Book Reviews


The two books under review can be considered as a single ‘act of publication’ in the sense that the second is an abridged version of the first, and the simultaneous appearance of both volumes is part of the concept chosen. I will refer to the individual volumes as A and B, and to the set as R. A is a textbook in the familiar, red-covered Cambridge ‘Textbooks in Linguistics’ series, the successor of Radford (1981, 1988). B has a larger page and a generally more inviting layout.

The review of a textbook is a quite different task from the review of a theoretical monograph. Whereas in the latter case the theory is the focus of the review, an evaluation of Chomsky’s minimalist program, the theory treated by R, would be completely out of place here. Rather, reviewing a textbook is like evaluating an NLP tool. In the context of Machine Translation, Krauwer (1993) argues that an evaluation should take as its object a pair <specification, tool>, and consider first to what extent the tool matches the specification, and secondly, to what extent the specification coincides with the actual problem. I will extend this general method to the evaluation of a textbook here.

Design criteria

The preface in A mentions “six main features which mark this book out as different from other introductions to syntax currently available”. In B an adapted list is presented in the preface. Taking this list as a design specification of the functionality of the tool Radford intended to produce, I will evaluate on the one hand the publication against these criteria, and on the other, the features against the practical use of a textbook.

1. Radford intends his textbook to be “suitable for use by true beginners”. This is evident in particular in Chapters 2 and 3, of which the first is entirely devoted to introducing syntactic categories, and the second to introducing the idea of tree structure. Radford’s approach stands in marked contrast to that adopted, for instance, by Haegeman (1994), who covers argument structure and the projection principle in Chapter 1. Of course, the felicity of Radford’s decision depends to a large extent upon the context of use. After an introductory linguistics course, it should not be necessary to explain extensively what a noun is. At the same time, Radford’s stated goal of bringing the user up to the level of understanding required for Chomsky (1995) imposes constraints on the minimum speed of advance. In this reviewer’s opinion, R requires a very long course or extremely fast learners to meet this goal. If the contents of Chapters 2 and 3 are as new to the learners as their wording suggests, it is hard to imagine treating the whole book in less than a year. This may be taken as a problem of either the textbook or the theory it introduces. However, at least in part, it is merely a consequence of the progress in theory
of grammar. In natural science it is generally accepted that it is impossible to have a one-semester introduction to a field of comparable specialization.

2. R takes a synchronic approach to the theory, introducing only the minimalist program of Chomsky (1995) without referring to earlier stages of Chomsky’s theory. This is one of two options, the other one being represented by, for instance, Ouhalla (1999). In general, the synchronic approach is more efficient in educating future researchers, while the historical approach gives better access to the entire body of literature. Thus, Radford’s textbooks are not ideal for computational linguists trying to understand useful insights gained in Chomskyan linguistics. Moreover, it is questionable whether the minimalist program represents a suitable stage of development of Chomsky’s theory to which a strictly synchronic approach might be taken. The way Lasnik (1999:1) qualifies the minimalist program (a strategy rather than an articulated theory) suggests it is not.

3. Radford illustrates cross-linguistic variation in terms of varieties of English. The problem he sees is that students cannot be assumed to know Spanish, Chinese or Chuckchee. As a consequence, prodrop is illustrated with Shakespearean English. Disadvantages of this approach are that it widens the gulf between textbook and literature, and creates the impression that the theory is English-specific. In this reviewer’s experience of teaching in non-English-speaking countries, students often have a prejudice that Chomskyan theory is too narrowly English-centred. These books will only strengthen this impression. However, for English-speaking countries and for English departments elsewhere, this is less of a problem. Teachers using R in a class with speakers of languages other than English, however, will have to face the challenge of explaining how these languages relate to the theory. Assuming that they do not want to claim that, for instance, German or French are not covered by the minimalist program (which is inconceivable from a Chomskyan point of view), they have to acquire this knowledge elsewhere.

4. R presents exercises with model answers and hints where necessary. The model answers in particular represent an extension compared to other textbooks. They are meant to counter the problem that students, even if they (think they) understand the chapter, “have little or no idea how to go about doing the exercises” (B:vii). The actual model answers, however, will in several cases increase the barrier to solving the exercises, rather than ease it, because they suggest an unreasonably high information standard which cannot be expected from students. An example is Exercise 2 in Chapter 6 of A (= Exercise 9 in B). In addition, model answers are regularly used to introduce new pieces of theory (e.g. for the analysis of relative clauses and topicalization in the workbook section of Chapter 7 of A).

