Design and Implementation of the WordNet Lexical Database and Searching Software†

Richard Beckwith, George A. Miller, and Randee Tengi

Lexicographers must be concerned with the presentation as well as the content of their work, and this concern is heightened when presentation moves from the printed page to the computer monitor. Printed dictionaries have become relatively standardized through many years of publishing (Vizetelly, 1915); expectations for electronic lexicons are still up for grabs. Indeed, computer technology itself is evolving rapidly; an indefinite variety of ways to present lexical information is possible with this new technology, and the advantages and disadvantages of many possible alternatives are still matters for experimentation and debate. Given this degree of uncertainty, manner of presentation must be a central concern for the electronic lexicographer.

WordNet is a pioneering excursion into this new medium. Considerable attention has been devoted to making it useful and convenient, but the solutions described here are unlikely to be the final word on these matters. It is hoped that readers will not merely note the shortcomings of this work, but will also be inspired to make improvements on it.

One's first impression of WordNet is likely to be that it is an on-line thesaurus. It is true that sets of synonyms are basic building blocks, and with nothing more than these synonym sets the system would have all the power of a thesaurus. When short glosses are added to the synonym sets, it resembles an on-line dictionary that has been supplemented with synonyms for cross referencing (Calzolari, 1988). But WordNet includes much more information than that. In an attempt to model the lexical knowledge of a native speaker of English, WordNet has been given detailed information about relations between word forms and synonym sets. How this relational structure should be presented to a user raises questions that outrun the experience of conventional lexicography.

In developing this on-line lexical database, it has been convenient to divide the work into two interdependent tasks which bear a vague similarity to the traditional tasks of writing and printing a dictionary. One task was to write the source files that contain the basic lexical data — the contents of those files are the lexical substance of WordNet. The second task was to create a set of computer programs that would accept the source

---

† This is a revised version of "Implementing a Lexical Network" in CSL Report #43, prepared by Randee Tengi. UNIX is a registered trademark of UNIX System Laboratories, Inc. Sun, Sun 3 and Sun 4 are trademarks of Sun Microsystems, Inc. Macintosh is a trademark of Macintosh Laboratory, Inc. licensed to Apple Computer, Inc. NeXT is a trademark of NeXT. Microsoft Windows is a trademark of Microsoft Corporation. IBM is a registered trademark of International Business Machines Corporation. X Windows is a trademark of the Massachusetts Institute of Technology. DECstation is a trademark of Digital Equipment Corporation.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

files and do all the work leading ultimately to the generation of a display for the user.

The WordNet system falls naturally into four parts: the WordNet lexicographers' source files; the software to convert these files into the WordNet lexical database; the WordNet lexical database; and the suite of software tools used to access the database. The WordNet system is developed on a network of Sun-4 workstations. The software programs and tools are written using the C programming language, Unix utilities, and shell scripts. To date, WordNet has been ported to the following computer systems: Sun-3; DECstation; NeXT; IBM PC and PC clones; Macintosh.

The remainder of this paper discusses general features of the design and implementation of WordNet. The ''WordNet Reference Manual'' is a set of manual pages that describe aspects of the WordNet system in detail, particularly the user interfaces and file formats. Together the two provide a fairly comprehensive view of the WordNet system.

Index of Familiarity

One of the best known and most important psycholinguistic facts about the mental lexicon is that some words are much more familiar than others. The familiarity of a word is known to influence a wide range of performance variables: speed of reading, speed of comprehension, ease of recall, probability of use. The effects are so ubiquitous that experimenters who hope to study anything else must take great pains to equate the words they use for familiarity. To ignore this variable in a lexical database that is supposed to reflect psycholinguistic principles would be unthinkable.

In order to incorporate differences in familiarity into WordNet, a syntactically tagged index of familiarity is associated with each word form. This index does not reflect all of the consequences of differences of familiarity — some theorists would ask for strength indices associated with each relation — but accurate information on all of the consequences is not easily obtained. The present index is a first step.

