BUILDING A LARGE THESAURUS FOR INFORMATION RETRIEVAL

Edward A. Fox** and J. Terry Nutter
Department of Computer Science
Virginia Tech, Blacksburg VA 24061

Thomas Ahlswede and Martha Evens
Computer Science Department
Illinois Institute of Technology, Chicago IL 60616

Judith Markowitz
Navistar International
Oakbrook Terrace, IL 60181

ABSTRACT

Information retrieval systems that support searching of large textual databases are typically accessed by trained search intermediaries who provide assistance to end users in bridging the gap between the languages of authors and inquirers. We are building a thesaurus in the form of a large semantic network to support interactive query expansion and search by end users. Our lexicon is being built by analyzing and merging data from several large English dictionaries; testing of its value for retrieval is with the SMART and CODER systems.

1. Introduction

Though computer systems aiding retrieval from bibliographic or full-text databases have been available for more than two decades, it is only in recent years that many people are becoming concerned about the serious limitations of those systems regarding effectiveness in finding desired references (Blair and Maron 1985). Like others, though, we are convinced that easy-to-apply automatic methods can help solve this problem (see argument in Salton 1986). Indeed, automatic approaches seem essential since many end-users want to search without involving trained intermediaries (Ojala 1986). However, since the fundamental issue is one of mismatch between language use of document authors and inquirers, leading to uncertainties regarding whether a particular item should be retrieved (Chen and Dhar 1987), we are also convinced that computational linguistics is essential for a complete solution.

Since many queries are simply sets of lexemes or phrases, and all queries can be reduced to that form, we believe that focusing on lexical and phrasal issues may be the most appropriate strategy for applying computational linguistics to information retrieval. While good results have been achieved in applying automatic procedures to find lexically related words based on local context in a particular collection (Attar and Fraenkel 1977), no reliable techniques exist for using collection statistics to build a thesaurus automatically, and manual construction of thesauri is both problematic and expensive (Svenonius 1986). Efforts to represent a significant portion of common sense knowledge will take years (Lenat et al. 1986), but developing knowledge resources to aid text processing is now feasible (Walker 1985). Encouraged by activities in identifying lexical and semantic information in other languages (especially since that reported in Apresyan et al. 1970), we decided to build a large, comprehensive lexicon for English that would include lexical and semantic relations (see comments below and survey in Evens et al. 1980) and be of use for information retrieval. Clearly this might also help with question answering (Evens and Smith 1978).

Lexical and semantic relations provide a formal means for expressing relationships between words and concepts. Most commercial dictionaries give explicit mention to the classical relations, synonymy and antonymy, but many other relations are used in the definitions. Perhaps the most common is taxonomy, the relation between a word and its genus term, as in *lion – animal*. Many noun definitions also contain the part-whole relation. Grading and queuing relations help describe words that come in sequences like *Monday – Tuesday – Wednesday* and *hot – warm – cool – cold*. Collocation relations are used to express the relationships between words that cooccur frequently like *hole* and *dig*. As a basis for automated thesaurus construction, we are trying to extract from machine-readable dictionaries triples consisting of words or phrases linked by a labelled arc representing the relation. We plan to include phrases as well as words for both practical and theoretical reasons. It is well known that thesauri that include phrases are much more effective than those without, and we are believers in the phrasal lexicon described by Becker (1975).

Our approach is first (see section 2) to apply text processing methods to machine readable dictionaries (Amsler 1984); next (see section 3) to analyze definitions from those dictionaries; then (see section 4) to merge that information (along with data in a large synonym file made available by Microlytics Inc.) into a large semantic network (see early work in Quillian 1968, and survey in Ritchie and Hanna 1984); and finally (see section 5) to test the utility of the resulting thesaurus using the SMART and CODER

* This material is based in part upon work supported by the National Science Foundation under Grant No's. IST-8418877, IST-8510069, and IRI-8703580; by the Virginia Center for Innovative Technology under Grant No. INF-85-016; and by AT&T equipment contributions.

**All correspondence regarding this paper should be addressed to the first author.

experimental retrieval systems. We discuss preliminary results of all these aspects of our research program.

2. Dictionary text processing

Since we wish to build a thesaurus with broad coverage, and since each dictionary has unique advantages, we plan to use several. The bulk of our work to date has been with the *Collins Dictionary of the English Language* (CDEL) and *Webster's Seventh New Collegiate Dictionary* (W7).

2.1. Collins Dictionary of the English Language

In 1985 we obtained a magnetic tape containing the typesetter's form of CDEL from the Oxford Text Archive and embarked upon the task of converting that to a fact base in the form of Prolog relations (Wohlwend 1986). CDEL is a large dictionary with about 82,000 headwords. Thus it is larger than W7 (which has roughly 69,000 headwords), and also has separate fields not found in W7, such as sample usages (roughly 17,000), first names, compare-to lists, related adjectives, and abbreviations. Like W7, there are on average two definitions per headword, though while W7 has at most 26 senses per entry there are many words in CDEL with more senses, on up through one entry with 50 senses.

