Aggressive Morphology for Robust Lexical Coverage

William A. Woods
Sun Microsystems Laboratories
1 Network Drive
Burlington, MA 01803
William.Woods@east.sun.com

Abstract

This paper describes an approach to providing lexical information for natural language processing in unrestricted domains. A system of approximately 1200 morphological rules is used to extend a core lexicon of 39,000 words to provide lexical coverage that exceeds that of a lexicon of 80,000 words or 150,000 word forms. The morphological system is described, and lexical coverage is evaluated for random words chosen from a previously unanalyzed corpus.

1 Motivation

Many applications of natural language processing have a need for a large vocabulary lexicon. However, no matter how large a lexicon one starts with, most applications will encounter terms that are not covered. This paper describes an approach to the lexicon problem that emphasizes recognition of morphological structure in unknown words in order to extend a relatively small core lexicon to allow robust natural language processing in unrestricted domains. This technique, which extends functionality originally developed for the Lunar system (Woods et al., 1972), has been most recently applied in a conceptual indexing and retrieval system (Woods, 1997; Ambroziak and Woods, 1998; Woods et al., 2000).

The system described here uses a collection of approximately 1200 knowledge-based morphological rules to extend a core lexicon of approximately 39,000 words to give coverage that exceeds that of an English lexicon of more than 80,000 base forms (or 150,000 base plus inflected forms). To illustrate the need for a robust extensible lexicon, a random sample of 100 words from the vocabulary of the million-word Brown corpus (Kucera and Francis, 1967), contained 24 words that were not included in a 300,000-word list of English word forms. This suggests that approximately 25% of the words in the Brown corpus would not be covered by an independent lexicon of even 300,000 words.

In a recent experiment, 54% of approximately 34,000 word types (numbers and hyphenated words excluded) from a 3.1-million-word corpus of technical literature would not be covered by our hypothetical 300,000-word lexicon. Many of these are special forms (e.g., *Nb2O3* and *Ti/tin*), and some are apparent misspellings (e.g., *auniprocessor* and *sychronized*), but the following are a sampling of fairly normal words that were not in the 300,000-word list:

busmaster
copyline
hereabove
preprocessing
uniprocessors
unreacted

2 Integrated, Preferential, Heuristic Morphology

There are a number of systems that have been used to describe natural language morphology for computational use. The most popular of these is perhaps the finite-state Kimmo system (Koskenniemi, 1983). Other approaches are described in (Sproat, 1992). The system described here differs from other systems in a number of dimensions. First, it is integrated with an extensive lexicon, a semantic ontology, and a syntactic analysis system, which it both consults and augments. For example, subsumption relationships in the semantic ontology enable the system to determine whether a proposed root is a container or a mental attitude, so that *cupful* is interpreted as a unit of measure (a kind of noun), while *hopeful* is interpreted as an adjective.

Second, it uses ordered preferential rules that attempt to choose a small number of correct analyses of a word (usually 1–3) from the many potential analyses that might be found. Finally, it uses rules that are heuristic in that they are not guaranteed to give correct analyses, but rather are designed to deal with various states of lack of knowledge and to make plausible inferences in the face of uncertainty. The focus is to use what it knows (or can infer) to determine a usable set of part-of-speech classifications for the word and to determine any root-plus-affix or internal compound structure that is apparent. If possible, it also assigns a semantic categorization to the word. It deals with unknown as well as known

roots, and it indicates relative confidences in its classifications when its rules indicate uncertainty in the result.

The role of the morphological analysis component in this system is to construct lexical entries for words that do not already have entries, so that subsequent encounters with the same word will find an already existing lexical entry. Thus, morphological analysis happens only once for each encountered word type that is not already in the core lexicon. The resulting lexical entries can be saved in a supplementary lexicon that is constructed as a side-effect of analyzing text. The rules of the morphological analysis system can ask syntactic and semantic questions about potential base forms. The system handles prefixes, suffixes, and lexical compounds (e.g., *bitmap* and *replybuffer*). It also handles multiword lexical items and many special forms, including Roman numerals, dates, and apparent phone numbers.

2.1 Morphological rules and the lexicon

The morphological analysis system makes use of a number of different kinds of morphological rules, applied in the following preferential order to words that are not already in the lexicon:

1. Morph-precheck for special forms
2. Phase one pass with suffix rules (allow only "known" roots in phase one)
3. Prefix rules
4. Lexical compound rules
5. Check of name lists and city lists for words not yet recognized
6. Phase two pass with suffix rules (allow unknown roots and default rules)

Generally, the rules are ordered in decreasing order of specificity, confidence and likelihood. Very specific tests are applied in Step 1 to identify and deal with "words" that are not ordinary sequences of alphabetic characters. These include numbers, alphanumeric sequences, and expressions involving special characters. Failing this, an ordered sequence of suffix rules is applied in Step 2 in a first pass that will allow a match only if the proposed root word is "known." The same list of rules will be applied later in a second pass without this known-root condition if an earlier analysis does not succeed. This issue of "known" roots is a subtle one that can involve consulting external lists of known words as well as words already in the lexicon, and can also consider certain derived forms of known roots to be "known," even when they have not been previously encountered. For example, if *fish* is a known word, then *fishing* is as good as known, so is considered a "known" root for this purpose. In general, suffix rules applied to "known" roots are more reliable than applications of rules to unknown roots or to words with no identifiable root.

