

Inductive Dependency Parsing

Joakim Nivre

(Växjö University)

Dordrecht: Springer (Text, speech, and language technology series, edited by Nancy Ide and Jean Véronis, volume 34), 2006, xi+216 pp; hardbound, ISBN 1-4020-4888-2, \$119.00, €89.95

Reviewed by

Christer Samuelsson

Columbia University

Philologists assure us that it's worth learning ancient Greek just to read Homer. For any linguist, it's definitely worth learning French, just to read Lucien Tesnière's *Elements de Syntaxe Structurale* (Tesnière 1959). For any serious dependency parsing student or professional, it would have been worth learning Swedish, just to read Joakim Nivre's *Inductive Dependency Parsing* if Nivre had not done the world the immense favor of writing his book in English, rather than in his native tongue of glory and heroes.

The book demonstrates Nivre's impressive ability to explain dependency grammar and dependency parsing clearly and succinctly to a wide audience. His overviews of dependency grammars (Section 3.1) and dependency parsing (Section 3.2) should be required reading in any course on the topics. He concisely sums up each relevant approach, often in a single sentence, and those wishing to learn more can rely on his extensive collection of references.

The remainder of Chapter 3 sets the board for the proposed novel MALT parser—the book's main innovation—by formalizing the dependency parsing task and specifying a particular type of deterministic shift-reduce parsing algorithm. Here, the set of parser actions are appropriate for dependency parsing, rather than for constituent parsing, and it is assumed that there is a one-to-one correspondence between dependency analyses and valid, complete sequences of parser actions. Nivre proves that the parsing algorithm correctly performs the formalized dependency parsing task, producing an acyclic, single-headed, projective dependency graph.

Whereas the short Chapter 1 consists of the obligatory introduction, motivation, and book overview, it is interesting to discuss Nivre's ambitious four criteria for a successful dependency parser, stated in Chapter 2:

1. **Robustness:** The parser must assign *at least* one dependency analysis to each sentence in any given text.
2. **Disambiguation:** The parser must assign *at most* one dependency analysis to each sentence in any given text.
3. **Accuracy:** The parser must assign *the correct* dependency analysis to each sentence in any given text.
4. **Efficiency:** The parser must process all given sentences in all texts in time and space linear in the size of each sentence.

Nivre draws the inevitable conclusion from the first two criteria:

Existence and Uniqueness: The parser must assign *exactly one* analysis to each sentence in any given text.

In Chapter 3, Nivre achieves the Accuracy criterion by presupposing the existence of an Oracle that always provides the correct parser action. In later chapters, he reverts to more traditional measures of Accuracy, while leaving the other three criteria intact. The Efficiency criterion is achieved by the deterministic and greedy (non-ambiguity-packing) nature of the shift-reduce parser. One could argue that even at a word level, deterministic, non-packing parsing is not always possible, but Nivre sticks to his guns.

In Chapter 4, Nivre replaces the omniscient Oracle of Chapter 3 in two logical steps. In the first step, the parser instead chooses the locally most advantageous parser action a^* according to some evaluation function $g(a)$:

$$a^* = \arg \max_a g(a)$$

This preserves the determinism and greediness of the parser, at the cost of potentially losing the globally best action or the gold-standard action. The latter is the parser action ultimately resulting in the reference dependency analysis of the training or testing data.

In the second step, an approximation $\hat{g}(a)$ of this unknown evaluation function $g(a)$ is learned from training data, using some machine-learning scheme. Thus the parser instead chooses the parser action \hat{a}^* :

$$\hat{a}^* = \arg \max_a \hat{g}(a)$$

which may very well deviate from the locally best parser action a^* , as well as from the globally best and the gold-standard parser actions. Nivre relates this learning scheme to maximizing likelihood estimates and conditional likelihood estimates, as well as to discriminative learning techniques. He also takes on the chore of relating the gold-standard parser action to the dependency analyses of the reference data.

The logical progression and the clarity with which this is done is one of the many strengths of this book. Many followers take the book this far as a point of departure for alternative explorations into inductive dependency parsing.

Nivre next investigates feature selection for creating the function $\hat{g}(a)$, then moves on to his choice of machine-learning framework—memory-based learning (MBL) (Daelemans and van den Bosch 2005)—ultimately leading to the proposed inductive shift-reduce parser: the MALT parser.

In Chapter 5, Nivre evaluates his MALT parser on the mandatory Penn Treebank parsing task. This is arguably a waste of time as: the Penn Treebank is a constituency treebank; its underlying linguistic theory is quite poor; its annotation consistency is even worse; and the rules for retrieving corresponding dependency analyses are, at best, inspired hacks.

Many researchers in the field have used Nivre's freely available MALT parser,¹ but have replaced MBL with some other machine-learning technique, such as conditional random fields. About half of the entrants of the dependency-parsing shared task of the

¹ <http://w3.msi.vxu.se/~nivre/research/MaltParser.html>.

2006 Conference on Computational Natural Language Learning (CoNLL X) (Marquez and Klein 2006) used this approach. That alone demonstrates the importance of this book.

The 2007 CoNLL conference will most likely exhibit non-greedy, ambiguity-packing versions of Nivre's inductive shift-reduce parsing scheme, in the vein of Masaru Tomita's generalization of (natural language) LR parsing (Tomita 1987).

Get Nivre's book; read it; and enjoy it! The excellent and thorough reference list alone is worth it, constituting a good ten percent of the book.

References

- Tesnière, Lucien. 1959. *Elements de Syntaxe Structurale*. Editions Klincksieck, Paris.
- Daelemans, Walter and Antal van den Bosch. 2005. *Memory-Based Language Processing*. Cambridge University Press, Cambridge.
- Marquez, Lluís and Dan Klein, editors. 2006. *Proceedings of CoNLL-X: Tenth Conference on Computational Natural Language Learning*, Brooklyn, New York, Association for Computational Linguistics.
- Tomita, Masaru. 1987. An efficient augmented context-free parsing algorithm. *Computational Linguistics*, 13(1–2): 31–46.

Christer Samuelsson has worked with natural language parsing at SICS, the University of the Saarland, Bell Labs, Xerox Research Centre Europe, Lehman Brothers, and a few start-ups. He is currently associated with the Center for Computational Learning Systems, Columbia University. Samuelsson's e-mail address is christer.samuelsson@gmail.com.

