EXTRACTION AND REPRESENTATION OF ENCYCLOPEDIC KNOWLEDGE FROM A DICTIONARY

by

Thomas James Godfrey

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science and Applications

APPROVED:

John W. Roach

John W. Roach, Chairman

Edward A. Fox

Edward A. Fox

Jane Terry Nutter

J. Terry Nutter

June, 1993

Blacksburg, Virginia

EXTRACTION AND REPRESENTATION OF ENCYCLOPEDIC KNOWLEDGE FROM A DICTIONARY

by

Thomas James Godfrey

Committee Chairman: John W. Roach
Computer Science

(ABSTRACT)

The software tool described in this thesis demonstrates a practical application of prototype theory to the representation of world or encyclopedic knowledge. The tool is designed to extract such knowledge from dictionary entries and to represent it in a network of frames. An application needing encyclopedic knowledge would rely on some separate utility program to draw information from the frames, translating frame data as necessary for its own use. The encyclopedic knowledge that can be extracted from a dictionary extends over an extremely wide range of topics, but it is very shallow, so the knowledge base of any final application would require further enrichment from other sources. However, a substantial part of the deficit might be overcome through similar automatic processing of more dictionaries and other published sources of encyclopedic knowledge.

Acknowledgements

As sole author of this thesis, I claim full responsibility for all its shortcomings but gratefully share all the credit for its merits with the many who have helped me along the way. Those I wish to single out for special recognition include the Creator and King of the universe, who designed my prototype and gave me life; my parents, Dr. Paul R. Godfrey and Mary Elizabeth Thompson Godfrey, who nurtured me throughout a happy childhood; my wife Beth, who persuaded me to pursue a degree in computer science; her parents, Mr. and Mrs. Douglas C. Richards, who financed my studies here at Virginia Tech; the chairman of my advisory committee, Dr. John W. Roach, who guided and encouraged me during my whole course of study at this university; and the other two committee members, Dr. Edward A. Fox and Dr. J. Terry Nutter, who contributed significantly to my progress. Many other former professors, fellow students, family members, and friends also deserve credit for having smoothed the way for me down through the years. May all such individuals, whether mentioned here by name or not, be assured of my deep appreciation for their part in this achievement. I also wish to express gratitude to the Longman Group Limited, which licensed the use, at Virginia Polytechnic Institute and State University, of their machine-readable version of the *Longman Dictionary of Contemporary English.*

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction and Problem Description

A. Introduction

Knowledge is information available for use. Without it, any active agent, whether a computer or a person, would have to rely completely on chance or guesswork to accomplish anything useful. If the goal is to have a computer perform a complex task reliably and with minimal human intervention, chance proves to be a poor substitute for knowledge, and the more ambitious the goal is, the more knowledge must be made available. This thesis addresses the problem of obtaining and representing general knowledge of the world, also called encyclopedic knowledge, for use in artificial intelligence applications.

Heavy emphasis will be placed on prototype theory as it relates to the representation of knowledge extracted by a computer from the *Longman Dictionary of Contemporary English* (LDOCE) in machine-readable form. This and other innovative features of our approach are summarized in Section C of this chapter. Later chapters cover these features in more detail and show

how they were applied in an original computer program, called Salient Information Viaduct (SIV).

B. Problem Description

There are actually two problems that this work confronts. Given a need for massive amounts of general information or world knowledge, how should it be extracted from trustworthy sources, such as a dictionary, and then how should it be represented for later use or assimilation by an application? Both of these questions involve a third, namely, what kinds of information should be extracted and represented? All three questions have many possible answers, of course, but some very specific alternatives will be offered for consideration and evaluated in the chapters that follow.

Once the knowledge base envisaged for this project has been finished, it will need to be converted to another form more suitable for information storage and retrieval. Although it lies beyond the scope of this thesis, an interesting aspect of this conversion step is the problem of merging new knowledge with old knowledge. The conversion itself should be reasonably straightforward and need not concern us here.

The problem that does concern us is not just getting and representing general knowledge. It is also important to get it rapidly, hence more or less automatically, by computer, and in large quantity, yet with a high degree of confidence in its accuracy. The familiar goals of speed, quantity, and quality compete with each other, so that none can be maximized without sacrificing one of the other two. The ideal is to attain overall excellence while keeping all three goals in balance.

The strategy for optimizing speed includes minimizing the need for human intervention during the knowledge extraction process and avoiding unnecessary work for the computer. It would be easy to bog down the whole process by making unreasonable demands on available resources, so the system designer must distinguish between what is feasible and what is merely wishful thinking.

It is mainly for this reason that only noun definitions are handled by the SIV program, and even a noun is excluded if it is determined to be too abstract. It seems that verbs, adjectives, and abstract nouns could all be handled in a similar way, but probably with a poorer return on the investment in the resources required to accommodate them.

Increasing the quantity of knowledge extracted necessarily increases the time required for the process, thus reducing the speed with which the whole operation is accomplished. Nevertheless, our quantity goal is well served by any increase in per item extraction speed that can be managed. The faster the extraction algorithm, the more knowledge we can expect to amass in a given time. Another aspect of our quantity strategy is choosing a dictionary with a fairly large number of entries.

Finally, the quality goal involves some hard decisions about what to set aside and what to include in the output representation as input material is scanned. Actually, very little is discarded, but those parts of the input that are not really "understood" by the program are clearly distinguished as such.

One other important aspect of this third goal relates to our choice of prototype theory as a basis for decisions about what to extract and represent. The assumption is that the noun definitions to be used as input are actually describing prototypes of nominal concepts, and the information extracted from them can be legitimately viewed as facts about some prototype. The claim, further developed in later chapters, is that certain information about prototypes, usually ignored in knowledge representation schemes, can also be extracted and neatly accommodated to build a knowledge base that reflects the intended meaning of source documents better than one that assumes a simplistic view of categories.

C. Summary of Innovations

The next chapter relates the present work to previous work in the same general field, but it may be difficult to glean from such a lengthy review what it is that particularly distinguishes this work as new and innovative or an extension of work already done by others. The main distinguishing features are:

* Connection to prototype theory
* High-speed parsing
* Non-specialized knowledge base

Perhaps the most important innovation has already been mentioned in the first section of this chapter, namely, the connection of our work to prototype theory. SIV is the first program specifically designed to extract from a dictionary information of particular interest from this theoretical perspective, including a wide range of encyclopedic knowledge of general or theory-neutral interest.

Another innovative feature of SIV is more technical than theoretical. The SIV parser does not produce a complete structural analysis of its input text, which partly accounts for its high speed, yet its output proves adequate for achieving a high level of success in extracting facts from the LDOCE. Similar parsing techniques have certainly been tried before, but SIV seems to be the first program that applies them generally to the extraction of a wide range of facts from a dictionary. Some significant aspects of the parsing algorithm also appear to be unique.

A final novel feature that should be mentioned concerns the overall goal chosen for SIV. It does not extract knowledge in a form specifically intended for some particular application. Instead, an actual application would absorb SIV output for its own purposes through a straightforward conversion or assimilation process. Our aim has been to concentrate resources on a small part of a larger task but to do just that small part well. This approach is certainly not unique to SIV, if all programs are considered, but SIV does appear to apply it uniquely to the extraction of encyclopedic knowledge from a dictionary.

D. Plan and Outline

The basic strategy for the work reported here included an initial review of pertinent literature and selection of the LDOCE as the source of knowledge. At that stage, a commitment was made to adopt the general tenets of prototype theory, with frames as the basic data structure for storing information, and ambitious goals for ideal representations were established accordingly. This was followed by a "hand analysis" of LDOCE entries to determine what kinds of information a computer should be able to extract, given our source. Then an actual program, called Salient Information Viaduct (SIV), was planned, implemented, documented, and finally tested by extracting knowledge from a representative sample of the whole dictionary. The work culminated in a careful analysis of results of the test and a final run using the entire LDOCE as input.

This description of that work is organized as follows. After the review of relevant and influential literature in Chapter 2, the thesis progresses from the underlying theory and ideals for representation of knowledge about prototypes (Chapter 3) to their practical implementation in actual data structures (Chapter 4), and then to results of the hand analysis of dictionary definitions to be used as input (Chapter 5). The SIV program itself is then described in some detail (Chapter 6), along with the results obtained when it was tested (Chapter 7).

Chapter 2

Previous Work

The review of literature in this chapter briefly surveys the work of other theorists and researchers who have chosen similar aims and topics.

A. Introduction

Our topic has connections to a wide range of interrelated fields, including syntax, semantics, ontology, lexicography, cognitive science, artificial intelligence, and computerized natural language understanding, to name just a few. It would be impractical to attempt here to review all the work in such fields that is somehow relevant. This chapter, therefore, limits coverage to work that has influenced the course of this project the most directly or that is considered to be notably similar in its aims and focus.

It should be emphasized that no publication was found that describes automatic extraction of a wide variety of encyclopedic knowledge from a dictionary either with orientation to the special demands of prototype theory or with reliance on such a simple yet effective parsing technique as the one adopted for this project. Even if such a publication does exist or soon will, this thesis should prove valuable for comparison.

B. The Cyc Project: Lenat and Guha

Of all the previous work reviewed, the Cyc project [Lenat and Guha, 1990] perhaps comes the closest to sharing our aims. That project, however, is extremely ambitious and far more than thesis sized, being sponsored at Microelectronics and Computer Technology Corporation (MCC) by a consortium of "far-sighted American companies" [1990: xvi]. Like our Salient Information Viaduct (SIV) project, the Cyc project has as one of its chief aims the creation of a database for encyclopedic knowledge, but it is inherently much more comprehensive and self-sufficient, even including its own rather sophisticated and diversified facilities for making inferences and managing distributed manual input of knowledge. The main reason for pursuing the ideas in this thesis, given the advances made in the Cyc project, is the hope that more of the total task can be accomplished automatically and more economically, while taking into account the true nature of prototypical categories.

The basic plan for Cyc is to load its "consensus reality knowledge base" [Lenat and Guha, 1990: 7] manually with about ten million "pieces of common sense knowledge" [1991:

84], or enough to reach "the crossover point where natural language understanding begins to be a more effective way of further enlarging it" [1990: 26]. Knowledge is organized into frames accessed through a distributed database. After a Cyc "knowledge editor" enters the common sense knowledge believed to be necessary to understand some text fragment, he can ask the system questions to probe the depth of its understanding [1990: 29]. Knowledge is also added automatically through a coordinated assortment of small inference engines, each one individually optimized for efficiency. These specialized "inference features" are policed by a global "truth maintenance system" with modules to detect and resolve any contradiction that may be introduced, always choosing the conclusion best supported so far [1990: 49-53].

The magnitude and even the existence of the Cyc project reveal the felt need for a large, general-purpose database for encyclopedic knowledge such as SIV is designed to help build, and although Cyc makes no special concession to prototype theory, the frame-based style of representing descriptive knowledge in Cyc has served as a model for SIV. Thus Cyc has contributed both motivation and inspiration. Although many other knowledge representation systems use frames, or some sort of "class-slot-object syntax" [MacGregor, 1991: 88], Cyc is exceptional, if not unique, in its requiring a knowledge base large enough to avoid the brittleness that plagues existing expert systems [Lenat and Guha, 1990: 3-4]. ("Brittleness" is the tendency for a system to fail when scaled up or applied outside its limited domain.) In the June 1991 ***SIGART Bulletin*** ("Special Issue on Implemented Representation and Reasoning Systems"), Cyc was the only one of the twenty-two systems described that relies on such a comprehensive knowledge base. Boose [1989: 17] offers another comparison of Cyc with many other systems, and Hayes [1985: 468] seems to support the general approach, arguing for "the construction of a formalization of a sizable portion of common-sense knowledge about the everyday physical world."

C. Knowledge Extraction: Amsler, Ahlswede, Alshawi, and Small

At least two dissertation projects have explored the challenging problem of parsing dictionary definitions [Amsler, 1980; Ahlswede, 1988], which is another major and indispensable aspect of the SIV project. One of these dissertations, however, gives only marginal attention to parsing, and neither of them is primarily concerned with the construction of a database for encyclopedic knowledge.

Amsler was most interested in the semantic structure of the definitions in a pocket dictionary, and he concentrated on

"ISA hierarchies" and the "knowledge of taxonomic relationships" contained in those definitions [Amsler, 1980: 1-2]. Although the "computational parsing of dictionary definitions" was too syntactic for his topic and beyond the means at his disposal, Amsler did propose a "preliminary phrase-structure grammar for parsing verb definitions" [1980: 109-10], which he had used experimentally on a small scale with some success. Both syntactic parsing and morphological parsing "to verb normal form" were considered as promising new directions in his chapter on "future work and speculation" and in his conclusions [1980: 107ff; 131].

One of the most significant aspects of Amsler's work with respect to the topic at hand is his early recognition of the value of dictionaries as a repository of "background information which humans use to perform complex tasks," a view that fueled his main motivation, "the desire to provide this 'artificial memory' for intelligent computer programs by deriving it from existing published dictionaries" [1980: xi].

Ahlswede's work was much closer than Amsler's to the work cut out for SIV, since it concentrated on automated extraction of "lexical-semantic relations" from a larger dictionary, and a substantial share of this work involved actually parsing dictionary definitions "with the specific intent of extracting word pairs linked by these relations"

[Ahlswede, 1988: 6-7], or even more specifically, "to produce relational triples for thesaurus-aided information retrieval" [1988: 130].

This author has adopted Ahlswede's view that "relational triples" such as those he extracted [1988: 130] may also constitute a kind of encyclopedic knowledge. In his discussion of the nature and scope of taxonomy, one of the "lexical-semantic relations," Ahlswede saw reasons to view it as either mostly lexical or mostly semantic, depending on the application, but felt no need to establish a clear boundary between lexical knowledge and world knowledge [1988: 42-43, 45]. His primary goal or vision was to pave the way for automated generation of lexicons for natural language processing systems [1988: 1], not construction of encyclopedic knowledge bases to be applied more generally. Ahlswede favored the lexical aspect and explained that "the end product of our work, the relational lexicon, is a specifically lexical knowledge base, containing linguistic knowledge as well as world knowledge" [1988: 43].

The fact that Ahlswede attempted so recently to extract massive knowledge automatically from a dictionary, for whatever purpose, demands that close attention be paid here to the results he achieved and to one conclusion he drew based on his experience, namely "that full natural-language parsing is not an efficient procedure for gathering lexical information in the form of relational triples" [Ahlswede, 1988: 168; Ahlswede and Evens, 1988b: 223]. This judgment seems amply justified by the ratio of effort to results that he reported.

After "a man-year or so of development time for the definition grammar" and 180 hours running a Vax 8300 in an attempt to parse over 8,000 definitions, only about two thirds of which were successfully parsed, Ahlswede obtained "5,271 taxonomies, a quarter or so of which were vacuous; several thousand modification arcs, for most of which the right hand side awaited further analysis; and a varied scattering of other relational arcs," whereas it took only "about three hours with UNIX utilities and interactive editing" to extract "11,596 relational triples representing six lexical-semantic relations from the intransitive verb definitions alone" [1988: 151-52]. For the more time-consuming effort, Ahlswede used the Linguistic String Parser (LSP) [1988: 2], which a team led by Naomi Sager had taken some twenty years to develop into "perhaps the most complete natural language processing grammar in existence," and Ahlswede spent still more time on "considerable modification to deal with the peculiar language of dictionary definitions" [1988: 101-2]. Because of our interest in nouns, it should be noted that Ahlswede achieved a 78 percent success rate using the adapted LSP to parse noun definitions, and there were 1.7 "parses per success" [1988:

126], with one success being claimed for every definition that was parsed at least once. Incidentally, most of Ahlswede's rather extensive bibliography and literature review are relevant here as well.

Although their excellent work had only minimal impact on the development of this thesis project, Alshawi [1987] and Boguraev and Briscoe [1987] have also tackled the problem of parsing and processing LDOCE dictionary entries. Their articles and others in the same issue of *Computational Linguistics* (vol. 13, nos. 3-4) provide an outstanding review of various projects to produce computerized lexicons or to extract information from dictionary entries. For instance, Jensen and Binot [1987] describe tools and techniques for automatically extracting information that is used to determine what a given prepositional phrase modifies. In a later article [1988], they explain plans to extend their approach to resolve other kinds of syntactic ambiguity as well. Another recommendable source of strategies for extracting information from dictionaries is Calzolari [1988: 80-94].

The parser described by Alshawi is perhaps the most similar to the one described below (in Chapter 6). One important difference is that it produces lists with a variable number of definition elements at the highest level. The SIV parser outputs lists with a fixed number of such elements. Some of these elements may be empty, but all of them are listed in a set order. In contrast, the Alshawi parser analyzes the LDOCE definition of the noun 'launch', for example, into three elements, namely (CLASS BOAT), (PROPERTIES (LARGE)), and (PURPOSE (PREDICATION (CLASS CARRY) (OBJECT PEOPLE))), whereas it parses the definition of 'hornbeam' into four elements: (CLASS TREE), (COLLECTIVE TYPE), (PROPERTIES (SMALL)), and (HAS-PART ((CLASS WOOD) (PROPERTIES (HARD)))) [Alshawi, 1987: 197]. This flexible arrangement of definition elements may have some advantages, such as preserving information about the order of definition elements, but an application would need to parse this kind of parser output further to identify a particular element of interest. With SIV parser output, the exact location of each top-level element can always be known in advance.

Neither parser can reveal the constituent structure of definitions in exhaustive detail, so they both produce "partial analyses of dictionary definitions" [Ahlshawi, 1987: 196], but Ahlswahi's parser is considerably better at structural analysis than the SIV parser is. Both parsers place more emphasis on semantic structure than on syntactic structure, with special attention to what Ahlshawi calls "the semantic head" of the definition (here called the genus term, as explained below in Chapter 5), and both were designed to handle input from the LDOCE in machine-readable form.

Because of those similarities, the results Ahlshawi reported should be of particular interest. That parser correctly identified the semantic head of 77 percent of a random sample of 500 LDOCE definitions, which apparently included definitions for nouns, verbs, and adjectives. Other information was extracted from 61 percent of the definitions, and the "additional information was judged to be correct" in 88 percent of those cases [Alshawi, 1987: 201].

Parsing dictionary entries is a special case within the larger realm of computerized natural language understanding. One thesis [Small, 1980] that addressed this more general problem deserves mention here. In the SIV project, knowledge is organized by the individual words of our language, which Small calls "active modular knowledge sources that coordinate the parsing process" [1980: 12]. His view seems to lend some support to our focus on words. In fact, Small's "Word Expert Parser" operates by means of "a distributed word-based control structure" [1980: 11], which implies that the usefulness and validity of categories named by words is by no means limited to ordinary communication. When Small says, "The analysis process cannot proceed without knowledge of the real-world and the ability to make inferences about that knowledge" [1980: 22; 97-98], he also recognizes one of the main motivations for having an encyclopedic knowledge base of the sort SIV would help build.

D. Knowledge Representation: Schank and Other Authors

Although words provide convenient pigeonholes for organizing knowledge about the world, they do not answer all questions about what kinds need to be included in our knowledge base. The study of lexical relations seems to imply that the pieces of knowledge can be quite simple, if not purely binary relations. Nutter [1989] presents an excellent compendium of such relations, hierarchically organized. Evens *et al.* [1980] offer a more extensive survey of the topic from a variety of viewpoints, and relations of particular interest for natural language processing are sketched in Evens *et al.* [1987] along with suggestions for extracting some of them from dictionaries.

Schank and some of his students can be credited with development of the idea that some kinds of knowledge have a less linguistic nature yet are also important for well-rounded reasoning capability. In particular, they introduced the concepts of "memory organization packets" or MOPs [Schank, 1982: 83, 95ff; Riesbeck and Schank, 1989: 34-36] and "thematic organization packets" or "thematic organization points" (TOPs) [Schank, 1982: 68, 110ff], both somewhat resembling the frames described by Minsky [1981] and the "prototypes" used in an actual expert system for medical consultations [Aikins, 1983: 176ff].

The MOPs were designed to "represent knowledge about classes of events, especially complex events" [Riesbeck and Schank, 1989: 34] as a set of norms and links. The norms specify such things as events, goals accomplished, and typical participants, while the links join MOPs to show their interrelations. Several distinct types of links are used, depending on the relation, with inheritance as a key feature. TOPs serve to represent "information about what usually happens within a certain high-level context," including expectational, static, and relational information [Schank, 1982: 68]. All of these structures take a higher-level or more macroscopic view of the world and collect the information relevant for that view. One would be hard pressed to find a single word to capture these more global concepts, yet they too play a crucial role in our ability to reason, as Schank and his students have pointed out so well.

Hayes also encourages us to look beyond the purely lexical aspects of our knowledge when he says, "the meanings of linguistic expressions are ultimately to be found in extra-linguistic entities: chairs, people, emotions, fluids ..." [Hayes, 1974: 19]. They are the real subject of our knowledge, so our representations must store more than merely information about words, which are themselves representations. Hayes also comments on the nature of substances, parts, and assemblies, noting that the focus on individuals in many systems of representation is too narrow [Hayes, 1974: 15; 1985: 481-82]. Bunt [1985] provides an excellent elucidation and formal analysis of the issues surrounding mass terms that Hayes raised in his articles.

E. Prototype Theory

Prototype theory provides a fresh view of the nature of categories, which is one of the most fundamental issues in the field of knowledge representation. Taylor nicely summarizes the classical view, which dates at least as far back as Aristotle, as follows: "(1) Categories are defined in terms of a conjunction of necessary and sufficient features ... (2) Features are binary ... either present or absent," from which we may conclude that "(3) Categories have clear boundaries ... (4) All members of a category have equal status" [Taylor, 1989: 22-24].

In recent years, prototype theory has been advanced as an alternative view of the nature of categories, one that avoids the inadequacies and over-simplifications that various researchers have attributed to the classical view. In particular, prototype theory rejects all four of the principles just stated, except possibly the second, but the binary features that it mentions, whether they really exist or not, are no longer granted a key role in category definition.

The word 'prototype' in the name of the theory implies that prototypes are its real subject, whatever light might also be shed on the nature of categories at the same time, but then what are prototypes? A prototype is commonly understood to be a particular instance of a certain kind of thing, specifically, one that is the earliest or best example of all things of that kind. Taylor prefers a "more abstract" sense of the term, namely, "a schematic representation of the conceptual core of a category" [Taylor, 1989: 59], and he goes on to say, "Entities are assigned membership in a category in virtue of their similarity to the prototype; the closer an entity to the prototype, the more central its status within the category" [1989: 60]. In this thesis, the terms 'prototype' and 'prototypical concept' will be used interchangeably in the more abstract sense elucidated by Taylor.

E.1. Background: Lakoff, Langacker, Rosch, and Taylor

Several authors have addressed technical and philosophical issues concerning categories and prototypes. Lakoff [1987] provides an up to date history of this branch of inquiry, where Rosch figures as an important pioneer, and at least two other authors [Langacker, 1987; Taylor, 1989] have also contributed major works to the field. Of course, many other writers have published supportive articles, and the one by Geeraerts [1988] is a fine example of these. Prototype theory is relevant to the work on SIV because the vast majority of nouns defined in any dictionary refer to prototypical concepts, rather than to some particular instance of some concept, and it is important to understand their true nature.

When the noun apple, for instance, is defined as "a hard round fruit with white juicy flesh and a red, green, or yellow skin" [LDOCE], the information provided in the definition is understood to refer to apples in general, not specially to any one piece of fruit, and not necessarily even to every individual apple. The definition does not precisely identify the set that contains all apples and only apples. Some apple might be brown and soft, due perhaps to decay or cooking, but these peculiar properties would not disqualify it from being an apple, whether it failed to fit the dictionary definition or not. It is important to understand that the color and hardness attributes mentioned in the dictionary apply best to the best and most typical actual apples, and at an idealized, mature stage in their development. There can be exceptions, and apples remain apples throughout their cycle of growth and decay.

What the dictionary definition actually provides, therefore, is not a set of necessary and sufficient conditions for membership in the apple category, but rather a brief description of the apple prototype, the concept that a speaker of English might associate with the word 'apple' and use as a basis for deciding whether any particular object should be called by that name. The more similar or directly relatable an object is to the apple prototype, the more likely it is to be called an apple, hence considered a member, to some extent, of the apple category.

The works by Lakoff, Langacker, and Taylor mentioned above share a theme grander than just prototypical categories. They are advancing a whole theory of linguistics, now known as cognitive linguistics, which is seen as an alternative to the quite mainstream "autonomous linguistics." Cognitive linguists reject the idea that some separate faculty of the mind is reserved for language processing, and they embrace the view that any adequate account of language must include some consideration of the contribution of other mental faculties, such as cognition and pragmatics, to the phenomenon of speech [Taylor, 1989: 16-20; Langacker, 1987: 2]. Although the larger debate is an interesting one that is still being fueled by current research [Hasegawa, 1993], its final outcome can be left undecided as far as the work on SIV is concerned. What is most relevant here is the illumination of the true nature of categories as established by the words of our language.

E.2. Prototypes and Fuzzy Boundaries

It has been common to posit a close connection between language and formal systems of logic that involve set theory. (See Böttner [1992: 243-45], Dowty *et al.* [1981: ix, 10, 20-21], Higginbotham [1989: 501], Kratzer [1989: 611-12, 614-16], May [1989: 388], Partee *et al.*[1990: xvii], van Benthem [1989: 437; 1991: 161-64], and references therein for several recent examples.) These systems typically assume that assertions are either true, false, or meaningless, with no half-truths allowed. Furthermore, the entities of interest are individuals and sets of individuals, and these either do or do not belong in a given set, which can always be precisely defined. Useful and convenient as these theoretical assumptions may be, they do not allow a convincing account of the assertions and categories of natural languages.

What we find instead are assertions that may exhibit degrees of truth and categories with fuzzy boundaries and more or less central (prototypical) or peripheral members [Lakoff, 1972: 183-85; Zadeh, 1965; Bobrowicz *et al.*, 1991: 137]. Lakoff [1972: 191] provides the following example of an assertion with intermediate truth value: "Approximately half of the prime numbers are of the form 4N + 1." This one is especially interesting, because all of the content words except the first ('approximately') can be given a precise mathematical definition. That one word, then, serves as a hedge, and it is only one of many hedges, the main subject of that article by Lakoff. A similar prototypical analysis has been applied to quantifiers as well [Lesmo and Torasso, 1987].

It is also easy to find examples of categories with fuzzy boundaries, evidently even easier than finding examples of categories that can be sharply defined, like the set of prime numbers (if we can agree that no number is only partly prime). Rosch asked subjects to rate sixty household items on a scale from 1 to 7 according to how good they are as examples of the furniture category and then compiled the results [1975: 229, reprinted in Taylor, 1989: 44]. The ratings range from 1.04 for chair (the highest rating) to 6.68 for telephone. The ratings gap between adjacent items listed in rank order is never greater than 0.48 (4.52 for shelf versus 5.00 for rug), and it is 0.20 or more in only seven cases, all but one of which occurs in the second half of the list. These results are even more impressive because of the high degree of agreement reported among the 200 subjects; the gradation is not merely an artifact of averaging, with each subject having sharp but slightly different boundaries for the furniture category. It is easy to see that this particular category is not an isolated, exceptional case.

Categories with fuzzy boundaries are considered prototypical. Some actual items may be more or less like the prototypical ideal. Thus some household items may be closer to the ideal piece of furniture; some cups more like the ideal cup, while others are much more like a bowl, say, and can only be marginally assigned to the cup category. Consider also how a cloth or garment may turn into a rag so gradually that the appropriateness of calling it a rag will be uncertain at some stage in the process. As Langacker says, "speakers do not adhere rigidly to criterial attributes in judging class membership" [1987: 16]. He gives the example of the baseball category. One might assume that an item must have certain attributes to qualify as a baseball, but in practice, such attributes are hard to find. Consider just one of Langacker's examples: "My baseball just exploded!"

Wierzbicka observes that "many scholars previously interested in meaning" have used fuzzy boundaries as a convenient excuse for abandoning the urgent task of studying and describing meaning. Having found that task unrewarding, they concluded that meaning cannot be described, due to the fuzzy boundaries, so they have turned instead "to other, less frustrating endeavours -- for example, to discussions of prototypes, ..." [1991: 76]. However valid her complaint may be, the fact remains that an interest in describing meaning can be quite complementary with an interest in prototypes and fuzzy boundaries, as Langacker and others have shown in their work.

It would exceed the scope of this chapter to elaborate further on the development of prototype theory as presented by Lakoff, Langacker, and Taylor, but some general comments on their three books may be helpful. Lakoff [1987] covers the topic in a largely informal fashion, with a heavy emphasis on various case studies on grammatical categories and linguistic expressions. Langacker [1987] fits the topic into a rather comprehensive and well-organized discussion of the cognitive linguistics framework. Taylor's well-written account [1989] is by far the shortest of the three, but it deals the most specifically with prototype theory, and most of the other topics included in the book are also very pertinent to the subject of this thesis. There are even chapters on polysemy, encyclopedic knowledge, and some important figures of speech (metonymy and metaphor).

F. Other Related Work

The publications briefly reviewed in this section and summarized below in Table 2-A all cover some particular project to produce a lexicon or thesaurus more or less automatically from machine-readable sources. Although these publications had little or no direct influence on the SIV project, each one rates mention here because of some notable similarity regarding its topic. Of course, many others could also have been included on the same basis, but some discretion had to be exercised to hold this section to a reasonable length. Evens [1989] may still be recommended for an excellent survey and extensive bibliography of the general topic of lexicon construction based on machine-readable dictionaries, even though it is rapidly growing out of date.

Some important differences from the present work will be noted for all publications selected here. Of course, comments underscoring the innovative aspects of this thesis should not be construed as necessarily critical of the approach chosen by other workers, nor as unappreciative of the advantages and extra functionality that their system may have. All contributions to the field should be welcomed.

Almost all of the knowledge extraction systems described below are concerned with the construction of a lexicon or thesaurus specifically intended for use in natural language processing, rather than a more general purpose store of knowledge.

Slator [1989a; 1989b; 1992] describes one system for building a lexicon that is "part of a larger system" for natural language text processing. However, it could readily serve "as part of a variety of differing systems of language analysis" [1989a: 93]. Like SIV, this one has been tested with input from the LDOCE, but knowledge extraction is accomplished through a combination of pattern-matching and deep syntactic parsing [1992: 393]. Slator reports a success rate of over 99 percent in parsing "the leading part of content word definitions (where the genus terms are found)" [1992: 394]. The latter article specifically mentions only four relations that are extracted.

