MORPHĒ: A Practical Compiler for Reversible Morphology Rules

John R. R. Leavitt
Center for Machine Translation
Carnegie Mellon University
Pittsburgh, PA 15213
`jrrl@cs.cmu.edu`

Abstract

Morphē is a Common Lisp compiler for reversible inflectional morphology rules developed at the Center for Machine Translation at Carnegie Mellon University. This paper describes the Morphē processing model, its implementation, and how it handles some common morphological processes.

1 Introduction

The input to the Morphē rule compiler is a rule file containing inflection rules, the specification of a discrimination network of morphological forms, and definitions of certain classes of strings and string mappings. This rule file can be compiled into either a word generation program or a word parsing program. The word generation program produces an inflected surface form from a feature structure,[1] while the word parsing program takes an inflected form and produces a set of feature structures for valid parses.[2]

2 The Processing Model

In Morphē the process of inflection is seen as consisting of two basic steps:

1. By making a series of feature- and orthographically-based decisions, choose an inflection procedure.
2. Apply that procedure to the uninflected root.

To implement the first step, Morphē uses a feature-based discrimination network with orthographically-based inflection rules at the leaves. Each node in the discrimination network specifies a set of features common to all of its descendants. For example, at the top of a subtree for nouns, a node might contain the features { `(cat noun)` } which would be inherited by the nodes for single-noun and plural-noun, and so on.

That Morphē explicitly divides feature-based decisions from orthographic decisions has two important consequences:

- The type of feature that may be checked in the discrimination network is not restricted. For instance, phonological and/or morphological features (e.g. paradigm) can be checked alongside syntactic features (e.g. category).
- A single morpheme can be split across several leaf nodes if feature tests below the morpheme level are necessary.

2.1 The Rule Formalism

As shown in Figure 1, a rule consists of a set of clauses, each of which contains orthographic pattern on the left-hand side and a set of inflection operations on the right-hand side.

- *Orthographic patterns.* The orthographically-based decisions are made by matching against regular expression-based patterns. Standard regular expression operations (i.e. Kleene closure, wildcards, etc.) are included. In addition, non-standard operations for matching against a pre-defined class of strings[3], and binding and retrieval of portions of the word[4] are included.
- *Inflection Operations.* The application of the inflection procedure is implemented as the sequential execution of the inflection operations in the right-hand side. The inflection operations include affixation, deletion, and the combined operation of "replacement" in prefix, suffix, and infix positions. Also included is an operation for performing regular string-to-string mapping within a word.[5]

3 Processing

During generation, processing begins with a feature structure entering the tree at the root node, and trickling down to the appropriate leaf node. Once at the leaf node, the word root is compared against each clause's orthographic pattern in turn. When a match is found, the inflection procedure for that clause is applied to the word root and the result is returned.

During parsing, processing begins with an inflected form entering the tree at each leaf node where the inflection rules are applied "in reverse" and the non-passing results discarded. Applying a rule "in reverse" means that the word is matched

[1] These feature structures are structurally the same as those used by the Generalized LR Parser/Compiler [Tomita et al., 1988] and the Generation Kit [Tomita and Nyberg, 1988], and can contain non-syntactic features.

[2] A detailed description of the Morphē program and rule file formalism and some example rule files are given in [Leavitt, 1992].

[3] This class matching is equivalent "alphabet subsets" in KIMMO [Karttunen et al., 1983], "restricted variables" in NABU [Slocum, 1988], and "string vars" in DIMORPH [Gibson, 1991].

[4] These mechanisms are similar to the binding and retrieval mechanisms used in Unix utilities such as "sed".

[5] String-to-string mapping is roughly equivalent to the "pairing-up [of] variables" in NABU.

```
(leaf-rule v+pres-part
  (((:or "x" "y") $)
   (+s "ing"))           ; verbs like perplex & carry
  ((C V (% GC) $)
   (+s %1 "ing"))        ; verbs like cut
  ((C "e" $)
   (rs "e" "ing"))       ; verbs like make
  ((C "ie" $)
   (rs "ie" "ying"))     ; verbs like die
  (:otherwise
   (+s "ing")))          ; verbs like dent
```

Figure 1: Inflection Rule for English Present Participle

against the inflected forms and the operations perform deinflection, rather than vice versa. After all clauses in all leaves have been tried, and presumably most results have been discarded, each remaining parse follows the network upwards, collecting the features of each node it traverses until a set of full feature structures arrives at the root node. When this process is finished, a lexicon check is made to ensure that only valid words (of the proper category, paradigm, etc.) are kept.