5. R includes a glossary to avoid the “terminological trauma” of a syntax course (B:ix). This is indeed a very practical feature, not commonly available in textbooks. It makes the books useful also as reference works, especially A, which has section references in the entries. It should be noted that the glossary is not given instead of, but in addition to, the subject index. The entries in the glossary are brief and to the point. Technical terms in the explanations are marked as references to other entries in the glossary.

6. Finally, probably the most remarkable feature of R is the fact that it consists of a full version A and an abridged version B. The abridged version is presented as a remedy against the problem that textbooks can often not be treated completely in a single course. The full version would then serve as a means of treating individual topics in more depth.

In at least one sense, R has succeeded remarkably well in realizing this conception. Reading B first, this reviewer hardly noticed that it was derived from another book. The most obvious feature of abridgement is found in the exercises. Keeping rigidly to a ration of two exercises per chapter, B sometimes offers exercises combining different tasks in different parts. The possibility of using A as an expanded version of B is not facilitated by an annoyingly large number of non-correspondences. A first class of problems concerns
non-correspondences in text structure. Both A and B consist of ten chapters, but A’s Chapter 8 corresponds to B’s Chapters 7 and 8 together. While this does not disturb the order of presentation, the fact that Chapter 5 in A corresponds to a number of sections in Chapters 3 and 4 in B makes finding passages in A corresponding to and deepening a particular passage in B a real puzzle. Even more problematic is a second class of mismatch, concerning terminology and analysis. Thus, B:43 introduces pronouns immediately as D, but A treats them as a separate category PRN from their introduction in section 2.4 until the discussion in section 4.7. Another example is the use of IP in most of B, whereas A uses TP. Whether the concept of an abridged version is an adequate tool in overcoming the time problem, already noted in the discussion of feature 1 above, is an open question. If A is (just) sufficient as a textbook to bring beginning students of linguistics up to the level required for reading Chomsky (1995), there seems to be a large risk that an abridged version B will be either too shallow to understand Chomsky (1995), or too dense to be treated in a single course, despite the reduction in number of pages. Only experience can tell to what extent these risks are avoided in practice.

**Style of presentation**

After this discussion on the basis of design criteria, I would like to turn to a number of general points concerning style of presentation. These points concern Radford’s writing style, his order of presentation of the material, and the level of simplification inherent in the genre of the textbook.

1. Radford’s writing style is generally dry and sometimes overly repetitive. The detailed, explicit step-by-step descriptions of many derivations are no doubt an advantage in its use by beginning students, but may turn off other users of a textbook. I can imagine that some readers like R’s style of humour as countering this, but to my taste such remarks as that explaining the concept of rational entity as “an entity capable of rational thought – hence e.g. not an expression denoting a politician” (A:327 and B:164) are irritating and out of place in a textbook (especially because in this case it is formally wrong).

2. The structure of R’s presentation is strongly phenomenon-driven rather than theory-driven. Thus, there are chapters devoted to head movement, operator movement and A movement, but elements of binding theory are spread rather casually among different chapters. This contrasts, for instance, with Ouhalla’s (1999) presentation, which has chapters for Case theory, binding theory and control, etc. The choice is mainly a matter of taste, but it interacts with other parameters along which textbook strategies may vary.

3. It is an inevitable property of textbooks that they simplify matters so as to make them gradually understandable. In the presentation of a complex theory, the start especially must have a kind of boot-strapping character. While probably not entirely avoidable in a textbook, simplifications should steer clear of outright errors. In R, Chapters 2 and 3 especially contain a number of such infelicities. Thus, B:61 states (or at least strongly suggests) that help you is a verbal projection because it has the same distribution as help.

**Conclusion**

The criteria advocated by R as design criteria for a solution to the textbook problem of developing a tool as an introduction to a complex theory restrict the readership of this publication. Setting the goal as a single stage of the theory, the most modern one at the time of publication, hardly suits researchers looking for solutions to particular linguistic problems
in Chomskyan linguistics. Restricting the discussion of cross-linguistic variation to different
varieties of English hardly suits the market outside monolingually English-speaking countries.
The goal of starting with true beginners and bringing them up to the level of current literature
is probably overly ambitious.