Another interesting example is the LEXIGRAM system, which is designed as a tool "to provide the lexicon and grammar system necessary for building any domain-specific NL [natural language] processing system" including a knowledge base [Dik *et al.*, 1992: 23]. Although the theoretical framework for LEXIGRAM is functional grammar, with no special attention given to prototype theory, Dik *et al.* do develop and exploit the notion of a "classificational hierarchy" of words in the dictionary, which they divide roughly into three groups: (1) top level words, (2) a larger group of ordinary words, and (3) domain-specific words, with words in higher groups being used to define those in lower groups [1992: 23, 36, 51]. The LEXIGRAM system extracts lexical information from noun, verb, and adjective definitions in the LDOCE as well, and work is beginning on a much larger Dutch dictionary [1992: 32, 37].

Boguraev and Neff [1992] also address the same general concern, namely "extraction of large-scale lexical information for the purposes of automated natural language processing" [1992: 110]. They report that their "dictionary entry parser ... can also deliver a mark-up representation of logical structure," and they have "carried out detailed analysis of several dictionary sources, monolingual and bilingual, followed by theory-driven search for lexical properties across entire sources" [1992: 112]. Boguraev has long been concerned with extracting lexical information from the LDOCE in particular. See Boguraev *et al.* [1987] for yet another description of such a project, this one "carried out within the theoretical framework of Generalized Phrase Structure Grammar" [1987: 193].

All publications mentioned in the remainder of this section report a system that relies on the LSP, the full-fledged syntactic parser briefly described above, whereas SIV employs a much faster and less sophisticated pattern-matching strategy for parsing dictionary entries. One of these systems produces "a large semantic network" specifically designed "to support interactive query expansion and search by end users" [Fox *et al.*, 1988: 101; Ahlswede *et al.*, 1988], and it includes information about phrases as well as words. In spite of its basic dedication to a special purpose, perhaps such an extensive knowledge base could easily be made to serve as a source of information for more general purposes as well.

A number of publications describe a long-term project to extract lexical and semantic information from several sources, including W7, for the purpose of building a lexicon or lexical database [Ahlswede, 1985; Ahlswede *et al.*, 1986, 1988; Ahlswede and Evens, 1988a; Fox *et al.*, 1988]. One of these cites the modification-taxonomy-queueing (MTQ) schema due to Oswald Werner as the theoretical model or framework for definition analysis [1985: 274]. Although another article states that one "major goal is to explore the role of relations other than taxonomy in the dictionary" [Ahlswede *et al.*, 1986], only a very few are explicitly mentioned, and the emphasis is on using the LSP to process definitions. Yet another article concentrates on "the development of an LSP grammar for adjective definitions ... and its use in the automatic extraction of relational information from W7 definitions" [Ahlswede and Evens, 1988a: 215]. This one names about 50 relations that can be extracted, which is an order of magnitude more than the number reported in either of the other two articles.

Table 2-A. Lexicon Construction Efforts

Author(s)	Source(s)	Parser	Purpose of Lexicon
Ahlswede *et al.*	W7, ...	LSP	language processing
Boguraev & Neff	various	not named	language processing
Dik *et al.*	LDOCE	LINKS	language processing
Fox *et al.*	W7, ...	LSP	database searches
Slator	LDOCE	combination	language processing

Chapter 3

Knowledge Representation

What kinds of encyclopedic knowledge of nominal concepts should ideally be represented in a database? This chapter proposes an answer inspired by recent work on prototype theory.

A. Introduction

There are many kinds of knowledge and many ways to represent any kind of knowledge, and this is particularly true of the kinds of knowledge used and stored by a computer. In this chapter, consideration is given to what seems to be the theoretical ideal for computer representation of world or encyclopedic knowledge.

In keeping with the focus on nominal concepts, we are concerned here only with representing knowledge about concepts represented by nouns. We recognize that other kinds of knowledge also need to be represented somehow in a complete system, and that those additional requirements could affect even the way that nominal concepts are represented, but the representation of just nominal concepts appears to constitute a substantial yet manageable chunk of the whole task.

Since this chapter concerns ideal representations, little regard is paid here to practical constraints imposed by the time, effort, and resources available for this thesis project, or even by the limitations of state of the art software and hardware. Later chapters will compare what can actually be achieved with the ideals suggested in this chapter.

B. Model and Approach

Before delving into the details of knowledge representation, it seems wise to consider the general approach that should be followed. For our ideal system, it may be tempting to model some way that knowledge is represented in nature. To be sure, other researchers have yielded to this temptation. For example, Schank writes, "Conceptual Dependency Theory was always intended to be a theory of how humans process natural language that was explicit enough to allow for programming it on a computer. ... Our method is to try to figure out how humans communicate with other humans and model these processes" [1975: 3,5]. A similar point of view appears in an essay by Minsky, who says, "I draw no boundary between a theory of human thinking and a scheme for making an intelligent machine" [1981: 247].

The model-nature approach is appealing, because it makes sense to reuse any brilliant solution to the problem of knowledge representation that we already find in service, if at all possible. Why should we bother with an artificial approach when knowledge is already being represented so cleverly in nature, especially within the human brain? The answer appears to be that we can hardly understand existing natural solutions to the problem we are considering, let alone implement them.

The situation is by no means peculiar to the field of knowledge representation, so it may be helpful to notice how the same dilemma has been resolved in other fields. Notice that our planes do not flap their wings, and our cars do not gallop on hooves. Natural solutions to problems of flight and locomotion are best in many ways, but in return for abandoning the tantalizing goal of mimicking those solutions with our mechanical devices, we get to have cars that outrun horses and vehicles that can fly us to the moon.

So we admit that our understanding of how knowledge is represented in the human (or animal) brain is too rudimentary and incomplete to serve as our model and prefer to pursue other, creative ideas, better suited anyway for the type of equipment at our disposal, in the hope that the desired result, if it is ever achieved, will be the best possible. The computerized representations that result could be even better than the natural ones in some respects, just as computers can perform certain clerical and arithmetic tasks with greater speed and accuracy than people can.

This is not to chide those who have preferred the more ambitious goal of imitating natural systems, since they have made substantial contributions to the field, nor even to reject all interest in cognitive studies, since they may provide some insight and inspiration, but our chosen approach is to consider ways to represent knowledge in a computer without regard to their neurological plausibility. Our goal is to simulate human cognitive competence, not to imitate the processes and data structures used in nature to implement that competence.

As Evens points out in her insightful discussion of "psychological reality vs. computational convenience," the choice will have an impact on our methodology: "Those who believe in the psychological reality of relations are very properly concerned with discovery procedures for determining them; those who view relations as a convenient lexical access method are content to invent them as needed" [Evens, 1988: 5-6].

Disinterest in how knowledge is represented in the brain does not entail a corresponding disinterest in the kinds of knowledge represented there. Since we do want to simulate human cognitive competence, the kinds of knowledge to be represented should correspond to kinds that humans have and be adequate for achieving a desired level of competence. We must also be concerned here with the way various kinds of knowledge about nominal concepts should be organized. How they actually are organized for this project and how the kinds of knowledge are actually represented will be explained in the next chapter.

The main idea guiding the SIV project is that most nouns defined in a dictionary correspond to prototypical categories with internal structure and fuzzy boundaries which can and should be represented to achieve better encyclopedic knowledge bases. Other nouns refer to individual instances belonging to prototypical categories, and still others refer to groups or sets of such instances, including samples of substances. Separate but similar types of representation are proposed to store knowledge about these individual and group instance concepts and the prototypical category concepts. The three types of representation will now be explained in more detail.

C. Individual Instance Concepts

The world we know is populated with innumerable individuals, whether people, animals, or things, and a high level of general intelligence requires that tabs be kept on many of them. There may be wide variation in the number of bits of information to store about these individuals, and it seems that no particular piece of information is absolutely essential. Many individuals do not even have a unique name, yet some information known about them may be worth storing. The problem of deciding whether a certain piece of information is important enough for permanent or even temporary storage is beyond the scope of this work, but we are still interested in identifying major kinds of information that should sometimes be represented.

It is often important to know the time of certain individuals' existence and their location, which may change as time passes. Many other important or noteworthy facts about them may also change in the course of time. There should be some provision, therefore, for representing such facts about each individual on file in such a way that each fact has a specified time during which it holds true. In some cases, a span may be unbounded, so that it may suffice to record just the approximate point in time when a fact either became true or ceased to be true.

Our interest is not limited just to individuals populating the real world of actual existence. For a broad sort of encyclopedic knowledge, there can also be important individuals that exist only in novels or other fictional works, or only in legend or in one's imagination [Hirst, 1989]. It might be important to store some information about Sherlock Holmes, for instance. To accommodate such individuals in our representation, we should also be able to specify their realm of existence. Their time spans may or may not be relevant, but if they are, they would be determined with respect to that same realm of existence.

Facts common to many individuals might be abstracted and stored separately as information about either an appropriate prototypical category or the entire group of individuals concerned, thus saving storage space. Along with the other relevant information, however, there should be a list of groups and prototypical categories to which the individual is known to belong. In the case of the categories, the degree of membership should also be noted. The individual will inherit the features and attributes of the group or category by default, but there should be provision for representing exceptional facts that may distinguish this individual from other members. For example, each individual cat might be assumed to have erect ears, since that is the kind cats generally have, but it should be possible to represent locally the fact that some particular cat has an exceptionally floppy ear. More will be said about group and category concepts in later sections.

The facts about an individual may involve various kinds of attributes. Some will be quite absolute, like the name, sex, or owner; others will be relative, with or without an associated scale and measure. Facts about height, weight, and hardness, for example, might be stated more or less exactly in terms of some unit, such as inches or pounds, or a scale, such as the Mohs' scale.

Though less specific or even vague, terms like 'tall', 'heavy', and 'soft' also convey useful information, perhaps the best available, so it should be possible to represent this kind of term also. These terms imply that the specific scale value for this individual, whatever it really is, would be above an approximate threshold for some category or group to which the individual belongs. For many attributes, such as friendliness, utility, and form, no exact specification may even be practical, so a vague term such as 'friendly' or 'useful' might be the only real option. Any of these relative terms can be made a little more exact by adding a degree term ('slightly', 'moderately', 'extremely') to indicate by how much the threshold is exceeded, or by comparison to another individual ('taller than Bill', 'ten pounds heavier than George', 'as soft as clay', 'bellshaped'), so our representation scheme should also accommodate modifiers like these.

Lakoff presents a formal analysis of various degree modifiers, such as 'very' and 'rather', and suggests that algebraic functions are involved in their semantics [1972: 215]. Whether or not his analysis or that conclusion is correct, it seems adequate for our purposes to associate such terms with approximate points or regions on a scale that measures degrees of deviation from the norm.

It has already been mentioned that some attributes are absolute, with little or no gradient apparent in their values. These can be represented as binary features, such as 'plus or minus male', or 'plus or minus animate'. This sort of representation scheme has been popular in the semantic analyses advanced by linguists. It would be convenient if all kinds of information about individuals could be represented in this way, but representations suitable for handling scales, measures, comparisons, and relative qualifiers are also desirable for handling the massive information that does not neatly fit the binary mold. Our theory is designed to accommodate all these kinds of information.

Another kind of information that we should be able to store can be even more complicated, since it involves the representation of events of arbitrary complexity. We may want to represent some act that an animate individual performed or to which any kind of individual was a party. This could involve representing even an entire story that featured the individual in some remarkable way. Of course, the story might be represented separately as a kind of information completely different from those we are considering here, but at least some pointer to that story representation should be included with the information about the individual. This would also be true of complicated statements about habitual, continuous, or repetitive actions involving the individual.

Ideally, we would want to store images and characteristic sounds of individuals (or entities) as well, and especially for inanimate things, perhaps even their properties of taste, smell, feel, and emotional impact. This sort of information might be approximated by descriptive terms, but they would not allow the fine comparisons and distinctions that would be possible with more detailed and elaborate characterizations, such as those furnished in modern multimedia dictionaries.

Still another sort of information about individuals that should be considered is the degree to which a piece of information is believed, possibly including some specification of the reasons for believing it to that degree. This degree of faith in each fact must be distinguished from the degree of intensity or appropriateness of some attribute, which is a separate though possibly related sort of information. For example, suppose that we want to represent a suspicion that some person is very angry. In this case, the degree of faith in our information about the anger of that person is low, while the degree of the anger itself, as represented, is high.

To summarize, we can now list the kinds of information that we would like to be able to store in our representation of individual instance concepts. They are listed in the order they were discussed above, which does not necessarily reflect their relative importance.

1. spans of time during which a fact is true (of the individual)
2. realm of existence (real, fictional, legendary, imaginary)
3. membership in a group or a prototypical category
4. exceptions to facts inherited through such membership
5. absolute attributes (name, sex, ownership, binary features)
6. relative attributes (location, weight, hardness, friendliness)
7. modification of relative attributes (degrees, comparisons)
8. remarkable events and stories
9. habitual, continuous, or repetitive (prototypical) actions
10. images, sounds, and other perceptual characterizations
11. degree and justification of faith in any of that information

Figure 3-1. Kinds of Information About Individual Instance Concepts

The kinds of knowledge listed in Figure 3-1 range over several important aspects of reality associated with nominal concepts, including time, space, categorization, relationships, and physical properties, so the coverage seems to be fairly adequate. Although the list could surely be expanded, it already establishes rather ambitious goals for knowledge representation.

D. Group Concepts

Like individuals, groups can have a real identity and existence in time and space, but whereas an individual is limited to being in a particular place at a particular time, a group may be scattered over a wide area at one point in time, then have its members gathered together at another. Its existence may also transcend the existence of any of its individual members. The U.S. Senate can be used to illustrate both of these characteristics. When it is convened for an important vote, all the senators may be located in a fairly small area; yet at other times, it would be pointless to try to specify the group's exact location. The Senate has existed and maintained its identity for over two hundred years, far longer than any senator's lifetime. On the other hand, even evanescent and transitory groups can qualify for treatment as a group concept (and require separate storage for their data). An example of this might be the passengers and crew of a particular hijacked aircraft. Their individual lifetimes would far exceed the span of time during which the group would exist as an entity of interest.

The difference between individuals and groups can be and sometimes is ignored. One might say that the Senate passed a bill, ignoring the dissenting votes of many of its members. On the other hand, an individual senator might be thought of as a collection of body parts or living cells, ignoring their collective identity. Having separate types of representation for groups and individuals should not interfere with this sort of reasoning, however, and they can be useful for maintaining the distinction when it is important.

It seems best to count as other instances of this same kind of concept both substance samples and actual instances of the concepts to which mass nouns refer. Consider paint as an example of a substance sample and the furniture that belongs in the White House as an instance of one mass noun concept. A sample of paint, like the Senate, can be either gathered into one place or distributed over many places. The same is true of the White House furniture. A sample of paint does not lose its identity with the gain or loss of some drops of paint, which might be considered the analog of the members of a group. This analogy is an oversimplification, since each member of the Senate is a discrete individual, incapable of subdivision without loss of identity, whereas even a fraction of a drop of paint is still a sample of paint.

This difference points to the need to maintain the distinction within the group type of representation, but it does not seem salient enough to justify adding an entirely new type. Note that White House furniture is much more like the Senate in this respect. A piece of that furniture can be gained or lost without really affecting the essential identity of the collection, yet a sawed off quarter of a desk, say, or even one of its drawers is not really a piece of White House furniture in the usual sense.

The kinds of information that would be stored for group concepts include all of those already discussed in the last section, perhaps with some differences in relative importance. We might not care about representing the sound of the U.S. Senate, for example, though this information might be of great interest in the case of a famous choir. Differences of this sort occur even among individuals (some individuals are eternally silent or make uninteresting noises), but there can still be a general tendency for some kinds of information to be less applicable to groups.

In addition to the kinds of information listed above, of course, we must also represent information about who belongs in a group of people or what belongs in a group of things or a substance sample. In some cases, particularly in the extreme case where the group has only two members, a simple list might be adequate. In other cases, the group may be too amorphous and poorly defined for this to be practical. The group of citizens represented by a particular senator would be a case in point. Our representation scheme must allow a more indefinite sort of member specification for this kind of group, which somewhat resembles the prototypical categories, further discussed and contrasted in the next section. The same sort of indefinite identification and characterization may also be desired even for groups whose members can be conveniently listed. This information would be like any other descriptive information, except that it would be distinguished as important for establishing group membership or identifying the particular substance sample.

We can now extend the earlier list (Figure 3-1) to show the new kinds of information to be stored in an ideal representation of group and substance sample concepts. They are again listed in the order in which they were discussed.

12. nature of members (substance sample or discrete individuals?)
13. list of members, if practical
14. descriptive information for establishing identity

Figure 3-2. Kinds of Information About Group and Substance Sample Concepts

A specification that tells whether we are dealing with an individual, a group, or a prototypical category may also count as a kind of information, and this can be included under point twelve in the list.

E. Prototypical Category Concepts

A prototypical category type of concept contrasts most sharply with an individual instance type of concept. For example, the name 'Benjamin Franklin' refers to a particular individual instance concept, namely an instance of a man and patriot, two terms that correspond to prototypical category concepts. The same name, of course, could also refer to some other individual besides the famous one, or even to something nonhuman, depending on the context of its use. The key point here is that a name (or proper noun), used in context, picks out a particular individual. In the example just cited, there is no real difficulty in drawing a sharp conceptual distinction between who is and who is not Benjamin Franklin.

The definiteness of individual instance concepts stands in clear contrast to the indefiniteness of nominal concepts like 'man' or 'patriot'. How old or how mature does a person have to be to be a man, or how patriotic to be a patriot? The answer is arbitrary. There are many degrees or measures of age, maturity, and patriotism, but none of them has any special claim as a boundary or threshold that can be precisely and objectively distinguished from the others. Even if some arbitrary threshold value can be established by law or fiat, the fact remains that the ordinary use of such terms does not rely on our knowing or accepting any specific threshold.

Prototypical category concepts also contrast sharply with individual instance concepts in the way they transcend the bounds of time and space, even more so than the group concepts do. To continue with the example given above, the U.S. Senate is an identifiable group whose establishment, at least, is definitely fixed in time and whose identity depends crucially on the existence of its members in some location at some time. In contrast, the prototypical term 'senate' refers more generally to any such government body, in any country, at any time. Since it does not have to be tied to any specific body, this senate concept had some validity as a hypothetical construct before the first senate was ever established and perhaps will retain its validity even after the last one is dissolved. The actual U.S.

Senate is just one particular instance of the prototypical senate category, which is in some sense timeless.

The question of whether any particular man is or is not a member of the U.S. Senate should in principle have a definite answer. The question of whether a particular government body is or is not a senate may not be so clear and definite. It depends on the definition, which, as has already been mentioned, can involve scalar characteristics with no clear cutoff points. In any particular case, tradition may have established the idea that a particular body is a senate; but in other cases a newly formed or newly considered body may be so marginally similar to those definitely called senates, that it can be difficult to decide whether to apply the term to it as well.

The difficulty of finding a precise definition for a prototypical category is not limited to choosing an exact point along some continuum of values. To be able to define 'man' or 'patriot' precisely, we would also need to settle on an exact set of criteria, choose threshold values for each one, and possibly also assign an exact relative weighting to each defining criterion. What must one do or be to qualify as mature or patriotic? Again, the answer is arbitrary within the vague limits determined by society. The context may influence weightings and even thresholds, so that what it takes to be patriotic in a time of war may be rather different from what it takes in a time of peace. To make matters even worse, it turns out that even the criteria themselves have similar, fuzzy boundaries. If government service is a criterion for patriotism, then how much and what kind counts? The illustration could be extended, but it should already be clear that nominal concepts like man and patriot do indeed have fuzzy boundaries.

The issue is not just academic. We are faced with the real problem of representing knowledge about such a vast and complex world that our only hope of managing general information about it lies in the manipulation of concepts with fuzzy boundaries. It is not enough to have a distinct name for every individual and identifiable group of individuals, especially since we must also deal with substances, like air, that are not individuated at all. If we know something about men or air, we can easily associate some descriptive attribute with them in a general way, provided we allow general terms for such things. In the real world, having to represent that information on a strictly group, individual, or substance sample basis would be quite awkward at best, but in return for the convenience of general terms we pay the price of establishing arbitrary categories, many of which need to be somewhat indefinite.

These categories with indefinite boundaries are termed prototypical categories. The idea is to claim that the noun man, for instance, as defined in a dictionary or used in an encyclopedic database, refers directly to a prototypical category only, not to any particular individual, however typical, nor even to a specific group. If our knowledge representation says that Benjamin Franklin was a man, it asserts that the individual identified by that name belongs to the 'man' category to some extent in some context, without implying that he was typical of all individuals in the category. It should logically follow from such an assertion that Benjamin Franklin shares, to some interesting extent, some descriptive properties that have been associated with members of the 'man' category in general.

By now it should be clear that prototypical categories, though distinct from actual instances of either individuals or groups, can themselves involve either individual, group, or substance concepts, the examples discussed above being 'man', 'senate', and 'air', respectively. Whether a noun refers to an actual instance or to a prototypical category depends on the context of its use. 'That man standing on the corner' is an individual instance, but 'any man over five feet tall' refers to a prototypical category, as does 'man' in the context of a dictionary definition. Similar examples could also be given using plural nouns. So far we have considered the reasons for distinguishing prototypical concepts from individual and group concepts, and we have emphasized the fuzziness of prototype definitions. What sorts of knowledge need to be stored for these prototypical categories?

It is not necessary to represent the span of time during which each attribute is true, but it can be important to note the general location in time and space of the members of the category, since these can be somewhat limited. It might be noted, for example, that galleys were in common use in medieval times or that penguins generally inhabit the southern hemisphere. This would not be understood to mean, of course, that no modern ship can be properly called a galley or that no bird in the northern hemisphere can be a penguin.

A specification of the normal realm of reality of category members should also be possible, but again allowing for exceptions and surprises. For example, it might be noted that unicorns are supposed to belong to a mythical realm, but if such a creature were ever found to exist in the real world, past or present, it would not be denied full status as an ordinary unicorn in other respects.

A group can be assigned to a prototypical category, as in the senate example above, but not vice versa. However, prototypical categories can belong to other more comprehensive prototypical categories in what is known as a taxonomic hierarchy, and information about position in this hierarchy should be representable. For example, a galley is a type of ship, which is a type of artifact. The extraction of information about taxonomic hierarchies from dictionaries has been studied extensively [Amsler, 1980; Ahlswede, 1988]. As before, we must also be able to represent exceptions. It may be known that cats generally have tails and that a Manx cat is a type of cat, but it should be possible to specify that a Manx cat generally does not have the expected tail.

Moving down the taxonomic hierarchy, we want to be able to specify a list or indefinite set of less inclusive categories. Any list may or may not be complete. For example, the 'car' category would include such lower categories as sedan, brougham, coupe, station wagon, police car, and so on. In principle, there should be no limit to how low a category can be in the hierarchy, which could include terms that are more and more specific even though no actual individual qualifies for inclusion. In actual practice, individual instance concepts can be considered the very bottom of the taxonomic hierarchy.

It has often been noted that certain categories in the hierarchy have a sort of privileged status, and the nouns that correspond to these are called basic level terms. These are neither at the top nor at the bottom of the hierarchy, but they are about as high as they can be and still have a recognizable form and specific function. We can have a very definite picture of a typical (or prototypical) giraffe, but it would be very difficult to form such a picture of a typical ungulate, which could be any hoofed mammal, thus placing the 'giraffe' category at or very near the basic level. Basic level terms tend to correspond to simple, unmodified and uncompounded nouns, which also points to their privileged status. See Taylor [1989: 46-51] for further discussion of the basic level concept.

Hierarchy level can also affect the fuzziness of category boundaries. This phenomenon is particularly evident among specimens of living things. It can be exceedingly difficult to determine the race of human beings, especially for the borderline cases, individuals of mixed ancestry, but the difficulty disappears when the problem is to decide whether an adult is a man or some other primate. The contrast is even sharper at still higher levels, such as plant versus animal, especially if we set aside creatures such as the euglena, which is a marginal member at best of either category.

Some scientists evidently feel that such distinctions should not be so clear throughout the history of life on earth, but morphological variation among biological specimens of the same basic kind appears to obey limits at present, leaving clear discontinuities between kinds [Marsh, 1976: 7-10, 76-

77; Lester and Bohlin, 1989: 88-89, 150-51], and the dearth, if not total absence, of indisputable transitional forms in the past suggests that the phenomenon is quite stable. "The known fossil record fails to document a single example of phyletic evolution accomplishing a major morphologic transition and hence offers no evidence that the gradualistic model can be valid" [Stanley, 1979: 39].

For instance, if bats really did acquire their wings gradually in the course of slow evolution, then the fossil of "the oldest known flying mammal" should belong to some shrew-like creature caught in transition to the form of modern bats. In fact, this fossil clearly evinces a bat [Jepsen, 1966; Gish, 1985: 108-10], supporting the hypothesis that shrews and bats have always been quite distinct. This is not to deny that the fragmentary remains of possibly extinct creatures can sometimes be difficult to identify with confidence, and some reservations may be necessary to account for superficial similarity, such as the striking resemblance of some moths to hummingbirds.

As suggested above, this phenomenon is not limited to living things. Consider again the domain of household items. Although a series of transitional items could be manufactured to bridge the gap between table and table cloth, making the distinction theoretically fuzzy, few if any such transitional artifacts actually occur in the real world since most of them would be useless. At a somewhat lower level in the taxonomic hierarchy is the distinction between a table and a chair. A high chair and a student's desk might qualify as transitional forms, since they combine table and chair functions, but a complete series of actual artifacts grading smoothly from table to chair can hardly be assembled. It becomes easier to get such a smooth intergradation and really fuzzy boundaries at still lower levels, especially below the basic level, where an end table contrasts with a coffee table, and a salad fork contrasts with a dinner fork. Discontinuities among non-living natural objects are also more pronounced at the higher taxonomic levels. Light and water do not intergrade at all, nor do brook and boulder, for example, but at the lower levels we do find nice intergradations among the numerous varieties of rock, or radiation, or bodies of water.

We still need to represent all the other kinds of descriptive information discussed in the section on individuals: absolute and relative attributes, modification of attributes, events, actions, and the perceptual characterizations. Once again it should be emphasized that these apply to category members generally, with exceptions freely allowed. Any image or sound represented for a category may be somewhat idealized, not necessarily associated with any individual member. The stored image of a goat, for example, may not show any specific color or pattern of marking. The same idea extends to other kinds of perceptual representations. As mentioned above, this sort of information may be missing entirely for categories at the highest levels in the taxonomic hierarchy.

One of the most important aspects of the representation of prototypical categories concerns the question of centrality and internal structure within categories. Some members are central members, clearly belonging to their category, while others are peripheral members, barely in the category if members at all. As suggested above, it is similarity to the prototype that determines the degree of centrality associated with a given member, but it appears that there is not just a single kind of similarity that counts, and the use of different types of criteria for similarity yields categories with members that are marginal in different ways. Such categories contain what we are calling internal structure.

Lakoff distinguished four types of criteria that relate to this issue. The first three, which he called **definitional**, **primary**, and **secondary**, are "capable of conferring category membership to a certain degree depending on various factors" [1972: 200]. We will shortly consider how these differ among themselves and give examples, but see Lakoff [1972: 195-201] for a fuller discussion.

Lakoff called his fourth criterion **characteristic though incidental**, "not capable of conferring category membership to any degree, but contributes to degree of category membership if some degree of membership is otherwise established" [1972: 200]. For example, when we say, "Esther Williams is a regular fish" [1972: 197], we are not placing her in the fish category at all. We are just saying that she possesses certain incidental characteristics of a fish (she "swims well and is at home in the water"), but this criterion cannot establish any claim to true fishhood by itself. Even if we take into account the three other types of criteria, considered next, we have no justification for calling Esther Williams a true fish.

A member can be placed in a category based on a **definitional** criterion even if a "primary criterion is below the threshold value for simple category membership" [Lakoff, 1972: 201]. For instance, "Richard Nixon is technically a Quaker" [198], means that he fits some technical definition of a Quaker. That is, Nixon is a member of the Quaker category, technically, even if his lacking other important qualities of a Quaker makes his membership somewhat marginal.

The **primary** type of criterion accounts for the difference in truth value between the last example and "Strictly speaking, Richard Nixon is a Quaker" [1972: 199], which is judged to be much farther from the truth. The phrase "strictly speaking" is supposed to indicate that all definitional and primary criteria are "above certain threshold values" [201]. Nixon fails to qualify as a Quaker, strictly speaking, because of his "religious and ethical views" that are deemed inconsistent with primary characteristics of true Quakers, even if he does meet all the definitional criteria.

The third type of criterion, called **secondary**, serves to explain the contrast between "Strictly speaking, a whale is a mammal" and "Loosely speaking, a whale is a fish" [1972: 199]. Consideration of primary criteria alone leads to a mammal label for whales, since they "breathe air" and give milk to their young, but one could ignore those criteria and place whales in the fish category instead, based on secondary criteria, namely "their general appearance and the fact that they live in water." The phrase "loosely speaking" in the example indicates that it is these secondary characteristics that are in view when we call a whale a fish.

Each piece of descriptive information for a given prototypical category could be associated with one of Lakoff's four types of criteria to allow some determination of centrality for putative members. Those pieces of information that are marked as definitional would apply to all members that are truly members to any extent. Those that are primary would not necessarily apply to marginal members, those that are secondary might fail to apply even to members closer to the center, and the number of applicable pieces of descriptive information marked as characteristic would help to show intermediate degrees of membership, somewhat blurring the major steps.

Finally, we should also represent degrees of faith in facts about categories. Even though the facts do not apply directly to individuals but to whole classes of individuals, it can be important to remember how much confidence can be placed in a given characterization of that class.

Therefore, all the kinds of information we want to represent for individuals and groups, we also want to represent for prototypical categories, though the exact nature and relative importance of these may again be different. In addition, we can add to the descriptive kinds of information the marking necessary for determining centrality as explained above.

F. Conclusion

Three types of nominal concepts have been proposed, and several different kinds of information, shown in the following finished list, have been suggested as desirable for each type:

1. spans of time during which a fact is true (of the individual)
2. realm of existence (real, fictional, legendary, imaginary)
3. membership in a group or a prototypical category
4. exceptions to facts inherited through such membership
5. absolute attributes (name, sex, ownership, binary features)
6. relative attributes (location, weight, hardness, friendliness)
7. modification of relative attributes (degrees, comparisons)
8. remarkable events and stories
9. habitual, continuous, or repetitive (prototypical) actions
10. images, sounds, and other perceptual characterizations
11. degree and justification of faith in any of that information
12. nature of members (substance sample or discrete individuals?)
13. list of members, if practical
14. descriptive information for establishing identity

Figure 3-3. Kinds of Information About Nominal Concepts

Many of these kinds of information are obviously needed and are commonly handled in knowledge representation systems, but others are more novel and idealistic. The goal has been to improve the quality of information available to any system that relies on encyclopedic knowledge.