Frequency of use is usually assumed to be the best indicator of familiarity. The closed class words that play an important syntactic role are the most frequently used, of course, but even within the open classes of words there are large differences in frequency of occurrence that are assumed to correlate with — or to explain — the large differences in familiarity. The frequency data that are readily available in the technical literature, however, are inadequate for a database as extensive as WordNet. Thorndike and Lorge (1944) published data based on a count of some 5,000,000 running words of text, but they reported their results only for the 30,000 most frequent words. Moreover, they defined a ''word'' as any string of letters between successive spaces, so their counts for homographs are untrustworthy; there is no way to tell, for example, how often *lead* occurred as a noun and how often as a verb. Francis and Kučera (1982) tag words for their syntactic category, but they report results for only 1,014,000 running words of text — or 50,400 word types, including many proper names — which is not a large enough sample to yield reliable counts for infrequently used words. (A comfortable rate of speaking is about 120 words/minute, so that 1,000,000 words corresponds to 140 hours, or about two weeks of normal exposure to language.)

DOCKET ALARM Find authenticated court documents without watermarks at docketalarm.com.

Fortunately, an alternative indicator of familiarity is available. It has been known at least since Zipf (1945) that frequency of occurrence and polysemy are correlated. That is to say, on the average, the more frequently a word is used the more different meanings it will have in a dictionary. An intriguing finding in psycholinguistics (Jastrezembski, 1981) is that polysemy seems to predict lexical access times as well as frequency does. Indeed, if the effect of frequency is controlled by choosing words of equivalent frequencies, polysemy is still a significant predictor of lexical decision times.

Instead of using frequency of occurrence as an index of familiarity, therefore, WordNet uses polysemy. This measure can be determined from an on-line dictionary. If an index value of 0 is assigned to words that do not appear in the dictionary, and if values of 1 or more are assigned according to the number of senses the word has, then an index value can be made available for every word in every syntactic category. Associated with every word form in WordNet, therefore, there is an integer that represents a count (of the Collins *Dictionary of the English Language*) of the number of senses that word form has when it is used as a noun, verb, adjective, or adverb.

A simple example of how the familiarity index might be used is shown in Table 1. If, say, the superordinates of *bronco* are requested, WordNet can respond with the sequence of hypernyms shown in Table 1. Now, if all the terms with a familiarity index (polysemy count) of 0 or 1 are omitted, which are primarily technical terms, the hypernyms of *bronco* include simply: *bronco* @→ *pony* @→ *horse* @→ *animal* @→ *organism* @→ *entity*. This shortened chain is much closer to what a layman would expect. The index of familiarity should be useful, therefore, when making suggestions for changes in wording. A user can search for a more familiar word by inspecting the polysemy in the WordNet hierarchy.

WordNet would be a better simulation of human semantic memory if a familiarity index could be assigned to word-meaning pairs rather than to word forms. The noun *tie*, for example, is used far more often with the meaning {*tie, necktie*} than with the meaning {*tie, tie beam*}, yet both are presently assigned the same index, 13.

Lexicographers' Source Files

WordNet's source files are written by lexicographers. They are the product of a detailed relational analysis of lexical semantics: a variety of lexical and semantic relations are used to represent the organization of lexical knowledge. Two kinds of building blocks are distinguished in the source files: word forms and word meanings. Word forms are represented in their familiar orthography; word meanings are represented by synonym sets — lists of synonymous word forms that are interchangeable in some syntax. Two kinds of relations are recognized: lexical and semantic. Lexical relations hold between word forms; semantic relations hold between word meanings.

WordNet organizes nouns, verbs, adjectives and adverbs into synonym sets (*synsets*), which are further arranged into a set of lexicographers' source files by syntactic category and other organizational criteria. Adverbs are maintained in one file, while nouns and verbs are grouped according to semantic fields. Adjectives are divided between two files: one for descriptive adjectives and one for relational adjectives.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

Hypernyms of *bronco* and their index values

Word	Polysemy
bronco	1
@→ mustang	1
@→ pony	5
@→ horse	14
@→ equine	0
@→ odd-toed ungulate	0
@→ placental mammal	0
@→ mammal	1
@→ vertebrate	1
@→ chordate	1
@→ animal	4
@→ organism	2
@→ entity	3

Table 1

Appendix A lists the names of the lexicographers' source files.