If no phase-one suffix rules apply, prefix rules are tried in Step 3 to see if an interpretation of this word as a prefix combined with some other "known" word is possible. Failing this, a set of lexical compound rules is tried, in Step 4, to see if the word is interpretable as a compound of two or more words, and failing that, lists of first and last names of people and names of cities are checked in Step 5. All of steps 3–5 are considered more reliable if they succeed than the phase-two pass of the suffix rules that comes in Step 6. This ordering allows prefixes and compounding to be tried before less confident suffix analyses are attempted, and avoids applying weak suffix analyses to known names. Various other ways to order these rules have been tried, but this order has been found to be the most effective.

2.2 Special form tests

Before trying pattern-based rules for suffixes, prefixes, and lexical compounds, the morphological analyzer makes a number of tests for special forms that require idiosyncratic treatment. These tests include the following:

- number (including integer, floating, and exponential notations, including numbers too large to be represented internally as numbers in the machine),
- Roman numeral (*vii*, *mcm*),
- ordinal (*1st*, *2nd*, *twenty-third*),
- alphanum (*Al2O3*, *79D*),
- letter (*b*, *x*),
- initial (*B.*),
- phone number (*123-4567*),
- hyphenated adjective (*all-volunteer*),
- ratio (*3/4*, *V/R*),
- multiword lexical item (*snake_in_the_grass*),
- special proper nouns (*gls@mit.edu*, */usr/bin*, *http://www.sun.com*, *C++*)

2.3 Pattern-action rules

Suffix rules in this system are pattern-action rules that specify:

1. a pattern of characters to match at the end of the word to be analyzed,
2. possibly a number of characters to remove and/or a sequence of characters to add to form a root (or base form),
3. a sequence of tests and action clauses indicating possible interpretations of a word matching this pattern.

These rules are organized into blocks that are typically indexed by a shared final letter, and are applied in order within a block until a rule is encountered that generates one or more interpretations. At that point, no further rules are tried, and the interpretations generated by that rule are used to construct a lexical entry for the analyzed word.

The following is an example of a fairly specific, but productive, knowledge-rich morphological suffix rule:

```
((f i s h) (kill 4)
   (test (plausible-root root))
   (cat nmsp
        (is-root-of-cat root '(adj n))
        eval (progn (mark-dict lex
                               'false-root
                               root t t)
                    (mark-dict lex
                               'kindof
                               'fish t t)
                    (mark-dict lex
                               'has-prefix
                               root t t)
                    (mark-dict lex
                               'root
                               'fish t t)
                    '-es)))
```

This rule matches a word that ends in *fish* and removes four letters from the end (the *fish* part) to produce a root word which it then tests to see if it is a plausible root (e.g., does it at least have a vowel in it?). If it gets this far, the rule will construct a category `nmsp` interpretation (a kind of noun), if the condition `(is-root-of-cat root '(adj n))` is true (i.e., if the root is a known adjective or noun). This rule deals with words like *hagfish* and *goatfish* and comes before the rules that handle words with *ish* as a suffix, like *doltish* and *oafish*. Incidentally, this rule doesn't apply to *oafish* because the hypothesized root *oa*, which would result from removing four letters, is not known to be an adjective or noun. When this rule succeeds, it specifies that the word will be assigned the category `nmsp`, a category indicating a word that has a mass sense, a singular count sense, and can also be used as a plural (e.g., *Goatfish are funny-looking.*). (The category `nmsp` comes from a collection of 91 syntactic categories, organized in a hierarchy based on generality, so that, for example, `nm` subsumes `nmsp`.) The action part of this rule specifies that (contrary to the usual case) the "root" obtained by removing characters from the end of the word (e.g., *goat*) is in this case a false root. The real root is *fish*, and the false root (*goat*) is actually a prefix. The rule also specifies that the word refers to a kind of fish and that the inflectional paradigm for this word is *-es* (thus allowing *goatfishes* as an alternative plural).