An analogy from image processing may clarify the approach. Some images consist entirely of squares, each having a uniform hue or shade of gray that corresponds to the average detected in that region of the scene as photographed. If the squares are relatively few and large, the picture can be seriously distorted and difficult to visualize. There can be no fuzzy edges in such an image. On the other hand, if the squares are innumerable and relatively tiny, the picture can be quite clear. There may be some distortion either way, but the finer-grained resolution is clearly more informative.

In the same way, a knowledge representation system that requires all nominal concepts to have sharp boundaries might be computationally convenient, even as necessary as the use of pixels in an image, but it is a fiction that distorts reality. Even if the fuzzy boundaries among nominal concepts, for example, must be represented by a series of discrete steps away from the prototypical center, at least the resolution is improved, and better information is made available to the system.

Chapter 4

Frames and Data Structures

This chapter provides a high-level description of facts and frames, the basic abstract data structures used to implement the ideas for knowledge representation that were presented in Chapter 3.

A. Introduction

The basic structure for all the knowledge represented in our database is a frame consisting of a set of facts. Each fact is represented by one slot and a list of entries, if any. In addition to the main database of frames for storing encyclopedic knowledge, there is also a smaller set of special frames, called the slot catalog, for organizing and managing information about the slots of ordinary frames.

Each frame collects information about some concept, called the head. If a slot has no list of entries, then it is called a closed slot and corresponds to a unary relation or simply a feature of the head of the frame. The head is supposed to have that feature if and only if the closed slot is present in the frame. On the other hand, if a slot does have a list of entries, then it refers to a relationship between the head and each of the fillers, the concepts corresponding to the listed entries. These entries are either pointers to embedded frames, or they are terminal entries at the bottom of a chain of embedding. Thus both frames and fillers (or slot entries) correspond to concepts, and slots correspond to relations among concepts.

The proposed knowledge base uses three types of frames. There is one frame, called the **sense frame**, for each nominal concept or distinct noun sense that has been processed. The term 'noun sense' is further elucidated in the next section. Another type of frame, called the **name frame**, represents the noun itself and contains pointers to corresponding sense frames as well as information common to some senses of the noun, such as dialect and usage labels. The third type of frame, which is described next, is the **detail frame.**

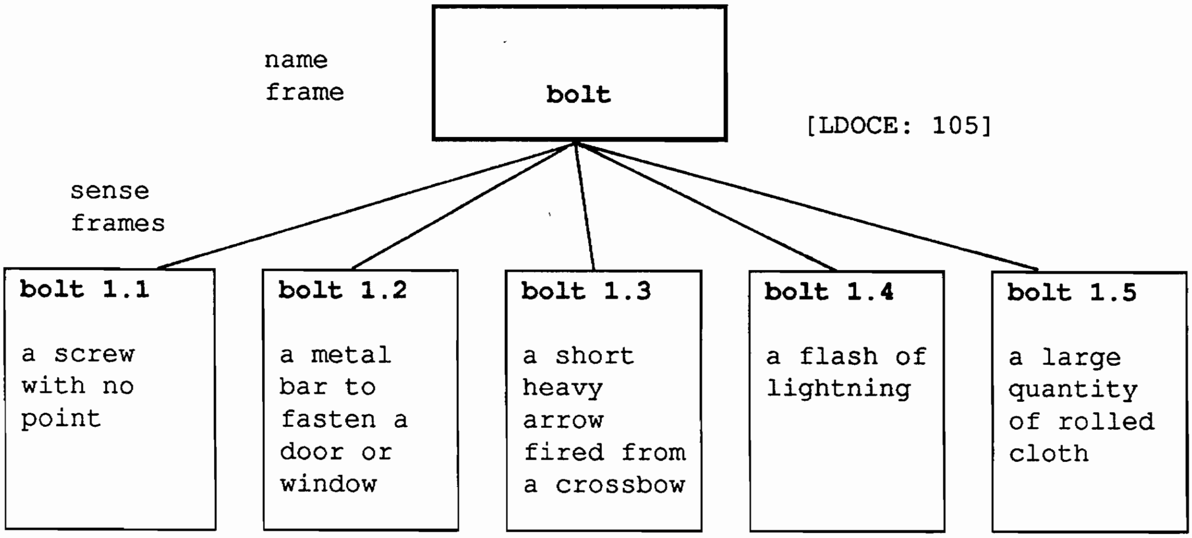
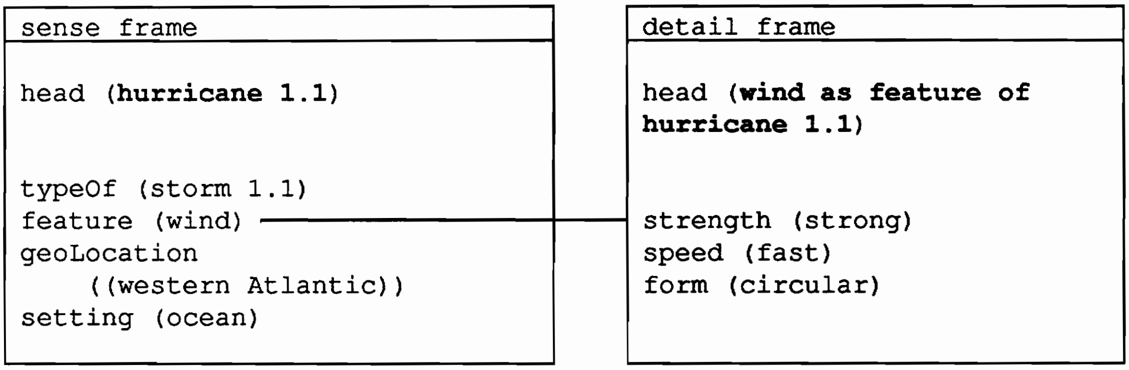


Figure 4-1. Five Sense Frames with Their Name Frame

Each detail frame collects information about a particular filler of a slot. For example, a hurricane is said to be "a violent storm with a strong fast circular wind in the western Atlantic ocean" [LDOCE: 513]. The sense frame for 'hurricane' should record the fact that it is a type of storm with a wind and that it occurs in the western Atlantic, but the details about the wind should go in a detail frame so that all of this information can be organized the same way (as slot/filler couples).



sense frame

head (**hurricane 1.1**)

typeOf (storm 1.1)
feature (wind)
geoLocation
((western Atlantic))
setting (ocean)

detail frame

head (**wind as feature of hurricane 1.1**)

strength (strong)
speed (fast)
form (circular)

Figure 4-2. Sense Frame with an Embedded Detail Frame

Notice that the filler of the **geoLocation** slot includes the modifier 'western' without resorting to a separate detail frame. This sort of simple modification is allowed when the kinds of possible modifiers are quite limited and explicitly specified as an option for the entries filling a particular slot.

All three types of frame have a slot called **head** to identify the concept described in the frame. The defined word or phrase, as spelled in its dictionary entry, fills the head slot of each name frame, and that same word or phrase plus a sense identifier fills the head slot of each sense frame. In detail frames, the head slot is filled by a frame-slot-filler combination serving as a pointer to a particular filler of some slot in some frame. Figure 4-1 shows the filler of each head slot in boldface, but no other slots are illustrated. Figure 4-2 includes all slot names with fillers given in parentheses.

Ideally, each entry filling a slot in a sense frame should point to another sense frame, so that there is no ambiguity about which sense is intended. In practice, an entry may remain a pointer to a name frame until the word can be disambiguated and the pointer redirected to the appropriate sense frame.

If frames are considered nodes of a graph with filled slots as labeled edges directed toward slot entries, the resulting structure can be seen as a network for temporary storage of encyclopedic knowledge relating to nominal concepts. A terminal entry, that is, a word without a corresponding frame in the network, could count as a degenerate frame to maintain the principle, and facts could be added to such a frame as new information becomes available for storage.

Taxonomic chains constitute one important type of path through such a network. These paths pass from one frame to another along edges corresponding to **typeOf** slots. The frame at the start of such a path lies at the lowest level in the taxonomic hierarchy, and successive frames at higher levels. (A sedan is a type of automobile, which is a type of vehicle, which is a type of artifact, ...) Other interesting paths follow edges corresponding to **analogue** or **synonym** slots.

B. Prototypes and Noun Senses

Prototypes correspond to sense frames, and each sense frame is associated with a particular noun sense. Since these frames will figure very prominently in our database, it is important to consider just what a noun sense is. That is the purpose of this section.

It is often difficult or impossible to draw objective distinctions among homonymy (such as 'bat', an animal, and 'bat', a stick of wood), one word with several senses (such as 'block', which might be either a city block, a building block, or a mental block) and one sense that covers a wide range of similar objects (such as 'ball', an object used in games, where the balls for pool, rugby, cricket, croquet, bowling, pinball, basketball, and table tennis, though all very distinctive, are perhaps still too similar to demand separate senses of the word 'ball'). One or more intermediate classifications may also be allowed so that separate but related meanings or subsenses can be grouped or one subsense further subdivided. The difficulty, then, is what to do when two meanings are assigned to a lexical form that superficially appears to be a single word. Should they be classified as homonyms ('bat'), or is there really just one polysemous word with two meanings ('block'), or should the two meanings be combined into just one with two subsenses ('ball')?

The decision appears to depend on some notion of similarity of senses, but useful objective criteria are often difficult to establish and apply consistently. In our treatment of the homonymy and polysemy continuum, we shall simply follow the arbitrary decisions adopted in the source dictionary, and each separate 'noun sense' for a sense frame corresponds to one meaning at the lowest level, where the dictionary makes the finest distinctions. It should be recognized, of course, that even finer distinctions in meaning could always be made, and the sets of senses, once determined, could be reorganized in various ways, perhaps according to frequency of occurrence [LDOCE: F30, F34], "basic" meaning first [LDOCE: F34], or historically oldest meaning first [W9: 10, 19].

Even a very specific subsense of a prototypical term defined in the dictionary is intended to cover a multitude of putative instances, none of which would really be identical. They would manifest various shades of meaning. Since the boundaries between subsenses are often fuzzy, it may be tempting to link shades of meaning somehow to the specification of fuzzy boundaries for prototypical concepts. One might take the five senses of bolt shown in Figure 4-1, for example, and mark the first sense as the most central, since it is supposed to be the "most common or most basic" [LDOCE F34], then assign the rest a more marginal or peripheral status in accordance with their sense number. This approach has been rejected in this thesis, however, for a couple of reasons.

First, the approach just suggested can introduce unnecessary distortion, making some senses more central than they really are. Continuing with the 'bolt' example, suppose we ignore the historical evidence that the crossbow projectile sense gave rise to the others [W9: 166] and accept what is now the most common sense as the most central, namely the threaded fastener sense, listed first. To be consistent, we now mark the roll of cloth sense as much more marginal than the crossbow projectile sense, but that is a rather dubious arrangement. Furthermore, the sense of a different word, like 'rivet', is treated as completely unrelated, even though it has as much or more in common with the most central sense of 'bolt' than the sliding bar sense, which is listed second.

Second, the resulting gradation does not seem to be what we really want anyway. The prototypical term 'bolt' should apply only marginally to individuals that fit one of the marginal senses, but we would like for a bolt of lightning or a bolt of cloth to be considered still very much a bolt, but just in the appropriate sense of the word. Apparently, sense distinctions documented in dictionaries do not readily map to shades of meaning naturally arranged around a central sense.

The important issue here, given the desire to accommodate prototypical categories, is deciding what those categories will be. Although a plausible case might be made for grouping some word senses from the dictionary to form a more inclusive or general category, it seems best to establish one prototypical category initially for each distinct noun sense found by the extraction program, with manual adjustment freely allowed. Once all desired adjustments have been made, each category will correspond to what we are calling a nominal concept. It should be understood that prototypical categories, even after adjustment, may be interrelated on the basis of their mutual similarity and that the more similar concepts are, the more difficult it is to distinguish between them, in general, regardless of how fuzzy their individual boundaries may be. (See Miller [1986] for a description of WordNet, a system that can relate sets of roughly synonymous terms in various ways.)

The fuzziness of prototypical category boundaries is largely due to the scales or relative measures used in their definition. It is understood that any individual measuring close to the specified point on a scale will generally be a more nearly central member of the category than one at a more extreme point on the scale. For example, a pebble is said to be "a small roundish smooth stone ..." [LDOCE: 757], so at least three scales are specified: size, shape, and texture. If a stone is either tiny or large, then it fits the pebble category more marginally than one that is just small, as specified. The rule is not entirely dependable, however. A razor is defined as "a sharp instrument ..." [LDOCE: 862], but if a particular razor is dull, perhaps because of abuse or a manufacturing defect, it does not seem to follow that it must be a marginal member of the razor category. Evidently, not all scales are of equal importance. Applicability markers, described below in the section on sense frames, should prove helpful in delineating some differences.

C. Name Frames and Their Slots

A name frame collects information about a word or phrase defined in the dictionary, so the head concept to which the frame refers is lexical or surface rather than semantic or underlying. In contrast, all sense frames refer to nominal concepts directly, and each one is normally limited to a particular noun sense, as explained in the last section.

Besides the head slot, name frames may have any of the slots described below. The exact name of the slot appears in boldface.

sense -- For economy, sense identifiers are not simply listed but are specified more compactly. The number of distinct senses at the highest level is listed first, unless there is only one, in which case the slot is filled by an empty list. If any sense is subdivided, the number of that sense is put in a sublist together with the number of its subsenses. If any subsense is also subdivided, then the pair of numbers for the high-level sense is augmented by another pair of numbers to identify the subsense and the number of its lower-level senses, and so on. For example, the entry (3,(1,2),(2,3,(1,2),(2,2))) specifies this subdivision of senses:

```
1     1.1...........first sense of homonym 1
      1.2...........second sense of homonym 1
2     2.1...2.1.1....first subsense of first sense of homonym 2
      ......2.1.2....second subsense of first sense of homonym 2
      2.2...2.2.1....first subsense of second sense of homonym 2
      ......2.2.2....second subsense of second sense of homonym 2
      2.3...........third sense of homonym 2
3     ..............homonym 3, with only one sense
```

Figure 4-3. An Example of Sense Subdivision

The sense identifiers for sense frames can be constructed automatically based on information entered in the sense slot. If a name frame lacks a sense slot, it is understood that only one sense is known to exist.

dialect -- The entries for this slot associate a given dialect label with certain senses of the defined word. If all senses share the same label or labels, then the list includes just the labels that apply, as in '(IndE PakE)'. If dialect labels apply to only a subset of the senses, then the label or labels are grouped with a list of the identifiers for all senses in the subset. For instance, the following dialect entry specifies that the first sense is British English but the second and third are American English:

```
(BrE (1 1 1) AmE (1 2 1) (1 3 1))
```

A missing dialect slot specifies that no particular dialect label applies to any of the senses.

usage -- The entries for this slot associate a given usage label, such as 'derog' or 'infml', with certain senses of the defined word in the same manner as explained above in the description of the dialect slot.

function -- The entries for this slot associate a given function or part of speech label with word senses, also in the manner explained above in the description of the dialect slot. This slot is actually redundant in the current project, since only certain nouns are being processed.

variant -- This slot associates the head with the lexical, regional, or stylistic variants given as fillers. In most cases, these variants simply show a different way to spell the defined word.

level -- One of the most remarkable features of prototypical categories is their hierarchical structure. The importance of knowing the level of a concept or its vertical position within the hierarchy has already been noted. For instance, one would expect to find less fuzzy boundaries for a prototypical category high in a hierarchy than for one at a level below it.

The entry for a level slot has three parts, first a level marker, then two numbers. If a noun occurs in the dictionary as part of a defined phrase, then it can be assumed that the concept corresponding to the simple noun is at the basic level while the concept for the phrase is below the basic level, so the level marker **basic** or **belowBasic**, respectively, appears in the entry filling the level slot for the noun or phrase. The first number after the level marker records the number of times the simple noun has been used in a defined phrase. The second number shows how often the same noun appears as the genus term (or main noun) in all the definitions processed so far. Genus terms tend to name concepts that are high in the hierarchy, so the **aboveBasic** level marker replaces a **basic** marker if the second count exceeds the first.

Since the level slot combines information from many definitions, the accuracy of its information should increase as more definitions are processed. Conversely, the information may be quite unreliable when it is based on only a small sample of definitions. One drawback of the design of the level slot is a gradual slowdown in processing speed caused by the need to keep updating the two counts in an expanding set of facts with these slots.

D. Sense Frames, Detail Frames, and Their Slots

Sense frames and detail frames share many of the same slots, so they are considered together in this section. Differences between them will be noted where necessary.

Since only nouns are being processed, each sense frame stores information about a particular nominal concept. Chapter 3 describes the three kinds of concepts: individual instance, group, and prototypical category. Nominal concepts of the third kind clearly predominate, since our source of information is a dictionary, but all three kinds can be handled. Detail frames inherit the classification of the sense frame to which they are ultimately attached, so no explicit class marking is necessary for detail frames unless there is a difference. Chapter 3 describes the various kinds of information that would be stored for these concepts in an ideal system, and this section describes slots designed to organize information that might actually be obtained from dictionary entries.

It might be argued that our slots and their fillers are nothing more than labels with names that may be deceptively meaningful in the eyes of a human observer but are actually quite meaningless as far as the system itself is concerned, since they have no practical connection to the real world. The criticism seems entirely valid when applied strictly to the shallow knowledge base of this scheme, but the theory (or hope) is that the meaning will become apparent and serviceable as the finished network is coupled with the missing pieces of a complete intelligent system, such as a powerful inference engine, for instance. Then it becomes possible to have rules that give different and appropriate results depending on conditions defined in terms of the structures of this partial system. The labels and structures derive real meaning primarily from their purpose and function in a total system. There can also be a valid complaint that even the meaning attached to our labels later will still fall far short of what a human reader would see in them. That alone is not a fatal flaw, however, provided the depth of meaning achieved yields satisfactory results and justifies development efforts.

Besides the obligatory head slot described in the introduction to this chapter, sense and detail frames can have a variety of other slots, several of which are described below. See Appendix A for a complete list of slots with a terse description of each.

nonPrototype -- The presence of this closed slot (a slot allowing no entries) indicates that the nominal concept named in the head concerns an actual instance of some prototypical individual, group, or substance, such as a particular person or object, or a specific group, or a sample of some substance. If this slot is absent, as it normally is in this project, it means that the nominal concept is a prototypical category. In either case, one of the following two closed slots further specifies the kind of concept.

group -- The presence of this closed slot (without entries) indicates that the nominal concept concerns neither a substance nor one individual entity but rather a group of them, either a prototypical group or some actual group. If neither this slot nor the substance slot (described next) is present, then it can be assumed by default that the concept concerns an individual entity in either a generic or specific sense.

substance -- This closed slot indicates that the nominal concept concerns either some substance, like air or gold, or some other mass noun, like clothing or furniture, rather than an individual or a group of individuals. This slot does not co-occur with the group slot, so there are only six combinations allowed for these first three slots:

1.	--	--	prototypical individual
2.	--	**group**	prototypical group
3.	--	**substance**	prototypical substance or mass noun
4.	**nonPrototype**	--	actual individual
5.	**nonPrototype**	**group**	actual group of individuals
6.	**nonPrototype**	**substance**	actual mass noun or substance sample

Figure 4-4. Combinations of **group**, **substance**, and **nonPrototype** Slots

synonym -- Each sense frame is linked directly to its own name frame. The synonym slot normally references yet another name frame linked in turn to a sense or range of senses that approximates the sense being described. The range could be narrowed by filling the synonym slot with the head of a sense frame rather than a name frame. The information stored in this slot is mostly semantic, in that it describes one meaning of the local name, but partly lexical in that it relates two names. In most cases, only one of the name frames will be linked directly to a sense frame with a synonym slot, and this will be the one with lower frequency or more limited applicability, perhaps restricted to only a particular dialect of English.

typeOf -- The entries filling this slot specify frames at higher levels in the taxonomic hierarchy, so the head of the local frame figures as a type of whatever the frame at the higher level describes. By default, facts about prototypical concepts higher in a taxonomic hierarchy are inherited by those at lower levels, and inheritance also applies to groups and their members.

essence -- This slot is so similar to the **typeOf** slot that they could be confused. For either slot, the filler tells what the head is, but in the case of the **essence** slot, it would not be correct to call the head a type of what the filler names. Instead, the filler identifies some other nominal concept that is the very essence of the head and that constitutes it. Consider 'creek' for example, which the LDOCE calls a body of water. In a sense, a creek is a body, and a creek is water. It is a type of body but not a type of water, which is actually just its essence. Water is what constitutes a creek.

equiMeasure -- This slot also tells what the head is, but in this case, there is little possibility of confusion. Each filler of this slot is a phrase, typically combining a number with some other measure term, which specifies an equivalent, perhaps in another system. A ready example would be '12 inches' filling the **equiMeasure** slot in a 'foot' frame. Note that a foot is 12 inches, but a foot is not a type of inches, nor are inches the essence of a foot, since the existence of a foot is quite independent of the existence of inches, at least in theory.

The remaining slots described in this section are used to represent information that may or may not apply across the board to all types, instances, members, substance or mass noun samples regardless of their centrality. The range of applicability for the fillers of these slots can be specified by grouping any of the entries with an applicability marker. These markers, listed in Table 4-A, designate the four types of criteria suggested by Lakoff [1972: 200] and discussed in Chapter 3.

Table 4-A. Applicability Markers

		Feature Applies to Instance If It Is		
Marker	Keyword in Definition	Central	Intermediate	Marginal
primary	esp.	yes	yes	?
secondary	usu.	yes	?	?
characteristic	often	?	?	?

Lakoff's first type of criteria, definitional, needs no marker, since it is the default. Definitional features apply to all instances that can be considered a type (or instance, or member, etc.) to any extent. A feature marked as **characteristic** does not necessarily apply to an instance at all, but to the extent that it does apply, the instance is considered a more nearly central or less marginal member of the prototypical category. The "keyword in definition" column in Table 4-A shows a word or abbreviation commonly used in LDOCE definitions that should trigger use of the corresponding applicability marker.

part -- It is common for dictionary definitions to include a description of some part or feature of a whole entity, and the entries of this slot list those parts or features. Usually, a detail frame is attached to record the description. There is no slot for listing features or body parts that are not especially remarkable. Since most animals with four legs can be assumed to have a head, neck, abdomen, and a tail as well, a dictionary is not likely to mention such features unless they are special in some way. Animals have a nose by default, for example, but an elephant is distinguished by having one that is extra long and prehensile, so these facts are recorded in a part slot and detail frame.

largerWhole -- This slot corresponds to the inverse of the **part** slot just described. The head in this case is a part of any nominal concept listed among the fillers of the slot. For example, a definition of 'trunk' might mention that it is the nose of an elephant, so 'elephant' would fill the largerWhole slot.

era -- The entries for this slot specify a general period of time during which the head exists. An entry may also be more specific, identifying a particular century, for instance.

habitat -- This slot lists the normal habitat or habitats of the plant or animal named as the head of the frame.

geoLocation -- The normal geographic (or extraterrestrial) location of the head is specified with this slot.

treatment -- Entries filling this slot, unlike almost all of the others, are complex verbal concepts, not simple nominal concepts. They specify some action that is normally or characteristically performed on the head. For example, the phrase 'made to fly' can be the filler of the **treatment** slot in a 'helicopter' frame, since one kind of treatment typically received by a helicopter is its being "made to fly" [LDOCE].

Other Slots -- In addition to the slots described individually above, there are a number of other absolute and relative attribute slots that may best be described as a group. Most of these should be rather self-explanatory in light of the examples already given above.

Some other more or less common absolute attribute slots are (1) **sex**, with a single entry, such as male or asexual, (2) **number**, with an integer indicating a number, such as the number of humps on a camel, and (3) **absenceOf**, with a (usually short) list of nominal concepts which are absent from the concept being described.

The inventory of relative attribute slots is considerably larger, yet they are so easy to understand that only a few further examples should suffice: (1) **color**, to specify the normal color of instances of this nominal concept, (2) **size**, to indicate their size in relation to other examples of the same type, whose average size serves as the standard of comparison, (3) **analogue**, to associate either the whole local frame with another frame for a similar concept or just a local feature with a similar feature in another frame, and (4) **attribute**, to list less common, still unclassified modifiers that apply to the head.

In addition to the applicability markers mentioned above, the entries of relative attribute slots can be grouped with degree markers to specify a particular amount of salience for a given quality. Only four explicit degrees are recognized, all suggested by words found in LDOCE entries. In scale order, most salient first, the markers are: **extremely**, **highly**, **considerably**, and **slightly**. A fifth degree, the implicit default, lies between the latter two.

Attribute entries are not always single modifiers. Sometimes several alternatives must be listed, or several modifiers may apply jointly, and it may or may not be possible to supply a complete list of them. Once again, markers are used, in this case, inserted at the start of the list of modifiers: (1) **oneOf** (one listed modifier applies), (2) **justThese** (all of these modifiers and no others in the same domain apply), and (3) **atLeastThese** (all of these and possibly others). The resulting list of modifiers or attributes, headed by the list marker, constitutes a single, complex entry.

E. The Slot Catalog

The slot catalog is essentially a table of information on the set of slots used by the SIV knowledge extractor. Each row of this table stores information about a particular slot in some name, sense, or detail frame. The columns of the table are (1) the name of the slot; (2) its type (name or sense); and (3) a terse, prose description of the intended meaning of the relation that the slot represents.

Since the slot catalog is designed to facilitate the management of name and sense frames, not for the storage of encyclopedic knowledge about the external world, very little needs to be said about it here. The entire catalog appears in Appendix A.

Of course, one could imagine special structures for information on the slot catalog as well, and the recursion could be endless. In fact, the bottom of the recursion is right here, and the idea of carrying it even this far is only to achieve a clearer view of the meaning, purpose, and function of all the diverse slots in the three major types of frames.

F. An Illustrative Example

In this section, several frames of all three types are presented in two formats: (1) in Figure 4-5 as blocks of single-spaced lines, each block representing one frame, and each line representing one slot, named at the beginning, with its fillers in parentheses, and (2) in Figure 4-6 as part of a network, with each frame shown in a box and related frames joined by connecting lines. Although there is considerable overlap, the two figures (based on definitions in LDOCE) do not show exactly the same frames and slots. Naturally, the example could be greatly expanded and elaborated, but it was kept rather simple to promote clarity.

Figure 4.5 Some Frames with Their Slots

```
head (cat)

sense (1 (1 3))
usage (derog (1 3 1))
function (noun)
```

```
head (woman)

sense (1 (1 3))
function (n)
```

```
head (person)

sense (1 (1 4))
function (n)
```

```
head (cat 1.1)

level (basic 2 2)
typeOf (animal)
size (small)
feature (leg fur claw)
```

```
head (woman 1.1)

typeOf (female)
attribute (human
     fully-grown)
```

```
head (person 1.1)

typeOf (being)
attribute (human)
```

```
head (cat 1.2)

typeOf (animal)
size (large)
```

```
head (cat 1.3)

typeOf (woman)
attribute (mean unpleasant)
```

```
head (Manx cat)

sense ()
function (n)
```

```
head (tiger)

sense (1 (1 2))
function (n)
```

```
head (lion)

sense (1 (1 2))
function (n)
```

```
head (Manx cat 1.1)

level (belowBasic)
typeOf (cat)
absenseOf (tail)
```

```
head (tiger 1.1)

typeOf (cat 1.2)
geolocation (Asia)
size (highly large)
attribute (fierce)
```

```
head (lion 1.1)

typeOf (cat 1.2)
color (yellowish-
               brown)
geoLocation
             (Africa)
```

```
head (lion 1.2)

typeOf (person)
attribute (famous
          important)
```

Figure 4.5 Some Frames with Their Slots (continued)

head (**Siamese cat**)

sense ()
function (n)

head (**Siamese cat 1.1**)

level (belowBasic)
typeOf (cat)
part (eye hair fur ear
foot tail face)

head (**eye as part of Siamese cat 1.1**)

color (blue)

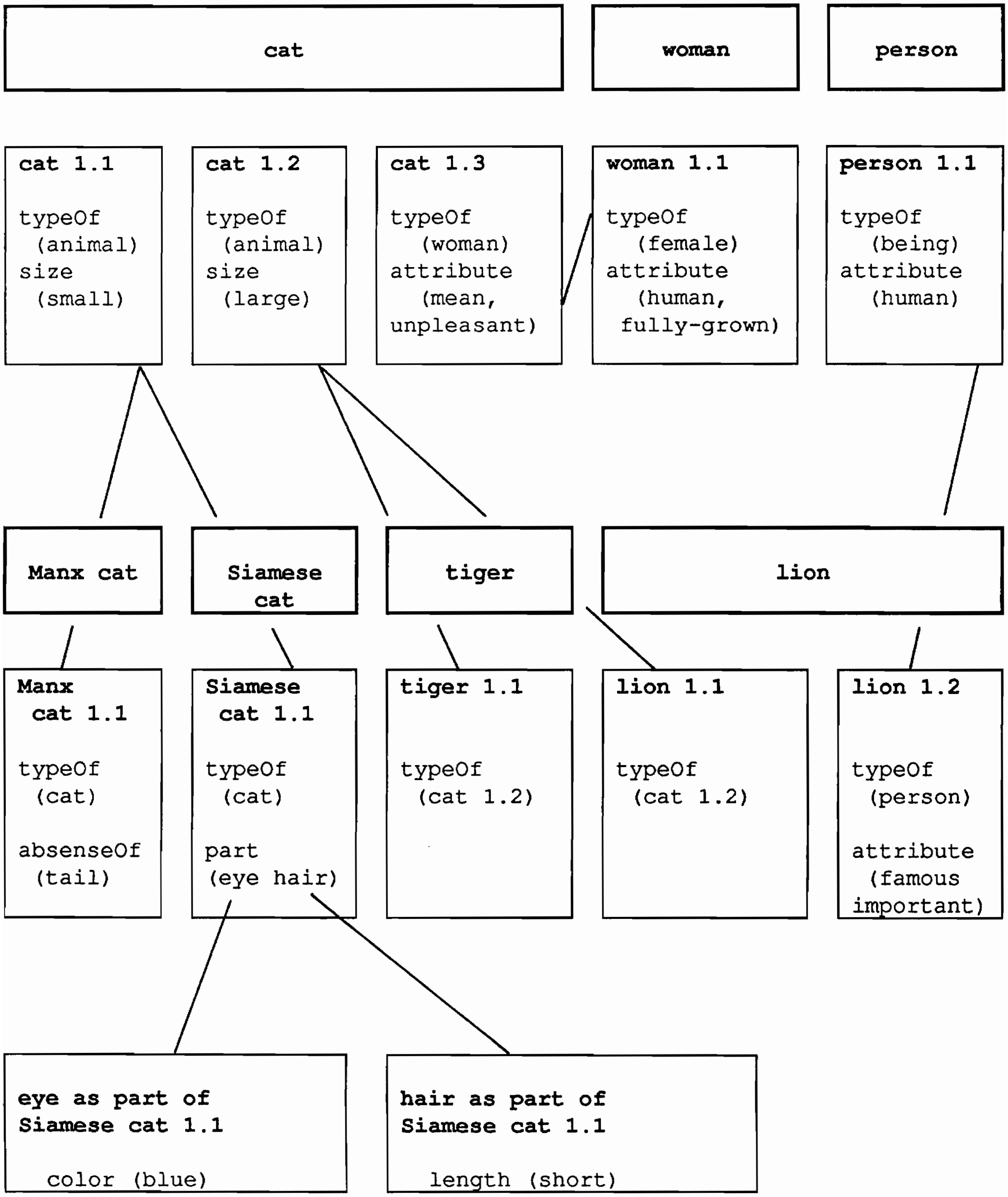
head (**hair as part of Siamese cat 1.1**)

length (short)

head (**fur as part of Siamese cat 1.1**)

color ((oneOf pale-grey light-brown))

Figure 4-6. Portion of a Frame Network



Chapter 5

Salient Information in Dictionaries

This chapter identifies various kinds of information actually available in LDOCE noun definitions and suitable for filling the data structures described above. Special attention is given to desired information about prototypical categories.

