Book Reviews

Finite State Morphology

Kenneth R. Beesley and Lauri Karttunen
(Xerox Research Centre Europe and Palo Alto Research Center)

Stanford, CA: CSLI Publications (CSLI
studies in computational linguistics,
edited by Ann Copestake) (distributed
by the University of Chicago Press),
2003, xviii+505 pp and CD-ROM;
hardbound, ISBN 1-57586-433-9, $85.00,
£59.50; paperbound, ISBN
1-57586-434-7, $40.00, £28.00

Reviewed by
Shuly Wintner
University of Haifa

Finite-state technology (FST) is a general term for the use of finite-state automata and transducers in computational linguistics and natural language processing (NLP). FST is very versatile, having been used very successfully for describing the phonology, orthography, and morphology of a large number of languages, as well as for solving practical problems such as morphological analysis and generation, language modeling for speech processors, shallow parsing, segmentation, and named-entity recognition. This technology is now very mature: Ever since it was observed by Johnson (1972) that the kind of phonological rules that are used by linguists denote, in fact, regular relations, and especially since the pioneering work of Koskenniemi (1983) and Kaplan and Kay (1994), much work has been invested in improving algorithms for finite-state networks and creating more regular-expression-like operators that can be compiled into finite-state networks.

It is therefore surprising that no textbook covering this technology in detail has previously been published: Beesley and Karttunen's *Finite State Morphology* is, to the best of my knowledge, the first textbook that is dedicated to this subject. Even general NLP textbooks spend relatively little space on FST (two pages in the case of Allen [1995], two sections out of twenty-five in the case of Jurafsky and Martin [2000]). The book is dedicated to a particular implementation, which comes with two regular expression languages, XFST and LEXC, and with compilers that can translate the expressions to extremely space- and time-efficient networks. Both systems were designed and implemented by Xerox and are provided (with compilations for Solaris, Linux, Windows, and Mac OS X) on a CD that accompanies the book.

The audience for the book is mainly linguists, and not necessarily computational linguists or computer scientists. The authors deliberately avoid mathematical definitions and specifications of algorithms, let alone proofs, in the text. However, mathematical correctness is not compromised, although it is doubtful whether readers with limited formal background would be able to appreciate it. Readers who *are* interested in the mathematics and in the computational aspects of the implementation will be left unsatisfied. Not only are they missing in the text, there are relatively few

references to such works (for example, there is hardly any reference to Mohri's [1997] works on sequential transducers, to Daciuk and others' works on incremental construction of lexicons [Daciuk et al. 2000], or to van Noord and Gerdemann's [2001a] work on transducers with predicates). There is also no mention of weighted finite-state networks and their uses in natural language processing.

For linguists, however, the book is full of useful advice. Not only does it provide a very gentle introduction to the field (chapter 1) and an extremely detailed and very well exemplified description of finite-state networks in general (chapter 2) and of the Xerox tools in particular (XFST in chapter 3 and LEXC in chapter 4), but it also provides invaluable insight into the process of developing large-scale finite-state networks, from the design and planning phase through maintenance, testing, and debugging (chapters 5 and 6). The core of the book consists of chapters 3 and 4, in which the authors describe in great detail the two main tools. Each and every operator is defined, explained, and demonstrated, usually with very illuminating linguistically motivated examples. The discussion is accompanied by useful exercises, many of which are solved in an appendix.

Programming with regular expressions is very different from programming in conventional languages (procedural, functional, or logic). The book provides an excellent exposition of the material, emphasizing not only the syntax and semantics of the two languages, XFST and LEXC, but also tips and tricks for clear and efficient network construction and common pitfalls to avoid. The detailed examples provide real-life morphological problems and the correct way to solve them. Thus, a considerable subset of Esperanto morphotactics is actually covered by examples in chapter 4. Examples are also drawn from Spanish, Portuguese, Irish, Arabic, Malay, and a variety of other natural languages.

The remainder of the book is dedicated to more marginal issues: flag diacritics, a special mechanism for tagging finite-state networks which is reminiscent of ATNs, are discussed in chapter 7, whereas provisions for nonconcatenative morphology, and in particular the compile-replace algorithm, are described in chapter 8. Some Xerox utilities are then listed in chapter 9, which closes the book.