The rules within a block are ordered in decreasing order of confidence and specificity. Thus, rules with conditions that check explicit inflectional paradigms of known roots are ordered before rules that guess the inflectional paradigm from the spelling of the root, and rules with more specific conditions are ordered before rules with less specific conditions so that the latter can assume that the former will already have been tested and rejected. The rules within a block of suffix rules will typically try for interpretations in roughly the following order:

1. inflected form of a known root satisfying a named inflectional paradigm (paradigmatic)
2. inflected form of a known word in right category with unknown inflectional paradigm
3. apparent inflected form of a known word of some other category
4. apparent inflected form of an unknown word
5. apparent derived form of a known root of the right category
6. apparent derived form of a known root regardless of category
7. apparent derived form of an unknown root
8. word with apparent syntactic category and perhaps suffix, without identifiable root
9. guessed noun (and perhaps verb also, if core vocabulary is not comprehensive)

The last rule in this sequence is a default guessing rule that depends on a flag that tells it whether it is running with a core lexicon that is believed to contain most nonobvious verbs. If so, then only the noun part-of-speech is assigned, but with a smaller core lexicon, the guessing rules would also assign a less likely interpretation as a verb, in order to provide a way for unknown verbs to be parsed correctly in sentences.

Prefix rules are similar in structure to suffix rules, except that the pattern is matched at the beginning of the word, and the rule blocks are indexed by the initial letter of the word. Lexical compound rules have a slightly different format and are called by a specialized interpreter that looks for places to divide a word into two pieces of sufficient size. The points of potential decomposition are searched from right to left, and the first such point that has an interpretation is taken, with the following exception: The morph compound analyzer checks for special cases where, for example, the first word is plural and ends in an *s*, but there is an alternative segmentation in which the singular of the first word is followed by a

word starting with the *s*. In such cases, the decomposition using the singular first word is preferred over the one using the plural. For example, the word *minesweeper* will be analyzed as *mine+sweeper* rather than *mines+weeper*. This preference heuristic is specific to English and might be different for other languages.

2.4 Recursive application of rules

When attempting to apply a rule to a word, the morphological analyzer can be applied recursively to analyze the hypothesized root. A simple caching technique is used to control the potential for combinatoric explosion and to block looping. This is sufficiently effective that the time required for morphological analysis is a negligible part of the time required for processing large amounts of natural language text. Protection against looping is especially important for a kind of morphological rule that derives one word from another without either of them being a root of the other in the usual sense (e.g., deriving *communist* from *communism* or *external* from *internal*). Operating in a loop-safe environment allows rules like these to identify the relationship between a new word and a known word in either direction, whichever of the two forms is encountered first.

3 Evaluation

Since analyzing a word is done once per unknown word type and consumes a negligible fraction of the overall text-processing time, speed of operation is not considered a factor for evaluation. The interesting dimension of evaluation deals with the coverage of the rules and the kinds of errors that are made. This was tested by applying the system to two word lists randomly selected from the Brown corpus and provided to me by Philip Resnik, using some sampling tools that he developed. The first of these (the token sample) consists of 100 word tokens selected randomly, without eliminating duplicates, and the second (the type sample) consists of 100 distinct word types selected randomly from the vocabulary of the Brown corpus. Prior to a single test run on each of these samples, neither the lexicon nor the morphological rule system had any exposure to the Brown corpus, nor had either of these word lists been looked at by the experimenter. Consequently, the results are a fair evaluation of the expected performance of this system on an unknown domain.

3.1 Grading rule performance

Since different syntactic category errors have different consequences for parsing text, it is useful to grade the syntactic category assignments of the analyzer on an A-B-C-D-F scale according to the severity of any mistakes. Grades are assigned to a lexical entry as follows:

A if all appropriate syntactic categories are assigned and no incorrect categories are assigned

B if all categories are correct, allowing for categorizing an adjective or a name as a noun or a noun as a name

C if an entry has at least one correct category and is correct except for missing a noun category or having a single extra category

D if there is more than one extra category or if there is a missing category other than one of the above cases, provided that there is at least one correct category

F if there are no correct categories

Both A and B grades are considered acceptable assignments for the sake of evaluation, since category B errors would allow a reasonable parse to be found. This is because the grammar used for parsing sentences and phrases allows a noun to be used as an adjective modifier and a proper noun to be used in place of a noun. One parser/grammar that uses this lexicon also allows any other category to be used as a noun, at the expense of a penalty, so that a C grade will still enable a parse, although with a penalty and a substantial likelihood that other false parses might score better. Similarly, a D grade increases the likelihood that a false parse might score better.

Separately, we measure whether count/mass distinctions are made correctly (for nouns only), and whether roots of derived and inflected forms are identified correctly. We are interested in the count/mass distinction because, like the common/proper noun distinction, it affects the grammaticality and likelihood of a noun phrase interpretation for a singular noun in absence of an explicit determiner.

