A Freely Available Wide Coverage Morphological Analyzer for English

Daniel Karp 1, Yves Schabes, Martin Zaidel, and Dania Egedi

Department of Computer and Information Science
University of Pennsylvania
Philadelphia PA 19104-6389 USA
dkarp/schabes/zaidel/egedi@cis.upenn.edu

Abstract

This paper presents a morphological lexicon for English that handle more than 317000 inflected forms derived from over 90000 stems. The lexicon is available in two formats. The first can be used by an implementation of a two-level processor for morphological analysis (Karttunen and Wittenburg, 1983; Antworth, 1990). The second, derived from the first one for efficiency reasons, consists of a disk-based database using a UNIX hash table facility (Seltzer and Yigit, 1991). We also built an X Window tool to facilitate the maintenance and browsing of the lexicon. The package is ready to be integrated into an natural language application such as a parser through hooks written in Lisp and C.

To our knowledge, this package is the only available free English morphological analyzer with very wide coverage.

1 Introduction

Morphological analysis has experienced great success since the introduction of two-level morphology (Koskenniemi, 1983; Karttunen, 1983). Two-level morphology and its implementation are now well understood both linguistically and computationally (Karttunen, 1983; Karttunen and Wittenburg, 1983; Koskenniemi, 1985; Barton et al., 1987; Koskenniemi and Church, 1988). This computational model has proved to be well suited for many languages. Although there are some proprietary wide coverage morphological analyzers for English, to our knowledge those that are freely available provide only very small coverage.

Working from the 1979 edition of the Collins Dictionary of the English Language available through ACL-DCI (Liberman, 1989), we constructed lexicons for PC-KIMMO (Antworth, 1990), a public domain implementation of a two-level processor. Using the morphological rules for English inflections provided by Karttunen and Wittenburg (1983) and our lexicons, PC-KIMMO outputs all possible analyses of each input word, giving its root form and its inflectional attributes. To improve performance, we used PC-KIMMO as a generator on our lexicons to build a disk-based hashed database with a UNIX database facility (Seltzer and Yigit, 1991). Both formats, PC-KIMMO and database, are now available for distribution. We also provide an X Window tool for the database to facilitate maintenance and access. Each format contains the morphological information for over 317000 English words. The morphological database for English runs under UNIX; PC-KIMMO runs under UNIX and on a PC.

This package can be easily embedded into a natural language parser; hooks for accessing the morphological database from a parser are provided for both Lucid Common Lisp and C. This morphological database is currently being used in a graphical workbench (XTAG) for the development of tree-adjoining grammars and their parsers (Paroubek et al., 1992).

2 Lexicons for PC-KIMMO

We used the set of morphological rules for English described by Karttunen and Wittenburg (1983). The rules handle the following phenomena (among others): epenthesis, y to i correspondences, s-deletion, elision, i to y correspondences, gemination, and hyphenation. In addition to the set of rules, PC-KIMMO requires lexicons. We derived PC-KIMMO-style lexicons from the 1979 edition of the Collins Dictionary of the English Language. The 90000-odd roots in the lexicon yield over 317000 inflected forms.

The lexicons use the following parts of speech: verbs (V), pronoun (Pron), preposition (Prep), noun (N), determiner (Det), conjunction (Conj), adverb (Adv), and adjective (A). Figure 1 shows the distribution of these parts of speech in the lexicon. The first column is the distribution of the root forms in the PC-KIMMO lexicon files, and the second column is the distribution for the inflected forms derived from the lexicons and stored in the database. For each word, the lexicon lists its lexical form, a continuation class, and a parse. The continuation class specifies which inflections the lexical form can undergo. At most, a noun root engenders four inflections (singular, plural, singular genitive, plural genitive); an adjective root, three (base, com-

1 We refer the reader to Karttunen and Wittenburg (1983) or Antworth (1990) for more details on the morphological rules.

2 Proper nouns were not included in the tables.
2.1 Adjectives

The continuation classes for adjective specify that the word can undergo the rules of comparative and superlative. For example, the lexicon entry for the adjective 'funky' is:

\[
funky \text{ A-Root2 } "\text{A(funky)}"\]

