

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/252133485>

# Computational Approaches to Morphology and Syntax

Article · January 2007

CITATIONS

117

READS

1,143

2 authors, including:



**Richard Sproat**

Google Inc.

200 PUBLICATIONS 6,804 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Language use in Autism Spectrum Disorders [View project](#)

# Computational Approaches to Morphology and Syntax

**Brian Roark and Richard Sproat**

(Oregon Health and Science University and University of Illinois at Urbana–Champaign)

Oxford: Oxford University Press (Oxford surveys in syntax and morphology, edited by Robert D. Van Valin Jr, volume 4), 2007, xx+316 pp; hardbound, ISBN 978-0-19-927477-2, \$110.00, £60.00; paperbound, ISBN 978-0-19-927478-9, \$45.00, £24.99

*Reviewed by*

*Noah A. Smith*

*Carnegie Mellon University*

Brian Roark and Richard Sproat have written a compact and very readable book surveying computational morphology and computational syntax. This text is not introductory; instead, it will help bring computational linguists who do not work on morphology or syntax up to date on these areas' latest developments. Certain chapters (in particular, Chapters 2 and 8) provide especially good starting points for advanced graduate courses or seminars. The text is divided into an Introduction and Preliminaries chapter, four chapters on computational approaches to morphology, and four chapters on computational approaches to syntax. The morphology chapters focus primarily on formal and theoretical issues, and are likely to be of interest to morphologists, computational and not. The syntax chapters are driven more by engineering goals, with more algorithm details. Because a good understanding of probabilistic modeling is assumed, these chapters will also be useful for machine learning researchers interested in language processing.

Despite the authors' former affiliations, this book is *not* an AT&T analogue of Beesley and Karttunen's (2003) pedagogically motivated text on the Xerox finite-state tools. This text is not about the AT&T FSM libraries or the algorithms underlying them (cf. Roche and Schabes 1997).

## 1. Chapter 1: Introduction and Preliminaries

The first chapter is a take-no-prisoners introduction to finite-state automata and transducers and their semiring-weighted generalizations. Algorithms (e.g., for FST composition) are discussed but not presented in detail. Epsilon removal, minimization, and determinization are mentioned but not defined. This material is probably too cursory to serve as a lone introduction for those wishing to fully understand weighted FSTs, but that lack of understanding will not be an impediment in the ensuing chapters because weights do not re-surface until chapter 6, in the context of  $n$ -gram models, and even then the algebraic view given here is not mentioned.

The chapter concludes with a defense of the place of finite-state models in linguistics, followed by a clear explanation of the trade-offs in computational linguistics (e.g., between computational cost, expressive power, and annotation cost).

## 2. Part I (Chapters 2–5): Computational Approaches to Morphology

These chapters are primarily an argument for the effectiveness of finite-state transducers in modeling natural language morphology.

Chapter 2 provides a laundry list of morphological phenomena, arguing that finite-state composition captures each of them, even in cases where there is a more obvious solution (e.g., finite-state concatenation for concatenative phenomena). Examples of many kinds of phenomena are given from diverse languages: prosodic restrictions in Yowulmne, phonological effects of German affixes, and subsegmental morphology in Welsh, to name a few. Importantly, the *compile-reduce* and *merge* operations are argued to be syntactic sugar for effects achievable by finite-state composition, so that even root-and-pattern Arabic morphology is explained in the same algebraic framework.

Reduplication effects, of course, challenge finite-state explanations, and so receive their own section. Extended (non-regular) computational models are presented alongside data from Gothic, Dakota, and Sye. The authors speculate that, in contrast with the commonly accepted Correspondence theory, Morphological Doubling theory (Inkelas and Zoll 1999), if correct, would imply that a non-regular “copying” process is not at work in reduplication. It is at this point that the reader may experience some discomfort; should the reduplication problem be addressed in syntax rather than morphology? Where exactly does the boundary lie? Readers hoping for a reassessment of this boundary, or even a new bridge over it, will not find it here.

Chapter 3 begins with Stump’s (2001) two-dimensional taxonomy of morphological theories, which appears rather divorced from the rich work on finite-state *computational morphology* in Chapter 2. The subtleties among the four types of theories (lexical vs. inferential and incremental vs. realizational, a more nuanced breakdown of the debate over “item-and-arrangement” vs. “item-and-process”) may be difficult to understand for the reader not trained in morphological theory, but resolution comes quickly. We are presented with a series of examples showing “proof-of-concept” fragmentary implementations (in AT&T’s *lextools*) of phenomena in Sanskrit, Swahili, and Breton to argue that lexical-incrementalist and inferential-realizational theories are computationally equivalent; both can be implemented using FSTs and can lead to the same models.

