Morphology and Computation

Richard Sproat
(AT&T Bell Laboratories)

Cambridge, MA: The MIT Press (The
ACL–MIT Press Series in Natural
Language Processing), 1992, xv +
295 pp.
Hardbound, ISBN 0-262-19314-0, $35.00

Reviewed by
Scott Stethem
Pacific Bell

This book undertakes the task of explaining the importance of an often-neglected field, computational morphology, while illustrating the field's large potential contribution to natural language applications. The intended audience is those with backgrounds in computer science, including automata and complexity theory, and linguistics, especially phonology and syntax. Familiarity with morphological concepts is not required, as Sproat includes a nice introduction to word structure and word formation. He draws upon a plethora of languages, using English to illustrate morphological phenomena where possible. The presentation varies from quite theoretical, with discussions of NP-hardness and finite-state machines, to highly 'commercial' (runs on machine ABC with operating system XYZ in *t* seconds).

Sproat begins with a motivation for studying computational morphology. Since most natural language systems work on limited subsets of just one language, system developers make numerous limiting assumptions about word formation and structure. The author mentions numerous applications (e.g., text-to-speech, word processing, document retrieval, machine translation) where morphology can contribute. I would have liked more detailed presentations of applications, emphasizing the practicality of morphological processing.

Seemingly simple questions like "What is a word?" have surprisingly complex answers. Chapter 2 nicely introduces the field of morphology, serving as a review for some, while being a fast-paced introduction for novices (listing excellent references for further study). Sproat gives due consideration to how morphological processes relate to orthography, phonology, syntax, and semantics.

Sproat's survey of morphology leads us through examples and theories of word structure and word formation. Word formation can be inflectional (i.e., using syntactic information like part of speech), derivational (e.g., *edit* into *editor*), and/or compounding (*dog house*). Additionally, the author goes beyond mundane examples of prefix/suffix concatenation, delving into infixes and such "exotic" phenomena as:

- circumfixation (i.e., discontinuous affixes);
- templates (e.g., the root-and-pattern of Semitic languages); and
- reduplication.

Chapter 3's discussion of computational morphology (CM) systems should have presented an objective comparison of how each system handles the various morphological phenomena of Chapter 2. Instead, the author spends the majority of this chapter presenting highly detailed descriptions of two-level finite-state morphology systems (collectively called URKIMMO). Despite the inordinate attention, however, he acknowledges the limitations of this approach to morphology. Sproat goes on record as saying "I do not believe for a moment that KIMMO-type systems are particularly psychologically realistic" (p. 183). Also, Sproat is dissatisfied with URKIMMO's concatenative premise; that is, the system's tremendous bias toward prefixes and suffixes. The details about finite-state morphotactics and phonology should have been relegated to an appendix. The computational complexity of CM systems like URKIMMO, while informative, belongs in an appendix as well.

The latter part of Chapter 3 discusses non-KIMMO morphology systems, including:

- a text-to-speech word decomposition module;
- Turkish spelling checking;
- a Spanish morphological analyzer; and
- word-based systems for on-line dictionaries.

The author goes on to identify major trends within computational morphology, and the areas needing more attention (i.e., the boundaries between morphology and other fields). For morphology's interaction with cyclic and lexical phonology, Sproat's comment on URKIMMO is "I do not see any straightforward solution to [cyclic rule application] within the two-level framework" (p. 211). Collaboration between morphology and syntax will be needed when "serious morphological analyzers are integrated into serious parsing systems" (p. 212).

The first part of the next chapter deals with morphological acquisition, determining the rules that govern word structure and word formation. Sproat discusses whether connectionism is contributing to this branch of computational morphology, mentioning a connectionist network that "learns" past-tense forms of English verbs. Sproat states "Rumelhart and McClelland 1986 is one of the better-known, and arguably one of the more successful, applications of connectionist models to natural language" (p. 216). Yet after presenting their system, he concludes that "there are a number of reasons to doubt that Rumelhart and McClelland's model is really doing what it is claimed to be doing" (p. 230). One reason cited that their system does not realistically model human language learning is that it sometimes produces bizarre past-tense verb forms, such as *membled* for *mail*.

The second part of this chapter revisits compound nominals, showing how CM systems can handle constructs like *life insurance company employee*. Representing knowledge about the world (or a domain) allows a powerful interaction between morphology and semantics. Examples are given of a CM system that allows compounds in database queries and a program that decides where to assign stress in a text-to-speech system.

The book *Morphology and Computation* shows that computational morphologists have only begun to explore the field. With new understandings in morphology and the related disciplines (e.g., syntax, semantics, phonology, psychology, and computer science), more systems are needed "to verify the adequacy of particular theoretical models" (p. 19). To create more robust CM systems, partnerships are needed with researchers having diverse backgrounds. Evolving from these collaborations, the next generation of morphological processing systems looks promising indeed.

Reference

Rumelhart, David E., and McClelland, James L. (1986). "On learning the past tenses of English verbs." In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Volume 2, edited by James L. McClelland, David E. Rumelhart, and the PDP Research Group, 216–271. The MIT Press.

Scott Stethem is a systems analyst at Pacific Bell, where he develops intelligent front-ends to databases. He has degrees in computer science from the University of Pennsylvania and the Johns Hopkins University. He presented the paper "Explanation-based learning from rule-governed features in phonological representations" at the 1991 AAAI Spring Symposium on *Machine Learning of Natural Language and Ontology*. Stethem's address is: 2600 Camino Ramon, Pacific Bell 4s450, San Ramon, CA 94583. E-mail: sesteth@srv.pacbell.com