A. Introduction

The frames and data structures described in the last chapter would serve little purpose if there were no practical way to populate them with facts of the sort that need to be represented and in sufficient quantity to avoid the brittleness problem noted by Lenat and Guha [1990: 3-4]. The frames could, of course, be filled in by hand, but that method is expensive, especially if the desired knowledge base is really large. Perhaps a major portion of the whole task could be accomplished by relatively cheap automatic processing.

That task is not limited to the organized entry of data into some knowledge base. It also includes locating and extracting the facts of interest from some trusted source of general knowledge. Once again, this source might be people, who would impart their own knowledge directly. Although considerable human input will surely be necessary in any case, our thesis is that large documents containing trustworthy information in machine-readable form could also serve as an important source of information. Dictionaries stand out as an excellent case in point, so consideration will be concentrated on them in this chapter, but other kinds of documents could also serve the same purpose, as explained in Chapter 8.

Of all the facts that might possibly be gleaned from dictionaries, many will be too trivial to keep without hopelessly cluttering memory and impeding the utilization of those that really matter. Besides that, many facts, whatever their usefulness, do not readily lend themselves to automatic extraction. This may be due to rarity or complexity. If each fact must be recognized by certain key words or word patterns in the source document, and the words or patterns for many such facts are all very rare, it probably will not pay to include them and bog down the extraction process. If recognizing a fact means having to identify a very complex pattern, again it may pay to ignore it, even if it is not so extremely rare.

In this chapter we consider various kinds of "salient" facts, namely those that promise to be useful, belong to a reasonably common type, and can be readily sifted from the wealth of information found in the whole dictionary. These criteria for salience are all far from absolute, and the threshold of tolerance can vary due to differences both in personal judgment and in availability of project resources. The final section of this chapter reviews the kinds of information about prototypes desired for the ideal system described in Chapter 3 and evaluates the LDOCE in particular as a repository of such information.

Still maintaining the earlier focus on facts about nominal concepts, we will center our attention here on salient information in noun definitions, but the handling of these definitions may suggest ways to handle the ones for other parts of speech, especially verbs, adjectives, and adverbs. Information found in definitions of articles, pronouns, prepositions, conjunctions, interjections, and certain common words in other categories is presumably much less amenable to the type of automatic processing envisaged in this thesis, but fortunately, such words all belong to small, closed classes. Besides the definitions, some lists and tables found in a dictionary could also supply valuable information, but these would require special handling as well.

B. The Genus Term

Almost every noun definition contains a single genus term, that is, some more general noun that covers a wider range of nominal concepts than the definiendum, which is the noun being defined. The genus term conveys extremely useful information because of the principle of inheritance. In general, whatever is found to be true of the genus term can be assumed to be true of the definiendum as well, not counting explicit exceptions. Since the genus term is the more general, it is also the more likely to be known and understood.

For example, suppose the definiendum is 'coati', and the genus term used in the definition is 'mammal' [W9]. The definition need not state explicitly that 'coati' refers to a furry or hairy vertebrate animal whose young live on milk, because all of this information follows, by inheritance, from the fact that a coati is a mammal.

If inheritance is one side of the coin, instantiation is the other. While knowledge of a definiendum is enriched by information about its genus term, specific information in the definition for a definiendum also enriches knowledge of the genus term through instantiation. Again using the 'coati' example, the definition in W9 mentions that this particular instance of a mammal has a flexible snout. In a system with inference capability, this fact could be added to all others gathered from definitions for other mammals to construct a more detailed description of the mammal concept itself.

As noted in Chapter 3, the more general term is more likely to have the less fuzzy boundaries, so this wealth of information can be useful in illuminating its naturally mysterious conceptual frontiers. We know that mammalian snouts can be flexible, since a coati has one, but from the fact that this needed special mention in the definition for coati, we can also conclude that mammalian snouts are generally quite rigid. Of course, this conclusion could be used by an inference engine to enrich the information on other mammals by appealing to the principle of inheritance, as explained above.

If the genus term is the only word in a definition, then it can be assumed that its range of reference is approximately the same as that of the definiendum, hence they are essentially synonyms. This is the case for about an eighth of all LDOCE definitions.

Since the presence of a single genus term in noun definitions is almost universal, the information it provides is fairly salient, but there are cases where no genus term can be identified or where more than one is provided, complicating the extraction process.

Examples of the first type of definition are gerund phrases, such as "being extravagant" in a definition for 'extravagance' [LDOCE], and noun clauses, such as "that which overflows" in a definition for the noun 'overflow' [LDOCE]. It hardly makes sense to describe extravagance as a kind of 'being', or overflow as a kind of 'that'. Although these abnormal structures might be recognized and paraphrased to yield a very generic genus term ('the *state* of being extravagant' or '*something* that overflows'), the incentive to do so does not seem very great.

The second type of definition can be fairly easy to accommodate if all genus terms are listed in the definition without intervening modifiers, as in "a group or party within a larger group ..." in a definition for 'faction' [LDOCE], where 'group' and 'party' are two alternative genus terms. If modifiers do intervene, however, it can be very difficult to identify more than one of the genus terms reliably due to the large number of possible patterns and their individual rarity. One simple example of this kind is "a small hook or hooked instrument" in a definition for 'crotchet' [LDOCE], where 'hook' and 'instrument' are the two genus terms.

A much more common complicating factor is the presence of a classifier, a word that masquerades as the genus term, when only the syntactic structure of the definition is considered. For instance, consider the definition for owl, "a type of night bird ..." [LDOCE]. Since the syntactic head of the noun clause is 'type', this might appear to be the genus term, but it does not really qualify; 'type' is hardly a more general term that includes owls. From a strictly semantic point of view, the classifier 'type' merely elaborates on 'bird', the real genus term in this example. It elaborates by specifying that the genus term, as modified in the definition, covers several distinct types or classes of birds, one of which constitutes the category of owls.

C. Attributive Modifiers

All modifiers of the genus term serve to narrow its range of reference to one that more closely fits that of the definiendum. Attributive modifiers, those that come before the genus term, are particularly easy to recognize and categorize. Since the part of the definition where these modifiers occur is usually devoid of other nouns, there is little uncertainty about which noun they modify. In contrast, modifiers that follow the genus term frequently modify some other noun, one that comes after the genus term.

The category of an attributive modifier indicates what kind of specific information it provides. For instance, the definition for 'quince' begins, "a hard fruit related to the apple ..." [LDOCE], where the adjective 'hard' serves as an attributive modifier of the genus term 'fruit'. In this case, the modifier clearly belongs to the category of words that describe firmness, and we can infer that the quince belongs nearer the hard extreme of the firmness scale for fruit. Attributive modifiers typically refer to some kind of scale or continuum, most exceptions being complex constructions like 'four-legged', 'sea-dwelling', and 'meat-eating'.

Like many others, the firmness scale has different absolute values at its endpoints depending on the genus term. To extend the same example, the firmness scale for fruit differs from the one for minerals, so even though quartz is much harder than a quince, its definition also uses the attributive modifier 'hard' in the same sense. In fact, a soft mineral could easily be harder than a hard fruit by any absolute measure of firmness.

That point, like some others made in this chapter, might seem more germane to a discussion of inference engines than to one about knowledge extraction, but it is important to realize that the apparent paradox can be resolved through reference to the genus term. The firmness facts make sense when properly interpreted, and similar observations apply as well to modifiers in other categories.

The information provided by many attributive modifiers will certainly meet the criteria for salience, but a couple of negative aspects should be mentioned. One is that the scales or qualities defined by these modifiers tend to be very coarse-grained. The presence of adverbial modifiers, described in a later section of this chapter, can supply a little more scale detail, but they are relatively uncommon. As a result, only two or three positions may be reliably identified along a given scale. This uncertainty in scale position is consistent with the fact that no single point on the scale can be correct for all instances of a given nominal concept being defined. Not all quinces are equally hard, and a cooked or rotten one could even be quite soft. Note that this individual uncertainty depends on the nominal concept involved. Individual samples of quartz, for instance, would not manifest the same variation in hardness that quinces do.

Another caveat concerns differences in the level of confidence that can be placed on the categorization of modifiers. A modifier like 'red' or 'green' is almost certain to refer to color, even though more figurative interpretations are technically possible. On the other hand, a modifier like 'warm' should normally refer to temperature, but it is not very hard to find examples where it does not. For example, it is used in a definition for 'love' and in one for 'overcoat', though in neither case does it literally describe the temperature of an object (interest or a coat, respectively). This is not necessarily a fatal flaw if the danger of misinterpretation is recognized and appropriate measures can be taken to tag or edit the misleading facts.

D. Prepositional Phrases

Appositive modifiers come after the genus term in the definition. The prepositional phrase is one type that can also come before the genus term, though this is very rare. Other types of appositive modifiers are relative clauses and verbal phrases, which are covered in the next section. Appositive modifiers often constitute a major percentage of the text found in a definition and may therefore contain a great deal of potentially useful information, but only the first in a series of these modifiers is sure to refer to the genus term. Later appositive modifiers may refer either to that term or to some other noun in a preceding phrase or clause.

The most distinctive part of a prepositional phrase is the preposition that introduces the phrase. Although prepositions occur frequently in definitions, the list of distinct prepositions is fairly short. Each preposition refers to a relationship that holds between its object and the nominal concept that the whole phrase modifies. Unfortunately, most prepositions are not very specific, so to determine what relationship is actually intended, it is usually necessary to consider what category of object is involved. Hence the information is less salient, more difficult to recognize and categorize.

Consider the phrases introduced by the preposition 'with' in these fragments of the definition of 'bullfinch' and 'bullfrog' [LDOCE]: "... songbird with a bright reddish breast ..." and "... frog with a loud unpleasant cry." In both definitions, 'with' refers to a relationship between its object ('breast' or 'cry') and the genus term ('songbird' or 'frog'), but it is rather vague. Since a breast but not a cry is a body part, we can conclude that a bright reddish breast is a part of the body of a certain kind of songbird and avoid the mistaken conclusion that a cry is a part of a certain kind of frog. In the latter example, the preposition actually relates the cry to the frog as its typical sound. (Compare the "standard basic lexical function" called "Son" by Mel'čuk and Zholkovsky [1988: 55, 63].)

The intended relationship to the genus term can often be easier to recognize when the prepositional phrase is combined with a participle. Two common examples are 'used for' and 'made with', where the range of likely relationships is already so small that the category of the object of the preposition can be safely ignored. There are also word sequences that are best treated as a unit functioning as a preposition, such as 'together with', 'instead of', and 'apart from', but these tend to be so uncommon and far from the genus term that the information they contain is almost always very difficult to extract efficiently.

Since they usually lack an active verb, prepositional phrases are most likely to contain static information about the genus term, such as its purpose, composition, or nature. In many cases, this information is just not salient enough and must be left unclassified or simply ignored.

E. Relative Clauses and Verbal Phrases

Except for prepositional phrases, practically all other appositive modifiers in a definition will contain some form of a verb, and the category of this verb is the key to whatever salient information may be available. Relative clauses are introduced by a relative pronoun, possibly preceded by a preposition. Subordinate clauses introduced by a subordinate conjunction are also possible, and both kinds of clauses use a finite form of the verb. Verbal phrases lack an explicit subject, and the verb is either an infinitive or a past or present participle. This great variety of syntactic structure and the fact that the key verb in these constructions belongs to an open class both militate against the easy extraction of information from them.

The obstacles are not quite insurmountable, however, and there really is a wealth of information waiting to be discovered in these structures, probably much more than any program can conveniently handle. The trick is to identify which elements correspond to a distinctive slot and which correspond to the filler of that slot. There are many possibilities, of course, with probably at least as many potential slots as there are different verbs or event concepts.

For example, a definition of 'lollipop man' includes the relative clause "whose job is to stop traffic" [LDOCE]. The subject of the clause suggests a **job** slot filled with 'to stop traffic'. The same slot could be filled based on information in a participial phrase that occurs in a definition for 'longshoreman': "employed to unload goods"

[LDOCE], the filler now being 'to unload goods'. On the other hand, one might choose to treat these as separate kinds of facts, one about an informal job, and one about formal employment, based on the exact wording used in the definitions.

One way to deal with the great variety of potential slots is to establish one that is very general, such as **typicalAction**, so that any clause or phrase that does not fit a specific slot can be saved anyway for later discrimination. Even the examples above could be categorized in this general manner. By studying the list of facts placed into the most general category, one can spot types of facts that are common enough to suggest a candidate for a new, more specific slot.

As suggested in the preceding section, prepositional phrases can figure prominently in verbal phrases, and certain combinations of verb and preposition can be readily identified to extract and categorize useful information. In addition to the examples cited there, consider also the following combinations: found in ... (typical location), living in ... (habitat), pulled by ... (tractive agent), worn by ... (typical wearer), worn over ... (underlying layer), etc. [LDOCE].

Verbal phrases with prepositions are fairly common, but there are also verbal phrases that lack them. The phrase may consist of nothing more than a verbal form, as in a definition for acquisition, "something or someone acquired" [LDOCE], or the verbal form may be followed by one or more adverbs or by a subordinate clause. The proper category for these constructions depends almost entirely on their verb, which belongs to a large class of words, so the temptation is to have a closed slot for each verb or simply to put all these facts into some generic category.

F. Information from Other Parts of a Definition

The main text portion of the definition of a noun is almost invariably a noun phrase, and the earlier sections of this chapter have covered the kinds of information that can be extracted from the major constituent elements of such a phrase. This section covers adverbs, which may also occur in the noun phrase core of the definition, and the various labels, symbols, examples, and other information that accompany the core and complete the part of an entire entry that is tied to a given sense of some definiendum.

Given our interest in prototypes, some adverbs demand special attention due to their importance in showing centrality, locating instances of the definiendum near the center or the margin of some prototype with fuzzy boundaries. The most common adverbs of particular interest in this regard are 'often', 'sometimes', 'especially', and 'usually'. The first two of these adverbs modify descriptive information in the definition that Lakoff [1972: 200] would classify as characteristic, as explained in Chapter 3, while 'especially' and 'usually' seem to mark primary and secondary characteristics, respectively, according to the same classification.

For example, in the definition for bullock [LDOCE], the genus term 'bull' is described as "often used for pulling vehicles," suggesting that a bull may very well be a true bullock, even if it is not used for this purpose, but bullocks characteristically do serve in this way, nonetheless. This characteristic is also incidental in the sense that a similar animal should not be considered a bullock to some extent simply by virtue of the fact that it serves that same purpose.

The phrase "used esp. for army calls" describing a "brass musical instrument" in the definition for 'bugle' [LDOCE] exemplifies a primary description, one that would not necessarily apply to marginal members of the category. There might be a brass musical instrument similar enough to the more central instances of the bugle prototype to be called a bugle, but such a marginal instance would be much less likely to be heard sounding an army call. The converse suggests that a description marked by 'especially' tends to be much less incidental than one that is only characteristic. If an instrument of any kind is used for army calls, then one might reasonably call it a bugle, in a sense, regardless of its other qualities.

The definitions for two senses of 'sideshow' [LDOCE] illustrate secondary descriptions. For the first sense, the genus term is 'show', and the description is "usu. with strange people." A true sideshow, even one fairly close to the prototypical ideal, might very well feature only perfectly ordinary people (though they would probably be doing something rather strange or remarkable). For the second sense, the genus term is 'activity', and the description is the attributive modifier 'usu. amusing'. A true sideshow in this sense could easily not be amusing, at least not to some observers.

Both secondary descriptions appear to be capable of "conferring category membership to a certain degree" [Lakoff: 200]. Although the effect is not so pronounced as with primary descriptions, any show with strange people could be termed a sideshow, in the first sense, and any amusing activity might reasonably be called a sideshow to some extent in the second sense of the word.

These correlations with the Lakoff criteria may be somewhat uncertain and questionable, but they seem to offer the best promise of getting information about centrality from dictionary definitions. Nevertheless, even if these precise correlations do prove to be largely invalid, marking them in our representations should at least preserve most of the information that the corresponding adverbs are supposed to convey.

As mentioned earlier, adverbs are also used to identify approximate positions along a scale or continuum of values. Some representative adverbs used for this purpose are 'extremely', 'very'/'highly', 'rather', and 'slightly'. All of these suggest positive displacement from any neutral position on the scale, and they are listed in order of decreasing displacement. At least this correlation seems to be fairly secure, and like all the adverbs, they are also comparatively easy to identify and extract from running text.

A dictionary definition normally contains quite a bit of other information besides what is provided in the definition proper, the text that says what the defined word or phrase means. Most of this accessory information is given in specially marked portions of dictionary entries, perhaps printed in a distinctive font, so it is quite readily identified and extracted if desired. Access to all parts of dictionary entries should improve as machine-readable versions become available with tags that are specifically designed to facilitate automatic identification of text elements [Amsler and Tompa, 1988; Blake *et al.*, 1992: 218-19; Fought *et al.*, 1993: 33; Goldfarb, 1990].

Of the various kinds of accessory information that are included, probably most are expressed in terms of code symbols or standard labels that can be briefly listed. Items of information that normally fall into this category include the following:

- part of speech of the definiendum
- predominant usage (literary, informal, slang, etc.)
- associated dialect
- grammatical or syntactic attributes (count noun, transitive verb,
  etc.)

There is often some default value understood if no symbol or label is used, so for instance one could take for granted that a word is used practically worldwide if no dialect label is included.

One kind of accessory information usually included is the pronunciation, including stress or perhaps intonation marking. This may be slightly more difficult to accommodate with the other material, due to the use of special characters, and it would presumably be of little interest to a computer system not designed to produce or recognize audible speech. Etymological information might also be ignored for lack of interest, but it is often provided and could be readily extracted if needed.

A dictionary entry may also include illustrative examples. Given our interest in enriching a database of knowledge about nominal concepts, it would seem that these could be a gold mine of deeper insights into their meaning, but unfortunately, that wealth of information is not so easily dug out. The problem here is that the examples are typically complete statements, not simple noun phrases or verb phrases, so all the problems of parsing unrestricted text arise, and then there is no easy way to relate the parse, if it can be trusted, to neat, recognizable facts about some particular concept or prototype. One approach to this problem is considered in the conclusion to this thesis.

Still other information can be obtained by reading between the lines, as it were, even though it is not provided explicitly in any one definition. A good example of this is the information stored in **level** slots, which is based on counts of noun occurrences in specific parts of all definitions, as explained in some detail in the preceding chapter.

G. Information About Prototypical Categories

It appears that every kind of salient information identified above can be information about a prototype or prototypical category. This section takes stock of the kinds of information desired for representation in an ideal knowledge base, as detailed in Chapter 3, and sorts out what is or is not available in the LDOCE. Nouns that refer to individual instances and to specific groups of such instances may be ignored here, since they do not correspond directly to prototypical categories, but they account for only a tiny percentage of all nouns defined in ordinary dictionaries (about one percent in the LDOCE).

In Figure 3-3 at the end of Chapter 3, fourteen different kinds of information were listed as desirable for an ideal system. Three of these are never (or at best, rarely) provided in LDOCE noun entries: (8) remarkable events and stories; (11) degree and justification of faith in the information; and (13) list of group members. Another kind, (10) images, sounds, and other perceptual characterizations, is also lacking in the LDOCE machine-readable files, but the printed edition is generously furnished with excellent sketches that could be scanned and stored as bit maps. All other kinds of information listed do appear to be included to some extent in the LDOCE.

The first two kinds of information are provided fairly consistently where appropriate: (1) span of time ("in former times," used in a definition of 'catapult', for instance) and (2) realm of existence ("imaginary" as an attribute of 'fairy' and "in children's stories" in the definition of 'ogre'). Antiquity, loosely defined, is the only span of time likely to be specified, and for almost all nouns, no time span is stated, which implies modern times by default. Similarly, the realm of existence is implicitly actual reality in nearly every case.

The third kind, (3) hierarchical organization or membership in a group or a prototypical category, is provided mainly through the genus term, but its level is likely to be very near the top of the hierarchy, with intervening levels between the genus term and the definiendum ignored. For instance, the definition of 'giraffe' places this creature in the animal hierarchy, skipping the ungulate and mammal levels. Some hierarchical level below that of the definiendum is occasionally provided through a short list of examples: "big cat ... such as a lion or tiger."

The fourth kind of information is often available in some form: (4) exceptions to facts inherited through membership in a category. An exception can be indicated rather subtly, as demonstrated above in the flexible snout example for 'coati', or more directly, as in "all people working ...

except officers" in a definition for 'crew', although this is rather rare.

The next two kinds, (5) absolute attributes (sex, number, etc.) and especially (6) relative attributes, constitute the main ingredients of LDOCE noun definitions. Attributes are expressed through modifiers of all kinds, including the attributive and appositive modifiers described above. In general, of course, these will be held nearly to the minimum necessary to identify the definiendum and to distinguish it from similar prototypes, so many potentially interesting attributes must be left out.

Given the high percentage of salient information of the latter kind, it is little wonder that (7) modification of relative attributes (degrees, comparisons) is another kind of information commonly included. This is usually expressed through a single adverb or, less often, through a subordinate clause.

Another kind of information, (9) habitual, continuous, or repetitive actions, is sometimes included, usually conveyed by means of a compound with a participle (such as 'grass-eating' in a definition for 'sheep') or a relative clause (such as 'which jumps along on its large back legs and which carries its young in a pouch' in a definition for 'kangaroo').

Every noun definition is supposed to specify (12) the nature of members of a prototypical category (whether substance sample or discrete individuals). The LDOCE "grammar code" U is interpreted to mean that the noun sense does not refer to discrete individuals.

The final type is (14) descriptive information for establishing identity. As far as prototypes are concerned, this type may be difficult to distinguish from absolute and relative attributes, but any description distinguished as important for establishing category membership should qualify. As noted in the last section, adverbs apparently yield some information about centrality, which applies to descriptions of all kinds, but they may distinguish a description as a primary or secondary characteristic, as required to recognize this last type of information.

The foregoing analysis of various kinds of information that can be extracted from a dictionary like the LDOCE would be incomplete without some consideration of the depth and breadth of that information. Since we are extracting information from a general-purpose dictionary, not one devoted to a particular field, such as medicine or electronics, we can expect to find concepts included that find application in every field or at least in many areas of interest. On the other hand, most, if not all, definitions do not include every kind of information identified in this thesis, and regrettably, the information that is included will also be incomplete, even when consideration is limited to a single kind of information.

Recalling the definition of apple, cited in Chapter 2, notice that explicit information is limited to relative attributes (e.g., shape and color), absolute attributes ("with ... flesh and ... skin"), and category membership ("a ... fruit"), so most of the different kinds of information found elsewhere are missing here. Continuing with the same example, consider the incompleteness of the information provided about relative attributes. The definition provides no information about the feel, smell, or taste of apples (unless 'juicy' somehow qualifies) nor about their weight or value, to mention just a few attributes that could have been added.

The reasons for such brevity are both obvious and sound, so the observations just made are not intended to criticize the dictionary. Conciseness is simply a limitation that should be taken into account when anyone plans to use a dictionary as a source of encyclopedic knowledge. It may be instructive in this regard to compare the length of the LDOCE definition of the noun bottle, specifically, sense 1, the primary sense, with the length of Wierzbicka's "explication" of roughly the same sense in terms of her "conceptual analysis" [Weirzbicka, 1991: 89-90]. The former contains 21 words, the latter 692, nicely illustrating Wierzbicka's suggestion that "the semantic structure of an ordinary human sentence may be about as simple as the structure of a galaxy or of an atom" [1991: 75].

To summarize, LDOCE noun definitions do contain 10 of the 14 types of desired information. Although this information is generally shallow, coarse-grained, and sometimes difficult to extract, it does cover a very wide range of topics. One might reasonably hope to use any full-sized dictionary to build a good skeletal framework for a complete database of world knowledge, but it would need to be fleshed out using other sources of information. One means of doing that is explored in Chapter 8.

Chapter 6

Implementation

The parser and knowledge extractor described in this chapter implement ideas introduced above. The description includes illustrative examples of input and output, along with high-level and low-level sketches of the algorithms employed. The complete computer program reads LDOCE entries and outputs a database of world knowledge facts derived therefrom.

A. Introduction

The theoretical foundations, abstract data structures, and technical ideas described in earlier chapters manifest their validity and usefulness only in some actual implementation. This is especially true if the results can be deemed satisfactory, since poor results may well be blamed on shortcomings of the implementation itself. This chapter describes the computational algorithms used in the Salient Information Viaduct (SIV) program to implement the concepts covered earlier.

SIV is written in VPI Prolog [Deighan and Roach] and is designed to run on a UNIX system. Readers wanting to experiment with the SIV software should contact the Computer Science Department at Virginia Polytechnic Institute and

State University to obtain access rights and the SIV user's guide.

The whole process of extracting database facts of the sort described in the last chapter boils down to identifying some structural parts of the dictionary entries given as input and then separating, labeling, and organizing those that may be of interest while suppressing all others. The task of identifying the parts, known as parsing, lies beyond the central thrust of the SIV project, but the unavailability of a suitable parser and the desire for a complete program dictated development of a simple one for SIV. Nevertheless, the SIV parser is configured as a separate module, so a better parser could be substituted if one should become available.

SIV completes the extraction process automatically, but since some human input may be desired, the program includes other modules that support interaction with the user to manipulate and enhance the results. These modules are of only peripheral interest, however, so this chapter will concentrate on parsing and knowledge extraction only.

Figure 6-1 charts the whole process described in more detail below. The left half of the figure covers the parsing or preparatory phase, which is the subject of the next section.

The right half of the figure covers the knowledge extraction phase, which is outlined in the final section.

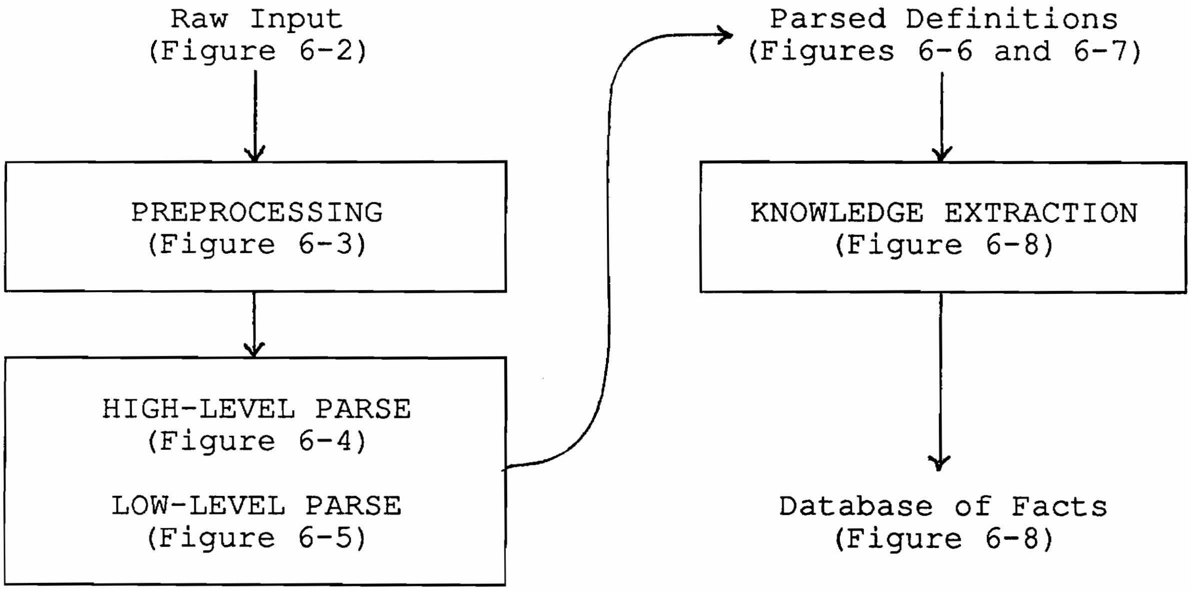


Figure 6-1. Overview of the Whole Process

B. Parsing Dictionary Entries

The various possible and existing schemes for automatic parsing suggest a continuum based on the amount of constituent structure represented in their output. At one extreme, what we may call the deep parser attempts to reveal the entire structure in detail. At the other extreme, what we may call the surface parser settles for some rational division into parts with minimal resolution of their underlying relationships. In general, of course, parsers near the deep extreme will yield the extra information at the expense of speed and resources. (See Mauldin [1991: 348] for a description of a "skimming" parser and its impressive performance in an information retrieval system.)

A major goal of the SIV project was to produce an efficient program with a favorable cost/benefit ratio, the benefit being useful knowledge base facts, the cost being the computer resources invested to obtain them. With this goal in mind, the demands on parser output were limited to what the knowledge extractor could readily use. This seems to be an important consideration even in view of rapid technological advances that are greatly reducing the cost of memory and CPU cycles. The goal of increasing the efficiency and total yield of knowledge extraction programs does compete with the goal of improving their accuracy when priorities are set on the allocation of additional resources.