Pedagogically, this is an extraordinary book. Its organization is excellent, no concept is used without being defined and exemplified, and key notions are repeated over and over again. Most chapters start with an introduction that summarizes the previous material and motivates the discussion and end in a summary. The authors' love for language, and in particular morphology, is evident everywhere, especially in the examples of "the mythical Bambona language" or "the fictional Monish language." It makes the book very enjoyable reading. It is evident that the book builds on many years of extensive experience with the technology in general and the Xerox tools in particular, and extensive experience *teaching* these issues; the authors do not hesitate to share this vast experience with their readers. For teachers of introductory NLP classes, as well as more advanced courses on FST, this book is a gold mine.

Who should buy this book? If you are a linguist who is planning to do some work with finite-state technology, then this book is a must. If you are not sure whether FST is for you, this book will most likely convince you that it is. If you are already working with some finite-state toolbox (such as the *FSM tools* from AT&T [Mohri, Pereira, and Riley 1998] or van Noord and Gerdemann's [2001b] *FSA utils*), then this book will provide insight into the vast possibilities that the technology offers and will help you place your own work in context. Specifically, if you already work with LEXC or XFST, this is the bible of the applications. However, if you are interested in the mathematics of finite-state networks or in the computational aspects of their implementations, you are probably better off with a book such as Roche and Schabes (1997).

A minor note: The book is extremely well-written, its language is fluent and lucid, and hard as I tried, I could not find a single error or typo. However, the repeated references to Xerox, the Xerox linguists, and the Xerox developers of the technology described in the book are exaggerated. The book would have been more fun to read without them.

References

Allen, James. 1995. *Natural Language Understanding*, second edition. Benjamin/Cummings, Redwood City, CA.

Daciuk, Jan, Stoyan Mihov, Bruce W. Watson, and Richard E. Watson. 2000. Incremental construction of minimal acyclic finite-state automata. *Computational Linguistics*, 26(1): 3–16.

Johnson, C. Douglas. 1972. *Formal Aspects of Phonological Description*. Mouton, The Hague.

Jurafsky, Daniel and James H. Martin. 2000. *Speech and Language Processing*. Prentice Hall, Upper Saddle River, NJ.

Kaplan, Ronald M. and Martin Kay. 1994. Regular models of phonological rule systems. *Computational Linguistics*, 20(3):331–378.

Koskenniemi, Kimmo. 1983. *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Department of General Linguistics, University of Helsinki, Helsinki, Finland.

Mohri, Mehryar. 1997. On the use of sequential transducers in natural language processing. In Emmanuel Roche and Yves Schabes, editors, *Finite-State Language Processing: Language, Speech and Communication.* MIT Press, Cambridge, MA, pages 355–381.

Mohri, Mehryar, Fernando Pereira, and Michael Riley. 1998. *A Rational Design for a Weighted Finite-State Transducer Library* (volume 1436 in Lecture Notes in Computer Science). Springer, Berlin/New York.

Roche, Emmanuel and Yves Schabes, editors. 1997. *Finite-State Language Processing*: *Language, Speech and Communication.* MIT Press, Cambridge, MA.

van Noord, Gertjan and Dale Gerdemann. 2001a. Finite state transducers with predicates and identity. *Grammars*, 4(3): 263–286.

van Noord, Gertjan and Dale Gerdemann. 2001b. An extendible regular expression compiler for finite-state approaches in natural language processing. In O. Boldt and H. Jürgensen, editors, *Automata Implementation* (volume 2214 in Lecture Notes in Computer Science). Springer, Berlin/New York, pages 122–139.

Shuly Wintner is a lecturer at the University of Haifa. His research interests involve various areas of computational linguistics, and in particular, the application of methods and techniques from computer science to the study of linguistic formalisms. He is also doing research on the morphology and syntax of Semitic languages, especially Hebrew. Wintner's address is Department of Computer Science, University of Haifa, 31905 Haifa, Israel; e-mail: shuly@cs.haifa.ac.il.

