALL,NOTE,PLACE,RUN,WORK |
| 14 | CHECK,DRIVE,LEAD,LIGHT,POINT, STOCK,TIME,USE |
| 13 | DESIGN,FAVOR,FIGURE,ISSUE,POST, PRESS,SPIRIT,WHEEL |
| 12 | CHARGE,FLASH,FLY,GRACE,MEASURE, RING,SERVICE,SHOT,THING,TOUCH, TWIST,VALUE,VOICE,WIND,WORD |
| 11 | AIR,DEPTH,DOUBLE,FEELING,FRONT, GALLERY,GROUND,LAP,LAW,LIFE,POWER, RESERVE,ROUND,SCALE,STYLE,TASTE, TONE,TRICK,UNION,VEIN,WEIGHT,WORLD |
| 10 | BEARING,BLOCK,CARD,CONNECTION, COURSE,CUT,DEAL,END,FACE,FLIGHT, FOOTING,GUARD,HEART,HEAT,HOLD, IMPRESSION,KEY,KNOT,MASS,MATTER, RANGE,REMEMBRANCE,RETURN,ROLL, SETTLEMENT,SLIP,STOP,STRAIN, STUFF,TEMPER,THOUGHT,TYPE,VIRTUE, WALK,WARD,WASH,WASTE,WIT |

Table 5-5 Most frequent ("ambiguous") Verb Main Entry Senses including no. Subsenses and Run-On's

| No. Senses | Main Entry(s) |
|---|---|
| 63 | GO |
| 35 | FALL, RUN |
| 31 | TURN, WORK |
| 30 | DO, DRAW |
| 29 | PLAY |
| 26 | GET |
| 24 | MAKE, STRIKE, TAKE |
| 21 | HAVE, PASS, TOUCH |
| 19 | GIVE |
| 18 | CARRY, HOLD, WIND |
| 17 | CATCH, SET |
| 16 | HANG, RAISE, RISE |
| 15 | BLOW, FLY, PULL |
| 14 | BREAK, CHECK, DROP, KEEP, LAY, PUT, SERVE, SETTLE, WHIP, WRAP |
| 13 | BEND, CALL, CLEAR, GATHER |
| 12 | CHARGE, DRESS, FEEL, FIX, FORM, GROW, INVEST, MEET, MOVE, PICK, PRESS, REST, SIT, TELL |
| 11 | BIND, CAST, DIVIDE, DRIVE, ENGAGE, ENTER, GAIN, ISSUE, LEAD, MIND, PACK, RECEIVE, REGARD, REPRESENT, SHOOT, TALK, THINK, WASH, WEAR, WRITE |
| 10 | BE, BRACE, CANCEL, CUT, DIP, DOUBLE, FILL, FIT, FREEZE, HIT, IMPRESS, INVOLVE, LEAVE, LOSE, MOUNT, PITCH, POINT, PRODUCE, REMOVE, RIDE, RING, ROLL, SINK, SPRING, STAND, STICK, SUSTAIN, TAP, THROW, TOP, TRUST, TUMBLE, WARRANT, WIN |

Table 5-6 Most Frequent ("ambiguous") Adjective Main Entry Senses including no. subsenses and Run-On's

| No. Senses | Main Entry(s) |
|---|---|
| 21 | DEAD,GOOD |
| 20 | DRY |
| 18 | FAIR,FREE,HARD |
| 17 | UP |
| 16 | FRESH,HIGH,OPEN,SHARP |
| 15 | CLOSE,HEAVY,THICK,WEAK,WHITE |
| 14 | FAST,GREAT,ROUGH,TIGHT |
| 13 | BAD,DULL,FALSE,FLAT,GROSS,LOW,SOLID |
| 12 | DELICATE,FLUSH,GENERAL,LIGHT,NEW,RIGHT |
| 11 | DEEP,FOUL,FULL,NATURAL,PLAIN,POSITIVE, SHORT,SOUND,SQUARE,WARM,WILD |
| 10 | DOWN,GOLDEN,PROPER,PUBLIC,REGULAR, REMOTE,ROUND,SIMPLE,STIFF,STRAIGHT, ULTIMATE,WHOLE |

### 5.1.3 Computational Generation of Noun Plurals

These complete lists of occurrences and main entries were to be intersected to provide a list of all occurring words which could also be main entries. From experience with the taxonomic concordance it was known that nouns occurring in a definition as taxonomically related to the word being defined were sometimes plurals, e.g., "buildings" in the definition of "plant-2.2a",

plant-2.2a - the land, buildings, and machinery used in carrying on a trade or business

To prevent these terms from being lost when possible definition genus terms were selected a program was written to generate possible plurals from all noun main entries and this list was intersected with the list of occurrences. There of course remained the usual confusions over noun-verbs such as "line" with plural "lines" equivalent to the 3rd person singular verb form. The elimination of these verbal forms occurring in noun definitions as candidates for noun genus terms would await human intervention.

## 5.2 Human Disambiguation

External funding (NSF Project MCS77-01315, Development of a Computational Methodology for Deriving Natural Language Semantic Structures via Analysis of Machine-Readable Dictionaries) had become available during the time it took to load the databases. Thus it was possible to hire a small group of graduate students to undertake the massive kernel identification and disambiguation task. The first sub-task was the development of a coding procedure which the disambiguators could use to record the syntactic and semantic decisions they had to make. The full discussion of the development of this coding procedure and the reasons for each coding convention is given in Amsler and White [1979] and only excerpts from that material will be presented here.

### 5.2.1 Syntactic and Semantic Scoring Conventions

For each singular or plural noun (or infinitive verb) that occurs in a noun (or verb) sense definition, the disambiguator was given a sense-definition text in the following form:

<ME><SN>..... = <KCT>.....<SDT>

where:

<ME> = Main Entry
<SN> = Sense Number
<KCT> = Kernel Candidate Term
<SDT> = Sense Definition Text

The kernel candidate term was that word in the definition text for which a disambiguation decision was to be made. For each singular or plural noun in a noun's sense-definition text, a line with the same main entry, sense number, and sense definition was given, with the noun to be considered appearing as the kernel candidate term (figure 5-1). For every infinitive verb in a verb's sense-definition text, a line with the same main entry, sense number, and sense definition was given, with the verb to be considered as the kernel candidate term (figure 5-2).

CRUISER-.2A..... = LIVING..... A MOTORBOAT EQUIPPED FOR LIVING ABOARD

CRUISER-.2A..... = MOTORBOAT..... A MOTORBOAT EQUIPPED FOR LIVING ABOARD

Figure 5-1 Noun Kernel Candidate Terms

TIE-2.5A..... = EQUAL..... TO MAKE OR HAVE AN EQUAL SCORE WITH

TIE-2.5A..... = HAVE..... TO MAKE OR HAVE AN EQUAL SCORE WITH

TIE-2.5A..... = MAKE..... TO MAKE OR HAVE AN EQUAL SCORE WITH

TIE-2.5A..... = SCORE..... TO MAKE OR HAVE AN EQUAL SCORE WITH

Figure 5-2 Verb Kernel Candidate Terms

Since the noun kernel candidate terms LIVING and MOTORBOAT; and the verb kernel candidate terms EQUAL, HAVE, MAKE, and SCORE, are all words which appear as MPD noun or verb main entries, a scoring line appears in the work forms for each.

The work forms were pre-sorted alphabetically by candidate terms. In the examples above, the line with LIVING as candidate term was included in the letter-L noun work-form set and occurred next to any other noun sense-definition that contained the word LIVING. Likewise the line with EQUAL as the verb infinitive candidate term was included in the letter-E verb work-form set and occurred next to any other potential verb sense-definition that contained the word EQUAL. Segregation of noun and verb kernel candidates was automatic because of the separate databases from which noun and verb candidates were derived.

The sorting of definition texts by candidate term presented the disambiguator with a means of doing all disambiguation of identical spelling forms at the same time. This allowed disambiguators to build a conceptual semantic model of the word to be disambiguated and assured maximal consistency between occurrences of one spelling form during disambiguation. The task of scoring kernel candidate terms was based upon three tests which human disambiguators were asked to make on each term (Figure 5-3).

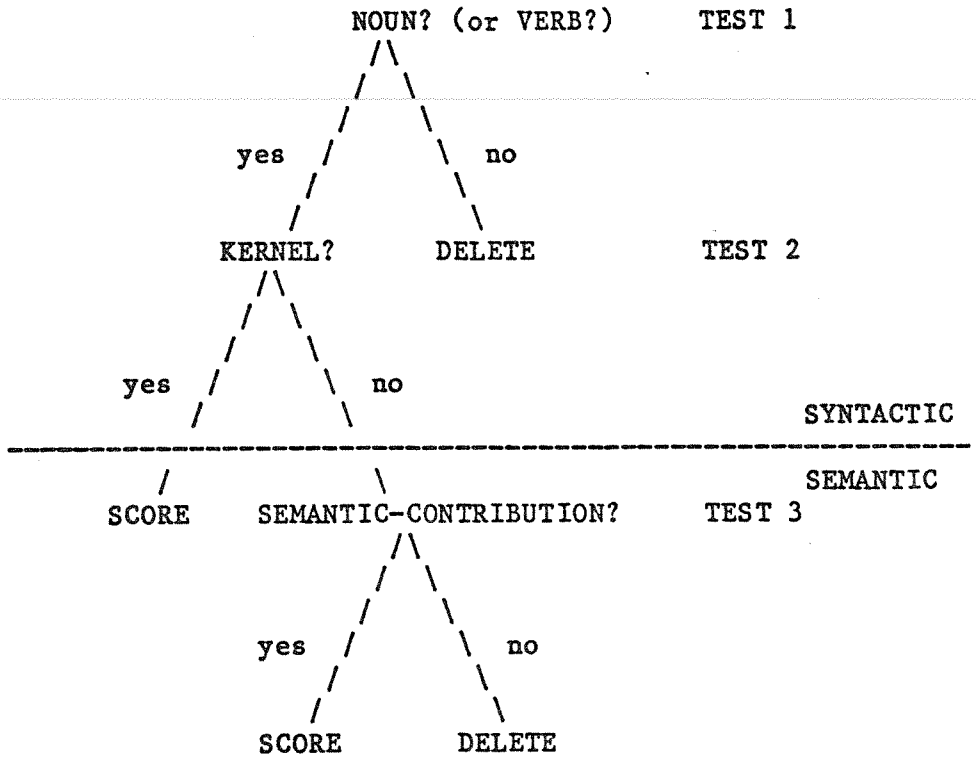



Figure 5-3 Key showing order of disambiguator scoring decisions taken from Amsler and White [1979].

The first test was simply to decide whether or not the candidate term was of the appropriate part of speech to be of further interest in the taxonomy. Thus only nouns occurring in noun sense definition texts or verb infinitives occurring in verb sense definition texts were eligible. In figure 5-1, of the two occurrences of the noun main entry CRUISER-.2A, only MOTORBOAT passes the first test, with LIVING being a verb form. For the verb entries of TIE-2.5A in figure 5-2, only MAKE and HAVE pass the first test, with EQUAL occurring as an adjective form and SCORE as a noun.

The second test the disambiguators were asked to perform was syntactic in nature and required their judgement as to how the sense definition text would be parsed during a linguistic analysis. Specifically, they were asked to determine the syntactic head of the noun or verb definition texts and to see whether the kernel candidate term was that head term. Syntactically a noun's definition is a noun phrase, and a verb's definition is a verb infinitive phrase.

At this point the disambiguators were keenly aware of the potential

problems that pronominal or other grammatical forms could cause in obscuring the semantic kernel of a definition. Some light can be shed on this problem if one considers the class of definitions for objects having a physical form, a specifiable substance or substances from which they are made, and/or a functional purpose for which they are used. Objects can have all three components specified in their definitions -- and while syntactically the focus may be upon the form of the object -- semantically or conceptually the composition or function of the object may be a more important term in its definition for taxonomic purposes.

atlas-.0a - a BOOK of MAPS

bateau-.0a - ANY of various small CRAFT

beam-1.1a - a large long PIECE of TIMBER or METAL

deuterium-.0a - a FORM of HYDROGEN that is of twice the mass of ordinary hydrogen

ribbon-.1a - a narrow FABRIC typically of SILK or VELVET used for TRIMMING and for BADGES

ring-1.1a - a circular BAND worn as an ORNAMENT or TOKEN or used for holding or fastening

To identify these problems, disambiguators were asked to also score words in a definition which would pass the first test, but otherwise would have been rejected for failing the second test. These scores were entered by first placing a / ("slash") after the disambiguation sense number of the word and then entering additional symbols to indicate the nature of the extenuating circumstances involved in the score.

The third test can be characterized as asking the disambiguators to decide whether the frame, "<main-entry> IS <kernel-candidate-term>" was an acceptable taxonomically true statement. Thus, "charity-.3a" is defined as "the giving of aid to the poor" and since "aid" is a noun the decision as to whether to disambiguate it appeared under the A nouns with AID as a kernel candidate. In this context "charity is aid" was found to be an acceptable taxonomic statement and an appropriate slash score would have been suffixed to the sense-number. This was an effort to score words in definitions which were perhaps not syntactically the head of the definition, but were semantically capable of serving in that role.

One additional request was made of the disambiguators at this point. Since they were deciding on the relative semantic acceptability of the candidate terms vs. the syntactic kernel, it was felt appropriate to indicate this acceptability with a two-valued scale. Indications as to whether the semantic kernel was more (/!) or less (/+) important than the true (and often unappropriate) syntactic kernel to the definition of the main entry were added.

Table 5-7 presents a summary taken from Amsler and White [1979] of the coding orthography used. While appearing quite complex in terms of all the options available, only a few of the options were used with high frequency. The three basic distinctions as presented in figure 5-3 were the major coding decisions. In terms of frequency of use, a word with a homograph, sense, and subsense alone, or one with this score suffixed with a /+ or a /! comprised the three most common disambiguations used.

Table 5-7 Coding Orthography for Scoring Disambiguated Words (from Amsler and White [1979])

| Position | Values | Significance |
|---|---|---|
| -1 | <word> | word from MPD |
| 0 | - | Separates Score from <word> |
| 1 | C or W | Collegiate or International |
| 2 | T or I | Transitive or Intransitive |
| 3 | \R or \I | Run-on or Inflected |
| 4 | <sense no.> | Sense + Subsense |
| 5 | V,N, or A | Verb, Noun or Adjective Run-on |
| 6 | <entry no.> | Homograph no. |
| 7 | .<sense no.> | Sense + Subsense |
| 8 | \-<> or *<> | verb + particle construction |
| 9 | /+,/!,/C or /N | less,more,causative,and negative flags |
| 10 | /? or /& | unknown, conjoined 'AND' |
| 11 | * | Collegiate appropriateness |
| 12 | % | addendum |

Conditions:

A) Either a value for 3 or a value for 6 must occur, but not both.