Wohlwend used various UNIX (trademark of AT&T Bell Laboratories) text processing tools and developed analyzers with *lex* and *yacc*. The main processing involved nine passes through the data by our analyzers, with a small amount of manual checking and correction between steps. By the fall of 1986 Wohlwend, France and Chen had extracted all suitable data from the CDEL tape, placed it in the form of facts that could be loaded into a Prolog system, and collected statistics regarding occurrences (Fox et al. 1986). Valuable information was present to aid in our work with the CODER "expert" information retrieval system (France and Fox 1986). Recently J. Weiss has completed manual checking and editing of the data, and has refined some of the automatic analysis (e.g., separating past tense forms from other irregulars). Later we will load the bulk of that into a semantic network and carry out further processing on the definition portions.

2.2. Webster's Seventh Collegiate Dictionary

We received our machine-readable W7 from Raoul Smith on five tapes in Olney's original format (Olney 1968). Our first piece of text processing was to compress this huge mass of data (approximately 120 megabytes) into a manageable format. Ahlswede wrote a C program *compress* which converted the tape data into a format based on that of Peterson (1982), with a few differences to simplify our analysis. The resulting version occupied 15,676,249 bytes.

The synonymy relation is particularly easy to recognize, since it is explicitly tagged — in the printed dictionary synonyms appear in small capitals. Thus the first step in adding relations to our lexical database was to extract the 45,910 synonymy relationships marked explicitly in this way using the UNIX *awk* utility and insert them in a table of word-relation-word triples (Ahlswede 1985).

Morphology also provides an analytical tool for the extraction of relations. One fruitful source is the prefix lists of words beginning with *non-*, *re-*, and *un-* that are printed in the dictionary but are never defined. These were left out of the Olney version of the dictionary, but one of our colleagues, Sharon King, typed in the lists and wrote routines that generate definitions for these words, e.g.,

readjust	vt	to adjust again
redefinition	n	the act or process of defining again
reexporter	n	that which exports again
unflattering	aj	not flattering

For many of these words it is possible to derive relational information automatically also:

redefinition	ACT	redefine
redefine	REPEAT	define
reexporter	AGENT	reexport
unflattering	ANTI	flattering

Almost twenty percent of the words and phrases presented in W7 do not have main entries of their own but appear as "run-ons" at the foot of other entries. Most of the run-ons are formed by adding productive suffixes to the main entry, such as *-ness*, *-able*, and *-ly*. Fortunately, the suffixes themselves are often defined in entries of their own that make clear how the root and the derived word are related. Word-relation-word triples can easily be generated between words derived using these suffixes and their roots.

We have long agreed with Becker's (1975) assertion that many phrases are best treated as lexical units. We were delighted to find that the editors of W7 apparently agree: out of 68,656 main entries, 8,878 are phrases of more than one word (separated by spaces). Another student is currently analyzing phrasal entries in W7, and looking for systematic relationships.

3. Analysis of Definitions

3.1. Preprocessing for the parser

The definition texts proper are the largest and (to us) the most interesting part of dictionary entries. Considerable preprocessing was necessary in order to prepare the W7 definition texts for parsing with the LSP. The Peterson/Ahlswede format uses semicolons as delimiters to separate several fields in a definition text. Since the LSP treats the semicolon as a word, we replaced the semicolon delimiters with spaces. The entry word, homograph number, sense number, and part of speech were necessary to identify the sense being defined and must be part of the text input to the LSP. Texts longer than 72 characters had to be broken up into multiple lines and texts had to be converted to all upper case. Finally, each text had to end with a period (separated from the last word by a space).

A shell script, *LSPformat*, used the UNIX utilities *sed* and *tr* to perform all of these operations except the breaking of long lines. Since no UNIX utility performed line breaks in quite the way we needed, we wrote a C program *foldx* which did. However, we have not made much use of *LSPformat* because we have found it useful to combine preprocessing of definition texts with the solution to another problem.

3.2. The LSP Word Dictionary

The LSP uses a lexicon of its own, the Word Dictionary, and cannot parse any text unless all words (strings separated by blanks) contained in the text are defined in the Word Dictionary. The Linguistic String Project has created a Word Dictionary of over 10,000 words (White 1983), but this cannot be used to parse even one W7 definition text without some additions.

In particular, W7's part of speech codes (*n*, *vi*, *vt*, *aj*, etc.) are absent from the Word Dictionary. These were easily added. The homograph and sense numbers were also absent. We found it convenient to generate a list of all the different homograph and sense numbers that appeared in W7 (there were 410) and enter them as "words" in the Word Dictionary. With these additions it became possible to parse a total of 8,832 of the 126,879 definition texts without adding any more entries to the Word Dictionary.

Identification of this subset required another special program, *showsubset*, which steps through all of the W7 definition texts, comparing each word with a list of entries in the Word Dictionary. It prints those texts out again, each word in upper case if it appears in the Word Dictionary and in lower case if it does not. A shell script, *showsubset.com*, combines the text processing functions of *LSPformat* with a call to *showsubset* to generate a complete set of definition texts in this mixed upper and lower case format. Definition texts consisting entirely of upper case words are then ready to parse.