Data-Oriented Parsing

Rens Bod, Remko Scha, and Khalil Sima'an (editors)
(University of Amsterdam)

Stanford, CA: CSLI Publications (CSLI studies in computational linguistics, edited by Ann Copestake) (distributed by the University of Chicago Press), 2003, xii+410 pp; hardbound, ISBN 1-57586-435-5, $80.00, £56.00; paperbound, ISBN 1-57586-436-3, $35.00, £24.50

Reviewed by
Dan Klein
Stanford University

Data-Oriented Parsing contains four parts, each of which will interest a different set of readers. The early sections give a good introduction to the data-oriented parsing (DOP) framework, while later sections present more recent work, including a substantial amount of work on lexicalized tree-adjoining grammars (LTAGs) and some work on structural models of translation.

1. Part I: Overview

Part I is a well-written, concise overview of DOP and stochastic tree-substitution grammars (STSGs). After a short introduction, Bod and Scha present the vanilla DOP model, in which all subtrees in the training corpus are considered STSG productions, with a subtree's probability proportional to its frequency. Next, Remko Bonnema and Scha address the issue of how to better estimate subtree substitution probabilities. The core difficulty in estimation is that a treebank is a collection of trees rather than STSG derivations. Bonnema and Scha first consider maximum-likelihood estimation, with which one tries to reconstruct which of the many derivation(s) produced each tree. Unsurprisingly, the maximum-likelihood hypothesis is the one in which each tree was generated in a single, atomic substitution; this hypothesis wastes no probability mass on unseen sentences. As a result, Bonnema and Scha turn to the uniform distribution over derivations, which is also well-founded but has the opposite bias (smaller subtrees take more probability mass because they can combinatorially occur in more derivations).

To round out the overview, John Carroll and David Weir discuss a hierarchy of models in the LTAG framework and present an empirical study of several statistical regularities which can tease apart the capacities of models along their hierarchy. For example, in transitive sentences, the subject and object are likely either to both be pronouns or to both be proper names. This is outside the (natural) locality of simpler models, such as probabilistic context-free grammars (PCFGs), but can easily be captured in more complex models. Carroll and Weir's chapter is one of the best in the book; one can easily get so lost in theoretical complexity concerns that one forgets about the real phenomena at stake.

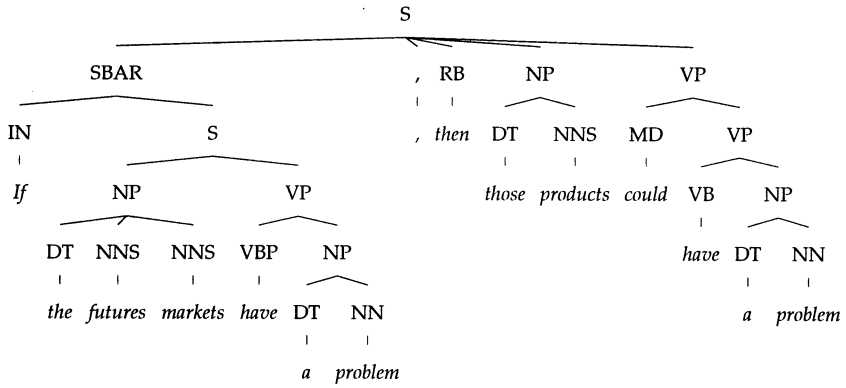

Figure 1
A tree containing multiple overlapping configurations whose regularity could usefully be modeled, such as the *if/then* pair, the sequence of tenses, and parallel lexical choices.

2. Comments on the DOP Framework

Overall, the DOP framework can usefully be contrasted with the more standard work on (lexalicalized) PCFG parsing (Collins 1999; Charniak 2000). In DOP, trees are built up using arbitrarily large substructures of previously seen trees. In contrast, most other work builds trees using highly local configurations. The upshot is that DOP models can capture many kinds of statistical correlations that standard models cannot. For example, consider the following sentence, parsed in Figure 1: *If the futures markets have a problem, then those products could have a problem*.

For standard parsing models, any correlated structures have to be local somewhere in the tree in order to be modeled. Here, we might thread the *if* and *then* up to the S node that dominates them, to capture the regularity of the *if* $\sim$ *then* construction, in the same way that lexicalized PCFG models thread lexical heads up through the tree (Klein and Manning 2003). But we might also want to capture the correlation between the antecedent and consequent tenses, the parallel structure of the two clauses, or any number of other possible regularities. We can cram all that into the S rewrite, of course, perhaps annotating it as

S-[*if/then*] $\rightarrow$
SBAR-[*if*,*futures*,*problem*,PRESENT] RB-[*then*] VP-[*products*]
VP-[*products*,*problem*,CONDITIONAL].

