A Dictionary and Morphological Analyser for English

G.J. Russell
S.G. Pulman

Computer Laboratory,
University of Cambridge

G.D. Ritchie
A.W. Black

Department of
Artificial Intelligence,
University of Edinburgh

1. Introduction and Overview

This paper describes the current state of a three-year project aimed at the development of software for use in handling large quantities of dictionary information within natural language processing systems.[1] The project was accepted for funding by SERC/Alvey commencing in June 1984, and is being carried out by Graeme Ritchie and Alan Black at the University of Edinburgh and Steve Pulman and Graham Russell at the University of Cambridge. It is one of three closely related projects funded under the Alvey IKBS Programme (Natural Language Theme); a parser is under development at Edinburgh by Henry Thompson and John Phillips, and a sentence grammar is being devised by Ted Briscoe and Clare Grover at Lancaster and Bran Boguraev and John Carroll at Cambridge. It is intended that the software and rules produced by all three projects will be directly compatible and capable of functioning in an integrated system.

Realistic and useful natural language processing systems such as database front-ends require large numbers of words, together with associated syntactic and semantic information, to be efficiently stored in machine-readable form. Our system is intended to provide the necessary facilities, being designed to store a large number (at least 10,000) of words and to perform morphological analysis on them, covering both inflectional and derivational morphology. In pursuit of these objectives, the dictionary associates with each word information concerning its morphosyntactic properties. Users are free to modify the system in a number of ways; they may add to the lexical entries Lisp functions that perform semantic manipulations, and tailor the dictionary to the particular subject matter they are interested in (different databases, for example). It is also hoped that the system is general enough to be of use to linguists wishing to investigate the morphology of English and other languages. Contents of the basic data files may be altered or replaced:

1. A 'Word Grammar' file contains rules assigning internal structure to complex words.
2. A 'Lexicon' file holds the morpheme entries which include syntactic and other information associated with stems and affixes.
3. A 'Spelling Rules' file contains rules governing permissible correspondences between the form of morphemes listed in the lexicon and complex words consisting of sequences of these morphemes.

Once these data files have been prepared, they are compiled using a number of pre-processing functions that operate to produce a set of output files. These constitute a fully expanded and cross-indexed dictionary which can then be accessed from within LISP.

The process of morphological analysis consists of parsing a sequence of input morphemes with respect to the word grammar. It is implemented as an active chart parser (Thompson & Ritchie (1984)), and builds a structure in the form of a tree in which each node has two associated values, a morphosyntactic category, and a rule identifier.

The system is written in FRANZ LISP (opus 42.15) running under Berkeley 4.2 Unix. Future developments will concentrate on improving its efficiency, in particular by restructuring the code. We also hope to produce an implementation in C, which should offer a faster response time.

2. Linguistic Assumptions

The grammatical framework underlying the linguistic aspects of the system is that of Generalized Phrase Structure Grammar, as set out in Gazdar et al. (1985). Morphological categories employed here correspond to the syntactic categories in that work, and the type of syntactic information present in dictionary entries is intended to facilitate the use of the system as part of a more general GPSG-based program. In developing our prototype, we have adopted many of the proposals made in that work. To that extent, certain assumptions about a correct analysis of English sentence syntax are built in to the lexical entries, but this should not preclude adaptation by users to suit different analyses.

Following what has become a general assumption in syntactic theory, we take the major lexical categories to be partitioned into four classes by the two binary-valued features [± N] and [± V]. The major lexical categories have phrasal projections; these are distinguished from their lexical counterparts by their value for the feature BAR. Lexical categories have the value 0, and phrasal categories (including sentences) have the value 1 or 2. Thus, a Noun Phrase is of the category:

((V -) (N +) (BAR 2))

In our analysis, 'bound morphemes', that is to say prefixes and suffixes, are distinguished from others by their BAR specification; the suffix *ing* is the sole member of the category:

((V +) (N -) (VFORM ING) (BAR -1))

As in other GPSG-based work, our analysis encodes the subcategorizational properties of lexical items in the value of a feature SUBCAT. Transitive verbs such as *devour* are specified as (SUBCAT NP), and intransitives such as *elapse* as (SUBCAT NULL).

As an example from the current analysis of how the system can operate to produce well-formed words, consider the familiar fact of English morphology that no word may contain more than one inflection. The word grammar must permit both *walked* and *walking*, but not *walkinged*. This is achieved by restricting the distribution of inflectional suffixes so that they attach to non-inflected stems only. A general statement of this type of restriction is made in terms of a feature INFL: stems specified as (INFL +) may take an inflectional suffix, while those specified as (INFL -) may not. The STEM feature described in section 4 provides one means of enforcing correct stem-affix combinations; if the suffixes *ed* and *ing* are specified with (STEM ((INFL +))), they

[1] This work is supported by SERC/Alvey grant number GR/C/79114.

will attach only to categories which include the specification (INFL +). *Walk*, as a regular verb, is so specified; *walked* and *walking* are therefore accepted. *Ed*, *ing*, other inflectional suffixes, and irregular (i.e. uninflectable) words, however, are specified as (INFL -). Our grammar assigns a binary structure to the words in question. In order for this method to prevent e.g. *walkinged*, the stem *walking* must also bear the (INFL -) specification. This it does, since we regard suffixes as being the head of a word, and as contributing to the categorial content of the word as a whole. If the INFL specification of the suffix is copied into the mother category, the STEM specification of a further suffix will not be satisfied. See section 4 for more discussion of these matters.