Chapter 4 gives an algebraic analysis of Koskenniemi’s (1983) “KIMMO” two-level morphological analysis system. Koskenniemi’s hand-coded morphology rules are argued to be a historical accident; if only computers had been more powerful in the 1980s, compilation of those rules into FSTs might have been automated, and in fact Kaplan and Kay had already developed the algorithms.<sup>1</sup> In the spirit of the previous chapter, Sproat and Roark also note that morphological accounts that use one, two, or more “cascaded” levels are all computationally equivalent rational relations under the finite-state approach, and that Optimality Theory can (under certain assumptions about constraints) be implemented with finite-state operations as well (Ellison 1994).

Chapter 5, “Machine learning of morphology,” focuses on unsupervised morphology induction methods. There is about a page of discussion on statistical language modeling approaches for disambiguation in agglutinative languages; no mention is made of the more recent use of discriminative machine learning in morphological disambiguation (Kudo, Yamamoto, and Matsumoto 2004; Habash and Rambow 2005; Smith, Smith, and Tromble 2005). The chapter focuses on the approaches of Goldsmith (2001), Schone and Jurafsky (2001), and Yarowsky and Wicentowski (2001). Although each approach is interesting on its own, little effort is made to unify work in this area, and none to bring the reader back full circle to finite-state models or the problem of inducing from data regular grammars (Stolcke and Omohundro 1993) or their weights

---

1 The authors rightly point out that Koskenniemi deserves much credit for building an implementation that aimed to have broad coverage, not merely a proof-of-concept.

(Eisner 2002). Another missed opportunity here is the recent introduction of Bayesian learning for word segmentation (Goldwater, Griffiths, and Johnson 2006).

Part I, in summary, aims to reduce many accounts of morphological phenomena to finite-state transducer composition, drawing on a wealth of illustrative examples. Twenty-two languages are listed in the language index at the end of the book, and, tellingly, all of them are discussed exclusively in Part I. These chapters are good diplomacy toward theoretical linguistics, showing how computational arguments can have theoretical implications.

### 3. Part II (Chapters 6–9): Computational Approaches to Syntax

In Part II, Roark and Sproat turn to models of syntax in computational linguistics. Because most research in this area has been on English, English parsing is what they present.

Chapter 6 covers finite-state approaches to syntax, including  $n$ -gram models and smoothing, class-based language models, hidden Markov models (though without a formal definition), part-of-speech tagging, log-linear models, and shallow parsing/chunking. The Forward, Viterbi, Viterbi  $n$ -best, Forward–Backward algorithms, and “Forward–Backward Decoding” (also known as posterior or minimum Bayes risk decoding) are covered with examples. This chapter is not as leisurely as the treatments of HMMs by Manning and Schütze (1999) or Charniak (1993), and it omits basic background on probabilistic modeling. For example, why must we ensure that an  $n$ -gram model’s total probability sums exactly to one? The answer relies on an understanding of perplexity and its use in evaluation, now in decline (cf. “stupid backoff” in Brants et al. 2007). The chapter does not reconnect with the algebraic view presented in Chapter 1; for example, the connection between HMMs and WFSAs is never expressed.

Chapter 7 introduces context-free grammars and their parsers, broken down into “deterministic” and “nondeterministic” approaches.<sup>2</sup> Probabilistic CFGs and treebanks are introduced informally alongside the latter, which may confuse some readers. Ambiguity is only presented as a natural phenomenon, not a problem of crude, over-generating grammars. The probabilistic CKY and Earley algorithms are presented. The Inside–Outside algorithm is presented in the context of Goodman’s (1996) maximum expected recall parsing (another instance of minimum Bayes risk). As in the case of the dynamic programming algorithms for HMMs in Chapter 6, the exposition is probably too brisk to be an introduction to the topic.

Chapter 8 contains a thoughtful discussion of many best practices in statistical parsing: treebank “decoration” techniques such as parent annotation and lexicalization, and the probability models underlying the parsers of Collins (1997) and Charniak (1997). Dependency parsing, unsupervised grammar induction, and finite-state approximations to PCFGs are allotted short sections.

Chapter 9 covers context-sensitive models of syntax. Unification-based parsing is presented at a high level, without formal details of unification or the differences between theories such as LFG and HPSG. The “lexicalized” models (TAG and CCG) are treated more thoroughly; pseudocode for a TAG dynamic programming parser is provided. There is brief treatment of Data-Oriented Parsing, reranking (a section that

---

2 These terms, though in wide use, are misnomers. All of these parsers are deterministic, since none involve randomness or nondeterministic behavior resulting from multiple processors. Here “(non)deterministic” refers to the *grammar*, not the parser.

would have been of more practical use in Chapter 8), and transduction grammars (i.e., grammars over more than one string, most frequently used in machine translation).

The abundance of dynamic programming algorithms in Part II leads to the question of whether such algorithms can be more easily taught (and unified) using recursive equations (Manning and Schütze 1999), or a more declarative framework (Shieber, Schabes, and Pereira 1995). Readers who prefer procedural pseudocode will find it here, though the book does not address implementation tricks for storing and indexing parse charts, or agenda-ordering methods to make parsing efficient.