B) Position 11 does not have a value unless position 1 also has a value.

C) Position 2 has a value if and only if position 1 has a value because only the Collegiate and Third International dictionaries designate verbs as transitive or intransitive.

Every scorable word had to have a value for position 4 or 7, but could have had empty values for any of the others. Only positions which were used had values. With the exception of position 6, null values usually indicated appropriateness for taxonomic use. Only words having scores for positions 7 or 6 and 7 were capable of taxonomic transitivity.

Each position represented a unique scoring decision by the disambiguator. The intent of the values available for scoring in each position are as follows:

1. Alternate source dictionary. Every effort was made to find an appropriate sense in the MPD to fit the usage of a word in a definition. When this effort failed the disambiguator looked up the entry in other Merriam-Webster dictionaries, first the Seventh Collegiate (C), and if no score was found there, the Third International (W).

2. Transitive vs. Intransitive Verb Sense. Verbs scored after recourse to the Collegiate or Third International dictionaries could either have been from the transitive (T) or intransitive (I) sequences of sense numbers listed in those dictionaries. This value could only occur if position 1 had a C or W and the word being scored was a verb having both a transitive and an intransitive series of definitions.

3. Entry types. When the kernel of a sense definition did not occur as a main entry in the MPD, but rather as a run-on (R) listed after the definition or as an inflection (I) of the main entry these scores were used. Some kernels scored with an I did not actually appear in boldface in the definition of a main entry.

4. Run-on sense number. On a few occasions a run-on listed at the end of an entry's definition had more than one sense. A run-on cannot also be a main entry (except as homographs), and so no word with a value for 4 may have a value for 7. However, it was theoretically possible for a run-on to have a value for 6, as when the run-on was appended to a word which had several entries, some of which were of the same part of speech. The resulting score, similar to \R<no.>Vl., never occurred in the actual scoring.

5. Part of speech of source main entry. This position was included in order to show the relationship between a run-on or inflection used nominally in noun definitions and the part of speech of the main entry. Ultimately it had been had intended to show those places in the taxonomy where noun trees were descended from verbs, a process which has considerable value for the semantic analysis of the lexical items involved, and which serves to join the entire vocabulary into an integrated network. Accordingly, only run-ons and inflections (words scored with values for 3) had values for 5.

6. Homograph number. Values for 6 indicate the SUPERIOR NUMBERS in the MPD which precede each main entry whenever there are other main entries with the same spelling.

7. Main entry sense number. Values for 7 were composed of the the

boldface sense numbers in the MPD and a letter which indicated which sub-section of a sense definition was referred to. 7 only had a null value when a scored word was a run-on or inflected form (i.e. 4 was \R or \I)

8. Particle-idiom information. Words or phrases (indicated in table 5-7 as <>) following the back-slash score contributed to the semantic validity of the ISA-link between the main entry and the kernel term. If the word was preceded by '-' it was part of a near-atomic phrase with the focal term. If the word was preceded by '*', the word was considered a semantically necessary part of the phrase with the main entry.

9. Non-kernel information. Words scored with values for 9 did not occur as the syntactic kernel of a main entry sense definition, but had a semantic ISA relationship to the main entry. The word was judged for its value to the ISA-relationship compared to the value of the syntactic kernel. If this term was semantically more important than the syntactic kernel, then the value for 9 was /!. If not more important, then the value for 9 was /+. A term which was semantically important and related to the main entry via a syntactic kernel which expressed negation or causation was scored with a /N or a /C respectively.

10. Problematic scorables. Words scored with values for 10 were either the kernel of a main entry sense definition, or had a value for 9, but were not considered appropriately scored at the time. If the word occurred in the definition as part of an "and"-conjunctive phrase, the value /& was entered in this position. If, at the time of scoring, no available sense seemed appropriate, the value /? was entered. Theoretically, both values could occur simultaneously (as /&?, but no such scores appeared).

11. Exclusive source information. Words scored with a * in this position were lexical-item phrases which appeared in the Collegiate (either as a run-on or as a main entry), but not in the MPD. This symbol was used to indicate that the choice of a Collegiate sense in such a case was not made solely for appropriateness of sense, but because of an exact match to the lexical item needed.

12. Tree version information. Words which were affixed with the symbol % appeared in the final version of the noun or verb trees, but did not appear in the initial version, for any of a number of reasons. This symbol was affixed computationally as part of the final tree-growing procedure.

### 5.2.2 Sense Decisions

As indicated in figure 5-2 showing the order of disambiguator scoring decisions, the selection of the appropriate sense number for a scorable kernel term was fundamentally a semantic decision. The disambiguator's decision was constrained by the available senses in the dictionary and assisted by seeing all the occurrences of the word to be considered for scoring together.

After determining that a kernel term was scorable, the disambiguator looked up the term in the MPD. From the senses given for the word, the decision as to the sense which best fitted that term's usage in the definition was made. This decision was then entered for the kernel term, along with any necessary special orthography.

It was quite often the case the disambiguators could not find a perfect fit between the available senses and the kernel term's usage. Often, a sense was considered to have all the necessary attributes for a fit, but additional attributes which made it too specific. In other instances, the disambiguators found a sense was too general to form a meaningful fit, or there was more than one sense which could be construed as appropriate. To some degree these difficulties were capable of being overcome IF sufficient time were given to study the senses involved and access to example sentences from the larger Seventh Collegiate and Third International Dictionaries was available. It is my contention that in every case only one sense was the correct one, and that whenever more than one sense was thought to be correct that insufficient understanding of the distinctions had been achieved. While the disambiguator was permitted to resort to the Seventh Collegiate dictionary for more appropriate senses, this was only done in the most hopeless circumstances. Normally, the disambiguator found compromises which resolved these problems by comparing the usage of a kernel term in one definition text with the usage of the same term in others.

In coding the noun "day", for example, the disambiguator had a total of 54 definitions using "day" to examine and 7 MPD senses of "day" from which to find the correct sense for each scorable usage. All the occurrences of the kernel term "day" were in front of the disambiguator at the same time, permitting a scoring decision by comparison of the usage of "day" in the definition to the usages in other lines already scored. If the usage was not perceived as exactly fitting any one sense of "day" better than the other senses, but a previous line with a very similar usage did fit, then the disambiguator would score the present line as s/he had scored the previous line. This process of referring to previously scored kernel terms is illustrated in the disambiguation protocol transcript given in appendix 4.

Thus the scoring task for ALL the usages of "day" was a collective one

and greater self-consistency was promoted, even at the possible loss of independence. This was appropriate because while reference to the definition text was part of the classification task, the ultimate goal of the disambiguations was to connect the semantically identical usages together by specifying a common disambiguation sense-number for each conceptually distinct usage set. The task of growing a taxonomy required all the disambiguations of a single spelling to be self-consistent and did not actually depend on whether the senses given in the dictionary were used exactly as the lexicographers had intended them to be -- as long as the disambiguator did use the text definitions of these senses consistently for different concepts which did derive from their text statements. Any task involving tens of thousands of decisions as to word senses had to place a premium on the speed of disambiguation and compromise the concept of "ultimate semantic truth" enough to get the scoring done in a matter of months, rather than years or decades.

It was expected that the individual decisions as to the correct taxonomic senses of word usages would be no more reliable than the decisions of biological taxonomists examining the flora and fauna of an unexplored continent for the first time. With certainty, entire treatises on the correct classification of individual genera can and will continue to be written and English language semantics will continue to revise any taxonomic conclusions based upon studies such as this. This study sought to provide initial data and provoke further painstaking investigation -- not proclaim definitive conclusions as to the structure of the whole English lexicon based upon one pocket dictionary's definitions.

A very limited test of the self-consistency of one disambiguator was made based upon the transcripted session reproduced in part in appendix 4. The disambiguator re-scored 37 occurrences of two words specifically selected to be of medium-level difficulty. They had previously disambiguated the words a few months before, but didn't recall anything about the task from then. 31 (84%) of the scores were identical, 3 (8%) were additions of a /+ score where the entry had been deleted as failing the "semantic contribution" test (figure 5-3) previously, and 3 (8%) more were differences between two senses which were noted as being hard to decide between during the transcripted session. Adding the /+'s in does not truly affect the validity of the tree produced, so one might characterize consistency as 84%; omissions 8%; and inconsistency 8%.

### 5.2.3 Frequencies of Disambiguated Senses

Once the disambiguators had scored all the kernels of the definitions which were going to be used in growing the taxonomies, a new statistic became available. This was the frequency with which each available sense of each kernel term had been used. Thus, rather than just being able to say that the spelling form of a word occurred so many times, now one could determine what semantic concepts were in fact being used. This was especially important because the most common words in the language also

have the most senses. Without knowledge of which senses of these words are being used one cannot adequately determine many aspects of their importance to the language. For example, if one knows which of their senses are actually frequently used as opposed to insignificant occurrences, then one can define a semantic core of the meanings needed to define the lexicon. West's dictionary [West 1953] made an effort at determining similar semantic percentages of use for its limited vocabulary, but the figures from this study now indicated the percentages of use by sense, of every kernel word that occurred in the noun and verb taxonomies.

Whereas one could previously only state that "MAKE" was the most frequent verb used to define other verbs, for example; now the data as to what sense(s) of "MAKE" were so used could be determined. The most frequent semantic concepts used to define other words in the Pocket Dictionary are presented for nouns in table 5-8 and verbs in table 5-9.

Table 5-8 Most Frequent Noun Kernels in Noun Definitions

| | | | |
|---|---|---|---|
| 789 | ONE 2.2A | 50 | SURFACE 1.1A |
| 630 | SOMETHING .0A | 49 | CONDITION 1.3A |
| 567 | PERSON .1A | 49 | OBJECT 1.1A |
| 444 | ACT 1.1B | 49 | STRIP 2.1A |
| 360 | STATE 1.1A | 48 | BIRD .0A |
| 350 | PART 1.1A | 48 | SHIP 1.1A |
| 339 | ONE 2.1A | 47 | BODY .4A |
| 225 | ACT 1.1A | 47 | INSTRUMENT .3A |
| 180 | DEVICE .2A | 47 | LANGUAGE .1A |
| 173 | MEMBER .2A | 47 | MAN 1.1B |
| 167 | GROUP 1.0A | 47 | MASS 1.1A |
| 150 | QUALITY .1A | 47 | MEANS 3.2A |
| 139 | PROCESS 1.4A | 45 | DISEASE .0B |
| 136 | PLACE 1.3A | 45 | LAND 1.1B |
| 122 | PIECE 1.2B | 44 | ACTION .3A |
| 110 | WOMAN .1A | 44 | DEGREE .2A |
| 109 | MATERIAL 2.1A | 42 | ACTION .2A |
| 104 | PERIOD 1.6A | 42 | MAMMAL .0A |
| 104 | PLANT 2.1A | 42 | PLANTS 2.1A/! |
| 96 | ELEMENT .3A | 42 | PRODUCT .1A |
| 93 | SOUND 2.2A | 42 | VEHICLE .3A |
| 92 | BODY .5A | 41 | LACK 2.0A |
| 90 | FABRIC .2A | 41 | PRACTICE 2.1A |
| 87 | INSTANCE 1.3A | 41 | USE 1.1A |
| 86 | INHABITANT .0A | 40 | BOOK 1.1A |
| 82 | INSTRUMENT .2A | 40 | COLOR 1.1B |
| 82 | PART 1.1A/+ | 40 | GAME 1.5A |
| 81 | AMOUNT 2.1A | 40 | MAN 1.1A |
| 78 | QUANTITY .1A | 40 | OFFICIAL 1.0A |
| 76 | NATIVE .0B | 39 | BLOW 4.1A |
| 74 | PLACE 1.2A | 39 | BOAT .0A |
| 67 | BUILDING .1A | 39 | MACHINE 1.2A |
| 67 | ROOM 1.3A | 39 | MOVEMENT .1A |
| 67 | UNIT .2A | 39 | QUALITY .4A |
| 66 | OFFICER .3A | 39 | STATE 1.2A |
| 66 | SUBSTANCE .2B | 38 | LIQUID 2.0A |
| 64 | TREE 1.1A | 38 | REGION .0A |
| 62 | POWER 1.2A | 38 | WORD 1.2A |
| 61 | ACTION .4A | 37 | CONTAINER .0A |
| 59 | SERIES .0A | 37 | THING .6A |
| 59 | STRUCTURE .2A | 36 | DISTANCE 1.1A |
| 58 | HERB .1A | 36 | MANNER .2A |
| 58 | SUBSTANCE .2A | 36 | TREES 1.1A/! |
| 58 | SYSTEM .1A | 35 | LAYER .2A |
| 54 | AREA .3A | 35 | OPENING .2A |
| 54 | GROUP 1.0A/! | 34 | HORSE .1A |
| 53 | STATEMENT .1A | 34 | LETTER 1.1A |
| 51 | SCIENCE .1A | 33 | FISH 1.1B |
| 50 | BRANCH 1.3A | 33 | HERBS .1A/! |
| 50 | GARMENT .0A | 33 | IMPLEMENT 1.0A |

Table 5-9 Most Frequent Verb Kernels in Verb Definitions

| | | | |
|---|---|---|---|
| 480 | MAKE 1.1A | 30 | GIVE 1.15A |
| 264 | CAUSE 2.0A | 30 | SUPPLY 1.3A |
| 214 | BE .1B | 30 | TAKE 1.1A |
| 137 | MOVE 1.1A | 30 | TREAT 1.5A |
| 107 | MAKE 1.10A | 30 | USE 2.2A |
| 80 | GIVE 1.9A | 29 | MAKE 1.2A |
| 75 | COVER 1.1A | 29 | SEPARATE 1.1A |
| 72 | PUT .4A | 28 | FORM 2.1B |
| 62 | BE .4A | 28 | SET 1.4A |
| 58 | BRING .2A | 27 | DRAW 1.7A |
| 57 | MOVE 1.3A | 27 | FURNISH .1B |
| 55 | STRIKE 1.2A | 27 | GIVE 1.4A |
| 55 | UTTER 2.1A | 27 | TAKE 1.18A |
| 51 | MAKE 1.2B | 25 | COME .1A |
| 47 | SET 1.2A | 25 | FORCE 2.3A |
| 45 | EXPRESS 3.1A | 25 | GET 1.5A |
| 44 | GO 1.1B | 25 | OBTAIN .1A |
| 44 | PUT .1A | 25 | PLACE 2.2A |
| 43 | BRING .3A | 25 | REMOVE 1.4A |
| 40 | COME .3A | 25 | TRAVEL 1.1A |
| 40 | SHOW 1.1A | 24 | CUT 1.1A |
| 39 | CHANGE 1.1A | 23 | BRING .1A |
| 38 | MARK 2.2A | 22 | DRIVE 1.1A |
| 36 | PROVIDE .4A | 22 | GET 1.1A |
| 34 | ARRANGE .1A | 22 | GO 1.1A |
| 33 | REMOVE 1.2A | 22 | SERVE 1.6A |
| 32 | SEND .1A | 21 | AFFECT 2.0A |
| 32 | SUBJECT 3.3A | 21 | ENCLOSE .1A |
| 30 | FASTEN .1A | 21 | REDUCE .1A |
| 30 | FEEL 1.7A | | |

## 5.3 The Tree Growing Process

The computational task of connecting a set of paired word-senses into a single data structure was handled via use of MACLISP, a version of LISP. This task has several problems inherent in its completion, some of which make LISP an ideal programming language, others which strain LISP's capabilities, and still others which require careful evaluation in order to avoid unpleasant surprises in attempting to output the assembled structure.