A brief overview or summary of the entire parsing process should help give perspective to the detailed descriptions that follow. The form of the entry for 'faience' (or some relevant part of it) is shown below in Figures 6-2 through 6-6 at various stages in the process. One word ('very') is added to the actual LDOCE definition to illustrate the way adverbs are moved.

```
((faience)
   (1 F0007200 !< fai *80 ence)
   (3 faI!"A : ns !, -!"Qns = feI!"Ans)
   (5 n !<)
   (6 U !<)
   (7 0 !< !< CE-- !< ----U)
   (8 a special type of clay !, made into cups !,
      dishes !, etc !. !, ornamented with bright
      colours !, and baked very hard))
```

Figure 6-2. Raw Input

```
((faience)
(n) (U)
(1) a special type of clay / made into cups / dishes /
etc @ / ornamented with bright colours / and baked very
hard)
```

Figure 6-3. Preprocessed Entry

Find main parts of whole entry.

```
(n) (U)

((1) a special type of clay / made into cups / dishes /
etc @ / ornamented with bright colours / and baked very
hard)
```

Figure 6-4. High-Level Parse

Find parts of individual definition.

```
((1) (1) a special type of clay / made into cups /
dishes / etc @ / ornamented with bright colours / and
baked very hard)
```

Fill gaps by adding prepositions or relative pronouns.
Move adverbs, modify definition format for parsing.

```
(special type of clay / made into cups / dishes / etc /
ornamented with bright colours / and baked hard <-
very)
```

```
Extract four items: (1) genus term           (clay)
                    (2) classifier            type
                    (3) simple modifier list (special)
                    (4) complex modifier list
                                             ((made into (etc cups
dishes))
                                              (ornamented with
bright colours)
                                              (and) (baked very
hard))
```

Match exceptional patterns.

```
(made into cups / dishes / etc / ornamented with bright
colours / and baked hard <- very)
```

Group conjuncts.

```
((etc cups dishes) ornamented with bright colours / and
baked hard <- very)
```

Match regular patterns.

```
((colours) / and baked hard <- very)
```

Adjust number of genus term and classifier.
Restore adverb positions.

```
(baked very hard)
```

Figure 6-5. Low-Level Parse

```
(faience
(1 1 1 (U) () () (n) (clay) type
      (special) ((made into (etc cups dishes))
      (ornamented with bright colours) (and)
      (baked very hard))))
```

Figure 6-6. Finished Output from Parser

PREPROCESSING

The SIV parser takes machine-readable LDOCE dictionary entries as its input, but these must conform to a special format, so a preprocessor program is needed to convert the original files as necessary. One entry as it would appear before and after this conversion step is shown in Figures 6-2 and 6-3, respectively. The special format has each entry as a separate Prolog list whose first element is itself a list that contains the definiendum ('faience' in the example above), which is the word or phrase to be defined. The body of the converted definition reads very much like the original, except that words or phrases printed in any special font appear as sublists. (No specific font identification is retained for use by the parser.) Also, some punctuation marks and certain other symbols are globally replaced to facilitate their handling in Prolog.

The parser takes a series of the converted entries and outputs a series of corresponding Prolog lists to be used as input for the knowledge extractor. As currently implemented, SIV uses only those entries where at least one sense is classified as a noun. In about two percent of the cases, a noun also gets another part of speech label, such as adjective, and these nouns are used as well.

HIGH-LEVEL PARSE

As each entry is processed, the part of speech label and other coded information positioned ahead of the actual definitions is scanned to determine whether this entry will be included in the output. If the entry qualifies, most of this coded information is identified, passed to the function that parses the definitions, and distinguished as common to all senses. The same parsing function also gets information obtained from the initial list that contains the definiendum. If it is a phrase, the spaces are replaced by underscores, and if any alternate form was included, it is identified as a synonym or lexical variant.

The high-level parsing function gets all the initial coded information (dialect, usage labels, etc.) plus all of the definitions that follow, that is, the final part of the whole entry beginning with the first sense number or, if there is no sense number, the first word used to define the only sense. It divides that final part into separate definitions, one for each sense or subsense, passing them one by one to the low-level parsing function along with the initial (common) coded information. Figure 6-4 shows the effect of high-level parsing on our faience example.

LOW-LEVEL PARSE

The low-level parsing function performs a number of steps as outlined and illustrated in Figure 6-5. It first scans the start of the definition it receives in search of any additional coded information. If any is found, it is merged, for this sense only, with the common information passed in from the upper level. The main parsing algorithm proceeds as described below, and then all parts identified are output in the format required by the knowledge extractor.

Fill gaps by adding prepositions or relative pronouns. The first part of the main algorithm attempts to insert prepositions and relative pronouns as necessary to eliminate gapping. For example, the phrase 'without newness or freshness' in a definition for 'bromide' [LDOCE] gets expanded to 'without newness or without freshness', and 'that creep or climb in trees' in a definition for 'creeper' [LDOCE] gets expanded to 'that creep or that climb in trees'.

When the parser operates in automatic mode, the added word is often inappropriate: 'that grows on sandy or waste land' in a definition for 'broom' [LDOCE] automatically expands to 'that grows on sandy or that waste land', for instance. Such unwanted expansions can be avoided, if the consequent slowdown is acceptable, by running in interactive mode, which lets the user decide where not to expand. Even in automatic mode, the expansion feature seems to do more good than harm, the affected phrases often being of no value to the SIV knowledge extractor with or without the extra word, but it is perhaps a close call.

Move adverbs, modify definition format for parsing. The definition is now scanned and changed as necessary to prepare for final parsing. These changes include (1) the marking of initial gerunds (e.g. "ablution ... the washing of ..." [LDOCE]); (2) the simplification of complex phrases like 'a (small) number of' (changed to 'few' or 'numerous'); (3) the deletion of determiners ('a', 'an', 'the') and periods; (4) removing parentheses around 'someone' or 'something' when initial; and (5) moving adverbs to the right over one word, or over up to three words if prepositions are found there, with a special symbol (<-, <--, or <---) inserted before the moved adverb to indicate how far to move it back to the left when parsing is completed. If adverbs were not moved over, a large number of patterns specified in the parser would require a duplicate pattern, identical except for the added adverb. As it is, a single basic pattern can cover cases with or without an adverb.

Extract four items. The modified definition is scanned again to extract up to four items. The one essential item is the head word or genus term. This identifies what sort of thing the defined noun is. For example, if the definiendum is 'tiger', we may expect to see 'cat' or 'animal' as the genus term used in the definition.

The genus term is not necessarily the syntactic head of the noun phrase that constitutes the definition. For example, if the definition for 'tiger' begins, 'a type of wild cat', then 'type' is the syntactic head. But 'cat' is still the genus term, and for parsing purposes, 'type' is considered a classifier, the second of the four items being extracted, though it comes first in the definition. Here it seems clear that a tiger is a kind of cat, not a kind of type, but there are other cases where the distinction is not so clear-cut. Consider 'sideboard', defined as "a piece of dining room furniture" [LDOCE]. Is it a kind of piece or a kind of furniture? Note that both 'sideboard' and 'piece' are count nouns, while 'furniture' is a mass noun.

The genus term stands between the third and fourth items to be extracted. Both of these items are lists of modifiers that supply specific information about the genus term and, indirectly, the definiendum. One list contains the relatively simple modifiers coming before the genus term, and the other list has the more complex modifiers that follow it in the definition. The simple list includes mostly ordinary adjectives, while mostly prepositional phrases, verbal phrases, and relative clauses populate the complex list, as explained in the last chapter. Of course, either list or both may turn out to be empty for some definitions.

The basic parsing strategy concentrates on identifying those four items in the final scan of the entire definition. Once the front of the definition has been inspected for the presence of some classifier, the remaining task amounts to a search for the beginning of complex modifiers, if any. Everything from the first one on goes into the complex modifier list. The last item before that point, which might be a list of conjoined items, is the genus term, and everything else, not including the genus term and any classifier, goes into the simple modifier list. With this basic strategy in view, the SIV parsing algorithm can now be described in a little more detail.

The object of the search for complex modifiers will be among a small list of items: (1) a preposition; (2) a participle; (3) a relative pronoun; (4) a subordinate conjunction; (5) certain heavy modifiers, such as 'present in'; and (6) a comma that cannot be recognized as a list separator. If none of these is found, the complex modifier list is simply left empty, and the last item scanned is used as the genus term. As soon as one of the items is found, the search continues for another, which if found will mark the beginning of a second complex modifier to be added to the list, and so on to the end.

Match exceptional patterns. Although the basic approach is quite simple, there are unfortunately many exceptions to the general rules, and these must be recognized and properly handled. For example, participles can occur among the simple modifiers before the genus term, and a word that has the form of a present participle may actually function as a gerund, which could be the genus term itself. One might complain that the ad hoc treatment of these exceptions in the SIV parser is too unsophisticated, but at least it avoids a rigorous identification of the function of each word, normally required by more conventional parsers, and its performance does seem to be commensurate with the demands of the SIV knowledge extractor.

When the final scan of the definition begins, a few of the leftmost items in the definition are checked to see if they match any of the exceptional patterns. In principle, even the rightmost item in the definition could be matched if some pattern is sufficiently long. The purpose of the exceptional patterns is to identify classifiers and to keep items together that would be erroneously split in the search for complex modifiers. For example, if the words 'together with' occur in a definition, we want to make an exception to the general rule that a preposition begins a new complex modifier, so 'together with' needs to be one of the exceptional patterns. If this pattern is found in a definition, then 'together' will not be mistaken for the genus term, and 'together with' will be stored as part of a single complex modifier.

Group conjuncts. If none of the exceptional patterns match, there is first a check to see if the first few items constitute a conjunct. For example, 'red / white / and blue' (where / represents a comma) would be changed to a list, '(and red white blue)' with the conjunction listed first.

Match regular patterns. Next there is a check to see if any of the regular patterns match the leftmost part of the definition. If so, the split is made accordingly, some of that leftmost part is stored and removed from consideration, and the parsing cycle continues with the remainder of the definition, and so on to the end. For example, if the leftmost part of the definition were 'animal with', then the pattern corresponding to the general rule that a preposition begins a new complex modifier would match, and those words would be split so that the first would be stored as either the genus term, if none had yet been found, or as the last word in the complex modifier now being stored.

If none of the regular patterns match either, then the leftmost item is simply stored and removed from consideration as parsing continues with the remainder of the definition. If the genus term has not yet been found, the item gets stored in the list of simple modifiers, else as part of some complex modifier.

Adjust number of genus term and classifier. When parsing is completed, if a classifier was identified, a plural classifier or genus term may be changed to singular. For example, if the definition begins, "any of a set of objects that ...," then 'any_of_set' would be the classifier, and 'objects' would be identified as the genus term, but it would be changed to the singular form 'object' at this point. Similarly, "any of various types of birds that ..." would have both 'types' as the classifier and 'birds' as the genus term converted to the corresponding singular forms.

Restore adverb positions. Next, if any adverbs had been shifted before parsing, they would now be restored to their original position in either list of modifiers. Then if the definition included any one-word synonym, it would now be inserted ahead of the longer definition as a separate definition with the same homonym, sense, and subsense numbers. Finally, all of the stored information is written to the output file in the required format, shown in Figure 6-6, and processing continues with the next definition for this entry, if any, or with the next entry, if any remain.

C. Knowledge Extraction

In the broadest sense of the term, knowledge extraction should include even the parsing phase of the process, with raw definitions or even their authors being the source from which knowledge is extracted through a series of procedures. In this section, however, the term will be taken in a much narrower sense, with the output of some parser, possibly massaged to produce the required format, being the immediate source. This simplification is convenient, because the earlier phases of the entire process, now being ignored, are relatively peripheral to the SIV project proper.

The input format that the SIV knowledge extractor requires is a series of Prolog lists, each beginning with one atom followed by one or more sublists of eleven specific elements. The extractor module has a feature that allows an optional check for compliance with that format before actual extraction begins, including a type check of each of the eleven embedded elements.

Each fact extracted from a definition will belong to a name or sense frame for the definiendum specified as the initial atom of the whole list or to a detail frame for some nominal concept mentioned in one of the embedded lists of modifiers. The slot and filler for the fact will usually be derived from a modifier in either the simple or complex list. For example, if the modifier is 'small', then the slot will be **size**, since size is what this modifier is used to describe, and the filler will be the word 'small', the modifier itself. The correlation is not always that straightforward, even for such a simple modifier, but this example provides a good general overview of the extraction strategy.

A brief summary of the entire extraction process is also provided below in Figures 6-7 and 6-8 as an aid to understanding the detailed explanations that follow it. Step descriptions in boldface type serve as cross-references to the summary. In Figure 6-8, examples in ordinary type starting at the left margin represent the part of the input that is used to identify the facts which are given on indented lines.

Note that the facts are not necessarily asserted to the database at that precise point in the process; they might be only appended to a list of facts to be asserted later. Facts are listed in slot-frame-filler form for the summary and get asserted to the database in a similar form. The description of frames in Chapter 4 mentions a **head** slot whose filler is the name of its frame, but no facts with a **head** slot ever need to be asserted. Instead, each slot/filler couple for a fact combines with the filler of the implicit **head** slot. This way, any fact can be directly associated with its frame, even if separated in processing from other facts that belong to the same frame.

```
(tiger
(1 1 1 () () () (n) (cat) type
       (very large fierce wild)
       ((that has yellowish fur)
       (with black bands) (across) (and) (that lives) (in Asia)))
(1 2 1 () () () (n) (person) ""
       () ((like such animal) (in (etc fierceness courage)))))
```

Figure 6-7. Entry as Input to the Extractor

Check genus term.
Update definiendum level.
Update sense identifier.

```
1 1 1
1 2 1
     sense       tiger      (1 (1 2))
```

Update labels.

```
() () (n)
     function    tiger      (n)
```

Construct sense frame identifier.

```
(tiger
(1 1 1 ...
     tiger__1
(tiger
(1 2 1 ...
     tiger__1_2
```

Scan grammar codes.
Check for synonym.
Scan genus term and classifier.

```
cat
     typeOf      tiger__1   (cat)
     level       cat        (above_basic 0 1)
person
     typeOf      tiger__1_2 (person)
     level       person     (above_basic 0 1)
```

Check list of simple modifiers.
Scan list of simple modifiers.

```
(very large fierce wild)
     size        tiger__1   (highly large)
     hazard      tiger__1   (fierce wild)
```

Check list of complex modifiers.
Scan list of complex modifiers.

```
(that has yellowish fur)
(with black bands)
     part        tiger__1   (fur bands)
     color      (tiger__1 part fur)    (yellowish)
     color      (tiger__1 part bands)  (black)
(like such animal)
     analogue    tiger__1_2 (animal)
```

Assert facts to database.

Figure 6-8. Steps for Knowledge Extraction

Check genus term. The SIV extractor ignores any definition featuring a genus term that is too abstract, specifically, an act, condition, fact, quality, state, or way. In most cases, these genus terms mark the definition of a noun derived from a related verb or adjective. That related word normally figures prominently in the definition of the noun, and very little additional information of interest is included. For instance, 'happiness' is defined as "the state of being happy" [LDOCE]. It seems best to postpone inclusion of such abstract nouns until the other parts of speech from which they are derived can also be properly accommodated. Definitions with a gerund as the genus term are accepted by the extractor even though they are technically abstract.

If a definition consists of a lexical variant only (the classifier being 'lex'), then one fact is immediately stored in the knowledge base. In this case, the slot is **variant** and the filler is the lexical variant itself as stored in the genus term. If the genus term from the input file is 'someone' or 'something', it is changed to 'person' or 'thing', respectively, before beginning the main extraction algorithm.

Update definiendum level. The first step in that algorithm is to update the level for the second part of a phrasal definiendum. That is, if the nominal concept being defined is expressed by more than one word, then all but the first word is assumed (perhaps incorrectly) to be a noun, and the filler of the **level** slot for that noun is updated. This filler contains a marker and two numbers, as explained in Chapter 4. In this case, the update is accomplished by incrementing the first number by one.

Update sense identifier. The next step is to update the sense identifier that will fill the **sense** slot of the frame for the definiendum, also described in some detail in Chapter 4. This update is based on the homonym, sense, and subsense numbers present in the input, with numbers in the identifier increasing as new definitions are encountered.

Update labels. The input also includes the dialect, usage, and function labels for each definition. The fillers of the corresponding name frame slots need to indicate which senses, if not all of them for the whole entry, are marked by those labels, so a list of sense identifiers is maintained for each type of label and updated at this point in the process.

Construct sense frame identifier. Next a Prolog atom is constructed to represent the name of each sense frame. This is the definiendum, with spaces replaced by underscores, and the sense identifier joined by a double underscore. This will be used along with the slots and fillers identified in the main part of the extraction program to form the triples that represent facts in the knowledge base.

Scan grammar codes. The grammar codes contain information that determines whether either of two closed slots (**substance** and **nonPrototype**) will be included, so they are scanned next. The **substance** slot is associated with the grammar code **U** and the **nonPrototype** slot is triggered by the code **the**.

Check for synonym. If there is no classifier and both lists of modifiers are empty, one fact is added to the sense frame at this point, the **synonym** slot being filled by the lone genus term. In this case, the next few steps are skipped, including the scan of modifiers but not the final addition of name frame slots.

Scan genus term and classifier. If the input definition proper contained more than just a genus term, that term is considered next, along with any classifier, and facts are added to the knowledge base as appropriate. The slots that may be identified at this stage include **typeOf**, **group**, **members**, **quantity**, and **groupOf**. If the genus term consists of a list of alternatives, the first of these is extracted for use in the main scan of the modifier lists. If not, the simple genus term is used for the same purpose. Before the modifier lists are checked and scanned, the **level** slot for the genus term is also updated as it was for the definiendum at the beginning the main extraction algorithm.

Check list of simple modifiers. The list of simple modifiers is checked first to see if it contains a phrase or any sequence words that must be processed as a whole. For example, the list may begin with the sublist '(or one more)', which would have originated from 'one or more ...' in the raw input. In this case, a **typeOf** fact is changed so that it has both singular and plural forms of the genus term as fillers.

A similar case arises in the LDOCE definition for 'credentials': "a letter or other written proof of ..." which yields 'proof' as the genus term and '((or letter other) written)' as the simple modifier list. The first item in this list contains an alternative genus term, so it is identified at this time and used to assert a new **typeOf** fact.

Scan list of simple modifiers. After the list of simple modifiers has been checked, items remaining in the list are considered in turn, facts are accumulated in a results list and added to the knowledge base at the end of the scan. If the current item is an embedded list, it is processed as though it were the main list of simple modifiers; after the embedded list is scanned, the scan continues with the rest of the actual main list. If the current term is a degree marker (highly, quite, slightly, etc.) or an applicability marker (esp, often, usu, sometimes), it is stored in a special buffer so that it will not be treated as an ordinary modifier. If the current item is a hyphenated form, there is a search for certain key words in the form. For example, if the second part of the form is 'eating', then a fact that has a **typicalFood** slot is stored in the accumulator. If a word that ends with the '-less' or '-like' suffix is the current form, then we store a fact that has an **absenceOf** or **analogue** slot, respectively.

If none of those special conditions apply, the main function that scans simple modifiers gets the next two words, or just the next word, if there is only one left in the list. This function contains a series of tests to determine which slot the current modifier will fill, based on its category. Many of these tests are as simple as the one mentioned above. For instance the modifier 'warm' is in the temperature category, so it fills the **temperature** slot. A few tests are more complex and take into account the genus term or the next word in the list of modifiers. For example, if the current word is in the habitat category (tree, lake, desert, etc.), and the genus term is in the creature category, then we have found something for the **habitat** slot, else for the **location** slot. If the current word does not pass any of the tests provided in the function, then it is used to fill the generic **attribute** slot.

Check list of complex modifiers. After the whole list of simple modifiers has been scanned, the list of complex modifiers is checked to see if any adjacent elements need to be combined or otherwise processed as a whole. Each element of the complex list is a sublist of words that compose some complex modifier, usually a phrase or subordinate clause. In some cases, one of these modifiers would be useless or incomplete by itself but serves a definite purpose when joined with its neighbor. For example, '((of great numbers) (of people))', the two complex modifiers in a definition for 'hecatomb' [LDOCE], combine at this point to form a single more manageable modifier '((of multitudinous people))' in preparation for the scan described next.

Scan list of complex modifiers. After the whole list of complex modifiers has been checked, the revised list is scanned element by element. The algorithm is very similar to the one for simple modifiers, but a bit more complex. One of the extra features of this scan is a count of number of modifiers already scanned. Those closest to the genus term are most likely to refer to that term and, indirectly, to the definiendum, so the tests in the main scan function can be sensitive to the relative position of the modifier currently under consideration. The algorithm allows some complex modifiers to be skipped over without incrementing the count. Once again, as modifiers are considered in turn, facts accumulate in a results list and get added to the knowledge base at the end of the scan.

If the sublist consists of a single word, it receives special treatment. If the word is just a conjunction isolated through the phrase expansion feature of the parser, it is simply ignored. If it is any other single word, it is used to fill the **unclAppos** (unclassified appositive) slot.

If the modifier contains at least two words, then we extract both the word at the end of the sublist and the middle part of it (excluding the front and the end). Consider the first complex modifier in the tiger example, '(that has yellowish fur)', where the middle part is 'has yellowish', and the word at the end is 'fur'. If the end is itself a list beginning with a conjunction, that conjunction and the first conjunct are also extracted and passed to the main function for the scan of complex modifiers. This pattern occurs in the last modifier in the tiger example, '(in (etc fierceness courage))', where the conjunction is 'etc' and the first conjunct is 'fierceness'. If the first word in the modifier sublist is an applicability marker, it is passed to that function also but is removed from the sublist before the scan begins.

There is a fairly long list of tests in the main scan function for complex modifiers. These tests can be fairly simple. For example, there might be a check for the words 'kept for' at the beginning of the phrase, which would trigger the storage of a fact that fills the **typicalUse** slot with the word extracted from the end of the sublist. There is often a check for modifiers of that word at the end to see if a detail frame can be constructed. This check reuses the function that scans simple modifiers to find the appropriate slot and filler for the head of the detail frame.

If the sublist begins with certain relative pronouns (that, which, or who), then there is also a fairly long series of tests specifically for relative clauses. Before these tests begin, the verb following the relative is extracted so that its tests can be sensitive to the type of verb. For example, if the verb is 'appear(s)', a fact is stored that fills the **appearance** slot with the end of the relative clause. There is also a smaller list of special tests if the sublist begins with 'used'. This avoids having to rematch that same word several times for the same complex modifier when trying the tests in the main list.

If none of the tests succeed, and if the phrase in the sublist does not include more than seven words, it is used to fill the generic **unclDescr** (unclassified description) slot. These facts are often worthless, but they are kept just in case some way might be found to sort out any that are of interest. In any case, it would be simple to isolate these and delete them all as a group.

This completes the description of the extraction algorithm for a given eleven-part definition element in the list of definition elements for an entry, which is a Prolog list that begins with a definiendum.

Assert facts to database. When all of the definitions have been processed, the finished lists for the dialect, usage, function, and sense labels are used to construct facts with correspondingly named slots. About one third of all noun entries in the LDOCE corpus contain definitions for only one sense and have a homonym number of 1. To save room in the database, this is considered the default, and no fact with a **sense** slot is actually asserted for such an entry. Hence the presence of a **sense** slot for a frame indicates multiple homonyms or senses.

Chapter 7

Results

A. Introduction

Computer programs can be proved correct or judged to be bug-free, but only the most trivial can ever attain perfection in the sense that no further improvement is possible, every nice feature having already been implemented. Hence the goal for the SIV program has not been perfection but a measure of adequacy for testing certain theories and concepts. There can be no definitive answers to questions about how practical and effective our general approach may be, since the variety, quality, and quantity of facts extracted might always be pushed at least a little higher. The temptation is either to stop short of a reasonable effort or to go too far beyond the point of diminishing returns. A point has been reached in this project where some suggestive answers can finally be offered based on the output of a program that seems to strike a fair balance between laziness and perfectionism.

This chapter summarizes the results of running the SIV program on a sizable cross-section of the LDOCE in machine-readable form. More detailed facts and figures are presented as tables in Appendix B, which parallels this chapter. Appendix C reports on a later run that used the entire LDOCE as input.

B. The Work and Test Samples

The whole LDOCE file is nearly sixteen megabytes large, so ten smaller segments were extracted from it for the purposes of program development and evaluation. Since the whole file has about 510,000 lines, it can be conveniently divided into ten parts with 51,000 lines in each part. To get the samples, the first 3,000 lines were extracted from each of the ten parts through the UNIX **sed** command, and then partial entries were trimmed from the beginning and end of each sample. In Table 8-A, the last line ("all") refers to the whole LDOCE file.

The 915,167 bytes in the ten finished samples amount to 5.73 percent of the big file. Several tables in Appendix B have a column of "projected" figures obtained by assuming that results from these samples are exactly proportional to the results that would be obtained from the whole file, with average item-per-byte rates remaining constant throughout.

The overview table, for example, projects that it would take 17,751 seconds (or about five hours) to process the whole LDOCE file. The extrapolated figures are certainly speculative and perhaps superfluous, given the information in Appendix C, which is based on an actual run with the whole dictionary.

Table 7-A. Sample Size and Location

Sample	Line Numbers	Lines, Words, Bytes	First Entry	Last Entry
	(First, Last)	(After Trim)		
0	000001, 3000	2989, 17765, 92268	AA	acquisitive
1	051001, 54000	2991, 17191, 87657	brolly	burdensome
2	102001, 105000	2992, 18035, 94079	cream cheese	crucifix
3	153001, 156000	2933, 18477, 95312	extinction	falls
4	204001, 207000	2957, 18023, 91827	have	helmeted
5	255001, 258000	2991, 17904, 90194	lockout	lowbrow
6	306001, 309000	2996, 17243, 89770	outright	pact
7	357001, 360000	2996, 16717, 86640	Q	radiography
8	408001, 411000	2931, 18723, 95577	shortfall	sightly
9	459001, 462000	2984, 17906, 91843	three-quarter	tinkle
all	000001, 509452	3087914, 15957744	AA	Zulu

All SIV modules were already running pretty well even before the ten samples became available. Eventually, six samples (0, 2, 4, 6, 8, and 9) were arbitrarily selected to be "work samples" for the final effort to fine tune the preprocessor and parser. At first, these two modules ran with just one of the work samples as input, problems were noted and corrected, the modules ran again with the same input, and so on in a cycle until no serious problems were noted. Then the process was repeated with the other work samples until the results from all of them were fairly satisfactory (certainly not perfect).

Next attention turned to improving the extractor. For this final phase of development, sample 9 was excluded from further service as a work sample and became one of the five "test samples" (namely 1, 3, 5, 7, and 9, leaving 0, 2, 4, 6, and 8 as the final set of work samples). Work on the extractor proceeded in the same cyclic pattern of trial, error, and correction, but it was often necessary to go back and debug preprocessor or parser code to improve input to the extractor for better performance. The test samples were ignored until the final test, since they would serve to demonstrate how well the extractor handles fresh input.

When preliminary results finally looked good enough to run that test, all ten samples were used as input for a single series of runs. A few minor problems appeared while running test samples 1, 3, and 7. There were only about one or two problems per sample, and in every case, only preprocessor or parser code needed more work. No troublesome area was deleted from any input file. After minimal fixes for just those few problems, the whole series ran to completion without further tinkering. Both Appendix B and the results analysis in the remainder of this chapter are based on that final test run with the ten samples.

C. Parser Results

The preprocessor took as input the raw samples identified in Table 7-A and output files with all the same entries in a form acceptable to the parser. The most tedious aspect of this effort was properly handling all the special code sequences for typesetting and representing special characters. A complete list of those codes and their function would have been very helpful. Since the program was implemented without such a list in hand, there are undoubtedly several less common codes or code combinations that it still does not interpret correctly. Appendix D reports the interpretation given to all codes recognized by the program.

The preprocessor counts the main entries it sees and also all the individual senses, not including subsenses. It found an average of 243.8 entries and 404.3 senses per sample. These became the input for the parser, which ignored entries with no noun senses. Considering all ten samples, 55.1 percent of the entries were nouns, including entries for multifunction words such as 'ablative', which can serve as a noun or an adjective. On the same basis, 60.3 percent of all senses passed to the parser were noun senses, an average of 243.9 senses per sample and 1.81 senses per noun. The count of just noun senses, however, does include subsenses, but they are so rare that several samples do not have any of them.

The work of the parser is complex and time-consuming. In fact, about half of the time required to turn raw dictionary entries into database files was devoted to the parsing phase of the process. Consequently, to evaluate parser performance, it was not practical to critique every feature of the parsed output. Instead, only the rate of success in finding the correct genus term was determined.

This measure has some important advantages. The basic unit of parser output is a noun sense definition, and each one is supposed to have a genus term, so the evaluation offers excellent coverage. The parser's strategy for finding the genus term is essentially the same as its strategy for splitting complex modifiers into chunks, and about three quarters of all noun sense definitions do contain some complex modifier. This implies that a measure of parser success in picking out the genus term should provide a fair indication of overall performance. Finally, it would be difficult to find a more objective measure for our purposes --or one much easier to obtain.

Two classes of noun sense definitions were excluded from consideration, and a third was considered separately. Of the total of 2,439 senses, 96 were merely a lexical variant of the definiendum, so these were excluded as too trivial. The 415 cases with both simple and complex modifier lists left empty were not excluded, since the parser often had to do at least some work to remove articles and group coordinate items (as in "an organization or activity," a definition for 'show'). Perhaps an article or conjunction could have been mistaken for a genus term. Twelve other sense definitions were excluded since they were erroneously marked as nouns in the source file, and the 53 multifunction sense definitions were considered separately. Some of these read like a noun definition, and some like an adjective or some other kind of definition. Since 6 definitions were both variant and multifunction, 155 definitions in all (6.4 percent) were excluded from the main evaluation.

Of the remaining 1,337 noun sense definitions in the six work samples and the remaining 947 in the four test samples, the parser picked the wrong genus term in 3 and 38 cases, respectively, which translates into a success rate of 99.8 percent for the work samples and 96.0 percent for the test samples. The differences in these figures for work and test samples, as well as scrutiny of the actual failures in the test samples, suggest that further development of the parser would result in somewhat better performance on fresh input. The high success rate even for the test samples, however, shows that it already works very accurately, according to the evaluation method just described. Almost half of all failures in the test samples appeared in sample 7, and the success rate hit 99 percent in test sample 3. This excellent result is fortunate, given the importance of having a correct parse for proper operation of the extractor.