However, the local configuration isn't very local anymore, and it quickly becomes impossible to estimate probabilities from reasonable amounts of data. So while standard models pick and choose what information to make available, DOP aims to exploit it all at the same time using large, overlapping substructures.

Here I would argue that any appraisal of DOP work must separate the fundamental idea—any substructure can be relevant to disambiguation—from the actual mechanisms used to execute this idea. The idea is clearly good and, I think, vastly underappreciated. The concrete DOP models, on the other hand, do not necessarily represent a perfect solution. Leaving aside the traditional criticism of DOP—that the subtrees' probabilities are generally estimated in objectionable ways—I think the more serious objection is that the various derivations are summed, modeling each parse as a mixture of alternative derivations. While other frameworks have

multiple derivations, notably TAG (discussed in this collection) and combinatory categorial grammar (Steedman 2000), these derivations often correspond to semantic ambiguities rather than spurious variations (which also happens, and when it does, it represents a challenge for these frameworks as well). My concern with the mixture model in DOP is that when there are several configurations whose regularity should affect the parse of the sentence, they are sometimes multiplied (conjoined) and sometimes added (disjoined). In the example above, the *if* $\sim$ *then*, *markets* $\sim$ *products*, and *have* $\sim$ *could have* paths all overlap at the top S node. They will therefore only show up in disjoint derivations, and so their contributions will be summed. On the other hand, nonoverlapping configurations like the respective correlations between *futures* and *markets* (in the antecedent) and the modal and infinitive (in the consequent) can show up together in the same derivations. In this case their scores are multiplied. However, in all these cases, the statistical regularities would naturally be seen as conjunctive. Moreover, in the DOP framework, the impact of a configuration is dependent not only on its estimated probability but also on which other subtrees it can tile with and in how many ways. For example, a large configuration with a terrible score can never directly knock out a parse tree; it can only knock out the relatively small number of derivations which employ it.

3. Part II: Computational Issues

Part II of this collection takes the general DOP framework as a given and treats computational issues inside that framework. First, Sima'an shows that finding the most probable parse of a sentence in the basic DOP model is NP-hard, as are several related problems. Lest all DOP researchers despair, the next several chapters present some hope: A chapter by Jean-Cédric Chappelier and Martin Rajman and then another by Bonnema present a Monte Carlo technique and a sampling technique (respectively) for finding the most probable parse. If you're willing to settle for the maximum-brackets parse, you're actually much better off. In the next chapter, Joshua Goodman presents an insightful, very efficient method in which he creates a PCFG whose nonterminal symbols contain indexes to the training treebank and then uses this PCFG to recover the maximum-brackets DOP parse.

Moving to memory-based learning, Guy de Pauw gives a memory-based approximation to DOP. The parsing figures aren't that high, but this approach makes much more explicit the ways in which large substructures drive DOP parsing. Finally, Ido Dagan and Yuval Krymolowski present a memory-based shallow parser with a more tenuous connection to DOP.

I should point out that most of the chapters in this part begin by declaring that Sima'an's NP-hardness result is a practical worry. I wasn't convinced; his proof is a clever reduction from 3SAT, but as with many clever NP-hardness reductions, the widgets that one uses to encode 3SAT instances don't look a lot like the kinds of subtree configurations that would actually come out of a treebank.

4. Parts III and IV: Recent Work

Part III leaves the realm of DOP primer and presents a collection of more recent work in both the DOP and LTAG frameworks. These chapters are more likely to be of interest to those who already know the majority of what's in parts I and II. To open part III, Sima'an describes Tree-Gram parsing, a model which sits somewhere between DOP and standard lexicalized parsing work, modeling lexicalized structural configurations other than local attachment, such as the path between two words in a parse tree. In a

pair of chapters on enriching DOP, Bod and Ronald Kaplan extend the DOP model to LFG parsing, and Günter Neumann extends it to HPSG.