### 5.3.1 The Input Data

The input may be formally characterized as a set of word-sense element pairs {(A1, B1), (A2, B2), ... (An, Bn)} where Ai and Bi are both selected from a set of all word-senses of a given part of speech occurring in a dictionary. The relationship between Ai and Bi, for any given i, may be expressed as "immediate descendant" or "genus/species" where Ai is an immediate descendant of Bi, or Ai is a species of genus Bi. By convention adopted from previous work on semantic networks [Simmons 1973; Simmons and Amsler 1975] I have used the relationship "token", symbolized by TOK, to represent such an infix relationship in "semantic triple notation", i.e., (Ai TOK Bi). Examples of the realization of this notation for actual word-sense elements include triples such as (MAMMAL-.0A TOK GIRAFFE-.0A), (MAMMAL-.0A TOK ZEBRA-.0A), etc. This notation can be readily extended into a form in which a list replaces the Bi component and all of the immediate descendants of any given Ai are enumerated in one pair, as (Ai TOK (B1, B2, B3, ... Bk)). An actual instantiation from the dictionary in this format is (MAMMAL-.0A TOK (AARDVARK-.0A BEAVER-.0A CHINCHILLA-.0A ... GIRAFFE-.0A ... ZEBRA-.0A)).

The relationship in the dictionary which corresponds to this "immediate descendant", "genus/species" or "token" (TOK) relationship is of course the relationship between a word-sense in its occurrence as main entry and the word-sense of its definition's kernel, as this word-sense was determined by the coding conventions discussed above.

I decided that the so-called slash-scores should also be included in the tree growth process. This was done because although they do not represent transitive relationships (there being no main entry defined in the dictionary with a slash score) and thus are restricted to the Ai portions of a word-sense pair, the growth process would nevertheless collect them together and enumerate them. This enumeration would display such elements in a manner permitting evaluation of their appropriateness as topmost tree nodes -- a most important observation for those nodes

marked with a /! score (i.e., nodes judged to have been better (semantic) kernels than the actual syntactic kernel selected for use without a slash score).

### 5.3.2 The Programming Language

The LISP programming language is ideally suited to representation of a structure assembled from a set of paired elements such as the input data. This is because LISP automatically handles the hash-coded access to the unique atoms Ai and Bi. Thus, each time Ai is referenced it is the same Ai regardless of whether these successive references were consecutive ones or 10,000 other word-sense nodes were created inbetween.

The principal limitation LISP has in this type of application is its requirement that all such word-sense nodes be resident in-core at one time. In this regard, SRI-International has apparently done some work on a "virtual atom package", though no publications on this project were available [Slocum 1979].

### 5.3.3 The Program

Ideally, a program should assemble the arcs of the tree, find the highest node and then enumerate the nodes encountered, traversing all the arcs downward. If, as does occur, there is no one single node spanning the entire set of nodes (the data when assembled is a forest instead of a single tree), then the program should by logical extension find all the highest nodes and traverse all of their arcs downward, tree by tree, until the entire forest was enumerated. This however is not always possible when one deals with real data.

In the formal definition of the data set I deliberately did not distinguish the word-sense elements which could be a member of the Ai's from those which could be a member of the Bi's. Except for the "slash scores" (as mentioned already), the two sets are not, in fact, exclusive. Except for the nodes which are "roots" or "terminals" of some tree, every other node must appear at least once as a left-hand member of an (Ai Bi) pair and at least once as a right-hand member of an (Aj Bj) pair.

It is also true that for some (Ai Bi) there is a pair (Aj Bj) such that Ai is the same word-sense as Bj, and Aj is the same word-sense as Bi. This configuration causes a loop or circuit to occur.

In terms of instantiations from the dictionary structure, this corresponds to a pair of words which are used as the kernels of each other's definitions. As such, the two word-senses involved are effectively reduced to one sense-meaning realizable by either of two different spelling forms; i.e., the two are synonyms.

The looping relation is often separated by a lengthy TOK path, i.e., A1 TOK B1 TOK A2 TOK B2 ... TOK Bn TOK A1. This of course makes it

impossible to assign to any of the members of such a circuit the role of "root" node, and makes it computationally very difficult to determine whether a word-sense node is a member of such a circuit or actually just a normal intermediate non-terminal node. All members of a loop appear to be intermediate nodes in structure, and would consequently be rejected as candidates for enumeration, producing a form of "rootless" tree which would never be found and thus never enumerated.

To avoid this type of problem I chose a more redundant, yet fully complete procedure for enumerating the structures in the assembled set of word-sense element pairs: an algorithm which simply enumerates the tree below every non-terminal word-sense element. This has an added benefit in that with complete enumeration of all non-terminals there is no requirement to provide an index to the trees for the purpose of locating specific intermediate nodes. To find out what is below a given non-terminal one merely has to look for that non-terminal in the alphabetically ordered forest of non-terminal trees. To find out what is above any non-terminal node, one can use the ordinary dictionary definitions as enumerated with their attached sense numbers to go upward one or more levels and then look at the appropriate higher non-terminal tree from that node downward.

The program always maintains a context stack of nodes whose descendants are being enumerated. Whenever any node is to be added to that context stack, a membership check is performed to see if this node would initiate a loop. If so, a warning message is output along with the contents of the stack at this point, the duplicate node is rejected, and the next node in the sequence of descendants that would normally have been selected after expanding the current (duplicate) node is immediately considered.

This procedure quickly terminates loops and marks their existence for later post-editing, while necessitating no significant back-tracking which would hinder enumeration of the entire tree.

The code for the MACLISP version of this program is given in figure 5-4. It is quite small and thus demonstrates the power of LISP in automatically handling what would be difficult bookkeeping and storage allocation/expansion tasks in other programming languages.

```
(defun in () (prog (toplist) (setsyntax 56 2 nil)
     (setq inf (openi ">udd>lrc>Amsler>tdata"))
     (setq dataout (openo ">udd>lrc>Amsler>output"))
     (defun str_help (fl sm) (structure) (return nil))
     (eoffn inf 'str_help)
 top (setq red (read inf))
     (setq red (list (car red) (cddr red)))
     (apply 'p2 red)
     (go top) ))
(defun p2 (a b) (progn (put a 'subs b)
  (cond ((get a 'upper) nil)
        (t (setq toplist (cons a toplist))
           (flag (list a) 'upper))) ))
(defun structure () (prog (stack) (setq indent 1)
 tax (cond ((null toplist) (return t))
           (t (princ '---------------' dataout) (terpri dataout)
              (linear (list (car toplist)))
              (setq toplist (cdr toplist))
              (go tax)))))
(defun linear (lis) (prog (g fir)
  tip (cond ((null lis)(return t))
            (t (prindent indent) (princ lpar dataout)
               (princ (setq fir (car lis)) dataout)
               (cond ((member fir stack)(terpri dataout)
                      (princ '   ****loop****' dataout)
                      (princ (cons fir stack) dataout)
                      (terpri dataout)
                      (setq lis (cdr lis))
                      (go tip)))
               (terpri dataout) (setq lis (cdr lis)) ))
      (cond ((setq g (get fir 'subs))
             (setq indent (add1 (add1 indent)))
             (setq stack (cons fir stack))
             (linear g)
             (setq stack (cdr stack))
             (setq indent (sub1 (sub1 indent)))
             (go tip))
            (t (go tip))) ))
(defun prindent (n)
  (cond ((zerop n) t)
        (t (princ blank dataout) (prindent (sub1 n))) ))
(defun put (a i v) (putprop a v i))
(defun flag (list prop)
  (mapc (function (lambda (a) (put a prop t))) list))
(setq blank '/ ) (setq lpar '/() (setq toplist nil)
```

Figure 5-4 MACLISP Tree-Growing Program

## 5.4 Addition of Definition Text

After the MACLISP program grew the forest of taxonomic trees there remained the task of adding definitions to the word-senses of the tree elements as they were arranged in the output. Since the number of nodes in the trees was so large (27,000 word-sense elements for nouns; 12,000 for verbs) it was not possible to store their definitions along with the nodes in-core, so they had to be added later using traditional data processing techniques.

### 5.4.1 Stages of Text Addition

The traditional data processing technique for adding this information is to perform the operation in multiple passes over the output data, rewriting the data in a format making mutual comparisons possible, sorting to rearrange data items, and merging related data items only when they are immediately adjacent to each other. The final output is then produced by resorting the resulting data into the original output format once again.

The two data sets I wished to merge were the output of the tree growing program and the definition text of the dictionary. The result was to be a data set containing the output of the tree, but accompanied by the definition text appropriate for each word-sense element.

By virtue of the same disambiguation data that made possible the tree-growing process, it was possible to enhance the definition texts before they were re-attached to the word-sense elements. The disambiguated sense-number tags for the nouns and verbs were respectively merged into the alphabetically arranged noun and verb definitions. This required: (a) rewriting the coding input forms to contain the tagged word-sense in place of the original untagged word-senses, and (b) merging the multiple occurrences of definition texts so as to include in one definition all the tagged words (in the case of multiple syntactic and/or semantic kernels) that were coded for a particular sense definition. Once this was accomplished, the definitions contained all the information that had been added during the disambiguation process, and provided a convenient cross-index to the other trees under which a given word-sense was also listed.

The dictionary definitions were then rewritten into the same format as the sequence-numbered lines of the output-tree file and merged with segments of the output tree. Such a segmentation was necessary since the output tree was too large to be processed as one file in our on-line environment. After sorting by word-senses common to both information

types, a program merged the definitions of the definition/output-tree file with the word-senses which occurred on the now alphabetically ordered lines of output-tree data immediately following the definition text. Definitions which did not match any word-sense element in a file were dropped. Likewise, word-senses which did not exactly match some definition sense did not have any definition text attached. This unfortunately occurred in some cases where slash-scores or apostrophes were involved. (The apostrophe was the LISP quote symbol and had not been specially protected).

Finally, the merged word-sense lines, which now included the appropriate definitions from the MPD, were resorted into the original output-tree order using the previously assigned sequence numbers.

As executed on the data from the disambiguated senses in the MPD, the task of adding definition text onto the tree structures may be seen as a process of several steps, each step having an output product associated with it. Figures 5-5 through 5-11 show a sample of the output from some of these steps for portions of the data under one node TIME-1.3A.

```
(TIME-1.3A
  (AGE-1.2A
  (ARMAGEDDON-.0B
  (CANDLELIGHT-.2A
  (COMMENCEMENT-.1A
  (CONVENIENCE-.4A
  (DEADLINE-.0A
  (JUNCTURE-.3A
  (MANANA-.0A
  (MEAL-1.2A
    (BREAKFAST-.0A
      (BRUNCH-.0A
    (BUFFET-3.2B
      (SMORGASBORD-.0A
    (DINNER-.0A
    (LUNCH-1.1A
      (BRUNCH-.0A
      (LUNCHEON-.0A
    (POTLUCK-.0A
    (SUPPER-.0A
    (TABLE-D'HOTE-.0A
```

Figure 5-5 Output-Tree Segment in Indented Format

Beginning with the tree structure in indented format (figure 5-6), a sequence-numbered file was created which also included depth-numbering information. The depth refers to the number of nodes above a given node in the tree in question and is equivalent to indentation/2 in the lines of output above. "-1 ----" was added between entries to properly note the beginning/end of consecutive trees.

```
009013 -1 ----
009014 00 TIME-1.3A
009015 01 AGE-1.2A
009016 01 ARMAGEDDON-.0B
009017 01 CANDLELIGHT-.2A
009018 01 COMMENCEMENT-.1A
009019 01 CONVENIENCE-.4A
009020 01 DEADLINE-.0A
009021 01 JUNCTURE-.3A
009022 01 MANANA-.0A
009023 01 MEAL-1.2A
009024 02 BREAKFAST-.0A
009025 03 BRUNCH-.0A
009026 02 BUFFET-3.2B
009027 03 SMORGASBORD-.0A
009028 02 DINNER-.0A
009029 02 LUNCH-1.1A
009030 03 BRUNCH-.0A
009031 03 LUNCHEON-.0A
009032 02 POTLUCK-.0A
009033 02 SUPPER-.0A
009034 02 TABLE-D'HOTE-.0A
```

Figure 5-6 Sequence-Numbered Format

The data in the format of figure 5-6 was then sorted and segmented into five conveniently-sized letter ranges: A through C, D through G, H through N, O through R, and S through Z; for merging with the definition texts.

The dictionary definition texts were taken from the disambiguators' scored coding sheets. Rather than enumerate this entire set for even these few words let me just show the processing for one word's definition set, BRUNCH-.0A. Figure 5-7 shows the relevant entries from the scored disambiguators' output forms involved.

From the B Nouns:

```
BRUNCH .0A..... = BREAKFAST .0A...  A LATE BREAKFAST , AN
                                    EARLY LUNCH , OR A
                                    COMBINATION OF THE TWO
```

From the C Nouns:

```
BRUNCH .0A..... = COMBINATION .2A.  A LATE BREAKFAST , AN
                                    EARLY LUNCH , OR A
                                    COMBINATION OF THE TWO
```

From the L Nouns:

```
BRUNCH .0A..... = LUNCH 1.1A......  A LATE BREAKFAST , AN
                                    EARLY LUNCH , OR A
                                    COMBINATION OF THE TWO
```

Figure 5-7 Coding Form Output

First, the coding form output was converted into definitions with disambiguated words in-context as in figure 5-8.