All three failures in the work samples can be traced to incompleteness of the parser's list of adjectives. In the definition "a severe searching test" (for 'crucible' in sample 2), the parser picked 'severe' instead of 'test'; in "an outer covering on a plant or animal" ('shuck'), it was 'outer' instead of 'covering'; and in "a regular going to and fro by air ..." ('shuttle'), it was 'regular' instead of 'going'. If the parser's short list of adjectives had included these less common modifiers, they would have been passed over in the search for the genus term. One might conclude that the adjective list should simply be completed to optimize parser performance, but the tiny increase in accuracy should be weighed against the runtime slowdown that the longer list would cause, and care would need to be taken not to include words in this list that might also function as a noun.

As expected, the success rate was significantly lower for the 47 multifunction non-variant sense definitions set aside for special consideration. For the 31 in the work samples, the parser found 23 good genus terms, a success rate of 74.2 percent, and for the 16 in the test samples, the figures were 9 and 56.3 percent, respectively. It would be easy to exclude the multifunction items from parser input, but it seemed good to see what would happen if they were retained, and there was little need to agonize over the decision, since they account for only 2.2 percent of all noun senses in our samples.

D. Extractor Results

The process of extracting facts from the files output by the parser and storing them in the SIV knowledge base consumed about a third of the total time required for the whole effort. Once again, the great bulk and complexity of material resulting from all that work precluded an exhaustive evaluation. A Prolog program was written and employed to count facts, with individual counts for each possible slot, so at least quantitative data are readily available. In addition, a large sample of facts was manually inspected for accuracy, as further explained below. This qualitative evaluation was much more tedious and subjective but perhaps more revealing and interesting. The next section of this chapter covers the results of a third kind of evaluation relevant to our interest in prototypes.

The automatic count actually included three separate totals for each slot. The first of these counted just unique slot-frame pairs, with no attention paid to the number of fillers. The second counted just unique slot-detail-frame pairs, again without regard to fillers. The third one was for unique slot-frame-filler triples, but this one did not include any facts with a name frame slot. The results of these three kinds of counts will now be considered in turn.

A total of 5,341 slot-frame pairs were extracted from the five work samples, and 5,927 were extracted from the five test samples, yielding a sum of 11,268 for all ten samples. That projects a total of 196,480 slot-frame pairs for the whole LDOCE. The extractor used non-abstract sense definitions only, and there were 1,106 of these in the work samples and 1,221 in the test samples, so an average of 4.83 pairs were extracted per definition for the work samples and 4.85 for the others. It seems significant that these figures are so close. If the extractor had been specially rigged to handle just some well-known input, without regard for more general requirements, the average should have been significantly smaller for the test samples, not larger.

Another criterion, based on the number of distinct slots used, might suggest that the extractor did a little worse with fresh input. Of the 81 slots listed in the slot catalog, 73 occur in at least one fact extracted from some work or test sample, but 72 distinct slots were detected in the work samples as opposed to only 67 in the test samples. On the other hand, only 14 slot-frame pairs involve a slot not detected in both sets of samples, which is about 1/800 of the total, so the significance of the observed discrepancy should be correspondingly tiny.

The count of slot-detail-frame pairs was 106 and 71 for the work and test samples, respectively, the average being 0.096 and 0.058 per source definition. Only 29 distinct slots were involved in this case, one of which was a name frame slot. Two thirds (19) of these did not occur in both sets of samples; 15 appeared in work samples only, and 4 appeared in test samples only (not counting occurrences in sense frames).

All of those detail frame results suggest that the extractor performed comparatively badly with fresh input, but once again the observation should be kept in perspective, with due regard to the small percentage of total work that underlies those figures. Only 1.57 percent of all slot-frame pairs involved a detail frame.

The scarcity of details can be attributed to at least two factors: (1) the program undoubtedly missed some opportunities to extract them, but (2) the input definitions themselves are apparently not very rich in the kind of information required. One example of a missed opportunity is the detail about tiger fur color included in the illustrative example in Chapter 6. (All other facts shown in that example were successfully extracted from test sample 9.)

Even though detail frames were comparatively uncommon, the frequency distribution of the slots involved clearly manifests the classical pattern, a few slots accounting for most of the cases, and many slots occurring only rarely. Specifically, the most common slot (**attribute**) appeared in almost a third of the cases, the five most common ones in 69.54 percent of the cases, the next eleven most common slots in 23.56 percent, and the other twelve in only 6.90 percent. (The one name frame slot was not counted in this analysis, but it was very rare, with only three cases in all.) It may be interesting to note that the most common slots for detail frames were very distinct from the most common slots among the much more abundant sense frames.

The totals for slot-frame-filler triples, or facts, did not include any of the name frames, which were very common but interesting for other reasons. The remaining facts that feature an **attribute**, **unclAppos**, or **unclDescr** slot were designated for separate consideration as "unclassified" facts. Naturally, facts with any of the 64 other sense frame slots are termed "classified" facts, meaning that they refer to some more specific relation. The distinction is somewhat arbitrary, however, since there is considerable variation among the so-called classified facts in their degree of specificity. Facts with **treatment**, **typicalUse**, and **typicalAction** slots, for example, group with the classified facts, even though they are close to the fuzzy borderline.

The extractor found 2,378 classified facts and 1,726 unclassified facts in the work samples, and 2,488 classified, 1,845 unclassified facts in the test samples, making a total of 4,866 classified facts and 3,571 unclassified facts. Putting both kinds of facts together, we get 4,104 from the works samples, 4,333 from the test samples, and 8,437 facts altogether. The average number of classified facts per non-abstract input definition was 2.15 for work samples, 2.04 for test samples, and 2.09 for all ten samples; the same averages were 1.56, 1.51, and 1.53 for unclassified facts. A total of 84,848 classified and 62,267 unclassified facts are projected if the whole LDOCE were used as input, still ignoring facts that involve name frame slots.

Intuitively, if the extractor had done worse with the test samples, a larger percentage of all facts from those samples should have gone into one of the unclassified categories reserved for miscellaneous modifiers and chunks of the parsed input. The actual percentage of unclassified facts was 42.1 for the work samples and 42.6 for the test samples, which implies that extractor performance was perhaps slightly better with the work samples but was practically the same for both samples. The difference is clearly less than the difference in average number of facts extracted per definition (3.71 versus 3.55), which remains unexplained. The difference in those averages, which is also rather small, might be at least partly due to richer information in work sample definitions.

The results of a frequency analysis, summarized in Table 7-B, show the same pattern mentioned above, a few high frequency items representing the bulk of all occurrences, with many low frequency items. There is also a significant skewing apparent in those figures. Notice that facts with low frequency slots were extracted comparatively better in the work samples, while facts with high frequency slots were extracted better in the test samples. It would be tempting to explain that the extractor was custom-designed to extract certain low frequency facts, given the advantage of familiar input, but that would not explain the reversed situation for the high frequency facts.

Table 7-B. Facts Grouped by Frequency of Slot

Slots \ Sample	Work	Test	Both	Percent
3	1317	1416	2733	56.17
7	525	598	1123	23.08
10	246	233	479	9.84
26	241	209	450	9.25
18	49	32	81	1.66
64	2378	2488	4866	100.00

Quite a bit of attention has been given to the question of how similar the results are for work and test samples. It seems to be an important question, because it sheds welcome light on the extent of our dependence on ad hoc solutions and the prospects for further improvement in generality. If the results were much poorer for the test samples, then one might hope that additional development effort might result in closing the gap. On the other hand, if the results were much poorer for the work samples, then one might conclude that the samples were simply too small and insufficiently representative, so that the work samples could include less informative definitions on the whole. As it is, with approximately the same level of performance with both sample sets, it seems safe to surmise that the algorithms and extraction strategies are reasonably general and should generate comparable results with any other sample from the same dictionary.

The results of a qualitative analysis of the results tell a story similar to that told by the quantitative analysis just considered. To probe the accuracy of the extractor's performance, all facts having any of 27 selected slots were inspected for correctness, and each fact was judged to be either right or wrong. Though admittedly subjective, decisions on how to grade a given fact were almost always reached with a high degree of confidence and with little room for doubt.

The choice of the 27 slots for the evaluation was arbitrarily based on the frequency analysis described in Table 7-B. The entire second and third highest groups were included, with 7 and 10 slots, respectively, and the 10 least common slots were added from the fifth group. These three special groups make up 23.1, 9.8, and 0.3 percent of the 4,866 classified facts, a total of 33.4 percent for all three groups.

None of the three slots in the highest frequency group were chosen. The **typeOf** slot, used in 41.0 percent of the facts, was almost always just the genus term picked out by the parser, so including those scores would have been giving the extractor credit for work pretty much finished by the parser. The **substance** and **synonym** slots appeared in 8.1 and 7.0 percent of the facts, but these were facts based on information drawn from specially coded information in the definitions. Extracting them was so straightforward that it seemed pointless to designate them for special scrutiny, especially since so many facts (738) were involved.

The results of checking the three chosen groups are shown in Table 7-C, which gives percentages in the "right" columns only, numbers of facts being given in all the others except the first. It is clear from these results that a fairly high degree of accuracy was attained, though there is certainly room for improvement. It seemed important to include some facts using the least frequent slots, but unfortunately, the number of these drawn from the test samples was so small (only five) that the margin of error in the 60 percent accuracy rating must be rather large. Conversely, the margin of error should be the smallest in the bottom row of figures, where the totals for all three groups are listed.

The results in Table 7-C apparently reveal another skewing effect similar to the one noted above. Ignoring the low, uncertain score in the test sample column, the extractor seems to have performed somewhat better with low frequency items than it did with high frequency items, this time in both samples, though the effect was rather more pronounced in the work samples. The more detailed tables presented in Appendix B show that this trend was not very smooth, and certain facts, mainly those that almost qualify as unclassified facts, received abnormally low scores for accuracy. Four such kinds of facts (**treatment**, **typicalUse**, **typicalAction**, and **location**) are included in the top row of Table 7-C scores, which probably explains the main reason for the skewing effect.

Table 7-C. Accuracy of Facts in Selected Slot Groups

Slots	Work	Wrong	Right	Test	Wrong	Right	Both	Wrong	Right
7	525	49	90.7	598	72	88.0	1123	121	89.2
10	246	8	96.7	233	23	90.1	479	31	93.5
10	19	0	100.0	5	2	60.0	24	2	91.7
27	790	57	92.8	836	97	88.4	1626	154	90.5

To conclude this section, a few more figures from the accuracy analysis will be considered. Combining results from all ten samples, but just the 17 slots in the first two groups that were checked, we note that 10 slots, used in 39.3 percent of the 1,602 facts in these groups, received a score of 93 percent or better. Five of these slots received a perfect score. Of the remaining 7, the lowest score was 80 percent even. To improve the average correctness of facts extracted, one could discard facts with certain low-scoring slots, or to avoid sacrificing breadth, one could concentrate development efforts on the less reliable slots, or perhaps somehow combine the two approaches. In any event, it seems fair to claim that even higher accuracy is attainable for world knowledge extractors like the one developed for SIV.

E. Prototypes in the Output

One big question has not been directly addressed in the quantitative and qualitative analyses described above. How well are prototypes actually represented in the final output of the SIV programs? Unfortunately, no exact answer to the question is possible, since it is so subjective by nature. Fortunately, some objective data bearing on the question can be offered for consideration, along with some personal opinions and general observations.

It has been assumed that almost all noun definitions actually refer to what we are calling prototypes and that most facts extracted by the SIV program are also facts about prototypes. (The rare exceptions are noun entries for a specific, named entity or event, such as the 'Creation' in sample 2 and the 'Hegira' in sample 4.) It should be emphasized that these are assumptions, not conclusions from our research. The purpose of this section is to show the extent to which desired information about prototypes was included in the results.

The rest of this section is organized in parallel with the last section of Chapter 5. All fourteen kinds of desired information about prototypes, as originally outlined in Chapter 3 (Figure 3-3) were examined there for possible availability in the LDOCE. All the kinds expected to be found in the LDOCE machine-readable files were actually extracted in some form and to some extent, except as noted below.

Of the first two kinds of information, only the first, (1) span of time, was actually extracted. This kind was realized as facts with an **era** slot. No slot was provided for facts about (2) realm of existence. There were only 17 facts altogether with an **era** slot. At least one of these (concerning the 'Oxford movement') specified the 19th century.

The **typeOf**, **group**, and **members** slots were used for facts based on the genus term, mentioned as a source of information of the third kind, (3) hierarchical organization or membership in a group or a prototypical category. There were 1,995 facts with **typeOf** slots and 19 each with **group** and **members** slots. The large number of **typeOf** slots is due to the fact that each definition had a genus term; it is less than the number of qualifying definitions because some genus terms were diverted to use in facts with **essence** or **variant** slots.

Facts with a **level** slot are an important source of information about hierarchical organization, and because of the link between hierarchy level and category boundary fuzziness, mentioned in Chapter 3, they may also provide some indication of how much fuzziness to expect for a given concept. It is true that this slot applies to nouns, not directly to nominal concepts, but it seems safe to assume that the corresponding concepts for any noun will be at about the same level in the hierarchy.

A total of 2,715 slot-frame pairs with a **level** slot were found in the ten samples, and 47,341 such pairs are projected for the whole LDOCE. Since it can be based on the genus term or some part of the definiendum, a fact extracted with a **level** slot may belong to a frame for a noun defined either inside or outside its source sample. Only 77.5 percent of the pairs are actually unique because of overlapping "outside" nouns.

Table 7-D describes the fillers that were extracted for **level** slots. Entries in the C1 and C2 columns represent the first or second count that is stored along with one of the three markers. The lowest value of variables used in these columns is indicated at the foot of the table. Recall that the first count shows how many times the word (or phrase) was included as part of a definiendum, and the second count shows how many times it occurred as the genus term in a definition.

Table 7-D does not show the number of frames for each count combination that occurred, but each combination could be considered a separate level in the hierarchy. For instance, **aboveBasic** 0 1 and **aboveBasic** 0 2 can represent two distinct levels, the second one slightly higher than the first. The total number of distinct count combinations is shown in the "levels" columns of the table. The number of frames for **aboveBasic** 0 3 and **aboveBasic** 0 4 are not specified separately but are included in the row for **aboveBasic** 0 m, which covers 19 count combinations altogether.

The total should be considered an upper limit only, since some levels are undoubtedly too close together to be considered as really distinct. On the other hand, these upper limits apply to the ten samples only. They should be somewhat higher for the whole dictionary.

The set of "inside sample" figures refers to just the frames for nouns defined in any of the ten samples, while the set of "outside sample" figures refers to the results of merging all those nouns that appeared as a genus term in some definition in some sample but were not themselves defined in that same sample. The third set of figures seems to indicate that a few "outside" nouns with respect to one sample were "inside" nouns with respect to another.

A rigorous analysis of the validity of **level** slot results would be difficult to arrange in any case, but it is perhaps premature now, since quality is supposed to improve with quantity, and we have only a small percentage of the LDOCE processed. However, a cursory examination of the mass of data already on hand suggests that the facts are generally quite plausible. Compound nouns like 'head of hair', 'hearing aid', and 'radiation sickness' are all reasonably marked as **belowBasic**. As expected, nouns marked **aboveBasic** include such common words as 'thing' (with counts 0 and 70), 'part' (0 58), 'place' (0 34), and 'piece' (0 30). Those with a high second count (indicating frequent appearance as a genus term) also generally seem to belong higher in the hierarchy than those with a low second count, though there are some surprises. Contrast 'person' (0 and 134, the highest second count) with 'object' (0 11) and 'aircraft' (0 1), for instance. The most common filler of the **level** slot has the **basic** level marker and both counts at 0. A few examples of nouns whose level slot has this filler are 'accommodation', 'brook', 'creek', and 'bromide'.

Table 7-D. Fillers of **level** Slots

			Inside Sample		Outside Sample		All Merged	
Marker	C1	C2	Frames	Levels	Frames	Levels	Frames	Levels
aboveBasic	0	1	29	1	390	1	409	1
	0	2	5	1	111	1	116	1
	0	m	-	-	129	19	131	19
	n	x	-	-	39	16	39	16
			34	2	669	37	695	37
basic	0	0	1033	1	2	1	1007	1
	1	0	-	-	100	1	100	1
	1	1	2	1	45	1	46	1
	2	x	1	1	20	3	21	3
	m	x	-	-	4	4	4	4
			1036	3	171	10	1178	10
belowBasic	0	0	228	1	4	1	232	1
			228	1	4	1	232	1
Totals:			1298	6	844	48	2105	48

n > 0, m > 2, x >= 0

The **absenceOf** slot was the only one that may be relevant to the fourth kind of information, (4) exceptions to facts inherited through membership in a category, and only 16 facts with this slot were found.

The next kind of information, (5) absolute attributes, was also rather rare. There were 12 facts with **equiMeasure** slots, 42 with **number** slots, and 16 with **sex** slots. Another kind of information dealing with attributes, (6) relative attributes, was much more common, since it claims most of the remaining classified facts, of which there were 2,871 altogether, not counting just the **typeOf** slots. Some unclassified facts might also be added to that total, in particular many of the 502 with an **attribute** slot.

Detail frames were one major repository of information of the next kind, (7) modification of relative attributes, and we have already seen that these frames were not very common. Only 177 were found altogether. There were also adverbs, however, extracted as degree markers like those described in Chapter 4, so a count was taken to see how common they are in the output.

The results of that count are shown in Table 7-E, which has five columns of figures (with hyphens equivalent to 0) for the number of adverbs or markers found in facts with a specified slot or group of slots. The first column is for those found with any name frame slot, the second one is for either of the three unclassified slots, the next two columns are for **typicalAction** and **treatment** slots, and the last one is for any other classified slot.

Row headings on the left side of Table 7-E identify the adverb or marker whose occurrences were counted. In one case, the extractor translated an adverb to a marker. That marker is shown first, then the "common" adverbs or markers that needed no translation, and finally the adverbs which were not translated but presumably should have been.

Adverbs and markers are indented in scale order and aligned with their translation. Another kind of information, (9) habitual, continuous, or repetitive actions, was mainly limited to facts with **treatment** or **typicalAction** slots, but these were fairly common, with 229 and 200 slots, respectively (still counting all ten samples).

Also rather common were facts with a **substance** slot, of which 395 were found. These facts are useful for establishing (12) the nature of members of a prototypical category (whether substance sample or discrete individuals). Frames that lack this fact refer to prototypes composed of discrete individuals by default, so this kind of information could be considered present in all frames in the output.

The final type of information is (14) descriptive information for establishing identity. As explained in Chapter 5, certain adverbs found in the definitions and incorporated into facts as applicability markers apparently yield some information about centrality by distinguishing a description as a primary or secondary characteristic. A special count like the one for degree markers was taken to gauge the extent to which this kind of information was successfully extracted. The results of that count are shown in Table 7-F, which reads like Table 7-E.

The preceding comparison of actual results with the ideals of Chapter 3 and the expectations of Chapter 5 would be incomplete without special attention to certain kinds of information particularly identified with prototype theory. Information about centrality and fuzzy boundaries is covered at least to some extent through the applicability markers reported in Table 7-F. Information about internal structure within categories is especially well covered by the many facts extracted with a **level** or **typeOf** slot, and facts with a **level** slot also give at least an approximate indication of how much fuzziness to expect. Table 7-D indicates that as many as 48 different levels were detected in the ten samples, and presumably even more would have appeared if the whole dictionary had been used as input.

Table 7-E. Count of Degree Markers

	Degree	Name-Fr.	Uncl.	Typ.Act.	Treat.	Other
Transl.	**considerably**	-	-	-	-	1
Common	**extremely**	-	-	-	-	-
	highly	-	8	-	-	20
	slightly	-	1	-	-	2
		-	9	-	-	22
Untr.	most	-	2	-	-	-
	very	-	18	1	2	-
	noticeably	-	-	-	-	-
	quite	-	1	-	-	-
	rather	-	6	-	2	-
		-	27	1	4	-

Table 7-F. Count of Applicability Markers

	Applicability	Name-Fr.	Uncl.	Typ.Act.	Treat.	Other
Transl.	**characteristic**	-	19	-	3	8
	primary	-	151	-	4	36
	secondary	-	38	-	2	15
		-	208	-	9	59
Untr.	often	3	3	-	-	-
	sometimes	-	3	1	-	-
	esp	41	9	-	2	-
	usu	6	9	-	-	-
	usually	-	2	-	-	-
		50	26	1	2	-

It was obvious from the start that no dictionary can provide all facts about any prototype nor even just all the salient facts that a human might be expected to know about the prototypes with which he is well acquainted. Wide and shallow coverage of world knowledge was expected, and that in a nutshell is what we find in dictionaries.

The evaluation offered in this section indicates that a large amount and great variety of prototype information is extracted by the SIV program when all kinds of such information are considered. Although a high percentage of this information would be of interest whatever the theoretical bias, the total amount and value of information uniquely related to prototype theory encourage further interest in our approach.

Chapter 8
Conclusion

The main idea underlying the work presented in the preceding chapters is that prototype theory can and should be applied to the representation of encyclopedic knowledge in computerized databases. The theory proposes that variably fuzzy boundaries between categories and internal structure within categories are important aspects of conceptual thought, which if properly represented allow a better, more accurate grasp of physical reality. The notion that this theory *can* be applied as stated has been tested by implementing a set of computer programs that extract world knowledge from a dictionary and represent it as required. The claim that the same theory *should* be applied as stated remains to be tested in some future application of a knowledge base enriched with the required information about prototypes.

Although the potential for applying prototype theory and its usefulness in a complete system are separate issues, they are not entirely unrelated. There is no point in debating usefulness if the potential is lacking for all practical purposes. Moreover, the amount of prototype information available in a complete system to test the theory will have a decisive impact on any conclusion that might be drawn from the effort. This underscores the relevance of the practical issue, addressed in this thesis, of how such information could be gathered in some automatic or semiautomatic fashion.

Even the question of potential is certainly not settled forever by a single thesis project. Although the present test is directly applicable only to non-abstract noun definitions in the LDOCE, it is important to note that these were not hand-selected as especially amenable to processing by the SIV program, so it seems safe to conclude that comparable results could be obtained on a much larger scale. However, even processing whole dictionaries may not yield enough information to make a suitable knowledge base feasible.

Since information about prototypes found in dictionaries is so shallow, with a general paucity of detail, one might naturally wonder if similar techniques can be applied to other sources of world knowledge available in machine-readable form [Ahlswede *et al.*, 1986: 66; Evens, 1989: 85]. A natural candidate would be encyclopedias. The language they contain is practically unrestricted, unlike that found in dictionaries, so the percentage of total information content that could be extracted in the same way would be considerably smaller, but since encyclopedias are typically so much larger than dictionaries, the possibility may merit further interest. The trick would be to identify phrases or other pieces of complete articles that could be understood with a fair degree of accuracy, ignoring the rest. Even specialized reference books and other sources of information might serve as input if the density of reliable information that can be extracted is deemed sufficiently high.

In spite of the formidable obstacles, information is already being extracted automatically from several kinds of ordinary text material besides dictionaries. For example, Velardi *et al.* [1989] describe a system that builds a "semantic lexicon" whose entries are derived automatically from "an on line database of press agency releases on finance and economics" [1989: 116]; Coates-Stephens [1991] tells about another system that extracts definitions of proper nouns from news stories; and Cavazza and Zweigenbaum [1992] describe yet another that gets information from technical reports on thyroid cancer. Delisle *et al.* list several other examples [1991: 326] and go on to describe their own system for automatically extracting facts from unconstrained text.

Besides the question of how much information about prototypes could be extracted automatically in a grander project, there is also the question of how good that information might be. Based on the results reported in this thesis, it is obvious that fairly accurate information about the core features of prototypes can be obtained from a dictionary.

In conclusion, this thesis has shown, apparently for the first time, that large amounts of accurate information can be extracted from a dictionary and represented in a database of frames that meet the major requirements of prototype theory, even without a deep parse of input text. Future work may well be directed at improving both the quantity and the quality of that information and at testing its usefulness in a complete system for a final application.

References

LDOCE *Longman Dictionary of Contemporary English: New Edition*. 1987. Essex: Longman Group.

W7 *Webster's Seventh New Collegiate Dictionary*. 1963. Springfield, MA: G&C Merriam Co.

W9 *Webster's Ninth New Collegiate Dictionary*. 1988. Springfield, MA: Merriam-Webster Inc.

Ahlswede, Thomas E. 1985. "A toolkit for lexicon building" in *Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics*, Chicago, IL, July 8-12, 1985. pp. 268-76.

----------. 1988. *Syntactic and Semantic Analysis of Definitions in a Machine-Readable Dictionary*. Illinois Institute of Technology dissertation.

Ahlswede, Thomas, and Martha Evens. 1988a. "Generating a relational lexicon from a machine-readable dictionary" in *International Journal of Lexicography*, vol. 1, no. 3, Fall 1988. pp. 214-237.

----------. 1988b. "Parsing vs. text processing in the analysis of dictionary definitions" in *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, Buffalo, NY, June 7-10, 1988. pp. 217-24.

Ahlswede, Thomas, Martha Evens, Kay Rossi, and Judith Markowitz. 1986. "Building a lexical database by parsing Webster's Seventh Collegiate Dictionary" in *Advances in Lexicography: Proceedings of the University of Waterloo (UW) Centre for the New Oxford English Dictionary 1st Annual Conference*, Waterloo, Ontario, November 6-7, 1985, Gayle Johannesen, ed. pp. 65-78.

Ahlswede, Thomas, Jeffrey Anderson, Martha Evens, James Neises, Sumali Pin-Ngern, and Judith Markowitz. 1988. "Automatic construction of a phrasal thesaurus for an information retrieval system from a machine readable dictionary" in *Proceedings of RIAO '88* (Recherche d'Information Assistée par Ordinateur), Cambridge, MA, March 1988. pp. 597-608.

Aikins, Janice S. 1983. "Prototypical knowledge for expert systems" in *Artificial Intelligence*, vol. 20, no. 2, February 1983. pp. 163-210.

Alshawi, Hiyan. 1987. "Processing dictionary definitions with phrasal pattern hierarchies" in *Computational Linguistics*, vol. 13, nos. 3-4, July-December 1987. pp. 195-202.

Amsler, Robert A. 1980. *The Structure of the Merriam-Webster Pocket Dictionary*. The University of Texas at Austin dissertation.

Amsler, Robert A., and Frank W. Tompa. 1988. "An SGML-based standard for English monolingual dictionaries" in *Information in Text: Proceedings of the University of Waterloo (UW) Centre for the New Oxford English Dictionary Fourth Annual Conference*, Waterloo, Ontario, October 26-28, 1988. pp. 61-79.

Blake, G. Elizabeth, Tim Bray, and Frank W. Tompa. 1992. "Shortening the OED: Experience with a grammar-defined database" in *ACM Transactions on Information Systems*, vol. 10, no. 3, July 1992. pp. 213-32.

Bobrowicz, O., C. Choulet, A. Haurat, F. Sandoz, and M. Tebaa. 1991. "A method to build membership functions: Applications to numerical/symbolic interface building" in *Uncertainty in Knowledge Bases*, Bouchon-Meunier *et al.*, eds. pp. 136-42.

Boguraev, Bran, and E. J. (Ted) Briscoe. 1987. "Large lexicons for natural language processing: Utilizing the grammar coding system of LDOCE" in *Computational Linguistics*, vol. 13, nos. 3-4, July-December 1987. pp. 203-18.

----------, eds. 1989. *Computational Lexicography for Natural Language Processing*. Essex: Longman Group.

Boguraev, Bran, Ted Briscoe, John Carroll, David Carter, and Claire Grover. 1987. "The derivation of a grammatically indexed lexicon from the Longman Dictionary of Contemporary English" in *Proceedings of the 25rd Annual Meeting of the Association for Computational Linguistics*, Stanford, CA, July 6-9, 1987, Candace Sidner, ed. pp. 193-200.

Boguraev, Branimir, and Mary Neff. 1992. "Text representation, dictionary structure, and lexical knowledge" in *Literary and Linguistic Computing*, vol. 7, no. 2. pp. 110-112.

Boose, John H. 1989. "A survey of knowledge acquisition techniques and tools" in *Knowledge Acquisition*, vol. 1, no. 1, March 1989. pp. 3-37.

Böttner, Michael. 1992. "State transition semantics" in *Theoretical Linguistics*, vol. 18, nos. 2-3. pp. 239-86.

Bouchon-Meunier, Bernadette, Ronald R. Yager, and Lotfi A. Zadeh, eds. 1991. *Uncertainty in Knowledge Bases* (proceedings of the Third International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU '90, Paris, France, July 2-6, 1990), vol. 521 in *Lecture Notes in Computer Science*. Berlin: Springer-Verlag.

Brachman, Ronald J., and Hector J. Levesque, eds. 1985. *Readings in Knowledge Representation*. Los Altos, CA: Morgan Kaufmann.

Bunt, Harry C. 1985. *Mass Terms and Model-theoretic Semantics*. Cambridge Studies in Linguistics, Vol. 42. London: Cambridge University Press.

Calzolari, Nicoletta. 1988. "The dictionary and the thesaurus can be combined," Chapter 3 in *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*, Martha Evens, ed. pp. 75-96.

Cavazza, Marc, and Pierre Zweigenbaum. 1992. "Extracting implicit information from free text technical reports" in *Information Processing & Management*, vol. 28, no. 5, September-October 1992. pp. 609-18.

Coates-Stephens, Sam. 1991. "Automatic acquisition of proper noun meanings" in *Methodologies for Intelligent Systems*, Ras and Zemankova, eds. pp. 306-15.

Deighan, John, and John Roach. n.d. *VPI Prolog User Manual*. Department of Computer Science, Virginia Tech, Blacksburg, VA.

Delisle, Sylvain, Stan Matwin, Jiandong Wang, and Lionel Zupan. 1991. "Explanation-Based Learning helps acquire knowledge from natural language texts" in *Methodologies for Intelligent Systems*, Ras and Zemankova, eds. pp. 326-37.

Dik, Simon C., Willem J. Meijs, and Piek Vossen. 1992. "LEXIGRAM: a functional lexico-grammatical tool for knowledge engineering" in *Linguistic Instruments in Knowledge Engineering*, van de Riet and Meersman, eds. pp. 21-54.

Dowty, David R., Robert E. Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*, vol. 11 in *Synthese Language Library*, Jaakko Hintikka and Stanley Peters, eds. Boston: D. Reidel Publishing Co.