The next three chapters essentially abandon the DOP framework and examine LTAG parsing. First, Aravind Joshi and Anoop Sarkar give a good introduction to the TAG and LTAG formalisms. Next, Srinivas Bangalore describes supertagging, a method of narrowing down the set of local configurations before parsing, which can greatly speed up LTAG parsing. At the end of the LTAG tour, David Chiang discusses the heuristic extraction of LTAG derivations from Penn Treebank trees and describes a broad-coverage statistical LTAG parser. Part III finishes with Lars Hoogweg extending DOP parsing with tree insertion, which broadens the kinds of substructures available in the DOP model. In particular, modeling insertion provides access to simplifications of existing subtrees which result from the removal of modifiers and also allows existing structures to be combined in a richer set of ways.

Part IV contains two chapters on using DOP for translation and one on unsupervised syntax learning. First, Arjen Poutsma presents a synchronous DOP model for translation (DOT). In this model, tree pairs are node-aligned, and one synchronously expands linked node pairs using compatibly linked subtree pairs, much as is done by Melamed (2003). While this approach seems like a good idea, empirical results are only presented on a corpus of 266 Verbmobil sentence pairs, and so it's hard to know how well it will work, or how efficiently. Second, Andy Way extends Poutsma's DOT model to richer LFG structures, with which linkages can more accurately reflect valid translational equivalences. Several models are proposed, but no results are presented.

In the final chapter of the collection, Menno van Zaanen describes the application of alignment-based learning to the task of inducing syntactic trees from raw data. The bulk of this chapter really has very little to do with DOP (despite a proposal to use alignment-based learning as a mechanism for dealing with unknown words), but it's interesting work, and worth reading in any case.

5. Conclusion

This collection would serve as a great introduction for the segment of the community which is interested in parsing but isn't up to speed on DOP (though Bod [1998] is an alternative introduction which is lighter on math and heavier on linguistic argumentation). Part III is also a good collection which contains just as many papers on recent TAG work as DOP work. Bottom line: For the people who feel the basic DOP framework is unsalvageably broken, this book isn't going to change their minds, but it's a comprehensive and thought-provoking collection that ranges from the original foundations to the highlights of recent work.

References

Bod, Rens. 1998. *Beyond Grammar*. CSLI Publications, Stanford, CA.

Charniak, Eugene. 2000. A maximum-entropy-inspired parser. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics*, Seattle, pages 132–139.

Collins, Michael. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia.

Klein, Dan and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan, pages 423–430.

Melamed, I. Dan. 2003. Multitext grammars and synchronous parsers. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, Edmonton, Alberta, Canada, pages 158–165.

Steedman, Mark. 2000. *The Syntactic Process*. MIT Press, Cambridge, MA.

Dan Klein is a Ph.D. student at Stanford University. His interests include the unsupervised learning of natural language structure, statistical parsing, inference in large dynamic programs, and large-scale machine learning for NLP. Klein's address is 353 Serra Mall, Room 448, Stanford University, CA 94305-9040; e-mail: klein@cs.stanford.edu.

Automated Essay Scoring: A Cross-Disciplinary Perspective

Mark D. Shermis and Jill C. Burstein (editors)
(Florida International University and ETS Technologies, Inc.)

Mahwah, NJ: Lawrence Erlbaum
Associates, 2003, xvi+238 pp;
hardbound, ISBN 0-8058-3973-9, $59.95
($24.50 prepaid)

Reviewed by
Lawrence M. Rudner
Graduate Management Admission Council

The vision of having effective algorithms score student essays should be appealing to the teacher, test publisher, and research scientist. Teachers would be freed of the burden of reading and hand-scoring maybe hundreds of student papers and consequently would be more likely to assign written questions and probe for deeper understanding. Test publishers would be able to score essays for less cost and conceivably provide higher-quality assigned grades. Research scientists, especially readers of this journal, would find this to be a fascinating area, one that merges research from multiple disciplines and having many avenues yearning for exploration.

Shermis and Burstein's is the first book dedicated to the topic of automated essay scoring (AES). As such, it is destined to be the seminal work in the area. The book is composed of thirteen chapters, each written by a different set of authors. Five of the chapters provide descriptions of five different approaches and form the heart of the book. There are also four chapters on psychometric issues, two on innovations, a formal introduction, and two introductory chapters. The authors are all authorities. The five approaches are described by their developers or major advocates.