Nevertheless, R has done a remarkable job with these textbooks. Taking the design criteria
explicitly stated in the preface as evaluation criteria, there are only some minor points of
criticism, most of them rather a matter of taste. R can therefore be said to keep his promises,
which is certainly more than can be said of most tools around.

References
59–66.
Ouhalla, J. (1999) Introducing Transformational Grammar: From Principles and Parameters to
Minimalism. Edward Arnold.
Radford, A. (1981) Transformational Syntax: A student’s guide to Chomsky’s Extended Stan-
dard Theory. Cambridge University Press.

PIUS TEN HACKEN
University of Basel
e-mail: pius.tenhacken@unibas.ch

Emmanuel Roche and Yves Schabes, Editors, Finite-State Language Pro-
pages.

This is a collection of articles, very diverse both in their content and in their quality. Their
common denominator, as the title suggests, is that they are all related to the processing of
natural languages with finite-state techniques. However, the articles vary in that some are
concerned with linguistic questions, some investigate computational issues and some describe
systems.

In the Introduction, the editors motivate the book by saying that “recent mathematical
and algorithmic results in the field of finite-state technology have had a great impact... on
natural language processing. As a result, a new technology for language is emerging...” (page
1). While it is true that some of the articles in this book demonstrate the usability of finite-
state techniques for natural language applications, especially for more ‘low-level’ applications
such as dictionary organization and maintenance (Chapter 6), morphological analysis and
generation (2, 6), part of speech tagging (7), information extraction (13), phonetic conversion
(14) or speech recognition (15), other authors attempt to use such technology for more
‘high-level’ applications, such as representing grammars (Chapters 3, 9 and 11) or parsing
(8). Moreover, some contributors attempt to show that finite-state technology is sufficient for
linguistic theory. For example, Roche states in Chapter 8 that “for the problem of parsing
natural language sentences, finite-state models are... one of the best formalisms at hand to
represent accurately complex linguistic phenomena.” This is a strong and rather controversial
claim, and I am not sure that it is substantiated by the articles in the book. In Chapter
11, Gross takes a different position: observing that linguistic theories usually compromise
coverage for generality, his goal is “to account for all possible sentences within a given corpus,
and this, with no exception” (page 330). While this objective seems feasible for morphology
or phonology, I doubt whether it can be achieved for syntax, let alone semantics.
The organization of the articles as chapters in the book seems arbitrary; I would classify the 14 chapters that follow the introduction into three groups. Three chapters (4, 10 and 12) deal with the mathematical and computational foundations of finite-state calculus. Four chapters (5, 7, 13 and 15) discuss practical applications of such techniques. The rest of the book focuses on linguistically motivated approaches that use finite-state mechanisms.

The Introduction (Chapter 1) presents some of the fundamental properties of finite-state automata and transducers, providing the essential definitions and outlining some of the commonly used algorithms, such as composition, intersection, determinization and minimization. It's a good introduction, although it's not clear whom it is aimed at: it is much too succinct for the novice, yet too basic for the expert. Also, as the remainder of the book is written by different authors, the introduction does not set a common notation to be used throughout the book. The lack of uniformity is evident even in the introduction itself; for example, transducers' edges are assumed to be labeled by words in definition 5, but definition 8 assumes they are labeled by letters. The main concepts are illustrated by examples from both formal and natural languages. Some are very useful, others are too trivial.

The articles that are dedicated to the mathematical foundations of finite-state technology are among the better ones in the book. In Chapter 4, Lauri Karttunen presents a thorough investigation into the ‘replace’ operator, starting with a good, concise survey of related approaches, a motivation and a summary of the results. Then, the simpler case of unconditional replacement is defined, followed by a helpful example and a list of special uses, including inverse, bidirectional and optional replacement. Karttunen then goes on to defining the more complex case of conditional replacement, this time providing insight into the process that led to the development of the results presented. The difficulties are explained and the solution is presented gradually, with precise definitions and verbal, informal descriptions. Again, several uses are presented with illuminating examples. The only drawback in this well written paper is the lack of linguistically motivated examples.