3. The Lexicon

The lexicon itself consists of a sequence of entries, each in the form of a Lisp s-expression. An entry has five elements: (i) and (ii) the head word, in its written form and in a phonological transcription, (iii) a 'syntactic field', (iv) a 'semantic field', and (v) a 'user field'. The semantic field has been provided as a facility for users, and any Lisp s-expression can be inserted here. No significant semantic information is present in our entries, beyond the fact that e.g. *better* and *best* are related in meaning to *good*.

Similarly, the user field is unexploited, being occupied in all cases by the atom 'nil'. It serves primarily as a place-holder, in that, while it is desirable to maintain the possibility for users to include in an entry whatever additional information they desire, the form which that information might take in practice is clearly not predictable.

The syntax field consists of a syntactic category, as defined by Gazdar et al. (1985), i.e. a set of feature-value pairs. Some of these are relevant only to the workings of the word grammar, and may thus be ignored by other components in an integrated natural language processing system. Their purpose is to control the distribution of morphemes in complex words, as described in the following section.

The content of a syntax field is often at least partially predictable. This fact allows us to employ as an aid to users wishing to write their own dictionary rules which add information to the lexicon during the compilation process. Recall that, in our analysis of English, the inflectability of a word is governed by the value in that word's category for INFL. Completion Rules (CRs) can be written that will add the specification (INFL -) to any entry already including (PLU +) (for e.g. *men*), (AFORM ER) (for e.g. *worse*), (VFORM ING), etc., thus removing the need to state individually that a given word cannot be inflected.

A second means of reducing the amount of preparatory work is provided in the form of Multiplication Rules (MRs). Whereas CRs add further specifications to a single entry, MRs have the effect of increasing the number of entries in some principled way. One application of MRs is to express the fact that nouns and adjectives do not subcategorize for obligatory complements. A MR can be written which, for each entry containing the specification (N +) and some non-NULL value for SUBCAT, produces a copy of that entry where the SUBCAT specification is replaced by (SUBCAT NULL).

The lexicon compiles into two files, one holding morphemes stored in a tree-shaped structure (cf. Thorne et al. (1968)), and the other holding the expanded entries relating to them. The compilation of a lexicon can take a considerable amount of time; our prototype incorporates a lexicon with approximately 3500 entries, which compiles in approximately ninety minutes.

4. The Word Grammar

The internal structure of words is handled by a unification feature grammar with rules of the form:

$$\text{mother} \rightarrow \text{daughter}_1\ \text{daughter}_2\ \ldots$$

where 'mother', 'daughter$_1$', etc. are categories. A rule which adds the plural morpheme to a noun might be given as shown below:

```
((BAR 0) (V -) (N +) (PLU +) (INFL -)) =>
            ((BAR 0) (V -) (N +) (INFL +))
            ((BAR -1) (V -) (N +) (PLU +) (INFL -))
```

The system provides two methods of writing rules in a more general form; variables and feature-passing conventions.

In our grammar, the category and inflectability of a suffixed word are determined by the category and inflectability of the suffix; in the rule below, ALPHA, BETA, and GAMMA are variables ranging over the set of values {+, -}:

```
((V ALPHA)(N BETA)(INFL GAMMA)(BAR 0)) =>
     ((BAR 0))
     ((V ALPHA)(N BETA)(INFL GAMMA)(BAR -1))
```

Since variables are interpreted consistently throughout a rule, the mother category and suffix will be identical in their specifications for N, V and INFL.

As an alternative to variables, feature passing conventions are also available. These relate categories in what Gazdar et al. (1985) term 'local trees', i.e. sections of morphological structure consisting of a mother category and all of its immediate daughters. The conventions refer to 'pre-instantiation' features; these are features present in the categories mentioned in the relevant rule. 'Extension' and 'unification' are meant in the sense of Gazdar et al. (1985), q.v.

The Word-Head Convention:

> After instantiation, the set of WHead features in the mother is the unification of the pre-instantiation WHead features of the Mother with the pre-instantiation WHead features of the Rightdaughter.

This convention is analogous to the simplest case of the Head Feature Convention in Gazdar et al. (1985). Although there is no formal notion of 'head' in the system, this convention embodies the implicit claim that the head in a local tree is always the right daughter. If the daughters are a prefix and a stem (as in e.g. *re-apply*), the WHead features of the stem are passed up to the mother. Features encoding morphosyntactic category can be declared as members of the WHead set, and *re-apply* is then of the same category as, and shares various sentence-level syntactic properties with, *apply*. If the daughters are a stem and a suffix, the category of the mother is determined not by the stem, but rather by the suffix. For example, *possible* and *ity* may be combined to form *possibility*, whose 'nouniness' is due to the category of the suffix.