As with other forms of artificial intelligence, the task of AES is to accomplish a human goal. This does not mean that the goal needs to be accomplished using the same techniques as humans use. In the case of AES, humans typically read a passage and look for certain prespecified key concepts defined in a scoring rubric. Readers call upon their content knowledge, literary experience, and language skills in evaluating the passage.

The computer cannot possibly score an essay the same way. Rather, AES seeks to use the computer's special capabilities. The computer can count surface features, examine individual words and phrases, look at word order, stem, identify stop words, parse each sentence, examine sentence-to-sentence relatedness, weigh different features, identify arguments, and compare each new essay to hundreds of prescored essays. The question is whether the results are adequate.

If the goal is to approximate human scores, then the answer is yes for all the approaches. Timothy Keith provides an extremely informative chapter on the predictive validity of several AES programs. The programs tend to yield impressively high correlations with the scores of human raters—generally between .70 and .90 and often between .80 and .85. Further, the correlations of AES with human raters cannot be distinguished from the correlations among human raters. As Ellis Page points out in a chapter describing Project Essay Grade (PEG), AES in a sense "passes the Turing test"—an outside observer cannot tell the difference between the machine and a human. Another way to look at the accuracy of AES is to examine the percentage of

agreement between AES and human scores. In practice, AES scores are considered to be comparable to human ratings if the two are within one point of each other: "adjacent accuracy." Several chapters report adjacent accuracies of 90–99%. By this criterion, it is fairly easy for an AES system to be adequate. Adjacency covers much of a scoring scale—half of a six-point scale and three-fourths of a four-point scale. Further, most scores are typically in the middle of the score range, again increasing the likelihood of obtaining a near-perfect adjacent accuracy.

As stated earlier, the heart of the book is the five chapters devoted to different methods. Three of the methods have been described in the professional literature and appear to be fairly mature—PEG, which is described by Ellis Page; e-Rater, which is described by Jill Burstein; and Intelligent Essay Assessor (IEA), which is described by Thomas Landauer, Darrell Laham, and Peter Foltz. Two additional approaches are presented. Leah Larkey and Bruce Croft present the details of a Bayesian approach based on the well-developed text classification literature. Scott Elliot provides a summary of studies conducted using Intellimetric. The five approaches are all quite different.

PEG and the Bayesian approach are the simplest. Using a large collection of surface features such as average sentence length, frequency of certain transitional words, number of semicolons, and word rarity, PEG yields extremely impressive AES–human rater correlations. These surface features appear to be effective proxies for the intrinsic variables that humans look for. The Bayesian approaches examine the probabilities of each token (typically a word or a stemmed word) being used in essays in each score group. Larkey and Croft present a wonderful analysis of a variety of approaches.

On the other end of the spectrum, IEA and e-Rater have a much deeper linguistic base. IEA examines content, style, and mechanics, with content expressed as independent measures of semantic quality and the amount of such content. E-Rater examines discourse structure, syntactic structure, and vocabulary usage. While the underlying mathematics is different, the two approaches share an underlying philosophy of relying on natural language processing rather than mechanical features.

If a reader is looking for an understanding of the approaches and potential of AES, this is the book to read. All the current approaches are presented in one volume. The authors do an excellent job of describing the philosophy and history of their approaches. Of particular interest are ideas for providing diagnostic and evaluative feedback that are sprinkled throughout the book. The chapters are, however, quite independent and in the wrong order. I suggest starting with the introduction, then moving to the descriptions of the approaches, psychometric issues in AES, and innovations in AES. The first two chapters, which are probably intended to provide a general framework and background, can be skipped without any loss.

Lawrence Rudner is the chief statistician with the Graduate Management Admission Council. He is the author of the Bayesian Essay Test Scoring sYstem (BETSY), which is available for research use at http://edres.org/betsy/. His research interests include AES, computer adaptive testing, and decision theory. Rudner's address is 1600 Tysons Boulevard, Suite 1400, McLean, VA 22102; e-mail: LRudner@gmac.com.