From the B Nouns:

```
BRUNCH-.0A          = A LATE BREAKFAST-.0A , AN EARLY LUNCH ,
                      OR A COMBINATION OF THE TWO
```

From the C Nouns:

```
BRUNCH-.0A          = A LATE BREAKFAST , AN EARLY LUNCH ,
                      OR A COMBINATION-.2A OF THE TWO
```

From the L Nouns:

```
BRUNCH-.0A          = A LATE BREAKFAST , AN EARLY LUNCH-1.1A ,
                      OR A COMBINATION OF THE TWO
```

Figure 5-8 Merged-in Word-Sense Tags

The tagged word-senses for the main entries of figure 5-8 were then sorted and each tagged word from a main entry which had been tagged for more than one syntactic/semantic kernel was merged into a single definition for that main entry.

BRUNCH-.0A, for example, was scored for BREAKFAST, COMBINATION, and LUNCH. These definitions appeared under the B's, C's, and L's in alphabetical order of their tagged words. These were merged into one definition for BRUNCH-.0A which preserved each of the individual tags, i.e.,

BRUNCH-.0A = A LATE BREAKFAST-.0A , AN EARLY LUNCH-1.1A , OR A COMBINATION-.2A OF THE TWO

This operation was carried out for every main entry which had been tagged for two or more words in its definition after which the data appeared as in figure 5-9. This format of the data is the closest to the original compact MPD form (figure 5-3) which was originally used to create the databases of chapter 6. With the addition of several tens of thousands of semantic disambiguations, it represents a significant computational advance over the plain undisambiguated data.

| | |
|---|---|
| AGE-1.2A | = THE TIME-1.3A OF LIFE AT WHICH SOME PARTICULAR QUALIFICATION IS ACHIEVED ; ESP |
| ARMAGEDDON-.0B | = THE SITE-.0A OR TIME-1.3A OF THIS |
| BREAKFAST-.0A | = THE FIRST MEAL-1.2A OF THE DAY |
| BRUNCH-.0A | = A LATE BREAKFAST-.0A , AN EARLY LUNCH-1.1A , OR A COMBINATION-.2A OF THE TWO |
| BUFFET-3.2B | = A MEAL-1.2A AT WHICH PEOPLE SERVE THEMSELVES ( AS FROM A BUFFET ) |
| CANDLELIGHT-.2A | = TIME-1.3A FOR LIGHTING UP |
| COMMENCEMENT-.1A | = THE ACT-1.1A% OR TIME-1.3A OF A BEGINNING |
| CONVENIENCE-.4A | = A SUITABLE TIME-1.3A |
| DEADLINE-.0A | = A DATE-2.4A OR TIME-1.3A BEFORE WHICH SOMETHING MUST BE DONE |
| DINNER-.0A | = THE MAIN MEAL-1.2A OF THE DAY ; ALSO |
| JUNCTURE-.3A | = A CRITICAL TIME-1.3A OR STATE-1.1A OF AFFAIRS |
| LUNCH-1.1A | = A LIGHT MEAL-1.2A USU. EATEN IN THE MIDDLE OF THE DAY |
| LUNCHEON-.0A | = A USU. FORMAL LUNCH-1.1A |
| MANANA-.0A | = AN INDEFINITE TIME-1.3A IN THE FUTURE |
| MEAL-1.2A | = AN ACT-1.1B OR THE TIME-1.3A OF EATING A MEAL-1.1A/+ |
| POTLUCK-.0A | = THE REGULAR MEAL-1.2A AVAILABLE TO A GUEST FOR WHOM NO SPECIAL PREPARATIONS HAVE BEEN MADE |
| SMORGASBORD-.0A | = A LUNCHEON OR SUPPER BUFFET-3.2B CONSISTING OF MANY FOODS ( AS HOT AND COLD MEATS , SMOKED AND PICKLED FISH , CHEESES , SALADS , AND RELISHES ) |
| SUPPER-.0A | = THE EVENING MEAL-1.2A WHEN DINNER IS TAKEN AT MIDDAY |
| TABLE-D'HOTE-.0A | = A COMPLETE MEAL-1.2A OF SEVERAL COURSES OFFERED AT A FIXED PRICE |
| TIME-1.3A | = A POINT-1.4A OR PERIOD-1.6A WHEN SOMETHING OCCURS |

Figure 5-9 Merged/Sorted/Merged Word-Senses Forming Single Definition

The data from figure 5-5 (sequenced taxonomic forest output) was then merged with the disambiguated definitions of figure 5-9, grouping all multiple occurrences of taxonomic output lines under their respective disambiguated text definitions. Figure 5-10 shows one of these groups as it appeared for BRUNCH-.0A.

```
                BRUNCH-.0A          = A LATE BREAKFAST-.0A , AN
                                      EARLY LUNCH-1.1A , OR A
                                      COMBINATION-.2A OF THE TWO
009025 03 BRUNCH-.0A
009030 03 BRUNCH-.0A
```

Figure 5-10 Input to Merging Program

The definitions text was added to each taxonomic occurrence and the data then sorted by sequence number and output as the fully completed trees such as in figure 5-11.

```
09013 -1 ----
09014 00 TIME-1.3A                = A POINT-1.4A OR PERIOD-1.6A WHEN
                                    SOMETHING OCCURS
09015 01 AGE-1.2A                 = THE TIME-1.3A OF LIFE AT WHICH
                                    SOME PARTICULAR QUALIFICATION IS
                                    ACHIEVED ; ESP
09016 01 ARMAGEDDON-.0B           = THE SITE-.0A OR TIME-1.3A OF THIS
09017 01 CANDLELIGHT-.2A          = TIME-1.3A FOR LIGHTING UP
09018 01 COMMENCEMENT-.1A          = THE ACT-1.1A% OR TIME-1.3A OF A
                                    BEGINNING
09019 01 CONVENIENCE-.4A          = A SUITABLE TIME-1.3A
09020 01 DEADLINE-.0A             = A DATE-2.4A OR TIME-1.3A BEFORE
                                    WHICH SOMETHING MUST BE DONE
09021 01 JUNCTURE-.3A             = A CRITICAL TIME-1.3A OR STATE-1.1A
                                    OF AFFAIRS
09022 01 MANANA-.0A               = AN INDEFINITE TIME-1.3A IN THE
                                    FUTURE
09023 01 MEAL-1.2A                = AN ACT-1.1B  OR THE TIME-1.3A OF
                                    EATING A MEAL-1.1A/+
09024 02 BREAKFAST-.0A            = THE FIRST MEAL-1.2A OF THE DAY
09025 03 BRUNCH-.0A               = A LATE BREAKFAST-.0A , AN EARLY
                                    LUNCH-1.1A , OR A COMBINATION-.2A
                                    OF THE TWO
09026 02 BUFFET-3.2B              = A MEAL-1.2A AT WHICH PEOPLE SERVE
                                    THEMSELVES ( AS FROM A BUFFET )
09027 03 SMORGASBORD-.0A          = A LUNCHEON OR SUPPER BUFFET-3.2B
                                    CONSISTING OF MANY FOODS ( AS HOT
                                    AND COLD MEATS , SMOKED AND
                                    PICKLED FISH , CHEESES , SALADS ,
                                    AND RELISHES )
09028 02 DINNER-.0A               = THE MAIN MEAL-1.2A OF THE DAY ;
                                    ALSO
09029 02 LUNCH-1.1A               = A LIGHT MEAL-1.2A USU.  EATEN IN
                                    THE MIDDLE OF THE DAY
09030 03 BRUNCH-.0A               = A LATE BREAKFAST-.0A , AN EARLY
                                    LUNCH-1.1A , OR A COMBINATION-.2A
                                    OF THE TWO
09031 03 LUNCHEON-.0A             = A USU.  FORMAL LUNCH-1.1A
09032 02 POTLUCK-.0A              = THE REGULAR MEAL-1.2A AVAILABLE
                                    TO A GUEST FOR WHOM NO SPECIAL
                                    PREPARATIONS HAVE BEEN MADE
09033 02 SUPPER-.0A               = THE EVENING MEAL-1.2A WHEN DINNER
                                    IS TAKEN AT MIDDAY
09034 02 TABLE-D'HOTE-.0A         = A COMPLETE MEAL-1.2A OF SEVERAL
                                    COURSES OFFERED AT A FIXED PRICE
```

Figure 5-11  Final Output

## 5.5 Display Problems

Once the forests of lexical nodes had been augmented with the definitions containing the merged disambiguator's codings, there still remained the problem of how to list these files in a usable form for subsequent analysis. The initial output of the forest had only been an indented display (figure 5-5) of word-senses. This format lacked definition texts and was hence incomplete. The final augmented version given in figure 5-11 had definitions, but the indented notation had been abandoned for processing reasons. A simple solution was to output the final tree with both definitions and indenting. Yet this was not entirely adequate since the definitions now spilled over standard line-printer width paper onto following lines and additionally had the same problems the original tree had concerning the inability to "track" the entries at the same depth across page boundaries. An expanse of blanks at the beginning of a line looks very much the same whether the actual depth is 8, 10 or 12 levels down. To solve this problem a program placing ! marks down the columns of indentation to facilitate vertical tracking was employed. A sample of this type of output is given in Figure 5-12.

```
0 TIME-1.3A     = A POINT-1.4A OR PERIOD-1.6A WHEN
0                 SOMETHING OCCURS
1 ! AGE-1.2A      = THE TIME-1.3A OF LIFE AT WHICH SOME
1 !                 PARTICULAR QUALIFICATION IS ACHIEVED
1 !                 ; ESP
1 ! ARMAGEDDON-.0B        = THE SITE-.0A  OR TIME-1.3A OF
1 !                         THIS
1 ! CANDLELIGHT-.2A       = TIME-1.3A FOR LIGHTING UP
1 ! COMMENCEMENT-.1A      = THE ACT-1.1A% OR TIME-1.3A OF
1 !                         A BEGINNING
1 ! CONVENIENCE-.4A       = A SUITABLE TIME-1.3A
1 ! DEADLINE-.0A = A DATE-2.4A OR TIME-1.3A BEFORE WHICH
1 !                 SOMETHING MUST BE DONE
1 ! JUNCTURE-.3A = A CRITICAL TIME-1.3A OR STATE-1.1A OF
1 !                 AFFAIRS
1 ! MANANA-.0A    = AN INDEFINITE TIME-1.3A IN THE FUTURE
1 ! MEAL-1.2A     = AN ACT-1.1B  OR THE TIME-1.3A OF
1 !                 EATING A MEAL-1.1A/+
2 ! ! BREAKFAST-.0A         = THE FIRST MEAL-1.2A OF THE DAY
3 ! ! ! BRUNCH-.0A    = A LATE BREAKFAST-.0A , AN EARLY
3 ! ! !                 LUNCH-1.1A , OR A COMBINATION-.2A OF
3 ! ! !                 THE TWO
2 ! ! BUFFET-3.2B  = A MEAL-1.2A AT WHICH PEOPLE SERVE
2 ! !                 THEMSELVES ( AS FROM A BUFFET )
3 ! ! ! SMORGASBORD-.0A       = A LUNCHEON OR SUPPER
3 ! ! !                         BUFFET-3.2B CONSISTING OF
3 ! ! !                         MANY FOODS ( AS HOT AND
3 ! ! !                         COLD MEATS , SMOKED AND
3 ! ! !                         PICKLED FISH , CHEESES ,
3 ! ! !                         SALADS , AND RELISHES )
2 ! ! DINNER-.0A    = THE MAIN MEAL-1.2A OF THE DAY ; ALSO
2 ! ! LUNCH-1.1A.0A         = A LIGHT MEAL-1.2A USU EATEN IN
2 ! !                         THE MIDDLE OF THE DAY
3 ! ! ! BRUNCH-.0A    = A LATE BREAKFAST-.0A , AN EARLY
3 ! ! !                 LUNCH-1.1A , OR A COMBINATION-.2A OF
3 ! ! !                 THE TWO
3 ! ! ! LUNCHEON-.0A = A USU. FORMAL LUNCH-1.1A
2 ! ! POTLUCK-.0A  = THE REGULAR MEAL-1.2A AVAILABLE TO A
2 ! !                 GUEST FOR WHOM NO SPECIAL PREPARATIONS
2 ! !                 HAVE BEEN MADE
2 ! ! SUPPER-.0A    = THE EVENING MEAL-1.2A WHEN DINNER IS
2 ! !                 TAKEN AT MIDDAY
2 ! ! TABLE-D'HOTE-.0A       = A COMPLETE MEAL-1.2A OF
2 ! !                         SEVERAL COURSES OFFERED AT A
2 ! !                         FIXED PRICE
```

Figure 5-12  Display Format for Portion of Noun Tree

There nevertheless remains a problem of gaining perspective on a tree which may span 10 or more pages. To assist in establishing one's location in the forest, a "trace" of the path above each entry starting a new page of the output is given at the top of that page. If the last entry in figure 5-12 had appeared instead at the top of the next page, the trace for that page would have been,

TIME-1.3A.....MEAL-1.2A.....TABLE-D'HOTE-.0A....

Another problem is the size of the output data for the whole noun forest as redundantly enumerated in the 153,000 lines of noun trees. The possibility of assembling a "minimal spanning forest" which would only include those trees which were not duplicated elsewhere has been considered, but any effort to display this must contend with the fact that it is a "tangled" hierarchy. To see what this means consider the tangled segment shown in figure 5-13.

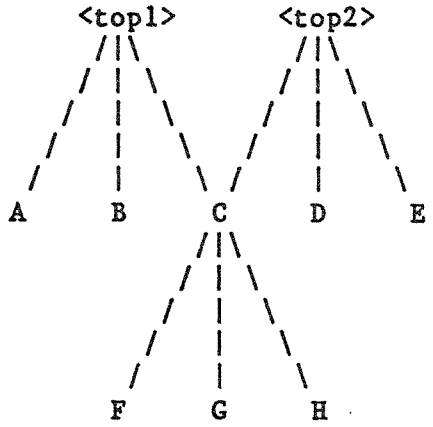



Figure 5-13 Tangled Hierarchy Segment

To enumerate this structure in a tree format such as figure 5-11 requires three top-level trees <top1>, <top2>, and C, each of which includes the exposition of all nodes below them. Thus the output would appear as in figure 5-14.

```
00 ----          00 ----          00 ----
01 <top1>        01 <top2>        01 C
02   A           02   C           02 | F
02   B           03   | F         02 | G
02   C           03   | G         02 | H
03   | F         03   | H
03   | G         02   D
03   | H         02   E
```

Figure 5-14 Linear Display of a Tangled Hierarchy

Within the requirements of the minimal spanning forest, the exposition of the separate tree for C would be eliminated, but the redundant exposition of the C tree appearing under <top1> and <top2> would remain. As C's tree might be several thousand nodes in extent itself the dilemma becomes clear. Thus, a minimal spanning forest might not in itself remove all the redundancy from display of large tangled hierarchies. One could, optimally, trace the duplicate downward path through C from <top1> and <top2> and note that one should look up the tree under C as a separate segment when details below that node are desired. However, this decision would be most annoying if C were a small tree and had less than a dozen nodes below it. Finally, if we introduce redundancy into the exposition of the minimal spanning forest to solve this problem we have nearly come full circle for it was the effort to remove this redundancy which prompted the development of the minimal spanning forest.