Further enlarging the Word Dictionary to include the entire defining vocabulary (as well as all the entry words) in W7 is a major effort, still in the early stages. There are two kinds of words occurring in W7 for which we need only trivial Word Dictionary entries. These are (1) words defined but not used in definitions, and (2) proper names, of which the most numerous and easiest to identify are scientific names. The LSP definition grammar requires no information about the word or phrase in the entry-word position except its part of speech (and it needs this only to avoid creating numerous "junk" parses). All proper names have a simple, fixed set of Word Dictionary parameters, thus their Word Dictionary entries are all identical except for the names themselves. Adding these two categories to the Word Dictionary would allow parsing of 29,692 definitions.

We have not yet added these words to the Word Dictionary because up to now we have concentrated exclusively on development of the definition grammar based on the original 8,832 word subset. There are logistical problems involved in dealing with a much larger Word Dictionary; furthermore there is at least one major and distinctive type of definition text (those involving biological taxonomy) missing from the subset.

3.3. Determining what relations to look for in definitions

One relation, taxonomy, is at least formally present in virtually all definitions. A definition text consists of a phrase (sometimes a very long one) consisting of one or more head words plus zero or more modifying phrases. The head word is a taxonomic superordinate of the word being defined. Sometimes the taxonomy relation is intuitively clear and useful:

> hole 0 1 n an *opening* into or through a thing
> curtain 1 1 n a hanging *screen* usu. capable of being drawn . . .

Sometimes the formal taxonomy is obscure or of little semantic value, and the important relations are to modifying words or phrases:

> customer 0 1a n *one* that purchases a commodity or service . . .
> (customer AGENT purchase)
> aged 0 1a aj *of* an advanced age
> (aged JSIMPL age)

Other times the head word is best understood as a pointer to another relation:

> baptistery 0 0 n a *part* of a church . . . used for baptism
> (baptistery PART church)
> agglutinate 2 1 vt to *cause* to adhere
> (agglutinate v-v-CAUSE adhere)

Computational study of dictionary definitions has focused heavily on taxonomy because it can often be identified fairly reliably without parsing the definitions (Amsler 1980, Chodorow et al. 1985). Our emphasis in identifying relations, whether for information retrieval or for other purposes, has been on relations other than taxonomy (Evens et al. 1987).

An important tool for identifying relations is the phrase count. A frequently occurring phrase, especially at the beginning of a definition, is often a "defining formula" (Smith 1985, Ahlswede and Evens 1987) that marks a relation.

Another useful tool was very simplified hand parsing of definitions. This consisted of manually identifying the headword(s) and blocking off the "left adjuncts" and "right adjuncts" (to use LSP terminology) of the headwords. For noun definitions, this was a very productive procedure; the right adjuncts of noun headwords, in particular, included many stereotyped phrase structures which could be associated with relations. Adjuncts in verb definitions were more varied and of less help in identifying relations.

3.4. Developing a grammar

The grammar of dictionary definitions is based on the grammar provided in Sager (1981). The great majority of changes to that grammar have been of two types:

1. Adaptations to the top-level syntax of W7. The LSP grammar is based on a medical sublanguage consisting of complete declarative sentences, as well as questions, imperatives, and some other sentence types. The texts of W7 are never complete sentences; rather they are phrases syntactically parallel to the words they define: noun phrases for nouns, infinitive verb phrases for verbs, etc.

It would be possible in principle to use existing LSP phrase structures to represent most definition texts, although some enhancement of the grammar would still be necessary to include the entry words, homograph and sense numbers, and part-of-speech symbols, as well as some syntactic peculiarities like the zeroed direct object which is part of every transitive verb definition. However, we have chosen, for greater ease of finding relational arcs as well as to make the grammar more efficient, to define special phrase structures for the various definition types. Shown below is part of the grammar for noun definitions:

```
<NDEF>       %%= <NDEF-HEAD> <NDEF-RN> .
<NDEF-HEAD>  %%= <PROHEAD> / <NOUNHEAD> .
<PROHEAD>    %%= ANY / ONE / SOMETHING .
<NOUNHEAD>   %%= <LN> <NVAR> .
<NDEF-RN>    %%= <OFX> / <PVINGO> / <PN> /
         <ASX> / <ASOFX> / <ASDFROMX> /
         <ADJINRN> / <WHETHS> / <PWHS> /
         <THATS-N> / <TOVO> / <TOBE> /
         <VENPASS> / -<VENO> / <VINGO> /
         <WHS-N> / NULL .
<OFX>        %%= <VN-P> <NDEF> . [note recursion]
<ASX>        %%= '(' AS <NSTG> ')' <NDEF-RN> .
<ASOFX>      %%= '(' AS <PN> ')' <NDEF-RN> .
<ASDFROMX>   %%= AS DISTINGUISHED FROM <NSTG> .
<VN-P>       %%= OF / FOR .
```