Word Sense Disambiguation: The Case for Combinations of Knowledge Sources

Mark Stevenson
(University of Sheffield)

Stanford, CA: CSLI Publications (CSLI studies in computational linguistics, edited by Ann Copestake) (distributed by the University of Chicago Press), 2003, xvi+175 pp; hardbound, ISBN 1-57586-389-8, $67.50; paperbound, ISBN 1-57586-390-1, $25.00

Reviewed by
Susan McRoy
University of Wisconsin–Milwaukee

In this book, Stevenson describes his work on applying and evaluating empirical methods for word sense disambiguation (WSD) in large texts. His approach combines several individually weak knowledge sources using a memory-based machine-learning algorithm. It contrasts with earlier methods that relied on hand-crafted rules to combine information from multiple knowledge sources, such as that of McRoy (1992), and with previous empirical approaches, such as that of Yarowsky (1992), that used small lexical samples. Previous rule-based approaches used many of the same knowledge sources and disambiguated about the same percentage of words but lacked a mechanism for evaluating disambiguation accuracy over large test sets. Large-scale empirical approaches to word sense disambiguation have become possible because of the availability of tagged text with categories from WordNet and a mapping, created by Knight and Luk (1994), from WordNet categories to senses from the *Longman Dictionary of Contemporary English (LDOCE)*.

According to the author, this book is based largely on his Ph.D. dissertation, with some extensions to address comments that were made by readers of the thesis and also some discussion of recent work. The book follows the traditional structure of a thesis: a discussion of the foundations of word sense disambiguation and a survey of earlier work by others, followed by a detailed presentation of the author's approach to the problem, his implementation of the approach, and its evaluation. It adds a foreword written by Yorick Wilks that succinctly describes the main contribution of the work and the methods that were used.

Stevenson is concerned with the problem of how to discriminate multiple meanings or senses of the same word. In particular, the book addresses the task of sense tagging, which Stevenson defines as the task of annotating all the words in a document with senses from a given lexicon. Sense tagging is useful for machine translation and may also be of some (but smaller) benefit to work on information retrieval. Stevenson distinguishes sense tagging from general semantic disambiguation and semantic tagging, which do not constrain the types of semantic annotations used, and also from sense disambiguation, which does not require that all words be disambiguated. (Alternatives to lexicon senses for annotation might be semantic features such as HUMAN or ANIMATE.) The author notes that because sense tagging is the most constrained word sense disambiguation task, "a solution to the sense tagging problem would also be a

solution to the other, more general, WSD tasks." Also, sense tagging could potentially replace other discriminators for ambiguities of pronunciation or part of speech.

The weak sources of knowledge that Stevenson's learning algorithm combines include selectional preferences, thesaurus or topic classes, and distributional information gleaned from a corpus. He quantitatively validates the importance of combining multiple sources by showing that while individually these sources could only provide accuracy at the 60% level, properly combined they could achieve accuracy as high as 92% (page 88). The 92% figure is based on disambiguating words only to the homograph level and counting all words. His algorithm achieves 70–90% accuracy when we consider only content words such as adverbs, adjectives, nouns, and verbs and disambiguation down to the level of individual senses.

The first half of the book provides an introduction to the task and concepts of word sense disambiguation, suitable for anyone who is beginning work in this area. For example, the first chapter characterizes different forms of word sense disambiguation, including sense tagging, which, as mentioned, is the focus of the book. The second chapter describes some earlier approaches to word sense disambiguation and the SENSEVAL evaluation framework. The third chapter describes the content and organization of lexical resources commonly used for WSD, including *LDOCE, Roget's Thesaurus*, and WordNet. Chapter 4 presents a general characterization of the application knowledge sources for WSD as either filters or partial taggers and discusses the information from *LDOCE* and WordNet that would be most useful for such algorithms. The framework itself seems to cover most, if not all, prior work. Thus, the author's goal seems to be to provide a way of unifying prior approaches, instead of choosing among them.

The remainder of the book describes Stevenson's approach to sense tagging and his evaluation of the results. Chapter 5 considers part-of-speech tagging using syntactic categories from *LDOCE* and the contribution that it can make to sense tagging. In Stevenson's experiments, 94% of words could be assigned the correct *LDOCE* sense (to the homograph level), using only information from the Brill tagging and a mapping from its tags to those of *LDOCE*. This result would seem to contradict Stevenson's claim that a combination of sources is needed to reach this same level of accuracy, although he did find that none of the other knowledge sources (simulated annealing, selectional preferences, or the broad context, as identified by subject codes) provides this accuracy individually.