Some treatment of this difficulty has been attempted by the lexicographers designing the ERIC Thesaurus of Descriptors in their seventh edition. The descriptors of the ERIC classification system are a tangled hierarchy of the same form as the MPD's lexical material. The designers devised several display techniques for their vocabulary, one of which produces an indented tree both upward and downward from every node. The node itself is represented as the left-most word in the dual-tree, with those lines above being nodes higher up in the hierarchy, one level indented to the right for each level ABOVE the node. Below the node, a similar tree is output, but here the descriptors are indented one level for each level below the node. The column alignment technique I employed (using exclamation marks) is mimicked by periods below the node described and colons above the node described (figure 5-15).

```
<top1>          <top2>          : <top1>
  . A             . C           : <top2>
  . B             . . F       C
  . C             . . G         . F
  . . F           . . H         . G
  . . G           . D           . H
  . . H           . E
```

Figure 5-15 ERIC-Thesaurus-type display of Tangled Hierarchy

This appears to work for a small tangled hierarchy such as the several hundred nodes in the ERIC classification system, but probably suffers from the same context problems of other exposition techniques when the individual trees begin to span several pages. In the ultimate analysis it may simply be the case that the data cannot be displayed with its full contexts and still remain intelligible. A form of "localized" display, in which a limited number of levels above or below is given might prove superior. This would require further experimentation.

# CHAPTER VI A TAXONOMY FOR ENGLISH NOUNS AND VERBS

## 6.1 "Tangled" Hierarchies of Nouns and Verbs

The grant, MCS77-01315, "Development of a Computational Methodology for Deriving Natural Language Semantic Structures via Analysis of Machine-Readable Dictionaries", created a taxonomy for the nouns and verbs of the Merriam-Webster Pocket Dictionary (MPD), based upon the hand-disambiguated kernel words in their definitions. This taxonomy confirmed the expected structure of the lexicon to be that of a "tangled hierarchy" [Fahlman 1975, 1977] of unprecedented size (24,000 noun senses, 11,000 verb senses). This data base is believed to be the first to be assembled which is representative of the structure of the entire English lexicon. (A somewhat similar study of the Italian lexicon has been done [Alinei 1974, Lee 1977]). The content categories agree substantially with the semantic structure of the lexicon proposed by Nida [1975], and the topmost elements of the verb taxonomy confirm the primitives proposed by the San Diego LNR group [Norman and Rumelhart 1975].

This "tangled hierarchy" (also termed a "tangled tree") may be described as a formal data structure whose bottom is a set of terminal disambiguated words that are not used as kernel defining terms; these are the most specific elements in the structure. The tops of the structure are senses of words such as "cause", "thing", "class", "being", etc. These are the most general elements in the tangled hierarchy. If all the top terms are considered to be members of the metaclass "<word-sense>", the tangled forest becomes a tangled tree (see figure 6-1).

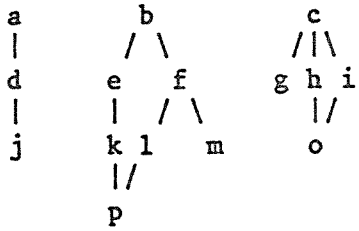



Figure 6-1a A Tangled Forest of Three Tangled Trees

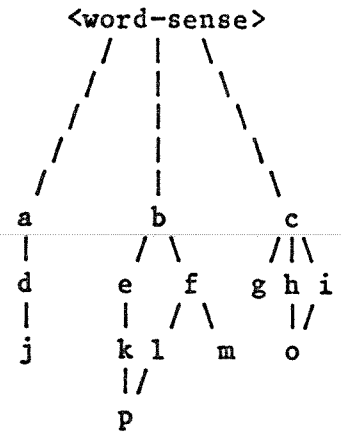



Figure 6-1b A Tangled Tree under the metaclass <word-sense>

The terminal nodes of such trees are in general each connected to the top in a lattice. An individual lattice can be resolved into a set of "traces", each of which describes an alternate path from terminal to top. In a trace each element implies the terms above it, and further specifies the sense of the elements below it.

The collection of lattices forms a transitive acyclic digraph (or perhaps more clearly, a "semi-lattice", that is, a lattice with a greatest upper bound, <word-sense>, but no least lower bound). If we specify all the traces composing such a structure, spanning all paths from top to bottom, we have topologically specified the semi-lattice. Thus the list on the left in figure 6-2 topologically specifies the tangled hierarchy on its right.

(a b c e f)
(a b c g k)
(a b d g k)
(a b c g l)
(a b d g l)
(a b c g m)
(a b d g m)
(a b d i)

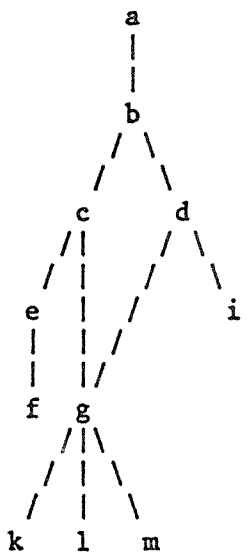



Figure 6-2 The Trace of a Tangled Hierarchy

### 6.1.1 Topmost Semantic Nodes of the Tangled Hierarchies

Turning from the abstract description of the forest of tangled hierarchies to the actual data, the first question which was answered was, "What are the largest tangled hierarchies in the dictionary?". The size of a tangled hierarchy is based upon two numbers, the maximum depth below the "root" and the total number of nodes transitively reachable from the root. Thus the tangled hierarchy of figure 6-2 has a depth of 5 and contains a total of 11 nodes (including the "root" node, "a"). However, since each non-terminal in the tangled hierarchy was also enumerated, it is also possible to describe the "sizes" of the other nodes reachable from "a". Their number of elements and depths are given in table 6-1.

Table 6-1 Enumeration of Tree Sizes and Depths of Tangled Hierarchy Nodes of Figure 6-2

| Tree Size | Maximum Depth | Root Node |
|---|---|---|
| 11 | 5 | a |
| 10 | 4 | b |
| 6 | 3 | c |
| 6 | 2 | d |
| 4 | 1 | g |
| 2 | 1 | e |

This example is given to demonstrate the inherent consequences of dealing with tree sizes based upon these measurements. For example, "g" has the most single-level descendants, 3, yet it is neither at the top of the tangled hierarchy, nor does it have the highest total number of descendants. The root node "a" is at the top of the hierarchy, yet it only has 1 single-level descendant. For nodes to be considered of major importance in a tangled hierarchy it is thus necessary to consider not only their total number of descendants, but whether these descendants are all actually immediately under some other node to which this higher node is attached. As we shall see, the nodes which have the most single-level descendants are actually more pivotal concepts in some cases.

Turning to the actual forest of tangled hierarchies, table 6-2 gives the size (number of nodes) and depth of the largest noun hierarchies and table 6-3 gives the sizes alone for verb hierarchies (depths were not computed for these).

Table 6-2 Sizes (Number of Nodes) and Maximum Depths of MPD Tangled Noun Hierarchies

| Size | Depth | | Size | Depth | |
|---|---|---|---|---|---|
| 3379 | 10 | ONE-2.1A | 493 | 8 | DEGREE-.1A |
| 2121 | 12 | BULK-1.1A | 477 | 15 | COURSE-1.1B |
| 1907 | 10 | PARTS-1.1A/! | 470 | 14 | WAY-.10A |
| 1888 | 10 | SECTIONS-.2A/! | 467 | 9 | SCOPE-1.2A |
| 1887 | 9 | DIVISION-.2A | 467 | 13 | PROCEDURE-.1A |
| 1832 | 9 | PORTION-1.4A | 465 | 10 | FORCE-1.1A/+ |
| 1832 | 8 | PART-1.1A | 464 | 10 | STRENGTH-.4A/+ |
| 1486 | 14 | SERIES-.0A | 463 | 9 | INTENSITY-.2A |
| 1482 | 18 | SUM-1.1A | 463 | 7 | EXTENT-.2A |
| 1461 | ** | AMOUNT-2.2A | 463 | 8 | DEGREE-.2A |
| 1459 | 8 | ACT-1.1B | 454 | 12 | METHOD-.1A |
| 1414 | ** | TOTAL-2.0A | 413 | 5 | SUBSTANCE-.2A |
| 1408 | 15 | NUMBER-1.1A | 408 | 6 | OBJECT-1.1A |
| 1379 | 14 | AMOUNT-2.1A | 388 | 7 | REGION-.0A |
| 1337 | 6 | ONE-2.2A | 388 | 8 | AREA-.3A |
| 1204 | 5 | PERSON-.1A | 378 | 6 | MASS-1.4A/! |
| 1201 | 14 | OPERATIONS-.1A/+ | 377 | 5 | PIECE-1.2A |
| 1190 | ** | PROCESS-1.4A | 373 | 5 | BUILDING-.1A |
| 1190 | 14 | ACTIONS-.2A/+ | 355 | 6 | MASS-1.1A |
| 1123 | 6 | GROUP-1.0A/! | 354 | 5 | EQUIPMENT-.2B/+ |
| 1101 | 12 | FORM-1.13A | 349 | 8 | CHARACTER-.2B |
| 1089 | 12 | VARIETY-.4A | 347 | 7 | QUALITY-.1A |
| 1083 | 11 | MODE-.1A | 338 | 5 | MECHANISM-.1B/! |
| 1076 | 10 | STATE-1.1A | 337 | 5 | THING-.6A |
| 1076 | 9 | CONDITION-1.3A | 337 | 4 | DEVICE-.2A |
| 1068 | 13 | MEASUREMENT-1.2A | 325 | 7 | TIME-1.1A/+ |
| 1068 | ** | DIMENSION-.1A | 298 | 5 | TREES-1.1A/+ |
| 1061 | ** | LENGTH-.1B | 298 | 6 | PERIOD-1.6A |
| 1061 | ** | DISTANCE-1.1A | 297 | 5 | MUSHROOMS-1.0A/+ |
| 1061 | 14 | DIMENSIONS-.1A | 296 | 4 | PLANT-2.1A |
| 1060 | 11 | SIZE-1.0A | 288 | 7 | EVENT-.1A |
| 1060 | 13 | MEASURE-1.2A | 277 | 5 | GROWTH-.2A/+ |
| 1060 | 10 | EXTENT-.1A | 274 | 4 | PRODUCT-.1A |
| 1060 | 14 | CAPACITY-.2A | 265 | 8 | PART-1.1A/+ |
| 869 | 7 | HOUSE-1.1A/+ | 260 | 8 | STATE-1.2A |
| 836 | 7 | SUBSTANCE-.2B | 254 | 5 | MEANS-3.2A |
| 836 | 8 | MATTER-1.4A | 248 | 12 | PROGRAM-1.2A |
| 741 | 8 | NEWS-.2A/+ | 248 | 11 | PLAN-1.2A |
| 740 | 6 | PIECE-1.2B | 245 | 8 | RESULT-2.1A |
| 740 | 7 | ITEM-.2A | 239 | 6 | SPOT-1.3A |
| 686 | 7 | ELEMENTS-.1A | 238 | 6 | LOCATION-.2A |
| 684 | 6 | MATERIAL-2.1A | 237 | 6 | SITUATION-.1A |
| 647 | 9 | THING-.4A | 236 | 5 | LOCALITY-.0A |
| 642 | 8 | ACT-1.1A | 236 | 5 | ARTICLE-.4A |
| 535 | 6 | THINGS-.5A/! | 235 | 4 | PLACE-1.3A |
| 533 | 6 | MEMBER-.2A | 231 | 6 | MATTER-1.4A/+ |
| 503 | 10 | PLANE-4.1A | 226 | 6 | PLACE-1.2A |
| 495 | 6 | STRUCTURE-.2A | 224 | 4 | UNIT-.3A |
| 494 | 10 | RANK-2.4A | 215 | 6 | ABILITY-.0B |
| 493 | 9 | STEP-1.3A | 210 | 6 | INSTANCE-1.3A |

(Note: ** = out of range due to data error)

Table 6-3 Sizes (Number of Nodes) of Topmost Tangled Verb Hierarchies

| Size | | Size | |
|---|---|---|---|
| 4175 | REMAIN-.4A | 233 | COME-.3A |
| 4175 | CONTINUE-.1A | 220 | BRING-.1A |
| 4087 | MAINTAIN-.3A | 220 | MAINTAIN-.3B |
| 4072 | STAND-1.6A | 219 | GO-1.16A |
| 4071 | HAVE-1.3A | 201 | SET-1.2A |
| 4020 | BE-.1B | 194 | PLACE-2.2A |
| 3500 | EQUAL-3.0A | 177 | EXPRESS-3.1A |
| 3498 | BE-.1A | 171 | SEND-.1A |
| 3476 | CAUSE-2.0A | 166 | PERFORM-.2A |
| 1316 | APPEAR-.3A/C | 165 | PUT-.4A |
| 1285 | EXIST-.1A/C | 159 | PROCEED-.5A |
| 1280 | OCCUR-.2A/C | 157 | GIVE-1.9A |
| 1279 | MAKE-1.1A | 153 | TAKE-1.9A |
| 567 | GO-1.1B | 150 | BE-.4A |
| 439 | BRING-.2A | 149 | CONTINUE-.1A |
| 401 | MOVE-1.1A | 149 | REMAIN-.4A |
| 366 | GET-1.1A | 140 | UNDERTAKE-.1A |
| 365 | GAIN-2.1A | 139 | HAVE-1.1A/C |
| 334 | DRIVE-1.1A/+ | 138 | RECEIVE-.1A/C |
| 333 | PUSH-1.1A | 138 | TAKE-1.18A |
| 328 | PRESS-2.1B | 137 | GIVE-1.15A |
| 308 | CHANGE-1.1A | 136 | FOLLOW-.4A |
| 289 | MAKE-1.10A | 128 | MOVE-1.3A |
| 282 | COME-.1A | 127 | COVER-1.1A |
| 288 | CHANGE-1.1A | 127 | FORM-2.1B/+ |
| 283 | EFFECT-2.1A | 125 | JOIN-.1A |
| 282 | ATTAIN-.2BN | 115 | HIT-1.1B |
| 281 | FORCE-2.3A | 115 | POINT-2.6A |
| 273 | PUT-.1A | 114 | KEEP-1.9A |
| 246 | IMPRESS-3.2A | 112 | TURN-1.11A/C |
| 245 | URGE-1.4A | 111 | DIRECT-1.2A |
| 244 | DRIVE-1.1A | 107 | PUT-.10A/-INTO-ACT |
| 244 | IMPEL-.0A | 106 | EXERT-.0A |
| 244 | THRUST-1.1A | 102 | STRIKE-1.2A |
| 235 | REACH-1.4A*TO | 102 | TOUCH-1.3A |

While the verb tangled hierarchy appears to have a series of nodes above CAUSE-2.0A which have large numbers of descendants, the actual structure more closely resembles that of figure 6-3.