2. Routines to identify and print out relational arcs. The following is an example of a routine which identifies the formula *used to/in/for* and prints out a relational triple (X) instr (Y):

```
T-W-N-USE = IN VENPASS%
     IF IMMEDIATE-NODE IS NDEF-RN  THEN $1.
 $1 = DESCEND TO LVENR; STORE IN X1;
      AT COELEMENT PASSOBJ X2 IF $USE-FORMULA
                              THEN $N-N-USE.
  $USE-FORMULA = BOTH X1 SUBSUMES 'USE'
      AND X2 SUBSUMES 'TO' OR 'IN' OR 'FOR'.
 $N-N-USE = DESCEND TO VERB OR NSTGO OR VINGO
     PASSING THROUGH STRING; DO GET-DWORD;
     WRITE ' instr '; DO WRITECORE; WRITE END
     OF LINE.
```

A few miscellaneous changes have also been made. For example, adverbs are used more freely in W7 than in the LSP grammar's sublanguage. Nouns appear more often without an article, and transitive verbs are more often used intransitively. The most spectacular difference between the sublanguages at this level is that W7 uses the conjunction *or* with a freedom that is barely if at all acceptable in standard English, especially in adjective definitions:

> abbatial 0 0 aj of or relating to an abbot . . .
> abaxial 0 0 aj situated out of or directed away from the axis

3.5. Results of parsing

2949 noun definitions, 1451 adjective definitions, 1272 intransitive verb definitions, and 2549 transitive verb definitions were parsed. The LSP's performance was significantly different in these four categories:

Part of speech	Percent success	Parses per success	Time (sec.) per parse	Arcs generated
nouns	77.63	1.70	11.05	26225
adjectives	68.15	1.85	10.59	5393
int. verbs	64.62	1.59	11.96	5438
tr. verbs	60.29	1.50	43.33	14059
average/total	68.65	1.66	18.89	51115

The count of arcs generated includes duplications; the total number of unique arcs was 25,178. These included 5,086 taxonomies, 7,971 modification relations (e.g., the definition "puppy n a young dog" yielded the modification arc (puppy) mod (young)), and 125 other relations, in three principal categories.

The first category was "traditional" relations such as taxonomy, part-whole, etc., which we felt were amenable to axiomatic treatment. Parsing produced relatively few of these: 334 causation arcs, 232 part-whole arcs, and a few hundred others.

The second category was a set of recurring syntactic relations that we speculated would prove to have consistent semantic significance. Some of these were familiar case relations: there were 448 verbal nouns, 124 adjectives derived in one way or another from verbs, etc. This category also included, for example, relations such as "v-obj", the relation between a verb and the direct object of its defining headword.

The third category consisted of syntactic relations which we simply noted with the idea of later doing cluster analysis to determine selectional categories in the dictionary, much as described by Sager (1981). These included 2,694 "permissible modifiers", adjectives modifier-noun pairs; 182 "permissible subjects", nouns appearing as subjects of verbs; and so on.

Definitions which failed to parse did so for a great many reasons; we may be near the point of diminishing returns in terms of refining the grammar to parse every definition. As the third column indicates, many definitions yielded multiple parses. Multiple parses were responsible for most of the duplicate relational arcs that were generated.

The quality of these parses is an important issue. A "good" parse is one consistent with the way competent English readers would agree to interpret the text; flaws include acceptance of ungrammatical forms (which must be corrected by changing the grammar) or, more often, resolution of syntactic or semantic ambiguities in ways which the human reader can recognize as not intended. Quality analysis of the 8,910 parse trees is still at an early stage.

It is not clear why transitive verbs took so much time to parse; our best guess at this time is that this was caused by the difficulty of locating the zeroed direct object (see above).

3.6. Extracting relational triples by text processing

As the "time per parse" column suggests, parsing definitions is slow. (Total parsing time for our 8,832 definitions was 176 hours.) It is also intensive with respect to development effort. Although we have by no means given up on parsing as a powerful tool of analysis for dictionary definitions, it seems unsuited for the task of finding relational triples. Consequently, we have experimented with a text processing approach based on the identification of defining formulas; it yields more triples in far less time, and in many cases their quality is much better.

Our initial text processing effort involved isolating intransitive verb definitions containing three of the most common intransitive verb headwords: *become* (774 occurrences in 8,482 definitions), *make* (526 occurrences), and *move*. *Become*, in intransitive verb definitions, is almost invariably followed by an adjective — in a handful of cases it is followed by a noun (marked by the article *a*) or an adverb which, in turn, precedes an adjective. These exceptions can be easily weeded out of the 774 definition texts. Conjoined adjectives to the right of *become* are also

easily identified and an *awk* program expands the definitions containing them into pairs (triplets, etc.) of definitions. These are then reduced to triple form:

ablate 0 0 vi to become ablated
(ablate 0 0 vi) incep (ablated)

abort 0 2 vi to become checked in development so as to remain rudimentary or to shrink away
(abort 0 2 vi) incep (checked)

abound 0 2 vi to become copiously supplied
(abound 0 2 vi) incep (supplied)

addle 2 1 vi to become rotten
(addle 2 1 vi) incep (rotten)

age 2 2b vi to become mellow or mature
(age 2 2b vi) incep (mature)
(age 2 2b vi) incep (mellow)

By this means we extracted 2,076 triples representing five relations, which we called *incep* (from *become*), *vncause* (from *make* preceding a noun), move (from *move*), *vacause* (from *make* preceding an adjective), and *vnbe* (from several head verbs followed by *as* and a noun phrase — the verb *be*, though occurring 389 times as a headword, governed a variety of definition structures and therefore conveyed no consistent relational meaning). The whole process took about three hours.