What Karttunen does to the replace operator, Mehryar Mohri does to sequential transducers in Chapter 12. In a very crisp paper, Mohri surveys several mathematical and computational properties of sequential and p-sequential transducers, and then demonstrates their uses in several areas of computational linguistics, such as phonology, morphology, lexicography, syntax and speech recognition. While the paper is not easy reading, it is clear and precise.

Chapter 10 is more computational than mathematical. It provides an algorithmic solution to a problem that is motivated by many of the other articles in this volume: intersecting several finite-state automata with a single, acyclic one. Pasi Tapanainen presents the naïve algorithm and then proceeds to improve it in several ways. After the final algorithm is presented, it is evaluated both by practical experimentation and by a theoretical complexity analysis.

The papers that describe practical applications of finite-state techniques are all interesting. In Chapter 5, Fernando Pereira and Rebecca Wright present an algorithm to compute a finite-state approximation of a context-free grammar. Since the motivation is to produce an efficient grammar out of a more linguistically motivated one, the approximation has to be sound. Thus, it is proven that if a string is generated by the context-free grammar, it is accepted by the finite-state approximation of the grammar. Moreover, the approximation is proven to be exact for certain families of grammars. The algorithm is explained so well, that the reader is left with the feeling that s/he would have done it exactly the same way. Fernando Pereira also co-authored (with Michael Riley) Chapter 15, which presents a framework for describing and implementing speech recognizers by combining weighted finite-state automata. Here, too, the algorithm is presented in a clear way, following a theoretical introduction to weighted transducers. The construction of the speech recognizer is discussed, followed by a description of the implementation and an outline of applications.

Emmanuel Roche and Yves Schabes present a finite-state part of speech tagger in Chapter 7. The tagger is inspired by Brill’s rule-based tagger, but it improves its efficiency considerably. This paper has everything: an interesting problem, a nifty solution, a formal presentation
of the algorithm, including a proof of correctness (in an appendix so as not to disturb the informal exposition), and even an empirical evaluation, showing a dramatic improvement over alternative methods.

Chapter 13 describes a finite-state transducer that is used to extract information from natural language texts. Hobbs et al. provide a detailed exposition of the system, concentrating on its modular construction and decomposing the language processing to several components, such as word recognition, phrase segmentation and event detection. The paper makes for easy reading and the structure of the system is presented in a clear way. A running example provides insight into the gradual construction of the output along the different stages of processing. However, the relation to finite-state techniques is only implied.

The rest of the book consists of articles that are somehow related to linguistics: either in their motivation or in their application. These articles vary greatly in their subject matter and in their quality. David Clemenceau’s Chapter 2, for example, is a technical description of a particular system for the morphological analysis of French; in Chapter 3, Kimmo Koskenniemi proposes two finite-state based formalisms for natural language processing, emphasizing the conceptual simplicity of a few representations and relations among them; and in Chapter 14, Eric Laporte discusses a transducer-based system for phonetic conversions, addressing the linguistic problems associated with such conversions and discussing several design alternatives.

In Chapter 6, Max Silberztein covers the issue of dictionary construction and maintenance with finite-state transducers. He then demonstrates how the finite-state representation of words and compound words can be used for lexical analysis and disambiguation. This is a very useful article for anyone who is interested in such applications, for any language with a reasonably constrained morphology. It remains to be seen whether such techniques can be successfully applied to languages with more complex morphology, where the ratio of inflected forms per base is in the thousands, rather than in the dozens.

Emmanuel Roche contributes Chapter 8, ‘Parsing with Finite-State Transducers’. The thesis is that “for the problem of parsing natural language sentences, finite-state models are both efficient and very accurate even in complex linguistic situations” (page 241). The paper assumes that a grammar for a natural language is given as a set of local contexts associated with each word. A finite-state transducer can then be constructed from each grammar entry, and the grammar is simply the union of these transducers. Parsing, then, consists in applying the grammar transducer to the input once, then applying it to the result of the first application, and so on, until a fixpoint is reached. Roche demonstrates this technique for several linguistic problems, such as support verb constructions, sentential complements and frozen expressions. But what is crucially missing is a formal discussion of the expressive power of the suggested technique, and an analysis of its complexity. Claiming that finite-state techniques are more adequate to parsing natural languages than are context-free grammars, one must show how context-free grammars can be encoded in finite-state transducers, and present an efficient parsing algorithm. Furthermore, the ‘analyses’ provided by the finite-state grammars are not really analyses in the linguistic sense of the word. Defining a modal as “a sequence that can be ignored during parsing” (page 255) is unacceptable. The whole notion of a grammar here is devoid of any relation to any linguistic theory. The question is, what purposes does the kind of analysis presented in the paper serve?