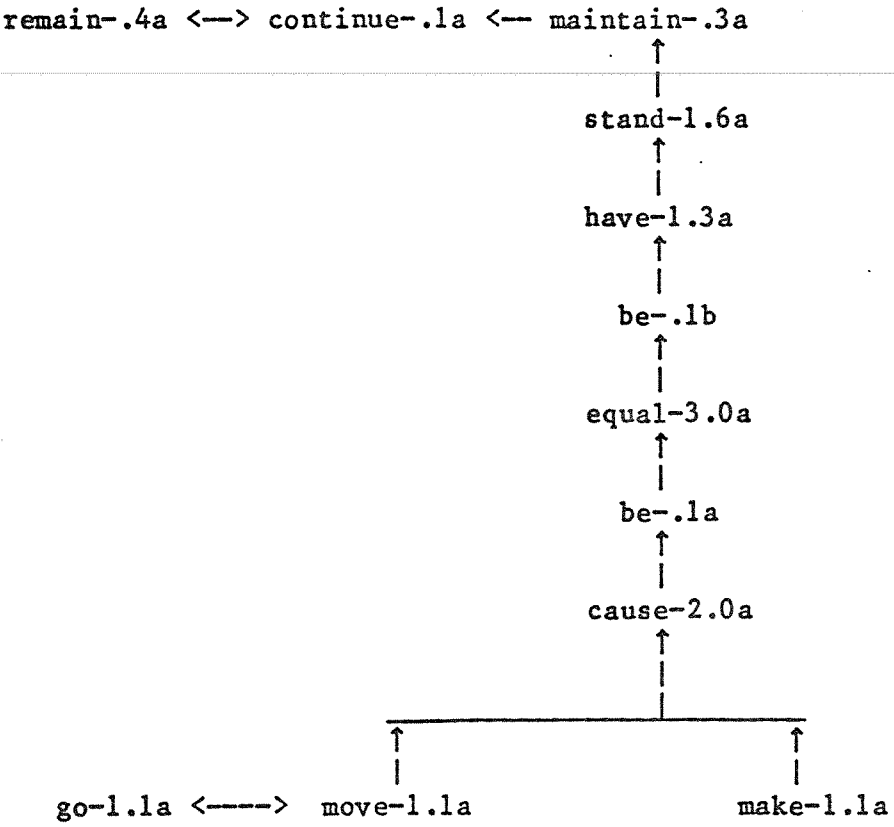



Figure 6-3 Relations between topmost Tangled Verb Hierarchy Nodes

Although the topmost nodes (REMAIN-.4A, CONTINUE-.1A, MAINTAIN-.3A) are ultimately connected to thousands of descendants, they have only one immediate descendant which connects them to the rest of the vast tree they dominate. Figure 6-3 shows this graphically. There is thus a chain of nodes whose ratio of immediate descendants to total descendants is very small until one reaches CAUSE-2.0A and MAKE-1.1A. CAUSE-2.0A has approximately 240 direct descendants and MAKE-1.1A has 480 direct descendants, making these two the topmost nodes in terms of the number of direct descendants, though they are ranked 9th and 13th in terms of the number of total descendants.

This points out in practice what the abstract tree of figure 6-2 showed as possible in theory, and explains the seeming contradiction in sizes occasioned by having a basic verb such as CAUSE-2.0A defined in terms of a lesser verb such as REMAIN-.4a. The total number of descendants is not necessarily as important as the structure of the tree under a given node. The number of immediate descendants and the ratio of

this number to the number of total descendants may be more significant than the size of the total tree a node dominates.

Why should this be so? The difficulty is explainable given two facts. First, the lexicographers HAD to define CAUSE-2.0A using some other verb, etc. This is inherent in the lexicon being used to define itself. Second, once one reaches the top of a tangled hierarchy one cannot go any higher -- and consequently forcing further definitions for basic verbs such as "be" and "cause" invariably leads to using more specific verbs, rather than more general ones. The situation is neither erroneous, nor inconsistent in the context of a self-defined closed system and will be discussed further in the section on noun primitives.

## 6.2 Noun Primitives

One phenomenon which was expected in computationally grown trees was the existence of loops. Loops are caused by having sequences of interrelated definitions whose kernels form a ring-like array [Sparck-Jones 1967; Calzolari 1977]. However, what was not expected was how important such clusters of nodes would be both to the underlying basis for the taxonomies and as primitives of the language. Such circularity is sometimes evidence of a truly primitive concept, such as the set containing the words CLASS, GROUP, TYPE, KIND, SET, DIVISION, CATEGORY, SPECIES, INDIVIDUAL, GROUPING, PART and SECTION. To understand this, consider the subset of interrelated senses these words share (figure 6-3) and then the graphic representation of these in figure 6-4.

GROUP 1.0A - a number of individuals related by a common factor (as physical association, community of interests, or blood)

CLASS 1.1A - a group of the same general status or nature

TYPE 1.4A - a class, kind, or group set apart by common characteristics

KIND 1.2A - a group united by common traits or interests
KIND 1.2B - CATEGORY

CATEGORY .0A - a division used in classification ; also
CATEGORY .0B - CLASS, GROUP, KIND

DIVISION .2A - one of the parts, sections, or groupings into which a whole is divided

*GROUPING <== W7 - a set of objects combined in a group

SET 3.5A - a group of persons or things of the same kind or having a common characteristic usu. classed together

SORT 1.1A - a group of persons or things that have similar characteristics
SORT 1.1B - CLASS

SPECIES .1A - SORT, KIND
SPECIES .1B - a taxonomic group comprising closely related organisms potentially able to breed with one another

Key:
* The definition of an MPD run-on, taken from Merriam-Webster Seventh Collegiate Dictionary to supplement the set.

Figure 6-4 Noun Primitive Concept Definitions

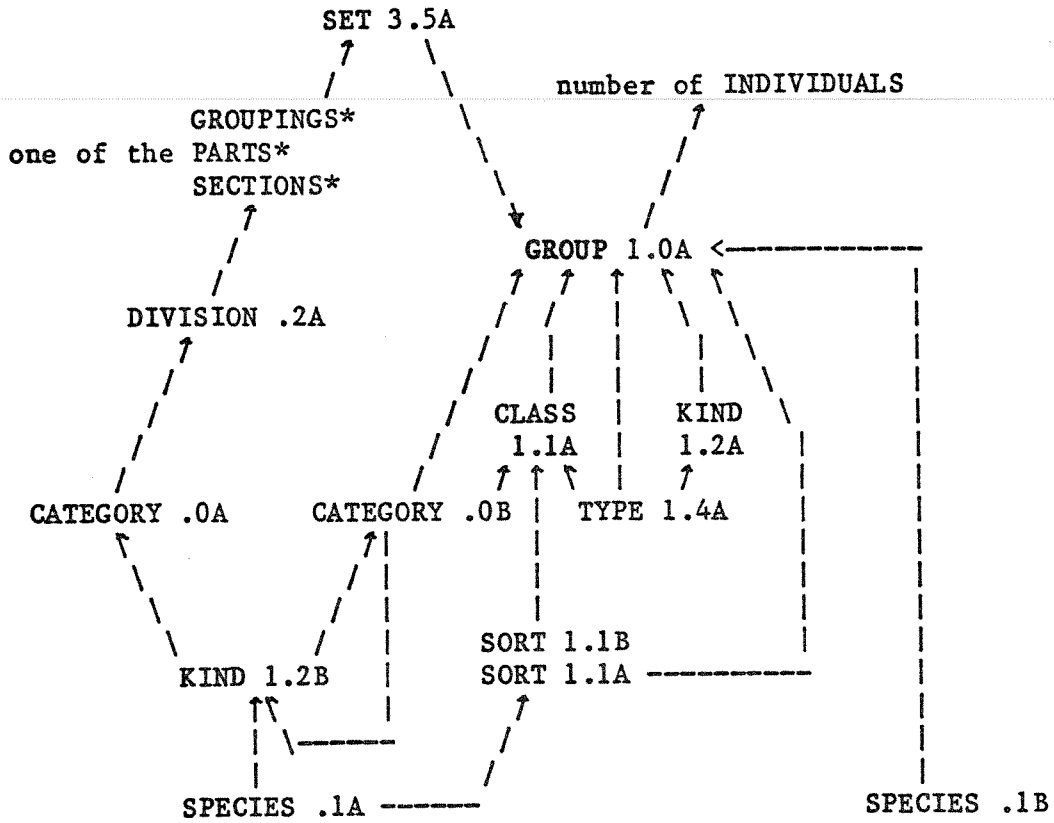



Figure 6-5 "GROUP" Concept Primitive from Dictionary Definitions

* Note: SECTIONS, PARTS, and GROUPINGS have additional connections not shown which lead to a related primitive cluster dealing with the PART/WHOLE concept.

This complex interrelated set of definitions comprises a primitive concept, essentially equivalent to the notion of SET in mathematics. The primitiveness of the elements is evident when one attempts to define any one of these words without using another of them in that definition. This essential property, the inability to write a definition explaining a word's meaning without using another member of some small set of near synonymous words, is the basis for describing such a set as a PRIMITIVE. It is based upon the notion of definition given by Wilder [1965], which in turn was based upon a presentation of the ideas of Padoa [1900], a turn-of-the-century logician.

The definitions are given; the disambiguation of their kernel's senses leads to a cyclic structure which cannot be resolved by attributing erroneous judgements to either the lexicographer or the disambiguator; therefore the structure is taken as representative of an undefinable primitive concept, and the words whose definitions participate in this complex structure are found to be undefinable without reference to the other members of the set of undefined terms.

The question of what to do with such primitives is not really a problem, as Winograd [1978] notes, once one realizes that they must exist at some level, just as mathematical primitives must exist. In tree construction the solution is to form a single node whose English surface representation may be selected from any of the words in the primitive set. There probably are connotative differences between the members of the set, but the ordinary pocket dictionary does not treat these in its definitions with any detail. The Merriam-Webster Collegiate Dictionary does the connotative differences between words sharing a "ring".

While numerous studies of lexical domains such as the verbs of motion [Miller 1972; Abrahamson 1975] (see also chapter 3) and possession [Gentner 1975] have been carried out by other researchers, it is worth noting that recourse to using ordinary dictionary definitions as a source of material has received little attention. Yet the "primitives" selected by Donald A. Norman, David E. Rumelhart, and the LNR Research Group for knowledge representation in their system bear a remarkable similarity to those verbs used most often as kernels in The Merriam-Webster Pocket Dictionary and Donald Sherman has shown (table 6-4) these topmost verbs to be among the most common verbs in the Collegiate Dictionary as well [Sherman 1979]. The most frequent verbs of the MPD are, in descending order, MAKE, BE, BECOME, CAUSE, GIVE, MOVE, TAKE, PUT, FORM, BRING, HAVE, and GO. The similarity of these verbs to those selected by the LNR group for their semantic representations, i.e., BECOME, CAUSE, CHANGE, DO, MOVE, POSS ("have"), TRANSF ("give","take"), etc., [Munro 1975; Rumelhart and Levin 1975; Gentner 1975] is striking. This similarity is indicative of an underlying "rightness" of dictionary definitions and supports the proposition that the lexical information extractable from study of the dictionary will prove to be the same knowledge needed for computational linguistics.

Table 6-4 50 Most Frequent Verb Infinitive Forms of W7 Verb Definitions (from [Sherman 1979]).

| | | | |
|---|---|---|---|
| 1878 | MAKE | 157 | FURNISH |
| 908 | CAUSE | 154 | TURN |
| 815 | BECOME | 150 | GET |
| 599 | GIVE | 150 | TREAT |
| 569 | BE | 147 | SUBJECT |
| 496 | MOVE | 141 | HOLD |
| 485 | TAKE | 137 | UNDERGO |
| 444 | PUT | 132 | CHANGE |
| 366 | BRING | 132 | USE |
| 311 | HAVE | 129 | KEEP |
| 281 | FORM | 127 | ENGAGE |
| 259 | GO | 127 | PERFORM |
| 240 | SET | 118 | BREAK |
| 224 | COME | 118 | REDUCE |
| 221 | REMOVE | 112 | EXPRESS |
| 210 | ACT | 107 | ARRANGE |
| 204 | UTTER | 107 | MARK |
| 190 | PASS | 106 | SEPARATE |
| 188 | PLACE | 105 | DRIVE |
| 178 | COVER | 104 | CARRY |
| 173 | CUT | 101 | THROW |
| 169 | PROVIDE | 100 | SERVE |
| 166 | DRAW | 100 | SPEAK |
| 163 | STRIKE | 100 | WORK |

The enumeration of the primitives for nouns and verbs by analysis of the tangled hierarchies of the noun and verb forests grown from the MPD definitions is a considerable undertaking and one which goes beyond the scope of this dissertation. A technique for identifying such groups can be defined however, as in figure 6-6.

Steps: (1) Determine candidates for inclusion in a set of primitives by the following procedures:

(a) A node may be in a primitive set if it was in a loop discovered during the operation of the tree-growing program and was labeled ***LOOP*** on the output.

(b) A node may be in a primitive set if it is a top-level element of a tree.

(2) For each primitive candidate accepted look up the definition of its kernel and add that to the primitive set. Apply this procedure repeatedly until no further new members of the set are encountered.

(3) Diagram the primitive family, noting circuits, and designate the nodes in the tree involved in the set as primitives, i.e., as equivalent to each other and without further expansion via conventional taxonomic techniques (see figure 3-1).