Evens, Martha Walton. 1988. "Introduction" Chapter 1 in *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*, Martha Evens, ed. pp. 1-37.

----------, ed. 1988. *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks.* Cambridge: Cambridge University Press.

----------. 1989. "Computer-readable dictionaries," Chapter 3 in *Annual Review of Information Science and Technology*, vol. 24. pp. 85-117.

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

Evens, Martha, Judith Markowitz, Thomas Ahlswede, and Kay Rossi. 1987. "Digging in the dictionary: building a relational lexicon to support natural language processing" in *Issues and Developments in English and Applied Linguistics*, vol. 2. pp. 33-44.

Fought, John, Marcia Wesler, Heather Davenport, and Carol van Ess-Dykema. 1993. "Extending SGML Concurrent Structures: Toward computer-readable meta-dictionaries" in *Literary and Linguistic Computing*, vol. 8, no. 1. pp. 33-46.

Fox, Edward A., J. Terry Nutter, Thomas Ahlswede, Martha Evens, and Judith Markowitz. 1988. "Building a large thesaurus for information retrieval" in *Proceedings of the Second Conference On Applied Natural Language Processing* (Association for Computational Linguistics), Austin, TX, February 9-12, 1988, Bruce Ballard, ed. pp. 101-8.

Geeraerts, Dirk. 1988. "Where does prototypicality come from?" in *Topics in Cognitive Linguistics*, Brygida

Rudzka-Ostyn, ed. Amsterdam: John Benjamins Publishing Co.

Gish, Duane T. 1985. *Evolution: The Challenge of the Fossil Record*. El Cajon, CA: Creation-Life Publishers.

Goldfarb, Charles F. 1990. *The SGML Handbook*, Yuri Rubinsky, ed. Oxford: Clarendon Press.

Hasegawa, Yoko. 1993. "Prototype semantics: a case study of the TE K-/IK- constructions of Japanese" in *Language & Communication*, vol. 13, no. 1, January 1993. pp. 45-65.

Hayes, Patrick J. 1974. "Some problems and non-problems in representation theory" in *Readings in Knowledge Representation*, Brachman and Levesque, eds. 1985. pp. 3-22.

----------. 1985. "The second naive physics manifesto" in *Readings in Knowledge Representation*, Brachman and Levesque, eds. 1985. pp. 467-85.

Higginbotham, James. 1989. "Elucidations of meaning" in *Linguistics and Philosophy*, vol. 12, no. 4. August 1989. pp. 465-517.

Hirst, Graeme. 1989. "Ontological assumptions in knowledge representation" Department of Computer Science, University of Toronto. November 1989.

Jensen, Karen, and Jean-Louis Binot. 1987. "Disambiguating prepositional phrase attachments by using on-line dictionary definitions" in *Computational Linguistics*, vol. 13, nos. 3-4, July-December 1987. pp. 251-60.

----------. 1988. "Dictionary text entries as a source of knowledge for syntactic and other disambiguations" in *Proceedings of the Second Conference On Applied Natural Language Processing* (Association for Computational Linguistics), Austin, TX, February 9-12, 1988, Bruce Ballard, ed. pp. 152-58.

Jepsen, Glenn L. 1966. "Early eocene bat from Wyoming" in *Science*, vol. 154, no. 3754. Dec. 9, 1966. pp. 1333-39.

Kratzer, Angelika. 1989. "An investigation of the lumps of thought" in *Linguistics and Philosophy*, vol. 12, no. 5. October 1989. pp. 607-53.

Lakoff, George. 1972. "Hedges: A study in meaning criteria and the logic of fuzzy concepts" in *Papers from the Eighth Regional Meeting, April 14-16, 1972*. Chicago Linguistic Society. pp. 183-228.

----------. 1987. *Women, Fire, and Dangerous Things: What Categories Reveal About the Mind*. Chicago: University of Chicago Press.

Langacker, Ronald W. 1987. *Foundations of Cognitive Grammar, Volume I, Theoretical Prerequisites*. Stanford, CA: Stanford University Press.

Lenat, Douglas B., and R. V. Guha. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. Reading, MA: Addison-Wesley.

----------. 1991. "The evolution of CycL, the Cyc representation language" in *SIGART Bulletin*, vol. 2, no. 3, June 1991. pp. 84-87.

Lesmo, Leonardo, and Pietro Torasso. 1987. "Prototypical knowledge for interpreting fuzzy concepts and quantifiers" in *Fuzzy Sets and Systems*, vol. 23, no. 3, September 1987. pp. 361-70.

Lester, Lane P., and Raymond G. Bohlin. 1989. *The Natural Limits to Biological Change*. Second Edition. Dallas: Probe Books.

MacGregor, Robert M. 1991. "Inside the LOOM description classifier" in *SIGART Bulletin*, vol. 2, no. 3, June 1991. pp. 88-92.

Marsh, Frank L. 1976. *Variation and Fixity in Nature: The Meaning of Diversity and Discontinuity in the World of Living Things, and Their Bearing on Creation and Evolution*. Mountain View, CA: Pacific Press Publishing Association.

Mauldin, Michael L. 1991. "Retrieval performance in FERRET: A conceptual information retrieval system" in *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval,* Chicago, IL, October 13-16, 1991, A. Bookstein, Y. Chiaramella, G. Salton, and V. V. Raghavan, eds. Special issue of *SIGIR Forum,* October 1991. pp. 347-55.

May, Robert. 1989. "Interpreting logical form" in *Linguistics and Philosophy*, vol. 12, no. 4. August 1989. pp. 387-435.

Mel'čuk, Igor A., and Alexander K. Zholkovsky. 1988. "The explanatory combinatorial dictionary," Sally B. Hankwitz, translator, Chapter 2 in *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*, Martha Evens, ed. pp. 41-74.

Miller, George A. 1986. "WordNet: A dictionary browser" in *Advances in Lexicography: Proceedings of the University of Waterloo (UW) Centre for the New Oxford English Dictionary 1st Annual Conference*, Waterloo, Ontario, November 6-7, 1985, Gayle Johannesen, ed. pp. 25-28.

Minsky, Marvin. 1981. "A framework for representing knowledge" in *Readings in Knowledge Representation*, Brachman and Levesque, eds. 1985. pp. 245-62.

Nutter, J. Terry. 1989. "A Lexical Relation Hierarchy," TR 89-6. Department of Computer Science, Virginia Polytechnic Institute and State University.

Partee, Barbara H., Alice ter Meulen, and Robert E. Wall. 1990. *Mathematical Methods in Linguistics*, vol. 30 in *Studies in Linguistics and Philosophy*, Gennaro Chierchia, Pauline Jacobson, and Francis J. Pelletier, eds. Boston: Kluwer Academic Publishers.

Ras, Zbigniew W., and Maria Zemankova, eds. 1991. *Methodologies for Intelligent Systems* (proceedings of the Sixth International Symposium on Methodologies for Intelligent Systems, ISMIS '91, Charlotte, NC, October 16-19, 1991), vol. 542 in *Lecture Notes in Artificial Intelligence* (subseries of *Lecture Notes in Computer Science*). Berlin: Springer-Verlag.

Riesbeck, Christopher K., and Roger C. Schank. 1989. *Inside Case-based Reasoning*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Rosch, Eleanor. 1975. "Cognitive representations of semantic categories" in *Journal of Experimental Psychology: General*. Vol. 104. pp. 192-233.

Schank, Roger C. 1975. *Conceptual Information Processing*. Amsterdam: North-Holland.

----------. 1982. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge: Cambridge University Press.

Slator, Brian M. 1989a. "Extracting lexical knowledge from dictionary text" in *Knowledge Acquisition*, vol. 1, no. 1, March 1989. pp. 89-112.

----------. 1989b. "Extracting lexical knowledge from dictionary text" in *SIGART Newsletter*, no. 108, April 1989. pp. 173-74.

----------. 1992. "Sense and preference" in *Computers & Mathematics with Applications*, vol. 23, nos. 6-9. March-May 1992. pp. 391-402.

Small, Steven. 1980. *Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding*. University of Maryland, College Park, thesis.

Stanley, Steven M. 1979. *Macroevolution: Pattern and Process*. San Francisco: W. H. Freeman and Co.

Taylor, John R. 1989. *Linguistic Categorization: Prototypes in Linguistic Theory*. Oxford: Clarendon Press.

van Benthem, Johan. 1989. "Polyadic quantifiers" in *Linguistics and Philosophy*, vol. 12, no. 4. August 1989. pp. 437-464.

----------. 1991. "General dynamics" in *Theoretical Linguistics*, vol. 17, nos. 1-3. pp. 159-201.

van de Riet, R. P., and R. A. Meersman, eds. 1992. *Linguistic Instruments in Knowledge Engineering* (proceedings of the 1991 Workshop on Linguistic Instruments in Knowledge Engineering, Tilburg, The Netherlands, January 17-18, 1991). Amsterdam: North-Holland.

Velardi, Paola, Maria Teresa Pazienza, and Stefano Magrini. 1989. "Acquisition of semantic patterns from a natural corpus of texts" in *SIGART Newsletter*, no. 108, April 1989. pp. 115-23.

Wierzbicka, Anna. 1991. "Semantic complexity: Conceptual primitives and the principle of substitutability" in *Theoretical Linguistics*, vol. 17, nos. 1-3. pp. 75-97.

Zadeh, Lotfi A. 1965. "Fuzzy sets" in *Information and Control*, vol. 8, pp. 338-53.

Appendix A
The Slot Catalog

Section E in Chapter 4 describes the slot catalog as a specialized data structure for SIV programs. It is pressed into service only when database files are merged by a SIV program accessible through its interface. The slot catalog is actually the file **slotcat.siv**, part of the complete SIV installation.

This appendix is a version of that file slightly modified to fit the format requirements for this appendix. The second column specifies the type of frame that is allowed to have the indicated slot. See Chapter 4 for a general discussion of slots and a fuller description of a few representative slots. An understanding of those descriptions may help clarify the briefer ones supplied here.

Table A-A. Slot Catalog

Slot Name	Frame Type	Description (f = filler; n = nominal concept)
ability	sense	n is capable of doing f
absence of	sense	f is absent from n
affected	sense	n typically affects f in some direct manner
analogue	sense	f is like n in some important way
appearance	sense	physical appearance of n (pretty, ugly, etc.)
attribute	sense	simple (unclassified) attribute of n
branch of	sense	n is a branch of f
cause	sense	n is caused by f
color	sense	color of n
dialect	name	dialect or language of name
dimension	sense	f describes the size of n but in only one or two dimensions (wide, tall, etc.)
dryness	sense	f tells how dry n typically is
effect	sense	an effect typically received or experienced by an observer (or hearer or experiencer) of n
equi_measure	sense	a measure equivalent to n, f being a list whose first element is a number
era	sense	past - n is outdated
essence	sense	f represents the substance, entity, or group that is or constitutes n (n not being just a type of f)
example	sense	f is an example of n
family of	sense	n is in the family of f
firmness	sense	firmness of n (hard, soft, etc.)
fit	sense	fit of n (as a tight or loose garment)
flavor	sense	the normal flavor of n
form	sense	physical form of n (hollow, solid, tight, loose)

continued

Table A-A. Slot Catalog (continued)

Slot Name	Frame Type	Description (f = filler; n = nominal concept)
frequency	sense	f indicates how frequently or commonly n occurs
function	name	f labels the part of speech of name
geo location	sense	geographical or celestial location of n
group	sense	n represents a group concept if slot is present
group of	sense	n is a group of f
habitat	sense	n typically dwells in the specified habitat
hazard	sense	a harmful, potential effect or hazard of n
ingredient	sense	f is an important ingredient of n
larger whole	sense	n is a part of f
level	name	location of name in taxonomic hierarchy
local_effect	sense	f is a feeling or sensation typically experienced as a result of n
location	sense	unclassified location of n
made from	sense	f is an important raw material for making n
maturity	sense	maturity of n (young, old, etc.)
members	sense	n is a group or set whose members are f
motility	sense	f describes the way n typically moves
movement of	sense	n is or involves the movement of f
non prototype	sense	n is an actual instance if slot is present
number	sense	number of n typical for a concept being described
part	sense	f is a part of n
pitch	sense	f describes dominant frequency of sound of n
place of use	sense	f is a place where n is typically used
position of	sense	n is a position, office, or rank held by f
potential use	sense	f describes some way that n might be used
produced by	sense	f produces, causes, or bears n

continued

Table A-A. Slot Catalog (continued)

Slot Name	Frame Type	Description (f = filler; n = nominal concept)
product	sense	n produces, causes, or bears f
quantity	sense	n represents a unit, measure, or quantity of something if slot is present
scope	sense	f indicates whether n is public or private, national or international, etc.
sense	name	nil - only one sense of word is on record, else a list that has highest homonym number, followed by sublist for each homonym with more than one sense, each sublist beginning with the homonym number, continuing with number of senses, and ending with similar sublists, one for each sense that has more than one subsense
sex	sense	the sex of n (male or female)
shape	sense	physical or geometric shape of n
size	sense	f gives size of n relative to others of same type
smell	sense	the smell, aroma, or fragrance of n
sound	sense	a description of the sound typically made by n
speed	sense	the typical speed of n when in motion
strength	sense	either the static or the dynamic strength of n
substance	sense	n is a substance if slot is present
synonym	sense	another name used in approximately the same sense
taste	sense	the normal taste of n
temperature	sense	f describes normal temperature of n
tension	sense	tension of n (as a tight or loose string)
texture	sense	the way n feels when touched
topic	sense	f is the topic or subject matter of n
treatment	sense	f is a phrase describing action performed on n

continued

Table A-A. Slot Catalog (continued)

Slot Name	Frame Type	Description (f = filler; n = nominal concept)
type of	sense	n is a type of f
typical action	sense	f is something that n typically does
typical food	sense	something that n typically eats
typical use	sense	f describes some way that n is used
typical user	sense	f is among the typical users of the n
uncl appos	sense	single word or modifier found after genus term
uncl descr	sense	complex attribute of n (found after genus term)
usage	name	f describes the way name is typically used
value	sense	the value normally ascribed to n
variant	name	a lexical, regional, or stylistic variant of name
virtue_neg	sense	f describes n as transgressing some standard of good and bad or right and wrong
virtue_pos	sense	f names some positive virtue of n based on some standard of good and bad or right and wrong
viscosity	sense	f describes the viscosity of n (thick, thin, etc.)
wealth	sense	f describes the economic status of n
weight	sense	f is the normal weight of n (f may specify unit of weight and count)

Appendix B

More Details on Results

The tables in this appendix expand on material presented in Chapter 7, which should be read first. Most tables have a details portion and a summary portion. The first has individual figures for each of the ten samples, and the second summarizes results obtained from just the work samples, from just the test samples, and from both sets of samples.

In all tables a single hyphen is equivalent to a zero. A double hyphen or empty space indicates that no meaningful figure can be entered. For example, if a column gives the percentage of facts judged to be correct, a double hyphen on the row for some type of fact means that no such fact was found.

The following tables are included in this appendix:

Run times reported in the overview table were recorded when SIV ran on a DECstation 5000/125.

Table B-A1. Sample Overview, Details

Sample	0	1	2	3	4	5	6	7	8	9
All Entries	241	275	243	217	228	254	256	273	217	234
Noun Entries	115	161	156	111	131	136	99	173	122	140
All Senses	434	425	432	379	360	433	406	372	375	427
Noun Senses	229	278	316	216	243	254	151	263	228	261
Non-abstract	200	276	308	203	235	242	146	249	217	251
Percent Nouns	47.7	58.5	64.2	51.2	57.5	53.5	38.7	63.4	56.2	59.8
" Noun Senses	52.8	65.4	73.1	57.0	67.5	58.7	37.2	70.7	60.8	61.1
" Non-abstract	46.1	64.9	71.3	53.6	65.3	55.9	36.0	66.9	57.9	58.8
1000 *.raw	92.3	87.7	94.1	95.3	91.8	90.2	89.8	86.6	95.6	91.8
Bytes: *.unf	33.1	31.9	36.2	28.7	28.7	31.0	30.0	29.5	30.4	31.8
*.fmt	23.4	28.0	35.0	21.2	24.2	25.3	16.1	25.8	23.3	26.0
*.db	33.5	46.5	54.6	32.2	37.6	38.4	26.5	43.2	37.6	42.1
Times: Format	17.2	17.0	20.2	16.6	15.5	17.5	16.2	15.7	17.1	18.0
Parse	42.6	51.3	69.4	49.2	44.7	56.3	30.5	50.9	44.1	48.4
Extract	26.8	43.6	50.2	30.9	33.7	38.8	25.1	41.6	31.8	37.3
Seconds Total	87	112	140	97	94	113	72	108	93	104
Bytes/Second										
Rates: Format	5359	5151	4657	5736	5937	5164	5536	5513	5579	5098
Parse	778	623	522	582	642	550	986	580	690	657
Extract	873	641	697	688	719	652	640	620	735	698
Form-Pars-Extr	1066	783	673	986	978	801	1251	801	1028	886
Senses/Second										
Rates: Format	25.2	25.0	21.4	22.8	23.3	24.8	25.0	23.7	21.9	23.7
Parse	5.4	5.4	4.6	4.4	5.4	4.5	5.0	5.2	5.2	5.4
Extract	7.5	6.3	6.1	6.6	7.0	6.2	5.8	6.0	6.8	6.7
Form-Pars-Extr	2.3	2.5	2.2	2.1	2.5	2.1	2.0	2.3	2.3	2.4
Senses: Total	229	278	316	216	243	254	151	263	228	261
Variant	11	5	12	14	14	12	6	10	2	10
Abstract	29	2	8	13	8	12	5	14	11	10
Mixed-Function	7	6	8	1	15	2	5	7	-	2
Senses / Noun	1.99	1.73	2.03	1.95	1.85	1.87	1.53	1.52	1.87	1.86
Classif. Facts	375	612	746	390	475	480	320	480	462	526
Unclass. Facts	288	412	510	310	354	360	218	381	356	382
Distinct Slots	45	54	65	44	58	54	52	54	51	55
Cl.F. / Sense	1.88	2.22	2.42	1.92	2.02	1.98	2.19	1.93	2.13	2.12
simpl G cmplx	35	93	93	51	46	50	42	69	56	82
simpl G -	15	34	29	14	23	25	13	28	25	29
- G cmplx	136	106	145	109	124	133	73	126	109	111
- G -	43	45	49	42	50	46	23	40	38	39

Table B-A2. Sample Overview, Summary

Sample	Work	Test	Both	(Percent)	Projected
All Entries	1185	1253	2438	100.0	42511
Noun Entries	623	721	1344	55.1	23435
All Senses	2007	2036	4043	100.0	70498
Noun Senses	1167	1272	2439	60.3	42529
Non-abstract	1106	1221	2327	57.6	40576
Percent Nouns	52.6	57.5	55.1		
" Noun Senses	58.1	62.5	60.3		
" Non-abstract	55.1	60.0	57.6		
1000 *.raw	463.5	451.6	915.2	5.7	15957.7
Bytes: *.unf	158.5	152.9	311.4		5429.6
*.fmt	122.0	126.3	248.3		4330.0
*.db	189.7	202.4	392.1		6837.5
Times: Format	86.2	84.8	171.1	16.8	2983
Parse	231.2	256.2	487.3	47.9	8498
Extract	167.5	192.1	359.6	35.3	6271
Seconds Total	485	533	1018	100.0	17751
Bytes / Sec.					
Rates: Format	5375	5324	5350		
Parse	686	597	639		
Extract	728	657	690		
Form-Pars-Extr	956	847	899		
Senses / Sec.					
Rates: Format	23.3	24.0	23.6		
Parse	5.0	5.0	5.0		
Extract	6.6	6.4	6.5		
Form-Pars-Extr	2.3	2.3	2.3		
Senses: Total	1167	1272	2439	100.0	42529
Variant	45	51	96	3.9	1674
Abstract	61	51	112	4.6	1953
Mixed-Function	35	18	53	2.2	924
Senses / Noun	1.87	1.76	1.81		
Classif. Facts	2378	2488	4866	57.7	84848
Unclass. Facts	1726	1845	3571	42.3	62267
Distinct Slots	72	67	73		
Cl.F. / Sense	2.15	2.04	2.09		
simpl G cmplx	272	345	617	25.3	10759
simpl G -	105	130	235	9.6	4098
- G cmplx	587	585	1172	48.1	20436
- G -	203	212	415	17.0	7236

Table B-B1. Parser Evaluation, Details

Sample	0	1	2	3	4	5	6	7	8	9
Noun Senses	229	278	316	216	243	254	151	263	228	261
Variant	11	5	12	14	14	12	6	10	2	10
Mixed-Function	7	6	8	1	15	2	5	7	-	2
Mismarked	-	2	-	1	3	1	-	3	-	2
Disqualified	18	13	20	16	26	15	11	20	2	14
Qualified	211	265	296	200	217	239	140	243	226	247
Wrong Genus	-	9	1	2	-	9	-	18	2	-
Percent Right	100	96.6	99.7	99.0	100	96.2	100	92.6	99.1	100
Mixed-Function	7	6	8	1	15	2	5	7	-	2
Qualified	7	6	8	1	9	2	5	7	-	2
Bad Genus	1	1	1	1	4	1	2	4	-	-
Percent Good	85.7	83.3	87.5	0	55.6	50.0	60.0	42.9	--	100

Table B-B2. Parser Evaluation, Summary

Sample	Work	Test	Both	(Percent)	Projected
Noun Senses	1428	1011	2439	100.0	42529
Variant	55	41	96	3.9	1674
Mixed-Function	37	16	53	2.2	924
Mismarked	5	7	12	0.5	209
Disqualified	91	64	155	6.4	2703
Qualified	1337	947	2284	93.6	39826
Wrong Genus	3	38	41		715
Percent Right	99.8	96.0	98.2		
Mixed-Function	37	16	53	100.0	924
Qualified	31	16	47	88.7	820
Bad Genus	8	7	15		262
Percent Good	74.2	56.3	68.1		

Table B-C1. Unique Slot-Frame Pairs, Details, Part 1

Slot \ Sample	0	1	2	3	4	5	6	7	8	9
absence_of	3	3	3	-	-	1	2	1	-	1
affected	1	1	1	-	4	-	-	-	3	-
analogue	3	2	3	-	2	1	3	-	3	3
appearance	1	1	5	3	1	3	-	2	2	3
cause	-	2	-	-	2	1	-	1	-	4
color	1	11	9	3	7	4	1	-	3	5
dimension	4	9	22	2	4	6	3	7	18	20
dryness	-	-	-	-	1	1	1	2	-	1
effect	2	4	4	4	1	1	2	2	3	2
equi_measure	-	-	1	-	1	2	-	6	-	1
era	2	1	6	-	1	-	1	3	1	2
essence	5	7	24	7	7	10	4	10	6	11
example	2	-	3	2	-	1	3	2	1	-
firmness	-	5	5	-	1	1	-	3	-	-
fit	-	-	-	-	1	-	2	-	-	-
flavor	-	-	2	-	-	-	-	-	-	-
form	1	2	7	3	2	2	1	1	-	2
frequency	1	1	1	-	2	1	2	3	1	-
geo_location	2	10	9	1	1	3	2	5	-	1
group	2	1	2	-	-	1	2	8	2	1
habitat	-	-	1	-	1	-	-	-	1	-
hazard	2	1	1	2	3	-	2	-	3	7
ingredient	2	1	2	1	1	-	3	1	4	2
larger_whole	2	6	6	8	12	6	4	10	11	4
local_effect	-	-	1	2	5	2	-	1	-	1
location	7	12	23	14	19	22	8	8	17	13
made_from	3	6	17	-	-	-	2	2	2	4
maturity	-	4	3	1	2	-	1	-	-	1
members	2	1	2	-	-	1	2	8	2	1
motility	-	-	3	-	-	1	-	-	-	1
movement_of	-	-	1	-	1	-	-	-	-	-
non_prototype	-	1	1	-	4	-	2	1	-	-
number	1	6	7	1	2	3	5	10	4	3
part	-	10	9	-	9	5	2	9	9	3
pitch	-	-	2	-	-	-	1	-	1	1
potential_use	-	-	-	1	1	1	-	-	-	-
produced_by	3	3	8	1	4	3	-	2	1	4
product	-	-	1	1	2	1	3	1	1	1
scope	-	1	3	-	2	1	2	2	1	-
sex	-	4	1	-	2	1	3	4	-	1
shape	1	13	18	6	-	6	1	6	12	2
size	1	26	14	12	6	14	7	7	8	12

Table B-C2. Unique Slot-Frame Pairs, Details, Part 2

Slot \ Sample	0	1	2	3	4	5	6	7	8	9
smell	-	-	-	1	-	-	-	-	-	-
sound	2	2	4	-	1	-	-	1	1	3
speed	-	2	2	-	-	2	1	1	4	1
strength	4	5	4	1	3	5	2	2	3	5
substance	52	49	44	41	43	27	29	36	27	47
synonym	24	38	31	23	35	28	15	29	32	27
taste	2	1	4	-	-	-	-	-	-	-
temperature	-	-	-	-	-	1	2	-	-	-
texture	-	7	2	1	1	1	-	-	1	1
topic	6	2	8	4	4	6	1	2	5	3
treatment	14	26	35	14	14	18	18	21	23	38
type_of	147	219	242	154	178	183	116	191	167	197
typical_action	14	19	18	11	23	27	16	21	9	21
typical_food	-	-	-	1	1	-	1	-	1	-
typical_use	19	28	30	16	15	21	23	19	18	21
typical_user	-	2	5	1	3	3	3	2	3	3
value	-	-	1	2	1	1	-	1	-	-
virtue_neg	2	7	9	3	5	3	-	1	3	1
virtue_pos	1	-	1	1	-	2	1	1	1	1
viscosity	-	-	2	-	-	-	-	-	-	-
wealth	-	-	-	-	1	-	-	-	-	-
weight	-	7	2	-	5	-	-	1	2	1
Classified:	341	569	675	349	447	434	305	457	420	488
attribute	30	63	43	41	34	41	27	50	43	60
uncl_appos	19	16	21	13	12	8	6	25	9	11
uncl_descr	110	159	180	122	126	128	83	144	130	143
Unclassified:	159	238	244	176	172	177	116	219	182	214
dialect	6	21	18	5	8	16	9	19	24	12
function	110	142	149	103	130	131	95	162	116	130
level	225	313	330	235	266	280	191	328	261	286
sense	55	57	56	41	41	39	34	45	33	53
usage	23	42	20	12	16	36	14	34	20	25
variant	6	5	8	6	8	8	6	10	2	10
Name Frames:	425	580	581	402	469	510	349	598	456	516
Sums:	925	1387	1500	927	1088	1121	770	1274	1058	1218

Table B-C3. Unique Slot-Frame Pairs, Summary, Part 1

Slot \ Sample	Work	Test	Both	Projected
absence_of	8	6	14	244
affected	9	1	10	174
analogue	14	6	20	349
appearance	9	12	21	366
cause	2	8	10	174
color	21	23	44	767
dimension	51	44	95	1657
dryness	2	4	6	105
effect	12	13	25	436
equi_measure	2	9	11	192
era	11	6	17	296
essence	46	45	91	1587
example	9	5	14	244
firmness	6	9	15	262
fit	3	-	3	52
flavor	2	-	2	35
form	11	10	21	366
frequency	7	5	12	209
geo_location	14	20	34	593
group	8	11	19	331
habitat	3	-	3	52
hazard	11	10	21	366
ingredient	12	5	17	296
larger_whole	35	34	69	1203
local_effect	6	6	12	209
location	74	69	143	2493
made_from	24	12	36	628
maturity	6	6	12	209
members	8	11	19	331
motility	3	2	5	87
movement_of	2	-	2	35
non_prototype	7	2	9	157
number	19	23	42	732
part	29	27	56	976
pitch	4	1	5	87
potential_use	1	2	3	52
produced_by	16	13	29	506
product	7	4	11	192
scope	8	4	12	209
sex	6	10	16	279
shape	32	33	65	1133
size	36	71	107	1866

Table B-C4. Unique Slot-Frame Pairs, Summary, Part 2

Slot \ Sample	Work	Test	Both	Projected
smell	-	1	1	17
sound	8	6	14	244
speed	7	6	13	227
strength	16	18	34	593
substance	195	200	395	6888
synonym	137	145	282	4917
taste	6	1	7	122
temperature	2	1	3	52
texture	4	10	14	244
topic	24	17	41	715
treatment	104	117	221	3854
type_of	850	944	1794	31282
typical_action	80	99	179	3121
typical_food	3	1	4	70
typical_use	105	105	210	3662
typical_user	14	11	25	436
value	2	4	6	105
virtue_neg	19	15	34	593
virtue_pos	4	5	9	157
viscosity	2	-	2	35
wealth	1	-	1	17
weight	9	9	18	314
Classified:	2188	2297	4485	78205
attribute	177	255	432	7533
uncl_appos	67	73	140	2441
uncl_descr	629	696	1325	23104
Unclassified:	873	1024	1897	33078
dialect	65	73	138	2406
function	600	668	1268	22110
level	1273	1442	2715	47341
sense	219	235	454	7916
usage	93	149	242	4220
variant	30	39	69	1203
Name Frames:	2280	2606	4886	85197
Sums:	5341	5927	11268	196480

No facts with any of these slots were extracted from any of the samples:

ability
branch_of
family_of
group_of
place_of_use
position_of
tension
quantity

Table B-D1. Unique Slot-Detail-Frame Pairs, Details

Slot \ Sample	0	1	2	3	4	5	6	7	8	9
analogue	-	-	2	-	-	-	-	-	1	-
color	-	2	1	-	2	2	1	-	1	3
dimension	1	3	3	-	1	1	1	4	7	2
dryness	-	-	-	-	-	-	-	1	-	-
effect	-	-	1	-	-	-	-	-	-	-
firmness	-	-	2	-	-	-	-	1	-	-
form	-	-	2	1	1	-	1	-	-	-
frequency	1	-	-	-	-	-	1	-	-	-
location	-	-	-	-	-	-	-	-	-	1
made_from	-	-	1	-	-	-	-	-	-	-
number	1	2	5	-	2	2	-	1	4	1
scope	-	-	-	-	-	-	1	-	-	-
sex	-	-	-	-	-	-	2	-	-	-
shape	-	-	-	-	-	1	-	1	5	-
size	-	1	-	3	2	2	1	1	-	1
speed	-	-	-	-	-	-	-	-	-	1
strength	-	-	1	-	1	-	-	-	2	-
taste	-	-	2	-	-	-	-	-	-	-
temperature	-	-	-	-	-	-	1	-	-	-
treatment	-	-	3	-	-	-	-	1	1	2
typical_action	-	-	-	-	-	-	-	-	1	-
typical_use	-	-	1	-	-	-	-	-	-	-
typical_user	-	1	1	-	-	-	-	1	1	-
value	-	-	-	1	-	-	-	-	-	-
virtue_neg	-	-	-	-	1	-	-	-	-	-
virtue_pos	1	-	-	-	-	-	-	-	1	-
weight	-	-	-	-	1	-	-	-	-	-
attribute	4	7	11	6	6	6	1	4	8	4
dialect	1	-	-	-	2	-	-	-	-	-
Sums:	9	16	36	11	19	14	10	15	32	15