The Word-Daughter Convention:

(a) If any WDaughter features exist on the Right-daughter then the WDaughter features on the Mother are the unification of the pre-instantiation WDaughter features on the Mother with the pre-instantiation WDaughter features on the Right-daughter.

(b) If no WDaughter features exist on the Right-daughter then the WDaughter features on the Mother are the unification of the pre-instantiation WDaughter features on the Mother with the pre-instantiation WDaughter features on the Left-daughter.

The subcategorization class of a word remains constant under inflection, but is likely to be changed by the attachment of a derivational suffix. Moreover, the subcategorization of a prefixed word is the same as that of its stem. The WDaughter convention is designed to reflect these facts by enforcing a feature correspondence between one of the daughters and the mother. When the feature set WDaughter is defined as including the subcategorization feature SUBCAT, the convention results in configurations such as:

```
((SUBCAT NP))             ((SUBCAT NP))
      ((V +)(N +))              ((SUBCAT NP))
      ((SUBCAT NP))             ((VFORM ING))
```

which show the relevant feature specifications in local trees arising from suffixation of an adjective with *+ize* to produce a transitive verb and suffixation of a transitive verb with *+ing* to produce a present participle.

The Word-Sister Convention:

When one daughter is specified for STEM, the category of the other daughter must be an extension of the value of STEM.

The purpose of this third convention is to allow the subcategorization of affixes with respect to the type of stem they may attach to. The behaviour of affixes that attach to more than one category can be handled naturally by giving them a suitable specification for STEM. If it is desired to have *anti-* attached to both nouns and adjectives, for example, the specification (STEM ((N +))) will have that effect, since both adjectives and nouns are extensions of the category ((N +)).

The user can define the sets WHead and WDaughter as he wishes, or, by leaving them undefined, avoid their effects altogether. The feature STEM is built in, and need not be defined. The effects of the Word-Sister Convention can be modified by changing the STEM specifications in the lexical entries, and avoided by omitting them.

5. The Spelling Rules

The rules are based on the work of Koskenniemi (1983a, 1983b, Karttunen 1983), though their application here is solely to the question of 'morphographemics'; the more general morphological effects of Koskenniemi's rules are produced differently. The current version of the system contains a compiler allowing the rules to be written in a high level notation based on Koskenniemi (1985). Any number of spelling rules can be employed, though our system has fifteen. They are compiled during the general dictionary pre-processing stage into deterministic finite state transducers, of which one tape represents the lexical form and the other the surface form.

The following rule describes the process by which an additional *e* is inserted when some nouns are suffixed with the plural morpheme *+s*:

```
Epenthesis
+:e <=> { < s:s h:h > s:s x:x z:z } --- s:s
or < c:c h:h> --- s:s
```

The epenthesis rule states that *e* must be inserted at a morpheme boundary if and only if the boundary has to its left *sh*, *s*, *x*, *z* or *ch* and to its right *s*. The interpretation of the rule is simple; the character pair ('lexical character : surface character') to the left of the arrow specifies the change that takes place between the contexts (again stated in character pairs) given to the right of the arrow. Braces ('{','}') indicate disjunction and angled brackets indicate a sequence. Alternative contexts may be specified using the word 'or'. Lexical and surface strings of unequal length can be matched by using the null character '0', and special characters may be defined and used in rules, for example to cover the set of alphabetic characters representing vowels.

The spelling rules are able to match any pair of character strings. It would for example be possible to analyse the suppletive *went* as a surface form corresponding to the lexical form *go+ed*. In this case, four rules would be needed to effect the change, and a better solution is to list *went* separately in the lexicon. In practice, the choice between treating this type of alternation dynamically, with morphological and spelling rules, and statically, by exploiting the lexicon directly, depends on the user's idea of which is the more elegant solution. While elegance may be in the eye of the beholder, computational efficiency is unfortunately not. It will generally be more efficient to list a word in the lexicon than to add spelling or morphological rules specific to small number of cases.

References

Gazdar, G., E. Klein, G.K. Pullum, and I.A. Sag (1985) *Generalized Phrase Structure Grammar.* Oxford: Blackwells.

Karttunen, L. (1983) "KIMMO - A General Morphological Processor", in *Texas Linguistic Forum 22*, 165 - 186. Department of Linguistics, University of Texas, Austin, Texas.

Koskenniemi, K. (1983a) "Two-level model for morphological analysis", in *Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe*, 683 - 685.

Koskenniemi, K. (1983b) *Two-level Morphology: a general computational model for word-form recognition and production.* Publication No. 11, University of Helsinki, Finland.

Koskenniemi, K. (1985) "Compilation of Automata from Two-level Rules", talk given at the Workshop on Finite-State Morphology, CSLI, Stanford, July, 1985.

Thompson, H. and G.D. Ritchie (1984) "Implementing Natural Language Parsers", in T. O'Shea and M. Eisenstadt (eds.) *Artificial Intelligence: Tools, Techniques and Applications.* New York: Harper and Row.

Thorne, J.P., P. Bratley, and H. Dewar (1968) "The syntactic analysis of English by machine", in D. Michie (ed.) *Machine Intelligence 3.* Edinburgh: Edinburgh University Press.