Figure 6-6 Technique III - Identification of Primitives

To see how this technique works in practice, consider the discovery of the primitive group starting from PLACE-1.3A.

place-1.3a - a building or locality used for a special purpose

The kernels of this definition are "building" and "locality". Looking these up in turn we have:

building-.1a - a usu. roofed and walled structure (as a house) for permanent use
locality-.0a - a particular spot, situation, or location

This gives us four new terms, "structure", "spot", "situation", and "location". Looking these up we find the circularity forming the primitive group.

structure-.2a - something built (as a house or a dam)
spot-1.3a - LOCATION, SITE
location-.2a - SITUATION, PLACE
situation-.1a - location, site

And finally, the only new term we encounter is "site" which yields,

site-.0a - location <* of a building> <battle *>

The primitive cluster thus appears as in figure 6-7.

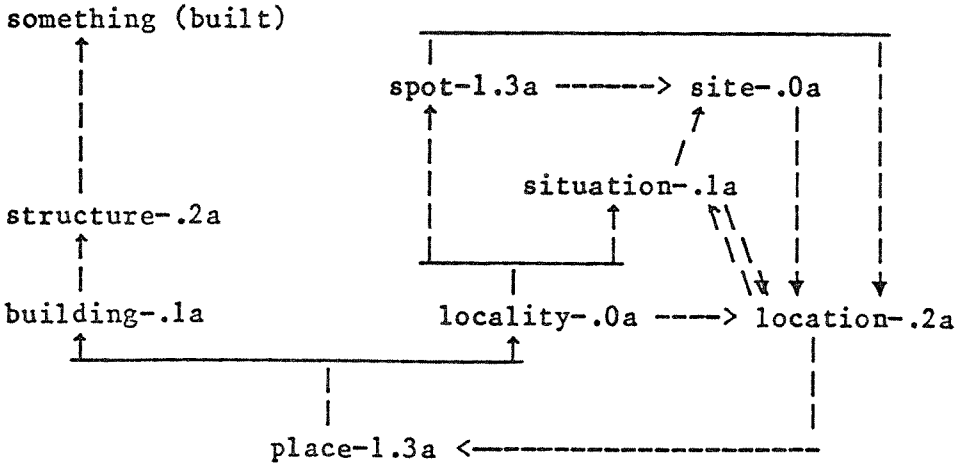



Figure 6-7 Diagram of Primitive Set Containing PLACE, LOCALITY, SPOT, SITE, SITUATION, and LOCATION

## 6.3 Nouns Terminating in Relations to Other Nouns or Verbs

In addition to terminating in "dictionary circles" or "loops", nouns also terminate in definitions which are actually text descriptions of case arguments of verbs or relationships to other nouns. "Vehicle" is a fine example of the former, being as it were the canonical instrumental case argument of one sense of the verb "carry" or "transport".

vehicle - a means of carrying or transporting something

"Leaf" is an example of the latter, being defined as a part of a plant,

leaf - a usu. flat and green outgrowth of a plant stem that is a unit of foliage and functions esp. in photosynthesis.

Thus "leaf" is not a type of anything. Even though under a strictly genus/species interpretation one would analyze "leaf" as being in an ISA relationship with "outgrowth", "outgrowth" has not a suitable homogeneous set of members and a better interpretation for modeling this definition would be to consider the "outgrowth of" phrase to signify a part/whole relationship between "leaf" and "plant".

Hence we may consider the dictionary to have at least two taxonomic relationships (i.e. ISA and ISPART) as well as additional relations explaining noun terminals as verb arguments. One can also readily see that there will be taxonomic interactions among nodes connected across these relationship "bridges".

While the parts of a plant will include the "leaves", "stem", "roots", etc., the corresponding parts of any TYPE of plant may have further specifications added to their descriptions. Thus "plant" specifies a functional form which can be further elaborated by descent down its ISA chain. For example, a "frond" is a type of "leaf",

frond - a usu. large divided leaf (as of a fern)

We knew from "leaf" that it was a normal outgrowth of a "plant", but now we see that "leaf" can be specialized, provided we get confirmation from the dictionary that a "fern" is a "plant". (Such confirmation is only needed if we grant "leaf" more than one sense meaning, but words in the Pocket Dictionary do typically average 2-3 sense meanings). The definition of "fern" gives us the needed linkage, offering,

fern - any of a group of flowerless seedless vascular green plants

Thus we have a specialized name for the "leaf" appendage of a "plant" if that plant is a "fern". This can be represented as in figure 6-8.

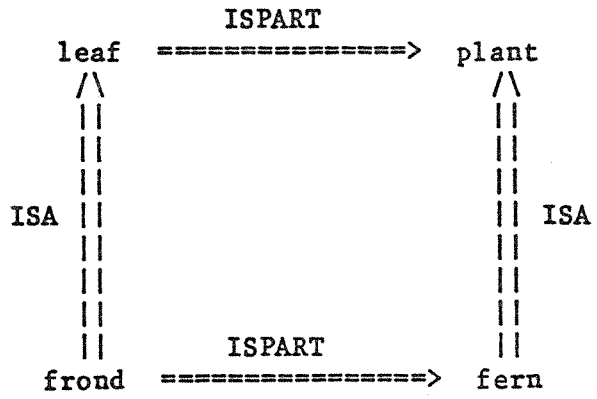



Figure 6-8 LEAF:PLANT::FROND:FERN

This conclusion that there are two major transitive taxonomies and that they are related is not of course new. Evens et al. [1976, 1977] have dealt with the PART-OF relationship as second only to the ISA relationship in importance, and Fahlman [1975, 1977] has also discussed the interaction of the PART-OF and ISA hierarchies. Historically even Raphael [1965] used a PART-OF relationship together with the ISA hierarchy of SIR's deduction system. What however is new is that I am not stating "leaf" is a part of a "plant" because of some need to use this fact within a particular system's operation, but because this has been "discovered" in a published reference source and results naturally from an effort to assemble the complete lexical structure of the dictionary.

The verb case argument relationship between "vehicle" and "carry" or "transport" likewise can be specialized. "Vehicles" transport "something"; "bookmobiles" (which are defined as being "trucks"; with "trucks" being "vehicles") transport "books". Likewise, "transport" is defined as,

transport - to convey from one place to another

And hence we now know that "vehicles" "convey something from one place to another". Looking down the "transport" tree we see that "transport" has a tree containing,

fly-1.12a - to transport by flying
sluice-2.3a - to transport (as logs) in a sluice
chauffeur-2.2a - to transport in the manner of a chauffeur
truck-4.1a - to transport on a truck
motor-2.0a - to travel or transport by automobile
raft-2.1a - to travel or transport by raft
freight-2.3a - to ship or transport by freight
bootleg-.0a - to make, transport, or sell (as liquor) illegally

This naturally leads to a similar diagram, figure 6-9, showing a relationship between "vehicle" and "transport".

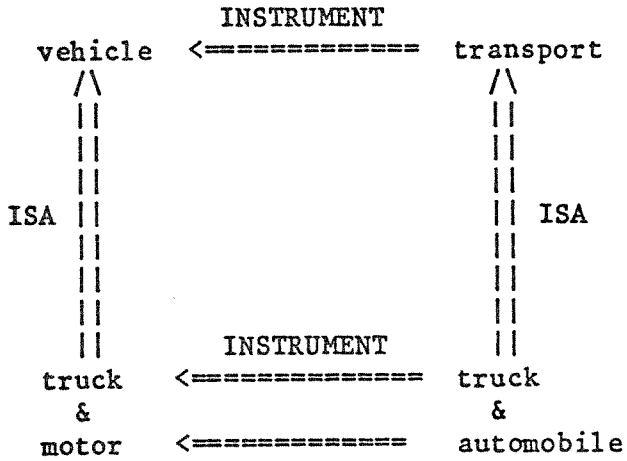



Figure 6-9 VEHICLE:TRANSPORT::TRUCK/MOTOR:TRUCK/AUTOMOBILE

There are two important differences here. First, the verb "transport" relates to "vehicle" via several different case-arguments. For instance, "in a sluice" refers to a non-actant INSTRUMENT, "in the manner of a chauffeur", "by freight", and "illegally" refer to the MANNER of transport (as well as "chauffeur" being the AGENT of "transport"). This illustrates the capability of a verb to tie together several nouns simultaneously, indicating verbs provide a powerful structural basis for uniting the dictionary.

Second, "raft", the noun, is an implied INSTRUMENT of "transport", the verb, because of the definition of "raft", the verb. Yet, the "vehicle" taxonomy does not include "raft" as a member. Looking to the dictionary we find the noun "rant" defined as,

raft-1.1a - a number of logs or timbers fastened together to form a float
raft-1.2a - a flat structure for support or transportation on water

The lack of a path between "vehicle" and "raft" indicates there is another means of creating taxonomies in the dictionary. A higher level concept can specify a relationship DOWNWARD as opposed to a kernel term specifying an upward ISA relationship. A "raft" is not a "vehicle" unless one asks whether a "raft" satisfies the definition of being a "vehicle". Thus a "raft" is a "vehicle" by deduction using a rule such as,

(transport INSTRUMENT x ==> x ISA vehicle)

which allows us to conclude that,

(transport INSTRUMENT raft ==> raft ISA vehicle).

The dictionary classification system can thus be extended by using such rules to logically add new elements to the existing explicit taxonomy. Additionally, a "raft" is defined as a "structure" for "transportation". "Transportation" is morphologically related to the verb "transport" and in section 10-1 this will be further discussed as the basis for extending dictionary knowledge.

## 6.4 Partitives and Collectives

As mentioned in section 6.3, the use of "outgrowth" in the definition of "leaf" causes problems in the taxonomy if we treat "outgrowth" as the true genus term of that definition. This word is but one example of a broad range of noun terminals which may be described as "partitives". A "partitive" may be defined as a noun which serves as a general term for a PART of another large and often non-homogeneous set of concepts. Additionally, at the opposite end of the partitive scale, there is the class of "collectives". Collectives are words which serve as a general term for a COLLECTION of other concepts.

In section 5.2.1 it was mentioned that the disambiguators often faced decisions as to whether some words were indeed the true semantic kernels of definitions, and often found additional words in the definitions which were more semantically appropriate to serve as the kernel -- albeit they did not appear syntactically in the correct position. Many of these terms were partitives and collectives. Figure 6-10 shows a set of partitives and collectives which were extracted and classified by Gretchen Hazard and John White during the dictionary project. The terms under "group names", "whole units", and "system units" are collectives. Those under "individuators", "piece units", "space shapes", "existential units", "locus units", and "event units" are partitives. These terms usually appeared in the syntactic frame "An _____ of" and this additionally served to indicate their functional role.

1.0 QUANTIFIERS

1.1 GROUP NAMES
- pair
- collection
- group
- cluster
- band (of people)
- bunch

1.2 INDIVIDUATORS
- member
- unit
- item
- article
- strand
- branch (of science,etc)

2.0 SHAPE UNITS

2.1 PIECE UNITS
- sample
- bit
- piece
- tinge
- tint

2.2 WHOLE UNITS
- mass
- stock
- body
- quantity
- wad

2.3 SPACE SHAPES
- bed
- layer
- strip
- belt
- crest
- fringe
- knot
- knob
- tuft

3.0 EXISTENTIAL UNITS

3.1 VARIANT
- version
- form
- sense

3.2 STATE
- state
- condition

4.0 REFERENCE UNITS

4.1 LOCUS UNITS
- place
- end
- ground
- point

4.2 PROCESS UNITS
- cause
- source
- means
- way
- manner

5.0 SYSTEM UNITS
- system
- course
- chain
- succession
- period

6.0 EVENT UNITS
- act
- discharge
- instance

7.0 EXCEPTIONS
- growth
- study

Figure 6-10 Examples of Partitives and Collectives (from [Amsler and White 1979]).

The breadth of the terms serving as collectives can also be shown by looking taxonomically at those words which are defined using GROUP as a collective or PART as a partive. Using an ad hoc feature classification system which seems appropriate to many of the members of the partive and collective set I have prepared the feature analysis tree of figure 6-11.

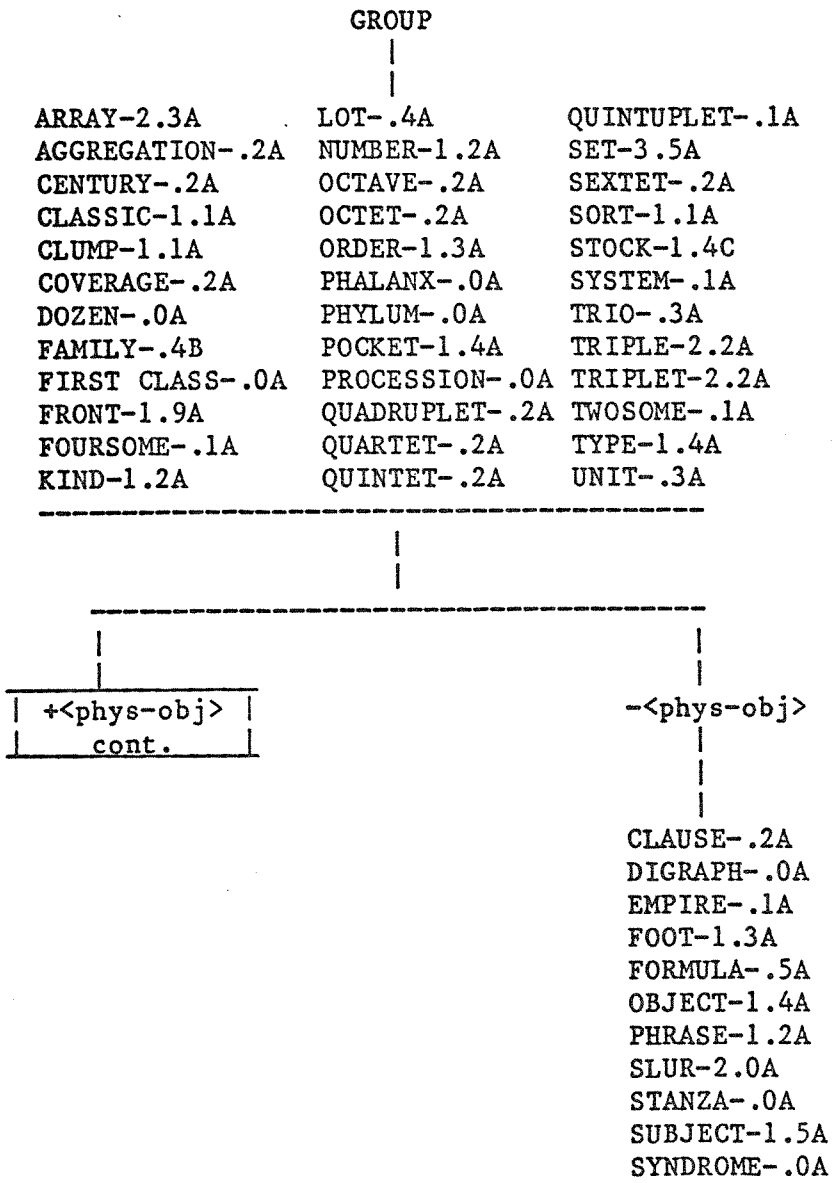



Figure 6-11 Componential "Feature" Analysis of GROUP Descendants

Note: Words at top in ARRAY-2.3A to UNIT-.3A list are unmarked for any specific features.