Atro Voutilainen’s Chapter 9 suffers from similar problems. When the introduction explicitly states that the author’s “perspective is linguistic rather than algorithmic” (page 283), one expects some sort of linguistic theory to be underlying the practical results. Unfortunately, this expectation is not fulfilled. Rather, this paper is a technical description of a regular description language and a particular grammar for English, defined in that language. The formalism seems to be easy to work with, albeit far from the requirements of most linguistic theories. The English grammar is described very superficially, more to demonstrate the formalism, and again the major question is: what are the advantages of the analyses produced by the system, and what are their linguistic merits?
In Chapter 11, Maurice Gross sets out a major undertaking: “to account for all possible sentences within a given corpus, and this, with no exception” (page 330). This goal is motivated by the observation that linguistically motivated grammars are unable to tackle a large number of ‘exceptions’, each requiring a special treatment. This is indeed a goal worth pursuing, but Gross does not explicate what is meant by ‘account for’, and the reader is left to assume that the meaning is ‘recognize’. This article presents a revolutionary approach to linguistics, which amounts to stipulating the entire set of data one is interested in. While this approach can be reasonable for morphology, as demonstrated in Chapter 6, it is inconceivable for syntax. Even the simple examples presented in the paper require systematic duplications of huge transducers, and the author refrains from elaborating on the size of transducers required for covering any real-size corpus.

The picture that emerges from reading the book can be summarized by quoting from Hobbs et al.’s Chapter 13: “The limitations of the applications of finite-state grammars to natural language processing have not yet been determined... these simple mechanisms can achieve a lot more than had previously been thought possible... We have shown that one can go a long way with simple techniques” (pages 389, 403). Indeed, finite-state techniques are provably useful for several natural language applications; but whether they are suitable for linguistically motivated characterization of natural languages remains to be seen.

SHULY WINTNER
University of Haifa
e-mail: shuly@cs.haifa.ac.il


Since the earliest days of natural language processing, researchers have tried to come up with ways of evaluating their systems and comparing the performance of different systems. Until the early 1990s, probably the most common way of demonstrating the quality of a system was to give examples of the linguistically interesting phenomena covered by the system. More quantitative approaches began to be developed in the early 1990s, leading to programs such as the MUC and ATIS system evaluations, and the PARSEVAL parsing evaluation (Sundheim 1991; Hirschman et al. 1993; Black et al. 1991). These programs were developed to provide quantitative ways of making cross-system comparisons, but they were extremely labor-intensive to carry out, and there was often some question as to their validity, in the sense of the relevance of the evaluation results to performance on real applications outside of the evaluation environment. In fact, the natural language processing field is still looking for valid cross-system comparisons that can be conducted with a reasonable amount of effort. This book is a useful but incomplete step in this direction. The designers of this evaluation paradigm have succeeded well in the goal of creating an approach which can be used across systems. The goal of an evaluation that can be conducted with a reasonable amount of effort is still in the future, as evidenced by the fact that only three of the eight development teams were able to complete the planned evaluation. In addition, this evaluation did not really try to address the question of validity for real world applications, since the participants were only focused on evaluating parsers, not full natural language processing systems.

