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.

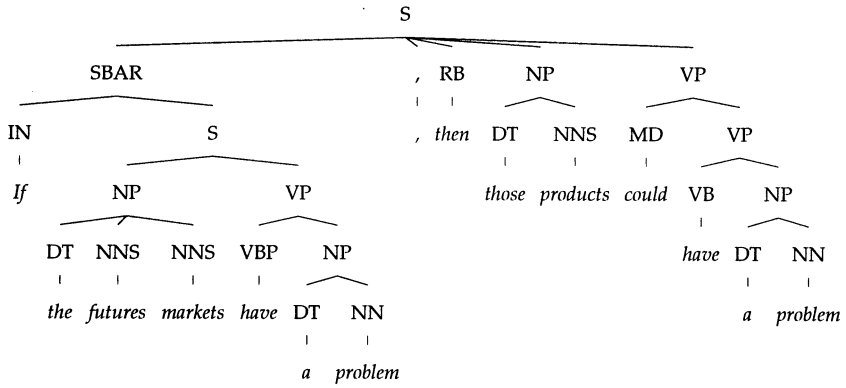

Figure 1
A tree containing multiple overlapping configurations whose regularity could usefully be modeled, such as the *if/then* pair, the sequence of tenses, and parallel lexical choices.

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.