Table B-D2. Unique Slot-Detail-Frame Pairs, Summary

Slot \ Sample	Work	Test	Both	Projected
analogue	3	-	3	52
color	5	7	12	209
dimension	13	10	23	401
dryness	-	1	1	17
effect	1	-	1	17
firmness	2	1	3	52
form	4	1	5	87
frequency	2	-	2	35
location	-	1	1	17
made_from	1	-	1	17
number	12	6	18	314
scope	1	-	1	17
sex	2	-	2	35
shape	5	2	7	122
size	3	8	11	192
speed	-	1	1	17
strength	4	-	4	70
taste	2	-	2	35
temperature	1	-	1	17
treatment	4	3	7	122
typical_action	1	-	1	17
typical_use	1	-	1	17
typical_user	2	2	4	70
value	-	1	1	17
virtue_neg	1	-	1	17
virtue_pos	2	-	2	35
weight	1	-	1	17
attribute	30	27	57	994
dialect	3	-	3	52
Sums:	106	71	177	3086

Table B-D3. Unique Slot-Detail-Frame Pairs, Frequency Analysis

Sample	Work	Test	Both	Percent
5 Slots:	63	58	121	69.54
11 Slots:	32	9	41	23.56
12 Slots:	8	4	12	6.90
28 Slots:	103	71	174	100.00

Slot \ Sample	Work	Test	Both	Percent
attribute	30	27	57	32.76
dimension	13	10	23	13.22
number	12	6	18	10.34
color	5	7	12	6.90
size	3	8	11	6.32
5 Slots:	63	58	121	69.54
shape	5	2	7	4.02
treatment	4	3	7	4.02
form	4	1	5	2.87
strength	4	-	4	2.30
typical_user	2	2	4	2.30
analogue	3	-	3	1.72
firmness	2	1	3	1.72
frequency	2	-	2	1.15
sex	2	-	2	1.15
taste	2	-	2	1.15
virtue_pos	2	-	2	1.15
11 Slots:	32	9	41	23.56
dryness	-	1	1	0.57
effect	1	-	1	0.57
location	-	1	1	0.57
made_from	1	-	1	0.57
scope	1	-	1	0.57
speed	-	1	1	0.57
temperature	1	-	1	0.57
typical_action	1	-	1	0.57
typical_use	1	-	1	0.57
value	-	1	1	0.57
virtue_neg	1	-	1	0.57
weight	1	-	1	0.57
12 Slots:	8	4	12	6.90

Table B-E1. Unique Slot-Frame-Filler Triples (Facts), Details, Part 1

Slot \ Sample	0	1	2	3	4	5	6	7	8	9
absence_of	3	4	3	-	-	1	2	1	-	2
affected	1	1	2	-	4	-	-	-	4	-
analogue	3	2	3	-	2	1	3	-	3	3
appearance	1	1	5	3	1	3	-	2	2	3
cause	-	2	-	-	2	1	-	1	-	4
color	1	11	9	3	7	4	1	-	3	5
dimension	4	10	23	2	4	8	3	7	18	23
dryness	-	-	-	-	1	1	1	2	-	1
effect	2	4	4	4	1	1	2	2	3	2
equi_measure	-	-	1	-	1	2	-	7	-	1
era	2	1	6	-	1	-	1	3	1	2
essence	5	9	27	7	8	12	4	10	6	11
example	4	-	7	3	-	1	4	2	1	-
firmness	-	5	5	-	1	1	-	3	-	-
fit	-	-	-	-	1	-	2	-	-	-
flavor	-	-	2	-	-	-	-	-	-	-
form	1	2	7	3	2	2	1	1	-	2
frequency	1	1	1	-	2	1	2	3	1	-
geo_location	2	11	10	1	1	5	2	5	-	1
group	2	1	2	-	-	1	2	8	2	1
habitat	-	-	1	-	1	-	-	-	1	-
hazard	2	1	1	2	3	-	2	-	3	8
ingredient	2	1	2	1	1	-	4	1	4	2
larger_whole	2	6	6	9	12	6	4	10	11	4
local_effect	-	-	1	2	6	2	-	1	-	1
location	7	14	27	15	19	25	8	9	20	14
made_from	3	6	17	-	-	-	2	2	2	4
maturity	-	4	3	1	2	-	1	-	-	1
members	2	1	2	-	-	1	2	8	2	1
motility	-	-	3	-	-	1	-	-	-	1
movement_of	-	-	1	-	1	-	-	-	-	-
non_prototype	-	1	1	-	4	-	2	1	-	-
number	1	6	7	1	2	3	5	10	4	3
part	-	10	11	-	10	5	2	9	11	4
pitch	-	-	2	-	-	-	1	-	1	1
potential_use	-	-	-	1	1	1	-	-	-	-
produced_by	3	3	8	1	4	3	-	2	1	4
product	-	-	1	1	2	1	3	1	2	1
scope	-	1	3	-	2	1	2	2	1	-
sex	-	4	1	-	2	1	3	4	-	1
shape	1	13	20	7	-	6	1	6	13	3
size	1	26	14	12	6	14	7	7	8	12
smell	-	-	-	1	-	-	-	-	-	-
sound	2	2	4	-	1	-	-	1	1	3
speed	-	2	2	-	-	2	1	1	4	1
strength	4	5	4	1	3	5	2	2	3	5
substance	52	49	44	41	43	27	29	36	27	47
synonym	33	41	38	31	42	36	15	33	42	32

Table B-E2. Unique Slot-Frame-Filler Triples (Facts), Details, Part 2

Slot \ Sample	0	1	2	3	4	5	6	7	8	9
taste	2	1	4	-	-	-	-	-	-	-
temperature	-	-	-	-	-	1	2	-	-	-
texture	-	7	2	1	1	1	-	-	1	1
topic	6	2	9	5	4	7	2	2	5	3
treatment	14	28	37	16	14	18	18	21	24	39
type_of	166	240	276	176	193	206	128	206	189	215
typical_action	14	23	20	14	25	31	16	23	9	25
typical_food	-	-	-	1	1	-	1	-	1	-
typical_use	22	32	36	17	15	22	23	19	19	23
typical_user	-	2	5	1	4	3	3	2	3	3
value	-	-	1	2	1	1	-	1	-	-
virtue_neg	3	9	10	3	5	3	-	1	3	1
virtue_pos	1	-	1	1	-	2	1	1	1	1
viscosity	-	-	2	-	-	-	-	-	-	-
wealth	-	-	-	-	1	-	-	-	-	-
weight	-	7	2	-	5	-	-	1	2	1
Classified:	375	612	746	390	475	480	320	480	462	526
attribute	36	72	47	44	41	48	28	63	53	70
uncl_appos	21	17	24	15	14	9	6	28	12	12
uncl_descr	231	323	439	251	299	303	184	290	291	300
Unclassified:	288	412	510	310	354	360	218	381	356	382
Sums:	663	1024	1256	700	829	840	538	861	818	908

Table B-E3. Unique Slot-Frame-Filler Triples (Facts), Summary, Part 1

Slot \ Sample	Work	Test	Both	Projected
absence_of	8	8	16	279
affected	11	1	12	209
analogue	14	6	20	349
appearance	9	12	21	366
cause	2	8	10	174
color	21	23	44	767
dimension	52	50	102	1779
dryness	2	4	6	105
effect	12	13	25	436
equi_measure	2	10	12	209
era	11	6	17	296
essence	50	49	99	1726
example	16	6	22	384
firmness	6	9	15	262
fit	3	-	3	52
flavor	2	-	2	35
form	11	10	21	366
frequency	7	5	12	209
geo_location	15	23	38	663
group	8	11	19	331
habitat	3	-	3	52
hazard	11	11	22	384
ingredient	13	5	18	314
larger_whole	35	35	70	1221
local_effect	7	6	13	227
location	81	77	158	2755
made_from	24	12	36	628
maturity	6	6	12	209
members	8	11	19	331
motility	3	2	5	87
movement_of	2	-	2	35
non_prototype	7	2	9	157
number	19	23	42	732
part	34	28	62	1081
pitch	4	1	5	87
potential_use	1	2	3	52
produced_by	16	13	29	506
product	8	4	12	209
scope	8	4	12	209
sex	6	10	16	279
shape	35	35	70	1221
size	36	71	107	1866
smell	-	1	1	17
sound	8	6	14	244
speed	7	6	13	227
strength	16	18	34	593
substance	195	200	395	6888
synonym	170	173	343	5981

Table B-E4. Unique Slot-Frame-Filler Triples (Facts), Summary, Part 2

Slot \ Sample	Work	Test	Both	Projected
taste	6	1	7	122
temperature	2	1	3	52
texture	4	10	14	244
topic	26	19	45	785
treatment	107	122	229	3993
type_of	952	1043	1995	34787
typical_action	84	116	200	3487
typical_food	3	1	4	70
typical_use	115	113	228	3976
typical_user	15	11	26	453
value	2	4	6	105
virtue_neg	21	17	38	663
virtue_pos	4	5	9	157
viscosity	2	-	2	35
wealth	1	-	1	17
weight	9	9	18	314
Classified:	2378	2488	4866	84848
attribute	205	297	502	8753
uncl_appos	77	81	158	2755
uncl_descr	1444	1467	2911	50759
Unclassified:	1726	1845	3571	62267
Sums:	4104	4333	8437	147116

Table B-E5. Unique Slot-Frame-Filler Triples (Facts), Frequency Analysis, Part 1

Sample	Work	Test	Both	Percent
3 slots:	1317	1416	2733	56.17
7 slots:	525	598	1123	23.08
10 slots:	246	233	479	9.84
26 slots:	241	209	450	9.25
18 slots:	49	32	81	1.66
64 slots:	2378	2488	4866	100.00

Slot \ Sample	Work	Test	Both	Percent
type_of	952	1043	1995	41.00
substance	195	200	395	8.12
synonym	170	173	343	7.05
3 slots:	1317	1416	2733	56.17
treatment	107	122	229	4.71
typical_use	115	113	228	4.69
typical_action	84	116	200	4.11
location	81	77	158	3.25
size	36	71	107	2.20
dimension	52	50	102	2.10
essence	50	49	99	2.03
7 slots:	525	598	1123	23.08
larger_whole	35	35	70	1.44
shape	35	35	70	1.44
part	34	28	62	1.27
topic	26	19	45	0.92
color	21	23	44	0.90
number	19	23	42	0.86
geo_location	15	23	38	0.78
virtue_neg	21	17	38	0.78
made_from	24	12	36	0.74
strength	16	18	34	0.70
10 slots:	246	233	479	9.84

Table B-E6. Unique Slot-Frame-Filler Triples (Facts), Frequency Analysis, Part 2

Slot \ Sample	Work	Test	Both	Percent
produced_by	16	13	29	0.60
typical_user	15	11	26	0.53
effect	12	13	25	0.51
example	16	6	22	0.45
hazard	11	11	22	0.45
appearance	9	12	21	0.43
form	11	10	21	0.43
analogue	14	6	20	0.41
group	8	11	19	0.39
members	8	11	19	0.39
ingredient	13	5	18	0.37
weight	9	9	18	0.37
era	11	6	17	0.35
absence_of	8	8	16	0.33
sex	6	10	16	0.33
firmness	6	9	15	0.31
sound	8	6	14	0.29
texture	4	10	14	0.29
local_effect	7	6	13	0.27
speed	7	6	13	0.27
affected	11	1	12	0.25
equi_measure	2	10	12	0.25
frequency	7	5	12	0.25
maturity	6	6	12	0.25
product	8	4	12	0.25
scope	8	4	12	0.25
26 slots:	241	209	450	9.25
cause	2	8	10	0.21
non_prototype	7	2	9	0.18
virtue_pos	4	5	9	0.18
taste	6	1	7	0.14
dryness	2	4	6	0.12
value	2	4	6	0.12
motility	3	2	5	0.10
pitch	4	1	5	0.10
typical_food	3	1	4	0.08
fit	3	-	3	0.06
habitat	3	-	3	0.06
potential_use	1	2	3	0.06
temperature	2	1	3	0.06
flavor	2	-	2	0.04
movement_of	2	-	2	0.04
viscosity	2	-	2	0.04
smell	-	1	1	0.02
wealth	1	-	1	0.02
18 slots:	49	32	81	1.66

Table B-F1. Accuracy Review,
Unique Slot-Frame-Filler Triples (Facts), Details

Sample	0	1	2	3	4	5	6	7	8	9
7 slots:	67	142	184	83	91	130	79	96	104	147
10 slots:	23	79	103	30	44	44	21	47	55	33
10 slots:	0	0	6	3	6	2	5	0	2	0
27 slots:	90	221	293	116	141	176	105	143	161	180

Slot \ Sample	0	1	2	3	4	5	6	7	8	9
treatment	14	28	37	16	14	18	18	21	24	39
typical_use	22	32	36	17	15	22	23	19	19	23
typical_action	14	23	20	14	25	31	16	23	9	25
location	7	14	27	15	19	25	8	9	20	14
size	1	26	14	12	6	14	7	7	8	12
dimension	4	10	23	2	4	8	3	7	18	23
essence	5	9	27	7	8	12	4	10	6	11
7 slots:	67	142	184	83	91	130	79	96	104	147
larger_whole	2	6	6	9	12	6	4	10	11	4
shape	1	13	20	7	-	6	1	6	13	3
part	-	10	11	-	10	5	2	9	11	4
topic	6	2	9	5	4	7	2	2	5	3
color	1	11	9	3	7	4	1	-	3	5
number	1	6	7	1	2	3	5	10	4	3
geo_location	2	11	10	1	1	5	2	5	-	1
virtue_neg	3	9	10	3	5	3	-	1	3	1
made_from	3	6	17	-	-	-	2	2	2	4
strength	4	5	4	1	3	5	2	2	3	5
10 slots:	23	79	103	30	44	44	21	47	55	33
typical_food	-	-	-	1	1	-	1	-	1	-
fit	-	-	-	-	1	-	2	-	-	-
habitat	-	-	1	-	1	-	-	-	1	-
potential_use	-	-	-	1	1	1	-	-	-	-
temperature	-	-	-	-	-	1	2	-	-	-
flavor	-	-	2	-	-	-	-	-	-	-
movement_of	-	-	1	-	1	-	-	-	-	-
viscosity	-	-	2	-	-	-	-	-	-	-
smell	-	-	-	1	-	-	-	-	-	-
wealth	-	-	-	-	1	-	-	-	-	-
10 slots:	0	0	6	3	6	2	5	0	2	0

Table B-F2. Accuracy Review,
Unique Slot-Frame-Filler Triples (Facts), Summary

Sample	Work	Test	Both	Projected
7 slots:	525	598	1123	19582
10 slots:	246	233	479	8352
10 slots:	19	5	24	418
27 slots:	790	836	1626	28353

Slot \ Sample	Work	Test	Both	Projected
treatment	107	122	229	3993
typical_use	115	113	228	3976
typical_action	84	116	200	3487
location	81	77	158	2755
size	36	71	107	1866
dimension	52	50	102	1779
essence	50	49	99	1726
7 slots:	525	598	1123	19582
larger_whole	35	35	70	1221
shape	35	35	70	1221
part	34	28	62	1081
topic	26	19	45	785
color	21	23	44	767
number	19	23	42	732
geo_location	15	23	38	663
virtue_neg	21	17	38	663
made_from	24	12	36	628
strength	16	18	34	593
10 slots:	246	233	479	8352
typical_food	3	1	4	70
fit	3	-	3	52
habitat	3	-	3	52
potential_use	1	2	3	52
temperature	2	1	3	52
flavor	2	-	2	35
movement_of	2	-	2	35
viscosity	2	-	2	35
smell	-	1	1	17
wealth	1	-	1	17
10 slots:	19	5	24	418

Table B-G1. Accuracy Review,
Wrong Fact Counts, Details

Sample	0	1	2	3	4	5	6	7	8	9
7 slots:	12	15	16	13	10	18	4	14	7	12
10 slots:	2	6	2	2	2	6	0	7	2	2
10 slots:	0	0	0	1	0	1	0	0	0	0
27 slots:	14	21	18	16	12	25	4	21	9	14

Slot \ Sample	0	1	2	3	4	5	6	7	8	9
treatment	5	5	4	3	4	7	1	7	-	5
typical_use	4	1	2	-	1	4	3	1	4	2
typical_action	1	3	5	5	4	4	-	3	1	2
location	2	5	3	4	-	3	-	3	2	2
size	-	-	-	-	-	-	-	-	-	-
dimension	-	-	-	-	-	-	-	-	-	-
essence	-	1	2	1	1	-	-	-	-	1
7 slots:	12	15	16	13	10	18	4	14	7	12
larger_whole	-	-	-	1	-	-	-	-	-	-
shape	1	1	1	-	-	2	-	1	1	-
part	-	-	-	-	2	1	-	1	-	-
topic	-	2	1	1	-	3	-	-	1	1
color	-	1	-	-	-	-	-	-	-	1
number	1	1	-	-	-	-	-	5	-	-
geo_location	-	1	-	-	-	-	-	-	-	-
virtue_neg	-	-	-	-	-	-	-	-	-	-
made_from	-	-	-	-	-	-	-	-	-	-
strength	-	-	-	-	-	-	-	-	-	-
10 slots:	2	6	2	2	2	6	0	7	2	2
typical_food	-	-	-	-	-	-	-	-	-	-
fit	-	-	-	-	-	-	-	-	-	-
habitat	-	-	-	-	-	-	-	-	-	-
potential_use	-	-	-	1	-	-	-	-	-	-
temperature	-	-	-	-	-	1	-	-	-	-
flavor	-	-	-	-	-	-	-	-	-	-
movement_of	-	-	-	-	-	-	-	-	-	-
viscosity	-	-	-	-	-	-	-	-	-	-
smell	-	-	-	-	-	-	-	-	-	-
wealth	-	-	-	-	-	-	-	-	-	-
10 slots:	0	0	0	1	0	1	0	0	0	0

Table B-G2. Accuracy Review,
Wrong Fact Counts and Right Fact Percentages, Summary

Sample	Work	Right	Test	Right	Both	Right
10 slots:	8	96.7	23	90.1	31	93.5
10 slots:	0	100	2	60.0	2	91.7
27 slots:	57	92.8	97	88.4	154	90.5

Slot \ Sample	Work	Right	Test	Right	Both	Right
treatment	14	86.9	27	77.9	41	82.1
typical_use	14	87.8	8	92.9	22	90.4
typical_action	11	86.9	17	85.3	28	86.0
location	7	91.4	17	77.9	24	84.8
size	-	100	-	100	-	100
dimension	-	100	-	100	-	100
essence	3	94.0	3	93.9	6	93.9
7 slots:	49	90.7	72	88.0	121	89.2
larger_whole	-	100	1	97.1	1	98.6
shape	3	91.4	4	88.6	7	90.0
part	2	94.1	2	92.9	4	93.5
topic	2	92.3	7	63.2	9	80.0
color	-	100	2	91.3	2	95.5
number	1	94.7	6	73.9	7	83.3
geo_location	-	100	1	95.7	1	97.4
virtue_neg	-	100	-	100	-	100
made_from	-	100	-	100	-	100
strength	-	100	-	100	-	100
10 slots:	8	96.7	23	90.1	31	93.5
typical_food	-	100	-	100	-	100
fit	-	100	-	--	-	100
habitat	-	100	-	--	-	100
potential_use	-	100	1	50.0	1	66.7
temperature	-	100	1	0	1	66.7
flavor	-	100	-	--	-	100
movement_of	-	100	-	--	-	100
viscosity	-	100	-	--	-	100
smell	-	--	-	100	-	100
wealth	-	100	-	--	-	100
10 slots:	0	100	2	60.0	2	91.7

Appendix C

Results from the Entire LDOCE

After the test run reported in Chapter 7 and Appendix B, which used only small samples of the LDOCE as input, the SIV program was finally used to process the entire LDOCE, and this appendix covers the results obtained. The main purpose of this longer run was to demonstrate the robustness of the SIV program. Chapter 7 should be read prior to study of the tables given in this appendix, as well as those in Appendix B.

There has been no effort to use the results in any other project, and SIV does not include any information retrieval features that would allow convenient browsing of facts. However, if the facts were all asserted within Prolog using the **createdb** and **loaddb** predicates, then ordinary Prolog queries could be submitted in the usual way, and fortunately, "the VPI Prolog system has been extended to include the facilities one normally finds in a relational database: most importantly, B+ tree indexing" [Deighan and Roach: 40].

In spite of the great bulk of data in the LDOCE, the SIV preprocessor and parser did handle their input as a single file even in this case, but constraints on the size of a Prolog temporary file necessitated splitting the one file output by the parser into five files to be used as input to the knowledge extractor. To save time, the SIV option to exclude duplicate facts was not selected, though it was selected for the shorter test. That is why, on a given row of the last table in this appendix, the figure in the first column is so frequently equal or very close to the figure in the third column.

The SIV program reported the following CPU time usage for modules processing the whole LDOCE on the same computer that was used for the shorter test:

```
Preprocessor........... 52 m 41.48 s
Parser.............2 h 31 m  6.77 s
Extractor..........2 h 57 m 21.15 s
------------------------------------------
Total..............6 h 21 m  9.40 s
```

The total CPU time was actually about six hours and 46 minutes, because the preprocessor program does not include the time to run **sed** commands in the time usage it reports, and this hidden time was observed to be about 25 minutes.

Although some additional development effort was necessary to handle special cases that had not been encountered or accommodated during the test with the ten samples, the conclusions reached through careful examination of the results from that earlier test (and reported in Chapter 7) should remain largely valid. The required changes to the program were deemed too minor to justify a comprehensive analysis of the new results. Hence only figures that could be obtained automatically or readily calculated are included here without further comment.

Table C-A. Overview

	LDOCE	Percent
All Entries	41122	100.0
Noun Entries	23607	57.4
All Senses	67772	100.0
Noun Senses	38939	57.5
Non-abstract	36837	54.4
1000 all.raw	15957.7	100.0
Bytes: all.unf	5438.0	
all.fmt	4220.3	
all.db	7244.8	
Times: Format	3161	13.8
Parse	9067	39.6
Extract	10641	46.5
Seconds Total	22869	100.0
Bytes / Sec.		
Rates: Format	5048	
Parse	600	
Extract	397	
Form-Pars-Extr	698	
Senses / Sec.		
Rates: Format	21.4	
Parse	4.3	
Extract	3.5	
Form-Pars-Extr	1.6	
Senses: Total	38939	100.0
Variant	1307	3.4
Abstract	2102	5.4
Mixed-Function	671	1.7
Senses / Noun	1.65	
Classif. Facts	82080	56.0
Unclass. Facts	64525	44.0
Distinct Slots	77	
Cl.F. / Sense	2.23	
simpl G cmplx	10863	27.9
simpl G -	3572	9.2
- G cmplx	19664	50.5
- G -	4840	12.4

Table C-B1. Slot Counts, Part 1

Slot	Unique Slot-Frame Pairs	Unique Slot-Detail Frame Pairs	Unique Slot-Frame Filler Triples
ability	48	-	48
absence of	512	7	517
affected	109	-	109
color	820	227	839
dimension	1660	319	1770
dryness	76	20	76
effect	265	16	265
equi measure	63	-	63
era	243	4	243
essence	1991	-	1991
example	144	-	255
firmness	390	57	390
fit	74	7	74
flavor	5	-	5
form	282	31	282
frequency	141	5	141
geo location	804	23	806
group	458	-	458
habitat	58	-	58
hazard	304	14	308
ingredient	368	8	368
larger whole	873	5	873
local effect	93	-	93
location	2534	40	2535
made from	857	91	861
maturity	257	21	257
members	458	-	458
motility	109	2	109
movement of	56	-	56
non prototype	219	-	219
number	466	245	467
part	1145	4	1149
pitch	35	1	36
place of use	22	-	22
position of	63	-	63
potential use	59	-	59
produced by	428	1	428
product	144	2	145
scope	283	23	283
sex	197	14	197
shape	1049	122	1073
size	2153	167	2155
smell	57	17	58
sound	162	4	165

Table C-B2. Slot Counts, Part 2

Slot	Unique Slot-Frame Pairs	Unique Slot-Detail Frame Pairs	Unique Slot-Frame Filler Triples
speed	149	6	150
strength	379	31	380
substance	7678	-	7678
synonym	3457	-	3627
taste	190	25	197
temperature	47	16	47
tension	7	5	7
texture	205	28	207
topic	748	-	748
treatment	4497	154	4513
type of	30723	3	33049
typical action	4066	4	4067
typical food	71	-	71
typical use	4070	7	4071
typical user	430	15	430
value	94	8	95
virtue neg	410	9	417
virtue pos	347	26	350
viscosity	33	1	33
wealth	29	1	29
weight	284	24	284
Classified:	79248	1917	82080
attribute	8930	1418	10170
uncl appos	2826	-	2826
uncl descr	51529	4	51529
Unclassified:	63285	1422	64525
dialect	2299	14	
function	22723	1	
level	30589	1	
sense	8370	1	
usage	4604	2	
variant	1307	-	
Name Frames:	69892	19	
Sums:	212425	3358	146605[*]

[*] Name frame facts are not included in the sum for this column (unique slot-frame-filler triples).

Appendix D

Codes in Raw Input Files

Chapter 7 mentions that a complete list of codes for typesetting and special characters was not available when SIV was implemented. This appendix reports all codes observed or recognized in the unprocessed (raw) LDOCE file along with the interpretation that was attributed to them in the program.

Table D-A shows the substitutes that were used in the preprocessed output files (not in the raw files) to represent special characters.

Table D-A. Special Character Substitutes

Substitute	Special Character	Substitute	Special Character
e%	e with acute accent	/	, (comma)
a`	a with grave accent	@	. (period)
e`	e with grave accent	%	' (apostrophe)
a^	a with circumflex accent	%%	" (double quote)
e^	e with circumflex accent	::	; (semicolon)
i:	i with diaeresis	||	! (exclamation mark)
c5	c with cedilla		

The "use" column in Table D-B shows the character or characters substituted for each recognized code handled through **sed** commands in the preprocessor. Square brackets enclose these codes and their substitute to reveal the position of surrounding spaces. The same column in Table D-C shows the interpretation given to all other recognized codes.

Table D-B. Codes in Raw Files Handled by **sed** Commands

Code	Use	Code	Use	Code	Use
[!']	[%]	[!1st]	[first]	[@0]	[fromto 0]
[!"]	[%%]	[!2nd]	[second]	[@1]	[fromto 1]
[!.]	[@]	[!3rd]	[third]	[@2]	[fromto 2]
[!,]	[/]	[!4th]	[fourth]	[@3]	[fromto 3]
[!;]	[::]	[!5th]	[fifth]	[@4]	[fromto 4]
[!!]	[||]	[!6th]	[sixth]	[@5]	[fromto 5]
[!|]	[|]	[!7th]	[seventh]	[@6]	[fromto 6]
[!+]	[+]	[!8th]	[eighth]	[@7]	[fromto 7]
[*31]	[e%]	[!9th]	[ninth]	[@8]	[fromto 8]
[*67]	[ae]	[!0]	[0]	[@9]	[fromto 9]
[*78]	[ ]	[!1]	[1]	[< -]	[< -]
[*B1]	[a`]	[!2]	[2]	[-]	[-]
[*B2]	[a^]	[!3]	[3]	[-)]	[-)]
[*B3]	[e`]	[!4]	[4]		
[*B4]	[e^]	[!5]	[5]		
[*B6]	[c5]	[!6]	[6]		
[*B8]	[i:]	[!7]	[7]		
		[!8]	[8]		
		[!9]	[9]		

Table D-C. Other Codes in Raw Files

Code	Use	Code	Use
*25	left bracket	*63	font change
*32	right bracket	*64	font change
*3E	start special symbols	*80	decimal, syllable separator
*3F	stop special symbols	*9F	one quarter symbol
*44	font change	*A0	one half symbol
*45	font change	*A1	three quarters symbol
*46	font change	*CA	start all caps
*53	degree symbol	*CB	stop all caps
*54	less than symbol	*CC	start pronunciation
*55	greater than symbol	*CD	font change
*58	pound sterling	!<	field separator

In addition, a number preceding a comma preceding a single zero was interpreted as an abbreviation for a group of three zeros. For example, 7 !, 0 !, 0 would be interpreted as 7,000,000 (seven million). Codes in the range *8A to *8F were ignored. Perhaps they are used to specify the sense number of a word used within a definition text.

Vita

Thomas James Godfrey was born on June 4, 1947, in Alexandria, Louisiana, the son of Paul Russell Godfrey, of southern New Jersey, and Mary Elizabeth Thompson Godfrey, of northwestern Missouri. He was reared and educated in Pineville, Louisiana, where he received a diploma from Pineville High School in May 1965, and both B.S. and B.A. degrees from Louisiana College in May 1969. His majors in college were mathematics and French/German, and his minors were physics and Latin, respectively.

In September 1973, after two and a half years of service in the United States Army, Godfrey began graduate studies in linguistics at the University of Texas at Austin, finishing there with a Ph.D. in December 1981. In 1975 he joined the Summer Institute of Linguistics (SIL) and Wycliffe Bible Translators (WBT) and married Beth Richards of Barker, New York. While in training for those organizations, Godfrey also took courses offered by SIL and accredited by The University of Texas at Arlington (1974-75) and the University of Oklahoma (1982).

Godfrey then served with SIL and WBT in Guatemala, working mainly on an extensive dialect survey of the Mam area and assisting with an Old Testament translation project for the Northern Mam people. During his furlough in 1988-89, Godfrey was on the staff of the International Computer Services Department of JAARS, Inc., in Waxhaw, North Carolina, working on dialect adaptation software.

Since moving to Blacksburg, Virginia, in June 1989, Godfrey has been studying computer science at Virginia Polytechnic Institute and State University. He is now employed by Industrial Computing Designs Corp. Thomas and Beth have four children: Rachel, Paul, Charis, and David.

Thomas James Godfrey