Chapter 6 (which is about 25 pages long) discusses Stevenson's implementation of a sense tagger using senses from *LDOCE* and the architecture based on combining the results of several filters and partial taggers. Neither algorithms for computing weak knowledge sources nor algorithms for combining weak knowledge sources are entirely new; however, they are explained succinctly here, along with some adaptations that were made. What is novel is the application of a machine-learning algorithm to combine the outputs of its filters and taggers. The particular algorithm is Daelemans et al.'s (1999) TiMBL memory-based learning system, which is explained in some detail.

Chapters 7 and 8 discuss the evaluation of word sense disambiguation algorithms. Chapter 7 describes several previous evaluation strategies, as well as providing important background on the mapping between WordNet senses and the senses of *LDOCE*, and also Stevenson's mapping of an evaluation corpus that was annotated with WordNet senses to a corpus that is tagged with *LDOCE* senses. Chapter 8 describes Stevenson's experiments to evaluate his own system.

The book would be most appropriate to students, researchers, or practitioners who are learning about word sense disambiguation for the first time or to anyone who has not considered word sense disambiguation since the late 1980s, when nonempirical

approaches were still common. The introduction is very basic and would be readily understandable by a student just taking her first course in computational linguistics. The background is similarly intended for those unfamiliar with word sense disambiguation, although the level of detail in some of the discussions is somewhat uneven. A key issue for Stevenson is how systems acquire disambiguation information, such as whether they require tagged or untagged corpora for training and whether they make use of a machine-readable dictionary. There is a discussion of the 1998 and 2001 SENSEVAL competitions (Kilgarriff and Palmer 2000; Preiss and Yarowsky 2001), which standardize the comparison of different approaches to word sense disambiguation.

The presentation of the book would have benefited from a bit more careful editing. The book contains numerous minor punctuation and grammatical errors and errors in the bibliographic references. The background section is uneven in its level of detail and its citations to related work. There are detailed discussions of Wilks's preference semantics, Cowie, Guthrie, and Guthrie's (1992) work using simulated annealing,[1] and Yarowsky's algorithm for tagging words with categories from *Roget's Thesaurus*. Readers would also likely have appreciated having a more comprehensive index, including, for example, entries for authors whose work has been cited.

References

Cowie, Jim, Joe Guthrie, and Louise Guthrie. 1992. Lexical disambiguation using simulated annealing. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France, pages 359–365.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. 1999. TiMBL: Tilburg Memory Based Learner, version 2.0, Reference Guide. Technical Report ILK 99-01, Induction of Linguistic Knowledge Group, Tilburg University, The Netherlands.

Kilgarriff, Adam and Martha Palmer, editors. 2000. Special issue on SENSEVAL. *Computers and the Humanities*, 34(1–2).

Knight, Kevin and Steve Luk. 1994. Building a large knowledge base for machine translation. In *Proceedings of the 12th National Conference of the American Association for Artificial Intelligence (AAAI-94)*, Seattle, WA, pages 773–778.

McRoy, Susan. 1992. Using multiple knowledge sources for word sense discrimination. *Computational Linguistics*, 18(1):1–30.

Preiss, Judita and David Yarowsky, editors. 2001. *Proceedings of the SENSEVAL-2 Workshop*, Toulouse, France.

Selman, Bart and Graeme Hirst. 1985. A rule-based connectionist parsing system. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, Irvine, CA, pages 212–221.

Selman, Bart and Graeme Hirst. 1987. Parsing as an energy minimization problem. In Lawrence Davis, editor, *Genetic Algorithms and Simulated Annealing* (Research Notes in Artificial Intelligence), Pitman, pages 141–154.

Yarowsky, David. 1992. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, Nantes, France, pages 454–460.

Susan McRoy's research spans a number of areas, including designing and implementing systems for tutoring and training; achieving robust human–machine communication; detecting and repairing communication errors; developing tools for real-time intelligent dialog systems; and investigating interactions among speech, gaze, and gesture during communications among people. McRoy's address is Department of Electrical Engineering and Computer Science, P.O. Box 784, University of Wisconsin–Milwaukee, Milwaukee, WI 53201; e-mail: mcroy@uwm.edu.

1 Readers interested in the history of the use of simulated annealing in NLP will also want to consider the work by Selman and Hirst (1985, 1987).