The entry consists of a word 'funky', followed by the continuation class 'A-Root2', and a parse 'A(~unky)'. The continuation class specifies that the word can undergo the normal rules of comparative and superlative, and the parse states that the word is an adjective with root 'funky'. The following is a sample run of PC-KIMMO's recognizer:

\[
\text{recognizer>>>funky}
\]

\[
funky \text{ A(funky)}
\]

\[
\text{recognizer>>>funkier}
\]

\[
funky+\text{er} \text{ A(funky) COMP}
\]

\[
\text{recognizer>>>funkiest}
\]

\[
funky+\text{est} \text{ A(funky) SUPER}
\]

The output line contains the root form and any affixes, separated by '+'. Thus, a '+' in the output indicates a morphological rule was used; its absence means no rule was used, and the parse was returned as found in the lexicon. PC-KIMMO will automatically add attributes such as COMP and SUPER to the parse, depending on the morphological rule matched by the surface form. But for irregularly inflected forms, special continuation classes indicate that the complete parse (viz., part of speech, root, and attributes) should be taken 'as is' from the lexicon entry. For example:

\[
\text{better} \text{ A.Root1 } "\text{A(good) COMP}"
\]

\[
\text{best} \text{ A.Root1 } "\text{A(good) SUPER}"
\]

\[
\text{good} \text{ A.Root1 } "\text{A(good)}"
\]

The class 'A.Root1' tells PC-KIMMO not to apply the morphological rules to 'better', 'best', and 'good'. Thus, 'gooder' is not recognized as 'good+er'.

2.2 Nouns

Inflections of nouns, such as the formation of plural and genitive, are handled by morphological rules (unless the formation is idiosyncratic). In the lexicon for nouns, the continuation class 'N.Root1' indicates that the formation of genitive applies regularly and that no other inflection applies. The continuation class 'N.Root2' indicates that the formation of the plural and of the genitive apply regularly.

\[
mice \text{ N-Root1 } "\text{N(mouse) PL}"
\]

\[
\text{mouse} \text{ N-Root1 } "\text{N(mouse) SG}"
\]

\[
\text{ambassador} \text{ N-Root2 } "\text{N(ambassador)}"
\]

Thus, the above lexicon entries are recognized as below:

\[
\text{recognizer>>>mice}
\]

\[
mice \text{ N(mouse) PL}
\]

\[
\text{recognizer>>>mouse}
\]

\[
\text{mouse} \text{ N(mouse) SG}
\]

\[
\text{recognizer>>>mouses}
\]

\[
\text{mouse+s} \text{ V(mouse) 3SG PRES}
\]

\[
\text{recognizer>>>mice's}
\]

\[
\text{mice+'s} \text{ N(mouse) PL GEN}
\]

\[
\text{recognizer>>>mouses'}
\]

\[
\text{mice+'s} \text{ N(mouse) PL GEN}
\]

\[
\text{recognizer>>>mouse's}
\]

\[
\text{mouse+'s} \text{ N(mouse) PL GEN}
\]

\[
\text{recognizer>>>ambassadors}
\]

\[
\text{ambassador+s} \text{ N(ambassador) PL}
\]

\[
\text{recognizer>>>ambassador's}
\]

\[
\text{ambassador+'s} \text{ N(ambassador) SG GEN}
\]

\[
\text{recognizer>>>ambassadors'}
\]

\[
\text{ambassador+s+'s} \text{ N(ambassador) PL GEN}
\]

2.3 Verbs

Given the infinitive form of a verb, the formation of the third person singular (+s), its past tense (+ed), its past participle (+ed), and its progressive form (+ing) is...
handled by morphological rules unless lexical idiosyncrasies apply. In order to encode all possible idiosyncrasies over the three verb endings, eight continuation classes are defined (see Figure 2). Each continuation class specifies the inflectional rules which can apply to the given lexical item.

| Continuation class | Applicable rules |
|--------------------|------------------|
| V_Root1            | none             |
| V_Root2            | +ed              |
| V_Root3            | +s               |
| V_Root4            | +s, +ed          |
| V_Root5            | +ing             |
| V_Root6            | +ing, +ed        |
| V_Root7            | +ing, +s         |
| V_Root8            | +ing, +s, +ed    |

Figure 2: Continuation classes for verbs

Examples of lexical entries for verbs follow:

- **admire** V_Root8 "V(admire)"
- **dye** V_Root1 "V(dye) PROG"
- **dye** V_Root4 "V(dye)"
- **zigzag** V_Root1 "V(zigzag) PROG"
- **zigzag** V_Root1 "V(zigzag) PAST WK"
- **zigzag** V_Root1 "V(zigzag) PPART WK"
- **tangoes** V_Root1 "V(tango) 3SG PRES"
- **tango** V_Root6 "V(tango)"
- **taught** V_Root1 "V(teach) PAST STR"
- **taught** V_Root1 "V(teach) PPART STR"
- **teach** V_Root7 "V(teach)"

Examples of runs follow:

- `recognizer>>admires` admires V(admire) 3SG PRES
- `recognizer>>admired` admired V(admire) PAST WK
- `recognizer>>teaches` teaches V(teach) 3SG PRES
- `recognizer>>taught` taught V(teach) PAST STR

2.4 Other Parts of Speech

Pronouns, prepositions, determiners, conjunctions, and adverbs are given continuation classes that inhibit the application of morphological rules. All of the morphological information is stored in the parse in the lexicon entry:

- `herself Pron "Pron(herself) REFL FEM 3SG"
- `it Pron "Pron(it) NEUT 3SG NOMACC"
- `behind Prep "Prep(behind)"
- `coolly Adv "Adv(coolly)"

PC-KIMMO recognizes them as follows:

- `recognizer>>teaches` teaches V(teach) 3SG PRES
- `recognizer>>taught` taught V(teach) PAST STR

3 Lexicons as a Database

PC-KIMMO builds in memory a data structure from the complete lexicon. Consequently, our large lexicons occupy more than 19 Mbytes of process memory. Further, the large size of the structure implies long search times as PC-KIMMO swaps pages in and out.

Thus, to solve both the time and space problems simultaneously, we compiled all inflectional forms into...
a disk-based database using a UNIX hash table facility (Seltzer and Yigit, 1991).

To compile the database, we used PC-KIMMO as a generator, inputting each root form and all the endings that it could take, as indicated by the continuation class. The resulting inflected form became the key, and the associated morphological information was then inserted into the database.

For example, the PC-KIMMO lexicon file contains the entry:

```
sa,v Root2 "W(saw)"
```

The class Root2 indicates that the noun 'saw' forms its plural, singular genitive, and plural genitive regularly. Thus, we send to the generator three lexical forms and the three suffixes for each inflection, extracting three inflected surface forms:

```
Lexical    saw+s saw+s saw+s+n
Surface    saws    saw's    saw
```

The root form of a noun is identical with the singular inflection, so we have a total of four inflected forms. Since we know which suffix we added to the root, we also know the attributes for that inflection. The inflected form becomes the key, while the part of speech, root, and attributes are stored as the content in the database. Hence, the lexicon entry for the noun 'saw' produces four key–content pairs in the database:

```
(saw, saw N SG), (saws, saw N PL), (saw's, saw N SG GEN), (saw's, saw N PL GEN).
```

Likewise, the verb lexicon contains the entries:

```
saw V Root8 "V(saw)"
saw V Root1 "V(see) PAST STR"
```

The continuation class Root8 indicates four inflections besides the infinitive: third-person singular (+s), past (+ed), weak past participle (+ed), and present participle (+ing). Hence, the generator produces:

```
Lexical    saw+s saw+ed saw+ing
Surface    saws    sawed    sawing
```

The class Root1 allows no inflections, but builds the inflection–feature pair directly: (saw, see V PAST STR).

Hence, morphological analysis is reduced to sending the surface forms to the database as keys and retrieving the returned strings. Figure 3 lists the database keys and content strings produced by the three lexicon lines given above. Note that distinct entries are separated by '#'. Since multiple lexical forms can map to the same surface form, the actual number of keys (ca. 292000) is less than the number of lexical forms (ca. 317000). Also, with the database residing on the disk, access times average 6 to 10 milliseconds, which greatly improves upon PC-KIMMO.

### 3.1 Implementation Considerations

The large number of keys implies a very large disk file. To reduce the size of the file, we take advantage of the morphological similarity in English between an inflected form and its lexical root form. Indeed, the root is often contained intact within the inflected form.