Each source file contains a list of synsets for one part of speech. Each synset consists of synonymous word forms, relational pointers, and other information. The relations represented by these pointers include (but are not limited to): hypernymy/hyponymy, antonymy, entailment, and meronymy/holonymy. Polysemous word forms are those that appear in more than one synset, therefore representing more than one concept. A lexicographer often enters a textual gloss in a synset, usually to provide some insight into the semantics intended by the synonymous word forms and their usage. If present, the textual gloss is included in the database and can be displayed by retrieval software. Comments can be entered, outside of a synset, by enclosing the text of the comment in parentheses, and are not included in the database.

Descriptive adjectives are organized into clusters that represent the values, from one extreme to the other, of some attribute. Thus each adjective cluster has two (occasionally three) parts, each part headed by an antonymous pair of word forms called a head synset. Most head synsets are followed by one or more satellite synsets, each representing a concept that is similar in meaning to the concept represented by the head synset. One way to think of the cluster organization is to visualize a wheel, with each head synset as a hub and its satellite synsets as the spokes. Two or more wheels are logically connected via antonymy, which can be thought of as an axle between wheels.

The Grinder utility compiles the lexicographers' files. It verifies the syntax of the files, resolves the relational pointers, then generates the WordNet database that is used with the retrieval software and other research tools.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

Word Forms

In WordNet, a word form is represented as the orthographic representation of an individual word or a string of individual words joined with underscore characters. A string of words so joined is referred to as a collocation and represents a single concept, such as the noun collocation *fountain_pen*.

In the lexicographers' files a word form may be augmented with additional information, necessary for the correct processing and interpretation of the data. An integer sense number is added for sense disambiguation if the same word form appears more than once in a lexicographer file. A syntactic marker, enclosed in parentheses, is added to any adjectival word form whose use is limited to a specific syntactic position in relation to the noun that it modifies. Each word form in WordNet is known by its orthographic representation, syntactic category, semantic field, and sense number. Together, these data make a ''key'' which uniquely identifies each word form in the database.

Relational Pointers

Relational pointers represent the relations between the word forms in a synset and other synsets, and are either lexical or semantic. Lexical relations exists between relational adjectives and the nouns that they relate to, and between adverbs and the adjectives from which they are derived. The semantic relation between adjectives and the nouns for which they express values are encoded as attributes. The semantic relation between noun attributes and the adjectives expressing their values are also encoded. Presently these are the only pointers that cross from one syntactic category to another. Antonyms are also lexically related. Synonymy of word forms is implicit by inclusion in the same synset. Table 2 summarizes the relational pointers by syntactic category. Meronymy is further specified by appending one of the following characters to the meronymy pointer: **p** to indicate a part of something; **s** to indicate the substance of something; **m** to indicate a member of some group. Holonymy is specified in the same manner, each pointer representing the semantic relation opposite to the corresponding meronymy relation.

Many pointers are reflexive, meaning that if a synset contains a pointer to another synset, the other synset should contain a corresponding reflexive pointer back to the original synset. The Grinder automatically generates the relations for missing reflexive pointers of the types listed in Table 3.

A relational pointer can be entered by the lexicographer in one of two ways. If a pointer is to represent a relation between synsets — a semantic relation — it is entered following the list of word forms in the synset. Hypernymy always relates one synset to another, and is an example of a semantic relation. The lexicographer can also enclose a word form and a list of pointers within square brackets (**[...]**) to define a lexical relation between word forms. Relational adjectives are entered in this manner, showing the lexical relation between the adjective and the noun that it pertains to.

DOCKET ALARM
Find authenticated court documents without watermarks at docketalarm.com.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.