We have also tried to extract taxonomies of intransitive verbs, assuming as a first approximation that the head verb constitutes a genus term. We have extracted 9,520 triples; but the quality of these was not as good as that of the other relations, since our head finding algorithm does not yet catch adverbial particles such as those in *give up* or *bring about*, or idiomatic direct objects as in *take place*.

We are now in the process of extracting relations by this process for nouns, adjectives, and transitive verbs. The improvement in performance time is less dramatic, particularly in the case of noun definitions, where postnominal phrases are much more common than in intransitive verb definitions. These phrases are harder to identify, so more manual intervention is necessary.

4. Constructing the semantic network

The SNePS semantic network system (Shapiro 1979b) is one of relatively few knowledge representation schemes that permit a unified representation of associative information and predicate-logic-style inference (for details on the logic, see Shapiro 1979a and Hull 1986) enhanced with default reasoning capability (Nutter 1983). In SNePS, every node represents a concept (concepts include lexemes, word senses, individuals, relations, propositions, and any other potential objects of the system's knowledge). Conversely, every concept is represented by a node, and no two nodes represent the same concept (although the concepts they represent may refer to the same object; for more on this and other principles underlying the design of SNePS, see Shapiro and Rapaport 1986). Logical and structuring information are carried on arcs. All explicit information about a concept is directly linked to its node, with structure sharing across propositions. It follows that every detected synonym of a given word sense, for instance, is connected to that sense by simply definable paths to the nodes representing the original word sense and the lexical relation SYNONYM. This simplifies finding related terms, eliminating much of the look-up necessary in schemes that are superficially more logic-like.

The lexicon we are forming contains three different kinds of information: lexical-semantic relations among specific word senses, axioms for lexical relations, and definitional assertions. Most of the semantic information is captured in the form of lexical relations between the sense being defined and some other word or word sense. Instances of lexical relations have a natural implementation in SNePS, with base nodes representing lexemes, word senses, and individual lexical-semantic relations and molecular nodes (non-base nodes labeled Mx for some integer x) representing propositions and rules about the base nodes. To give content to individual lexical relations, relation axioms are included for the various relations. For instance, the lexical relation CHILD has a single axiom which says that if an x is a child y, then any x is a y and any x is young. While most word sense meanings determined from definitions can be represented fully by lexical relations, occasionally some aspect of meaning will remain after full analysis of the lexical relations between this and other word senses. In that case the entry is completed by a definitional assertion.

The SNePS representations of instances of lexical relations, the relation axioms, and assertions completing definitions are not only compatible but linked, and hence can readily be combined. In addition, because these various kinds of information all have representations as SNePS subnetworks, they are all immediately available to the inferencing package (SNIP) without conversion or instantiation.

Thus in most cases the semantic representations of the different kinds of lexical information is straightforward, and has the advantage that all information is uniformly available to the inference system. There is a second inferential advantage which derives from the nature of the semantic network representation. There are two different kinds of inference available in semantic networks: rule-based inference and path-based inference. Rule-based inference, the more common form of inference in most SNePS applications, involves predicate-calculus-like reasoning based on the existence of rule nodes (nodes representing complex predicate calculus formulas). Insofar as SNIP operates like a (slightly exotic) notational variant of predicate logic, the inference in use is rule-based. In rule-based reasoning, the semantic information resides in the nodes; the arcs are used for accessing and to manage implementation. Path-based inference can be viewed as the precise inverse: inference which relies only on the existence of paths (that is, concatenations of arcs) from one node to another. Conceptually, rule-based inference represents conscious reasoning from principles, while path-based inference represents (unconscious) reasoning by traversing associational links. The most natural implementation for path-based reasoning is hierarchical inheritance, but it can be applied more generally, for example to locate synonyms. In choosing related terms for expanding retrieval queries, it turns out that path-based

reasoning is by far the more important. Path-based inference is more efficient than rule-based inference, because given a starting point it eliminates the need to search the network for unifying matches. That is, where path-based inference is possible the system does not have to look for rules which might apply; it need only traverse a very limited subgraph from a given starting point along a limited set of predefined paths.

As a simple example, consider the following definitions (quoted directly from the *Oxford Advanced Learner's Dictionary of Current English*; in each case, the definition given is the first sense for the first homonym; pronunciations, references and examples have been omitted).

sheep n (pl unchanged)	grass-eating animal kept for its flesh as food (mutton) and its wool.
wool n [U] 1	soft hair of sheep, goats, and some other animals . . .
ram n 1	uncastrated male sheep.
ewe n	female sheep.
lamb n 1 [C]	young of the sheep . . .