Hence, instead of storing the root, we store the number of shared characters along with any differing characters, and reassemble the root from the inflected form on each database query. Further, despite the large set of attributes, relatively few combinations (ca. 80) are meaningful, and can be encoded in a single byte. Since a large proportion of roots are wholly contained within the surface form, and since 92% of the keys have one lexical entry, the average content string is only three bytes long. Consequently, the total disk file is under 9MBs. We anticipate further compaction in the near future.

### 3.2 Accompanying Utilities

Besides the PC-KIMMO lexicons, we currently maintain the database file and an ASCII-character "flat" version for on-line database browsing. One program converts the lexicons into the database format, while others dump the database into the flat file or reconstruct the database from the flat file. We have also built a X Windows tool to perform maintenance on the database file (see Figure 4). This tool automatically maintains the consistency between the flat file and the database file. We have built hooks in C and Lisp (Lucid 4.0) to access either the database or PC-KIMMO from within a running process.

### 4 Obtaining the Analyzer

The PC-KIMMO lexicons, the database files, the LISP and C access functions, programs for converting between formats, and the X Window maintenance tool are
available without charge for research purposes. Please send e-mail to zaide16cis.upenn.edu or write to either Yves Schabes, Martin Zaidel, or Dania Egedi.

5 Conclusion

We have presented freely available morphological tables and a morphological analyzer to handle English inflections. The tables handle approximately 317000 inflected forms corresponding to 90000 stems.

These tables can be used by an implementation of a two-level processor for morphological analysis such as PC-KIMMO.

However, these large tables degrade the performance of PC-KIMMO's current implementation, requiring about 18 Mbytes of RAM while slowing the access time.

To overcome these shortcomings, we created a morphological analyzer consisting of a disk-based database using a UNIX hash table facility. With this database, access times average 6 to 10 milliseconds while moving all of the data to the disk. We also provide an X Window tool for facilitating the maintenance and access to the database.

The package is ready to be integrated into an application such as a parser. Hooks written in Lisp and C for accessing these tables are provided.

To our knowledge, this package is the only available free English morphological analyzer with very wide coverage.

Bibliography

Evan L. Antworth. 1990. PC-KIMMO: a two-level processor for morphological analysis. Summer Institute of Linguistics.

G. Edward Barton, Robert C. Berwick, and Eric Sven Ristad. 1987. Computational Complexity and Natural Language. MIT Press.

Lauri Karttunen and Kent Wittenburg. 1983. A two-level morphological analysis of English. Texas Linguistic Forum, 22:217–228.

Lauri Karttunen. 1983. KIMMO: A two-level morphological analyzer. Texas Linguistic Forum, 22:165–186.

Kimmo Koskenniemi. 1983. Two-level morphology: a general computational model for word-form recognition and production. Technical report, University of Helsinki, Helsinki, Finland.

Kimmo Koskenniemi. 1985. An application of the two-level model to Finnish. In Fred Karlsson, editor, Computational Morphosyntax: Report on Research 1981-1984. University of Helsinki.

Kimmo Koskenniemi and Kenneth W. Church. 1988. Complexity, two-level morphology and Finnish. In Proceedings of the 12th International Conference on Computational Linguistics (COLING'88).

Mark Liberman. 1989. Text on tap: the ACL data collection initiative. In Proceedings of DARPA Workshop on Speech and Natural Language Processing, pages 173–188. Morgan Kaufman.

Patrick Paroubek, Yves Schabes, and Aravind K. Joshi. 1992. XTAG - a graphical workbench for developing tree- adjoining grammars. In Third Conference on Applied Natural Language Processing, Trento, Italy.

Margot Seltzer and Ozan Yigit. Winter 1991. A new hashing package for UNIX. In USENIX.

Stuart M. Shieber, 1986. An Introduction to Unification-Based Approaches to Grammar. Center for the Study of Language and Information, Stanford, CA.

