Adaptive Parsing: Self-Extending Natural Language Interfaces

Jill Fain Lehman
(Carnegie Mellon University)

Boston: Kluwer Academic Publishers
(The Kluwer International Series in
Engineering and Computer Science;
Natural Language Processing and
Machine Translation, edited by Jaime
Carbonell), 1992, xiii + 240 pp.
Hardbound, ISBN 0-7923-9183-7, $64.00,
£43.50, Dfl 145.00

Reviewed by
Julia Johnson
University of Regina

An adaptive parser responds to unfamiliar utterances by augmenting the grammar with rules that will permit the same utterance to be parsed the next time it is encountered. The kind of grammatical components that the system learns are idiosyncratic vocabulary and syntactic structure. Jill Fain Lehman presents the design, implementation, and evaluation of an adaptive parser in this book based on her Ph.D. thesis work at Carnegie Mellon. The book would be useful both for researchers, in highlighting open questions in the area of adaptive language acquisition, and for practitioners, for building an adaptive parser. Since it is a thesis, some points appear to be argued unnecessarily. On the other hand, Lehman's book is free of typing errors and well written.

Chapter 1 considers the tradeoff between computational complexity and coverage. Computational complexity results from increased ambiguity in a grammar that tries to anticipate all of the linguistic forms it might encounter over a large cross section of users. Lehman's notion of adaptive parsing concerns the linguistic behavior of one user at a time, thereby limiting computational complexity.

Chapter 2 presents a model of adaptive parsing intended for the frequent user in a task-oriented domain. A deviation results when the current grammar cannot parse an utterance. An input is characterized by its deviation level, corresponding to the number of errors committed.

In Chapter 3, a claim is made that frequent users in task-oriented domains tend to exhibit self-bounded linguistic behavior, which places a natural limit on the degree of extensibility required of the system. This claim is substantiated by using hidden-operator experiments.

Chapters 4 through 6 discuss the assignment of meaning to an utterance. The meaning of a nondeviant utterance is represented by a sequence of update actions to be performed on the database. For a deviant utterance, the parse tree is augmented with recovery actions that can be considered to be an explanation for why the parse failed. Algorithms for detecting deviations and producing explanations are presented in stages as definitions of predicates that are progressively generalized. A detailed example of the parse of a particular utterance and the explanations generated is provided.

Chapter 7 focuses on choosing the best explanation for a deviant input. Clarifications requested of the user focus on meaning, not explanation. In the words of the author, "If we wish to assume any expertise on the part of the user it should be task expertise, not linguistic expertise." Explanations of a deviant utterance are grouped into equivalence classes. Two explanations are equivalent if the meanings they give the utterance have the same effect on the database. Chapter 8 discusses modification of the grammar to recognize a new form in response to the recognition of that form as a deviant input.

Informal arguments are given throughout the book based on efficiency considerations for one design decision compared with another. Chapter 9 is dedicated specifically to the subject of evaluating the adaptive interface against performance criteria that include response time and the number of rejected inputs. The grammar that exists prior to interaction with any user is called the kernel grammar. A comparison is made between the number of alternative phrasings requested in on-line experiments with users when the kernel grammar is used alone and when the kernel grammar with adaptation is used on the same set of inputs. Unresolved issues are discussed in Chapter 10.

Chapter 11 concludes the book with a summary description of the main features of the system, which include deviation detection, error recovery, and grammar augmentation. A need is identified for comparison of the implemented techniques of adaptive language acquisition with other theories of parsing and grammar representations.

The design of the system would be particularly appealing to those wishing to build a natural language interface, because the main subproblems are handled in a uniform manner. The notion of an annotated parse tree serves as a focal point between the process of providing an explanation of a deviant input and that of augmenting the grammar with rules that will permit an unfamiliar utterance to be parsed. The annotated parse tree serves as output for the process of forming explanations and as input to the process of adapting the grammar. Likewise, there is a smooth transition between the process of providing a meaning for a nondeviant input and that of providing a meaning for a deviant input. The meaning of a nondeviant input derives from a parse tree without annotation. Hence, a nondeviant input is considered as a special case of a deviant input. Another interesting feature is that deviation detection is handled as a special kind of input error.

Lehman has illustrated that the problem of providing an adaptive parser for frequent users in task-oriented domains is an achievable subgoal toward that of providing a highly perceptive and intelligent natural language interface. The general organization of the system highlights the need for research on many open questions, and is therefore a useful tool for guiding our research directions. In particular, the least-deviant-first parsing algorithm and the particular case frame representation of the grammar that were used in the implementation are not necessary and, as suggested by the author, should be compared with other theories.

In the implemented system, many problems are circumvented by assuming a very simple task domain (the scheduling of events such as meetings, lunches, and travel). Therefore, many problems are simplified, such as the number of possible meanings for a natural language input. The problem of grouping explanations so that those within a group all have the same effect on the database is greatly simplified by choosing a domain in which there are few possible updates. A powerful and domain-independent language for representing explanations is needed, and a theory for updating databases from ambiguous inputs needs further development (Davidson 1987).

The author suggests that her model could also be used to learn discourse-level phenomena, morphology, and semantics. Another interesting future direction would be to permit an adaptive parser to withdraw rules from the grammar.

Many key questions remain open. What is the best kernel grammar for reducing the amount of extensibility required of the system? Lehman does not reference the work of Marsh and Friedman (1985), who argue that it is easier to pare down a broad coverage grammar for a specialized sublanguage than to build up a grammar from a kernel that has been developed for a restrictive domain. How do we compare alternative kernel grammars for their capability of being extended in the directions that interactions with users will indicate? The hidden-operator experiments with frequent users in a task-oriented domain show that the language understood by the system and that employed by the user tend to converge. Is there a kernel grammar that would permit that same convergence behavior for casual users? The performance improvements realized with adaptive parsing over a particular kernel grammar without adaptation were not strong. Would a greater improvement have resulted with a different kernel grammar? In Lehman's system, each specific user has his or her own adapted kernel. Could the system be extended to permit a group of users involved in a common task or using a specialized common sublanguage to share an adapted kernel?

References

Davidson, James Edward (1987). "Interpreting natural language updates." Technical report STAN-CS-87-1152, Department of Computer Science, Stanford University.

Marsh, Elaine, and Friedman, Carol (1985). "Transporting the Linguistic String Project system from a medical to a navy domain." *ACM Transactions on Office Information Systems,* **3**(2), 121–140.

Julia Johnson obtained her Ph.D. in Computer Science at the University of British Columbia. Her research interests are in the design of natural language interfaces in general, and methods for resolving ambiguities in particular. She is an assistant professor in the Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2. e-mail: johnson@cs.uregina.ca