From these definitions, we extract (among others) the following lexical relations:

sheep	TAX	animal
sheep	TFOOD	grass
wool	PART	sheep
wool	PART	goat
wool	TAX	hair
ram	MALE	sheep
ewe	FEMALE	sheep
lamb	CHILD	sheep

A simple post-processor transforms each triple of the form *x R y* into a SNePS User Language command of the form "(build arg1 x rel R arg2 y)", which results in creating a molecular node representing the proposition that x bears the relation R to y, yielding the network shown in Figure 1 (simplified by omitting relation axioms and the network linking nodes representing lexemes to those representing a particular word sense). Since all SNePS arcs have inverses, simple arc traversals from the node for *sheep* will locate such related terms as *ewe*, *ram*, *lamb*, *wool*, *animal*, and so on. Likewise, starting from a query containing *wool*, the system can rapidly find *sheep*, etc. A more complete network, including other definitions, would allow going down as well as up the taxonomic tree, finding e.g. "merino" and other sheep varieties from either *sheep* or *wool*.

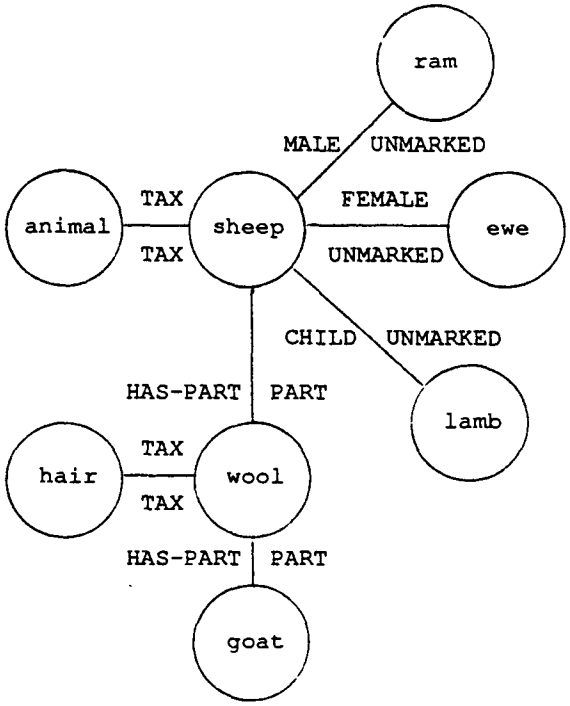

Figure 1. Representation of Lexical Relations involving *Sheep*

5. Testing with SMART and CODER

Several studies have been undertaken regarding the use of lexical and semantic relations in information retrieval. Though one investigation involved use of a special system constructed at IIT (Wang et al. 1985), most of the other work has involved the SMART system. The first version of SMART ran on IBM mainframes; a more modern form was developed to run under the UNIX operating system (Fox 1983b). In SMART, queries and documents are represented simply as sets of terms, so a multi-dimensional vector space can be constructed wherein each term is associated with a different dimension in that space. Queries and documents can then be associated with points in that space, and documents can be retrieved if "near" to the query. But since queries are typically short, it can be valuable to expand a query with terms related to the original set (especially due to variations in naming practices like those considered in Furnas et al. 1982).

In our first experiment, involving a small collection of 82 documents, we found a mild improvement in system performance when all types of related terms (except antonyms) were involved in query expansion (Fox 1980). Similar benefits resulted when using a different, larger collection (Evens et al. 1985). In two later studies we used SMART but worked with Boolean queries. Query expansion then involved "ORing" in related terms with the original ones. Once again, improvements resulted, especially when the p-norm scheme for interpreting Boolean queries was applied (Fox 1983a, 1984). In all of these studies, lexical-semantic relations were identified manually for all query terms that were expanded.

In other recent work with SMART, Fox, Miller and Sridaran used the same Boolean queries, but varied the

source of related words. They compared with the base case of original queries the results of the following sources for expansion: all words based on manually derived lexical-semantic relations, all words (except for antonyms) taken from the *Merriam Webster Thesaurus*, and all words (except those in a "stop" list) from the definition appearing in a dictionary for the correct word sense. All expansion schemes gave better results han the base case. While the lexical-semantic relation method seemed best overall, the dictionary results were comparable and the thesaurus approach was only slightly worse.

We are convinced that much larger improvements are possible if end-users can be more directly involved in the process, so they can decide which words should be expanded, and can select which related terms to include from the lists produced from our thesaurus. Testing this hypothesis, however, requires a more flexible processing paradigm than we have employed in the past. Furthermore, we believe that inferencing using the information in the semantic network we are building can allow us to develop an effective automatic or semi-automatic scheme for "intelligent" query expansion. The CODER system should support these approaches.