A List of Attributes

|   |   |
|---|---|
| 1SG | 1st person singular |
| 2SG | 2nd person singular |
| 3SG | 3rd person singular |
| 1PL | 1st person plural |
| 2PL | 2nd person plural |
| 3PL | 3rd person singular |
| 2ND | 2nd person |
| 3RD | 3rd person |
| SG | singular |
| PL | plural |
| PROG | progressive |
| PAST | past tense |
| PPART | past participle |
| INF | infinitive or present (not 3rd person) |
| PRES | present |
| STR | strongly inflected verb |
| WK | weakly inflected verb |
| GEN | genitive (+ 's) |
| NOM | nominative case |
| ACC | accusative case |
| NOMACC | nominative or accusative case |
| NEG | negation |
| PASSIVE | passive form (for "born") |
| to | contracted form verb + to |
| COMP | comparative |
| SUPER | superlative |
| MASC | masculine |
| FEM | feminine |
| NEUT | neuter |
| WH | wh-word |
| REFLEXIVE | reflexive |
| REF1SG | 1st person singular referent |
| REF2ND | 2nd person referent |
| REF2SG | 2nd person singular referent |
| REF2PL | 2nd person plural referent |
| REF3SG | 3rd person singular referent |
| REF3PL | 3rd person plural referent |
| REFERENCE | masculine referent |
| REFERENCE | feminine referent |
Nous présentons un analyseur morphologique de l'anglais. Les tables morphologiques incluent plus de 317000 formes fléchies, dérivées de 90000 racines.

Les tables ont été construites à l'aide de dictionnaires électroniques (en particulier “Collins Dictionary of the English Language, 1979 edition”) distribuées par ACL-DCI (Liberman, 1989).

Les tables sont disponibles dans deux formats. Le premier format peut être utilisé avec un analyseur morphologique à deux niveaux tel que PC-KIMMO (Antworth, 1990). Dans le deuxième format, toutes les formes fléchies ont été insérées dans une base de donnée sur disque à l'aide d'un utilitaire sur UNIX (Seltzer et Yigit, 1991). Un outil pour X Window permet d'accéder et de modifier cette base de données est aussi disponible.

L'analyseur peut être utilisé par un autre programme tel qu'un analyseur syntaxique. Les tables peuvent être accédées en Lisp et C.

### Tables pour PC-KIMMO

Nous avons utilisé les règles morphologiques de l'anglais écrites par Karttunen et Wittenburg (1983). A l'aide de ces règles et de dictionnaires, nous avons créé des lexiques qui peuvent être utilisés par PC-KIMMO (une implementation d'un analyseur morphologique à deux niveaux (Antworth, 1990)). La Table 1 comporte le nombres de racines ainsi que le nombres de formes fléchies qui peuvent être reconnues.

| Categories      | Racines | Formes fléchies |
|-----------------|---------|-----------------|
| Pronom (Pron)   | 92      | 93              |
| Préposition (Prep) | 148   | 150             |
| Déterminant (D) | 100     | 100             |
| Conjonction (Conj) | 64    | 64              |
| Adverbe (Adv)   | 6992    | 7176            |
| Nom (N)         | 50370   | 199303          |
| Adjectif (A)    | 20550   | 65146           |
| Verbe (V)       | 11880   | 45445           |
| TOTAL           | 90196   | 317477          |

Figure 1: Nombre de Racines et de Formes Fléchies.

### Base de Données

PC-KIMMO charge la totalité du lexique en mémoire sous la forme d'une structure de données qui permet de factoriser les prefixes communs des mots. Avec nos lexiques chargés, PC-KIMMO occupe environ 19 mega octets. L'espace mémoire est trop important et de plus le temps d'accès n'est pas satisfaisant.

Nous avons donc compilé toutes les formes fléchies sous forme de base de donnée sur disque avec l'aide d'un utilitaire UNIX (Seltzer et Yigit, 1991). Cette utilitaire permet d'éliminer PC-KIMMO tout en réduisant l'espace mémoire (200 kilo octets) et le temps d'accès (entre 6 et 10 millième de secondes).

Ces tables sont maintenues sous forme de base de données et aussi sous forme de texte. Des programmes permettent la transformation de ces tables d'une forme à l'autre. Nous avons écrit un outil pour X Window (Figure 2) qui permet d'accéder et de modifier cette base de données est aussi disponible.

![Figure 2: Utilitaire pour la Base de Données Morphologiques](image)

### Distribution

Nous distribuons ces tables ainsi que les utilitaires sans frais avec un contrat de non-commercialisation. Veuillez contacter par courrier électronique zaidel@cis.upenn.edu ou écrire à l'une des personnes suivantes: Yves Schabes, Martin Zaidel ou Dania Egedi.