4 Handling Common Morphological Processes

This section explains how common morphological processes are handled by Morphē.

- *Affixation.* Prefixation, suffixation, and infixation are handled directly by the +p, +s, and +i inflection operators. To determine the insertion point, infixes must be placed either before or after some portion of the word that was bound during pattern matching.
- *Deletion.* Word initial, word final, and word internal deletion are handled directly by the -p, -s, and -i inflection operators. As with infixation, some bound part of the word must act as an "anchor" for the deletion.
- *Gemination and Reduplication.* Since expressions may be bound during pattern matching, bound expressions can be affixed to the word to create the effects of gemination or reduplication. For example, when forming the present participle, certain English verbs repeat the final consonant before adding the suffix "ing" (e.g. "cut" → "cutting"). This simple twinning is encoded by the third clause in the above sample rule. Reduplication, as found in Warlpiri [Sproat and Brunson, 1988], or Latin [Matthews, 1974], can be handled in a similar manner (i.e. by binding the appropriate portion of the root and retrieving it during affixation).
- *Paradigmatic Alternation.* Alternations that consists of a single mapping of one string to another, such as the "-fe/-ve" alternation for the plural of English nouns like "wife" or "knife" can be handled by a single replacement operation. Alternations that consist of a number of related alternation, such as the {"-us/-i" "-um/-a" "-a/-ae"} alternation for the plural of English nouns like "octopus", "spectrum", and "vertebra" could be handled as separate cases, but it is convenient to be able to refer to the entire class of alternations. The map operator invokes a string-to-string mapping on a bound portion of a word.[6] Alternations such as vowel rounding in the comparative forms of German adjectives, and consonant and vowel alternation in Rumanian, can be handled by this method.
- *Suppletion.* Morphē currently handles suppletion by requiring suppletive forms (e.g. "went" for "go") to be included in the lexicon. In this, it is not unlike many other system, such as KIMMO and DIMORPH.

5 Current Uses and Future Research

Morphē is presently being used for French and German generation morphology in the Kant project, a knowledge-based machine translation system being developed at Carnegie Mellon University [Mitamura et al., 1991]. In addition, a rule file has been developed for English and one is currently being designed for Spanish. Future research will be directed towards morphological phenomena that cannot currently be handled in an elegant fashion. Certain types of suppletion, such as irregular stems with regular endings in Latin, should be handled more generally and with less reliance on the lexicon as a storehouse of irregularities. In addition, the design of mechanisms appropriate to the handling of prosodic inflection will also be investigated.

6 Acknowledgments

I would like to thank Alex Franz, Nicholas Brownlow, and Deryle Lonsdale for their comments on drafts of this paper.

References

[Gibson, 1991] Gibson, E. (1991). DIMORPH: A Morphological Analyzer. Technical Report CMU-CMT-91-128, Center for Machine Translation, Carnegie Mellon University.

[Karttunen et al., 1983] Karttunen et al., L. (1983). KIMMO: A two level morphological analyzer. In *Texas Linguistic Forum 22*.

[Leavitt, 1992] Leavitt, J. (1992). *The MORPHĒ User's Guide*. Center for Machine Translation, Carnegie Mellon University.

[Matthews, 1974] Matthews, P. (1974). *Morphology*. Cambridge University Press, Cambridge, England.

[Mitamura et al., 1991] Mitamura, T., Nyberg, E., and Carbonell, J. (1991). An Efficient Interlingua Translation System for Multilingual Document Production. In *Proceedings of Machine Translation Summit III*.

[Slocum, 1988] Slocum, J. (1988). Morphological processing in the NABU system. In *Proceedings of the Second Applied Natural Language Processing Conference*, pages 228–234.

[Sproat and Brunson, 1988] Sproat, R. and Brunson, B. (1988). Constituent-Based Morphological Parsing: A New Approach to the Problem of Word-Recognition. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*.

[Tomita et al., 1988] Tomita, M., Mitamura, T., and Kee, M. (1988). *The Generalized LR Parser/Compiler User's Guide*. Center for Machine Translation, Carnegie Mellon University.

[Tomita and Nyberg, 1988] Tomita, M. and Nyberg, E. (1988). *The Generation Kit and the Transformation Kit*. Center for Machine Translation, Carnegie Mellon University.

[6] It should be noted that binding the appropriate portion of the word is a nontrivial task and may require a place marker in the root to help the pattern locate it. This is, however, an artifact of language, not the inflection model.