Building upon early efforts to build intelligent retrieval systems (Guida and Tasso 1983, Pollitt 1984) and learning from experiences with similar systems (Croft and Thompson 1987), we have been developing the CODER (COmposite Document Expert/effective/extended Retrieval) system (Fox and France 1987, Fox 1987) for the last three years. Though part of that effort deals with new approaches to automatic text analysis (Fox and Chen 1987), in the current context the most important aspect of CODER is that it is built as a distributed collection of "expert" modules (according to the models discussed in Belkin et al. 1987) programmed in Prolog or C, to support flexible testing of various AI approaches to information retrieval. Weaver and France have developed modules for handling lexical and semantic relations and a server module providing access to our version of the contents of CDEL. In the future, a module will be added to interface CODER with the SNePS semantic network so that further experiments can be undertaken.

6. Conclusions and Future Work

Based on preliminary investigations, it appears that lexical-semantic relations can be used to give small improvements in the effectiveness of information retrieval systems. Work with *Webster's Seventh New Collegiate Dictionary* and the *Collins Dictionary of the English Language* has demonstrated that text processing and natural language parsing techniques can be used to extract and organize important data about English that will be of value for information retrieval and for a variety of natural language processing applications. Furthermore, it is clear that the SNePS semantic network system can be used to store that type of data in a form that will permit both rule and path-based inference.

Future work will include further processing of dictionaries (including the *Longman Dictionary of Contemporary English* and others from the Oxford Text Archive), merging the resulting output into a large semantic network, extending the capabilities of SNePS to handle a very large thesaurus, integrating SNePS with the SMART and CODER systems, and further testing of the utility of that thesaurus to support interactive information retrieval sessions.

REFERENCES

Ahlswede, Thomas, 1985. "A Tool Kit for Lexicon Building." In *Proc. 23rd Annual Meeting of the Association for Computational Linguistics*, 268-276.

Ahlswede, Thomas and Martha Evens, 1987. "Generating a Relational Lexicon from a Machine-Readable Dictionary." In *Visible Language*, W. Frawley and R. Smith (eds.), Oxford University Press, to appear.

Amsler, Robert A., 1980. "The Structure of the Merriam-Webster Pocket Dictionary." Ph.D. Dissertation. TR-164, Computer Science Dept., Univ. of Texas, Austin, Dec. 1980.

Amsler, Robert A., 1984. "Machine-Readable Dictionaries." *Annual Review of Information Science and Technology*, 19:161-209.

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

Attar, R. and Aviezri S. Fraenkel, 1977. "Local Feedback in Full-Text Retrieval Systems." *J. ACM*, 24(3): 397-417.

Becker, Joseph, 1975. "The Phrasal Lexicon." In *Theoretical Issues in Natural Language Processing*, R. Schank and B. Nash-Webber (eds.), ACL, 38-41.

Belkin, N. et al., 1987. "Distributed Expert-Based Information Systems: An Interdisciplinary Approach." *Information Processing and Management*, to appear.

Blair, D.C. and M.E. Maron, 1985. "An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System." *Commun. ACM*, 28(3):289-299.

Chen, Hsinchun and Vasant Dhar, 1987. "Reducing Indeterminism in Consultation: A Cognitive Model of User/Librarian Interactions." In *Proc. AAAI-87*, 285-289.

Chodorow, Martin S., Roy Byrd, and George Heidorn, 1985. "Extracting Semantic Hierarchies from a Large On-Line Dictionary." In *Proc. of the 23rd Annual Meeting of the Association for Computational Linguistics*, 299-304.

Croft, W. Bruce and Roger Thompson, 1987. "I3R: A New Approach to the Design of Document Retrieval Systems." *J. Am. Soc. Inf. Sci.*, in press.

Evens, Martha W. and Raoul N. Smith, 1978. "A Lexicon for a Computer Question-Answering System." *Am. J. Comp. Ling.*, No. 4, Microfiche 83: 1-96.

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

Evens, Martha W., James Vandendorpe, and Yih-Chen Wang, 1985. "Lexical-Semantic Relations in Information Retrieval." In *Humans and Machines:*

The Interface Through Language, S. Williams (ed.), Ablex, Norwood, NJ, 73-100.

Evens, Martha W., Judith Markowitz, Thomas Ahlswede, and Kay Rossi, 1987. "Digging in the Dictionary: Building a Relational Lexicon to Support Natural Language Processing Applications." *IDEAL* (Issues and Developments in English and Applied Linguistics) 2(33-44).

Fox, E.A., 1980. "Lexical Relations: Enhancing Effectiveness of Information Retrieval Systems." *ACM SIGIR Forum*, 15(3):5-36.

Fox, Edward A., 1983a. "Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types." Dissertation, Cornell University, University Microfilms Int., Ann Arbor MI.

Fox, Edward A., 1983b. "Some Considerations for Implementing the SMART Information Retrieval System under UNIX." TR 83-560, Cornell Univ., Dept. of Comp. Sci..

Fox, Edward A., 1984. "Improved Retrieval Using a Relational Thesaurus Expansion of Boolean Logic Queries." In *Proc. Workshop Relational Models of the Lexicon*, Martha W. Evens (ed.), Stanford, CA, to appear.