3.2 Sampling rule performance

The morphological analyzer has been applied to the words from the two sample word lists that were not already in its core lexicon. There were 17 such words from the token sample and 72 such words from the type sample. Of the 17 unknown token-sample words, 100% were graded B or better (88% A and 12% B); 85% of the roots were identified correctly (all but one); 85% of the count noun senses were found (all but one); and 100% of the mass noun senses were found. Token-sample performance is not a very challenging test for a morphological analyzer because it is biased toward a relatively small number of frequently occurring word types. Token-sample performance is used to assess the per-token error rate that one would expect in analyzing large amounts of running text. In contrast, type-sample performance

Table 1: Syntactic category performance of the analyzer.

Category Grade	A	B	C	D	F	B or better
Number	62	8	1	0	1	70
Percent	86%	11%	1.5%	0%	1.5%	97%

Table 2: Count/mass distinction performance of the analyzer.

Count/mass	Good count	Extra count	Good mass	Missing mass
Number	39	1	14	1
Percent	100%	2.6%	93%	6.7%

Table 3: Root identification performance of the analyzer.

Detect root	Good	Wrong	Debatable	Missing	Extra
Number	57	1	1	0	1
Percent	95%	1.7%	1.7%	0	1.7%

gives a measure of the expected performance on new words the analyzer is likely to encounter.

For the 72 words in the type sample that are not covered by the lexicon, Tables 1–3 show the syntactic category performance of the analyzer and its abilities to make count/mass distinctions and identify roots.

Notes on incorrect or debatable analyses:

1. One N (noun) for a probable name (*Tonio*), counted as B.
2. Two NPR(proper name) for abbreviations; (*A.V.* may be ADJ, *W.B.* is correct), counted as one B and one A.
3. One wrong root when suffix *ism* was identified as root of *hooliganism* in a hypothesized compound *hooligan+ism* (arguably justifiable as a kind of *ism*, which is known in the lexicon, but counted as an error anyway). Reanalyzing this word after *hooligan* is a known word gets the correct interpretation.
4. One debatable root in the hyphenated phrase *reference-points* whose root was listed as *points* rather than *reference-point*. This is due to a bug that caused the hyphenated word rules to incorrectly identify this as a verb, rather than a noun (counted as F for syntax).
5. One extra root for *embouchure* from *embouche* (but a correct form of the French root?).
6. One missing category N for *bobbles*, which was given category V but not N because the core lexicon incorrectly listed *bobble* only as a verb (counted as C for syntax). This is corrected by adding the missing category to the lexical entry for *bobble*.

4 Conclusions

We have described an approach to robust lexical coverage for unrestricted text applications that makes use of an aggressive set of morphological rules to supplement a core lexicon of approximately 39,000 words to give lexical coverage that exceeds that of a much larger lexicon. This morphological analyzer is integrated with an extensive lexicon, an ontology, and a syntactic analysis system, which it both consults and augments. It uses ordered preferential rules that attempt to choose a small number of correct analyses of a word and are designed to deal with various states of lack of knowledge. When applied to 72 unknown words from a random sample of 100 distinct word types from the Brown corpus, its syntactic category assignments received a grade of B or better (using a grading system explained herein) for 97% of the words, and it correctly identified 95% of the root words. This performance demonstrates that one can obtain robust lexical coverage for natural language processing applications in unrestricted domains, using a relatively small core lexicon and an aggressive collection of morphological rules.

References

Jacek Ambroziak and William A. Woods. 1998. Natural language technology in precision content retrieval. In *International Conference on Natural Language Processing and Industrial Applications*, Moncton, New Brunswick, Canada, August. www.sun.com/research/techrep/1998/abstract-69.html.

Kimmo Koskenniemi. 1983. Two-level model for morphological analysis. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 683–685, Los Angelos, CA. Morgan Kauffmann.

H. Kucera and W. Francis. 1967. *Computational Analysis of Present-Day American English.* Brown University Press.

Richard Sproat. 1992. *Morphology and Computation.* MIT Press, Cambridge, MA.

William A. Woods, Ronald M. Kaplan, and Bonnie L. Nash-Webber. 1972. The lunar sciences natural language information system: Final report. Technical Report BBN Report No. 2378, Bolt Beranek and Newman Inc, Cambridge, MA, June. (available from NTIS as N72-28984).

William A. Woods, Lawrence A. Bookman, Ann C. Houston, Robert J. Kuhns, Paul A. Martin, and Stephen Green. 2000. Linguistic knowledge can improve information retrieval. In *(these proceedings).*

William A. Woods. 1997. Conceptual indexing: A better way to organize knowledge. Technical Report SMLI TR-97-61, Sun Microsystems Laboratories, Mountain View, CA, April. www.sun.com/research/techrep/1997/abstract-61.html.