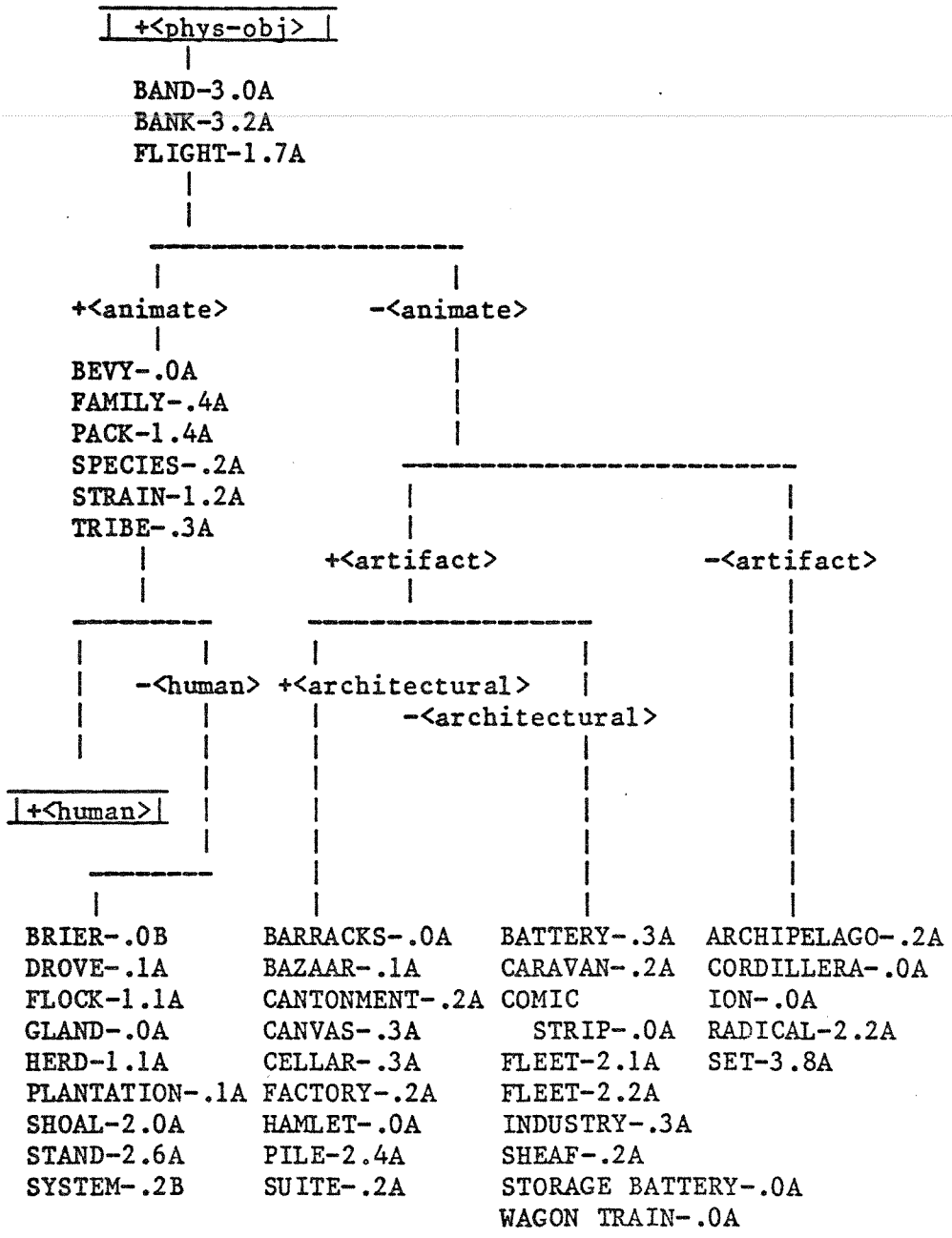
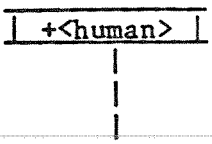



Figure 6-11 (cont.) Componential "Feature" Analysis of GROUP Descendants




| | | |
|---|---|---|
| BOARD-1.6A | ENSEMBLE-.4A | ORCHESTRA-.1A |
| BRIGADE-.2A | FACTION-.1A | ORDER-1.1A |
| CABAL-1.0A | FAMILY-.1A | PANEL-1.2A |
| CARAVAN-.1A | FAMILY-.2A | PARTY-.1A |
| CAST-2.7A | FAMILY-.3A | PARTY-.1B |
| CHORUS-1.2A | FIFTH COLUMN-.0A | PARTY-.2A |
| CIRCLE-1.5A | FLOCK-.1.2A | PARTY-.3A |
| CLAN-.0A | FOLD-1.1A | PUBLIC-2.2A |
| CLAQUE-.0A | FOLLOWING-2.0A | QUARTET-.2A |
| CLASS-1.4A | GANG-1.1A | QUINTET-.2A |
| CLASS-1.4B | HERD-1.A | RACE-3.2A |
| CLIQUE-.0A | HUDDLE-2.1A | RING-1.4A |
| CLUB-1.3A | INTEREST-1.4A | SECT-.3A |
| COHORT-.1A | JUNTA-.0A | SHIFT-2.3A |
| COLLEGIUM-.0A | KINDRED-1.1A | SQUAD-.1A |
| COLONY-.3A | LOBBY-1.2A | SQUAD-.2A |
| COMPANY-.4A | MACHINE-1.4A | STAFF-.4A |
| CONGREGATION-.1B | MANAGEMENT-.4A | STRATUM-.2B |
| CORPS-.2A | MESS-1.2A | SYNDICATE-1.1A |
| COTERIE-.0A | MIND-1.6A | TABLE-1.4A |
| CULT-.3B | MINORITY-.2B | TEAM-1.2B |
| DEFENSE-.5A | MISSION-.2A | TRIBE-.1A |
| DEMIMONDE-.2A | MONOPOLY-.3A | TRIBE-.2A |
| DYNASTY-.2A | MUSTER-2.2A | TROUPE-.0B |
| ELITE-.0B | NATIONALITY-.6A | UNDERGROUND-3.2A |
| EMBASSY-.2A | OLIGARCHY-.2B | VOTE-1.4A |

Figure 6-11 (cont.) Componential "Feature" Analysis of GROUP Descendants

# CHAPTER VII FUTURE WORK AND SPECULATION

## 7.1 Grammars for Dictionary Definitions

There are three common definition formats. These are the single text definition (approximately 1/2 of all definitions), e.g.,

a-1.0a - the 1st letter of the English alphabet
abate-.1a - to put an end to <abate a nuisance>

the synonymous cross-reference definition (approximately 1/4 of all definitions), e.g.,

abbey-.1a - MONASTERY , CONVENT
abase-.0a - HUMBLE , DEGRADE

and the combination of a single text definition followed by a synonymous cross-reference subsense (approximately 1/8 of all definitions), e.g.,

ability-.0a - the quality of being able
ability-.0b - POWER , SKILL
abandon-1.0a - to give up
abandon-1.0b - FORSAKE , DESERT

Other types of definition senses are restricted to portions of the remaining 1/8 of all definition types in frequency and include usage notes, e.g.,

abbé-.0a - a member of the French secular clergy -- used as a title

be-.4a - -- used with the past participle of transitive verbs as a passive voice auxiliary <the door was opened>

as well as text and synonymous cross-reference definitions conjoined by words such as "esp" and "also", e.g.,

abatement-.2a - an amount abated ; esp
abatement-.2b - a deduction from the full amount of a tax

abbreviate-.0a - SHORTEN , CURTAIL ; esp
abbreviate-.0b - to reduce to an abbreviation

abortion-.0a - a premature birth occurring before the fetus can survive ; also
abortion-.0b - an induced expulsion of a fetus

beat-1.4a - OVERCOME ; also
beat-1.4b - SURPASS

Within these definition forms, the text definition is clearly the only one requiring extensive parsing effort. It is this type of definition with which we shall be concerned hereafter.

Dictionary text definitions are a specialized form of ordinary text. I mention this because it is tempting to believe they are either a formal logical language or normal descriptive English. They are neither of these, and consequently can only be parsed with slightly less effort than any other form of text.

One reason for the complexity of text definition structure is that each such definition is a single phrase or sentence. To accomplish this the lexicographers use nominalization and other morphological transformations (e.g. gerunds in hyphenated adjectives as the "ant-eating" of aardvark's definition) to incorporate sentences about the word being defined into the noun phrases used in the single definitional phrase.

In addition to these morphological transformations, dictionary definitions share three parsing problems of ordinary text to varying degrees. These are (1) semantic disambiguation, (2) anaphora resolution, and (3) nominal compounding. Much work upon these problems has already been undertaken by other researchers.

Edward Kelly and Philip Stone [1975] made significant progress on automatic disambiguation of word senses and this technique could be applied to dictionary entries as well. The determination of dictionary anaphora also appears to be solvable by current systems, mostly because the dictionary uses standard patterns or "templates" for definitions, and there are only a limited number of these.

One difficult problem for automating dictionary parsing which does remain is the question of compound nouns, such as "car thief", "ice cream", etc. While some of these are given status as defined entries, (e.g. "ice cream" is defined separately from "ice" and "cream"), most others are not so treated. James Rhyne [1975] has made progress in this area demonstrating that such compounds are explainable under case-grammar theories by showing many such pairs are the result of the deletion of a verb related by specifiable case argument patterns to the nouns involved. Thus, "car thief" is explainable by understanding that a thief is the AGENT of the verb STEAL, and that this verb forms compounds using its THEME and AGENT. From this we see that a "car thief" is "a person who steals cars". Similar analyses can be developed for many other compounds

and indeed knowledge of dictionary verb relationships would greatly facilitate the bootstrapping of this operation, yielding more information about how to understand compound nouns which in turn reveals more about how to automate further dictionary analyses.

The computational parsing of dictionary definitions has not been undertaken during this dissertation because my major interests have been semantic rather than syntactic. It also did seem nearly impossible that an adequate grammar for 4.5 megabytes of text could be developed that would correctly assign semantic components of definitions, or perform disambiguation. Additionally, the lexicon for parsing the contents of the dictionary would have had to contain between 10-20 thousand main entries and thus have been both too expensive and difficult to manage within the existing LISP programming environment.

There does however remain a reasonable question as to what a dictionary grammar would look like. Figure 7-1 gives a preliminary phrase-structure grammar for parsing verb definitions. It has been experimentally applied to only a dozen difficult definitions, but because of the dictionary's patterning of definitions should therefore be adequate to parse several hundred or thousand other definitions without further alteration. It is a grammar based upon the conjunctions AND and OR. Nearly every constituent in a dictionary verb definition has been converted from a ordinary part-of-speech category into a conjunctive phrase. AND and OR are the dominant syntactic form over nouns, verb infinitives, adverbs, gerunds, and continue to dominate the higher level structures such as NP's, PP's and VP's as well. Examples of parses based upon this grammar are given in figures 7-2 and 7-3.

The goal of the grammar is to determine the "kernel" or "kernels" of a verb's definition. It works well enough to provide a tentative conclusion that very strong syntactic "defining formulas" [Olney 1967] are indeed being used in the dictionary and that a purely syntactic grammar can be developed to perform this task (assuming of course that the current grammar is not adequate). However, the problem of using the MPD as the parser's lexicon has not yet been worked out.

```
<S> ::= <NP2><vb> { <PP> / <adv><PP> / <adv> }
<vb-def-phrase> ::= <vb-kernel-set>
                    { "so that" <S> / <PP> / <AVP><PP> /
                     <AVP> }n   { <prep> }
<vb-def> ::= { "to" } <vb-def-phrase>
             { ( "and" / "or" ) { "to" } <vb-def-phrase> }
<vb-kernel-phrase> ::= <vb-kernel> <NP-set>
<vb-kernel-set> ::= <vb-kernel-phrase>
                    { ( "and" / "or" ) <vb-kernel-phrase> }n
                ::= <vb-kernal> { ( "and" / "or" )
                    <vb-kernel> } { <NP-set> }
<vb-kernel> ::= <vb> <vb-prep>*
            ::= <vb> "to" <vb-kernel-set>
            ::= <vb>
<NP2-set> ::= <NP2> { ( "and" / "or" ) <NP2> }n
<NP-set> ::= <NP> { ( "and" / "or" ) <NP> }n
<NP> ::= { <art> } { <adj-set> } <N-set>
<PP> ::= <prep> { ( "and" / "or" ) <prep> }n <NP2-set>
     ::= <prep> <gerund-phrase>
<N> ::= <n> { <n> }n
    ::= <pn>
<N-set> ::= <N> { ( "and" / "or" ) <N-set> }n
<adj-set> ::= { <adv> } <adj>
              { "," { "and" / "or" } <adj-set> /
                ( "and" / "or" ) <adj-set> }n
<NP2> ::= { <art> } { <adj-set> } <N-set> { <PP> }n
<AVP> ::= "as" <gerund-set>
      ::= <adv>
<gerund> ::= <vb> "+ing" <gerund-prep>*
         ::= <vb> "+ing"
<gerund-set> ::= <gerund> { ( "and" / "or" ) <gerund> }n
<gerund-phrase> ::= <gerund-set> { <prep> }1

Key:
* means context-sensitive, i.e. agreement with the <vb> is needed.
/ separates exclusive alternatives within a rule
(,) surround required elements
{,} surround optional elements
<,> surround meta-symbols
"..." surround literal strings
n following brackets means "any finite number of occurrences"
1,2,3,etc. following brackets means exactly that number of
           occurrences, e.g. {<prep>}1 means optionally 1
           preposition (or none if the option is not used)
```

Figure 7-1 Syntactic "Kernel" Grammar for Parsing Dictionary Verb Definitions