Fox, Edward A., 1987. "Development of the CODER System: A Testbed for Artificial Intelligence Methods in Information Retrieval." *Information Processing and Management*, 23(4).

Fox, Edward A. and Qi-Fang Chen, 1987. "Text Analysis in the CODER System." In *Proc. Fourth Annual USC Computer Science Symposium: Language and Data in Systems*, Columbia SC, 7-14.

Fox, Edward A. and Robert K. France, 1987. "Architecture of an Expert System for Composite Document Analysis, Representation and Retrieval." *Int. J. of Approximate Reasoning*, 1(2).

Fox, Edward A., Robert C. Wohlwend, Phyllis R. Sheldon, Qi Fan Chen, and Robert K. France, 1986. "Building the CODER Lexicon The Collins English Dictionary and Its Adverb Definitions." TR-86-23, VPI&SU Computer Science Dept., Blacksburg, VA.

France, Robert K. and Edward A. Fox, 1986. "Knowledge Structures for Information Retrieval: Representation in the CODER Project." In *Proceedings IEEE Expert Systems in Government Conference*, October 20-24, 1986, McLean VA, 135-141.

Furnas, George W. et al., 1982. "Statistical semantics: how can a computer use what people name things to guess what things people mean when they name things." In *Proc. of the Human Factors in Computer Systems Conference*, Gaithersburg, MD, Assoc. for Computing Machinery, New York.

Guida, G., and C. Tasso, 1983. "An expert intermediary system for interactive document retrieval." *Automatica* 19(6): 759-766.

Hull, Richard G., 1986. "A New Design for SNIP the SNePS Inference Package." SNeRG Technical Report 86-10, Department of Computer Science, SUNY at Buffalo.

Lenat, Doug et al., 1986. "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks." *The AI Magazine*, 65-84.

Nutter, J. Terry, 1983. "Default reasoning using monotonic logic: a modest proposal." In *Proc. AAAI-83, The National Conf. on AI*, Washington D.C., 297-300.

Ojala, Marydee, 1986. "Views on End-User Searching." *J. Am. Soc. Inf. Sci.*, 37(4), 197-203.

Olney, John, 1968. "To All Interested in the Merriam-Webster Transcripts and Data Derived from Them." Systems Development Corporation Document L-13579.

Peterson, James L., 1982. "Webster's Seventh New Collegiate Dictionary: A Computer-Readable File Format." TR-196, Dept. of Comp. Sci., Univ. of Texas, Austin.

Pollitt, S.E., 1984. "A 'front-end' system: an Expert System as an online search intermediary." *Aslib Proceedings*, 36(5): 229-234.

Quillian. M. Ross, 1968. "Semantic Memory." In *Semantic Information Processing*, Marvin Minsky ed. Cambridge, MA: MIT Press, 227-270.

Ritchie, G.D. and Hanna, F.K., 1984. "Semantic Networks: a General Definition and a Survey." *Inf. Tech.: Res. Dev. Applications*, 3(1):33-42.

Sager, Naomi, 1981. *Natural Language Information Processing*." Addison-Wesley, Reading, MA.

Salton, G., 1986. "Another Look at Automatic Text-Retrieval Systems." *Commun. ACM*, 29(7): 648-656.

Shapiro, Stuart C., 1979a. "Generalized Augmented Transition Network Grammars for Generation from Semantic Networks." In *Proc. of the 17th Annual Meeting of the Association for Computational Linguistics*, 25-29.

Shapiro, Stuart C., 1979b. "The SNePS Semantic Network Processing System." In *Associative Networks*, Nick Findler (ed.), Academic Press, NY, 179-203.

Shapiro, Stuart C. and William Rapaport, 1986. "SNePS Considered as a Fully Intensional Propositional Semantic Network." In *Proc. AAAI-86, Fifth National Conf. on AI*, Phil. PA, 278-283.

Smith, Raoul N., 1985. "Conceptual primitives in the English lexicon." *Papers in Linguistics* 18: 99-137.

Svenonius, Elaine, 1986. "Unanswered Questions in the Design of Controlled Vocabularies." *J. Am. Soc. Inf. Sci.*, 37(5):331-340.

Walker, Donald E., 1985. "Knowledge Resource Tools for Accessing Large Text Files." In *Proc. First Conference of the UW Centre for the New Oxford English Dictionary: Information in Data.* Nov. 6-7, 1985, Waterloo, Canada, 11-24.

Wang, Yih-Chen, James Vandendorpe, and Martha Evens, 1985. "Relational Thesauri in Information Retrieval." *J. Am. Soc. Inf. Sci.*, 36(1): 15-27.

White, Carolyn, 1983. "The Linguistic String Project Dictionary for Automatic Text Analysis." In *Proc. Workshop on Machine Readable Dictionaries*, SRI, Menlo Park, CA.

Wohlwend, Robert C., 1986. "Creation of a Prolog Fact Base from the Collins English Dictionary." MS Report, VPI&SU Computer Science Dept., Blacksburg, VA.