These chapters are neither a gentle introduction to probabilistic modeling of syntax for linguists nor a handbook for the language engineer who wants to build an efficient, competitive parser. (There is also no advice on the relative merits of today's parsers available for download.) The audience that will find Part II most valuable will be researchers who understand the principles of probabilistic modeling but want a more up-to-date view of statistical parsing than offered by Manning and Schütze (1999), with more coverage of advanced topics than Jurafsky and Martin (2008). This group might include structured machine-learning researchers interested in the nuances of natural language parsing and computational linguists who do not work on syntax but want to keep up with the area.

#### 4. Conclusion

The two major parts of this book stand as clear, up-to-date, and concisely written summaries of particular sub-fields in computational linguistics: finite-state morphology and English syntactic processing. The book does a fine job of elucidating the trade-offs that make computational linguistics a tightrope act, and therefore serves as good diplomacy for researchers in related fields. At 112 and 146 pages, respectively, either of the parts is readable on a half-day plane or train trip.

Today, the strongest bridge between morphology and syntax is the Chomsky hierarchy, which is mentioned frequently in this book (but never depicted). The contrast between Parts I and II implies blueprints for more bridges: data resources to support more powerful learning algorithms for morphology (as we have seen in syntax), a stronger influence of non-English data on computational syntactic modeling (as we have seen in morphology), and practical ways to accomplish the amalgamation of morphology and syntax. This reviewer believes *Computational Approaches to Morphology and Syntax* will re-introduce the two sub-communities to each other and help each to leverage the successes of the other.

#### References

- Beesley, Kenneth R. and Lauri Karttunen. 2003. *Finite State Morphology*. CSLI Publications, Stanford, CA.
- Brants, Thorsten, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 858–867, Prague.
- Charniak, Eugene. 1993. *Statistical Language Learning*. MIT Press, Cambridge, MA.
- Charniak, Eugene. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 598–603, Providence, RI.
- Collins, Michael. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 16–23, Madrid.
- Eisner, Jason. 2002. Parameter estimation for probabilistic finite-state transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 1–8, Philadelphia, PA.
- Ellison, T. Mark. 1994. Phonological derivation in Optimality Theory. In

- Proceedings of the 15th International Conference on Computational Linguistics*, vol. 2, pages 1007–1013, Kyoto.
- Goldsmith, John. 2001. Unsupervised acquisition of the morphology of a natural language. *Computational Linguistics* 27(2):153–198.
- Goldwater, Sharon, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney.
- Goodman, Joshua. 1996. Parsing algorithms and metrics. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 177–183, Santa Cruz, CA.
- Habash, Nizar and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging, and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 573–580, Ann Arbor, MI.
- Inkelas, Sharon and Cheryl Zoll. 1999. Reduplication as morphological doubling. Technical report 412-0800, Rutgers Optimality Archive.
- Jurafsky, Daniel and James H. Martin. 2008. *Speech and Language Processing* (2nd edition). Prentice Hall, Upper Saddle River, NJ.
- Koskeniemi, Kimmo. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Ph.D. thesis, Department of General Linguistics, University of Helsinki, Helsinki, Finland.
- Kudo, Taku, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to Japanese morphological analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 230–237, Barcelona.
- Manning, Christopher D. and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.
- Roche, Emmanuel and Yves Schabes (editors). 1997. *Finite-State Language Processing*. MIT Press, Cambridge, MA.
- Schone, Patrick and Daniel Jurafsky. 2001. Knowledge-free induction of morphology using latent semantic analysis. In *Proceedings of the 5th Conference on Computational Natural Language Learning*, pages 67–72, Toulouse.
- Shieber, Stuart and Yves Schabes and Fernando C. N. Pereira. 1995. Principles and implementation of deductive parsing. *Journal of Logic Programming* 24(1–2):3–36.
- Smith, Noah A., David A. Smith, and Roy W. Tromble. 2005. Context-based morphological disambiguation with random fields. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 475–482, Vancouver.
- Stolcke, Andreas and Stephen Omohundro. 1993. Hidden Markov model induction by Bayesian model merging. In Stephen José Hanson, Jack D. Cowen, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems*, vol. 5. Morgan Kaufmann, San Mateo, CA, pages 11–18.
- Stump, Gregory T. 2001. *Inflectional Morphology: A Theory of Paradigm Structure*. Cambridge University Press, Cambridge, UK.
- Yarowsky, David and Richard Wicentowski. 2001. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, pages 207–216, Toulouse.

Noah A. Smith is an assistant professor at Carnegie Mellon University. He conducts research in statistical models and learning algorithms for natural language processing, including morphology and syntax, as well as applications such as machine translation and question answering. Smith's address is Language Technologies Institute, School of Computer Science, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213; e-mail: nasmith@cs.cmu.edu; URL: [www.cs.cmu.edu/~nasmith](http://www.cs.cmu.edu/~nasmith).