The book describes a multi-system exercise in parser evaluation aimed at defining new and useful methods for parser evaluation. The organizers of the exercise invited eight research teams to test their parsers on the same corpus of 600 test sentences from software manuals, and to report on the results at a workshop held at the University of Limerick, Ireland, in May 1995. The participants were: UMIST (ALICE), the University of Helsinki (ENG-CG), the University of Limerick (LPARSER), the University of Manitoba (PRINCIPAR), the University of York (RANLT), Xerox Research Centre, Grenoble (SEXTANT), the National University of Singapore (DESPAR) and the University of Nijmegen (TOSCA). The goal was
to evaluate multiple systems using the same corpus and reporting on the same metrics, thereby
providing the basis for a reliable cross-system comparison of the various systems. Because the
original reporting of results at the workshop was not standardized, it was difficult to compare
the systems directly, and so the participants were asked to carry out a more standardized
analysis by applying the parsers to a 60-utterance subset of the original corpus and to report
the results in a standard format. The results of this second exercise are reported in this book.
Each system underwent a three-phase evaluation process, to separate the contributions of
the parser, the lexicon, and the grammar to the overall results. The first phase measured
performance with the parser’s original grammar and lexicon, the second phase permitted
additions to the lexicon, and the third phase permitted changes to both the lexicon and the
grammar. The reported results also include a standard set of metrics which are meaningful
across systems: (1) analysis of a set of particular constructs; (2) coverage, in the sense of how
many utterances in a corpus received an analysis; (3) efficiency (processing time); and (4)
accuracy. This degree of standardization is important for making meaningful comparisons
between the capabilities of the systems. The fourth set of results, the correctness of analyses,
was based on the following fairly straightforward metrics:

1. verbs recognized
2. nouns recognized
3. compounds recognized
4. phrase boundaries recognized
5. predicate-argument relations recognized
6. prepositional phrases attached
7. coordination-gapping analyzed.

These metrics are presented in nicely standardized tables in each paper that break out the
percentage of correct analyses for each construction in each corpus, and averages over corpora
and over constructions. The standardization of the results is very helpful when comparing
systems. The corpora selected consisted of selections from three software manuals. This corpus
is representative of a subset of real-life parsing problems, but relatively restricted in style, well-
edited, and consisting largely of fairly short sentences. A more challenging test would have
been to measure the performance of these systems on corpora with more complex sentences,
such as newspaper text, or more spontaneous, unedited, corpora, such as outputs from a speech
recognizer, or e-mail messages. Furthermore, a 60-sentence corpus is clearly inadequate for
a meaningful evaluation, and can be justified only if this exercise is aimed at no more than
investigating the evaluation technology, not performing a true comparison across systems.

The book also contains two general chapters, one by Dekang Lin advocating the use of a
dependency-based approach to comparing parsers, and one by Eric Atwell on comparative
evaluation of grammatical annotation models. The dependency-based approach was adopted
by the development teams as a standard way of comparing results.

The biggest problem with this book is that it ultimately does not provide enough informa-
tion to allow the reader to draw conclusions about either the quality of the systems or the
usefulness of the evaluation process. For example, all but three (ENG-CG, PRINCIPAR and
SEXTANT) of the teams were unable to complete the three planned phases of the evaluation.
It is likely that this failure was due in most cases to lack of resources rather than lack
of interest in completing the evaluation. Lack of resources, however, could result from any
one of at least four situations: (1) completing the evaluation was a prohibitive amount of
work for some teams because of their systems were poorly designed; (2) there were funding
disparities among the teams that made completing the evaluation possible for some teams
but not for others; (3) some teams may have failed to invest in good development tools,
thus slowing down the process of modifying the lexicon and grammar; or (4) the developers
were absorbed by other priorities. If completing the evaluation, (recall that this means adding
lexical items and grammar rules, which is almost always necessary for real applications) was
difficult because of poor system design, then this is a reflection on the quality of the systems.
in a way that funding disparities, lack of tools, or competing activities are not. However, we simply do not know why not all teams were able to complete all the evaluation phases.

No system improved dramatically across the phases, and it is also not clear what that means. The performance of the best of the systems hovered around the low 90%, in each phase of the evaluation, but again, it is not clear what this means for real applications. Is it good enough because in a complete system semantics and domain knowledge will compensate for parsing problems? Is it good enough because only a small amount of work on a specific domain would bring the performance up into the high 90%? Or is it not good enough because this error rate is unacceptable for some intended application? Or is it good enough for some applications (text classification) but not for others (database queries).

It is also worth pointing out that a valid evaluation which can be expected to play a role in decisions outside of the lab also requires that the test data be previously unseen by the system, which is not the case in this evaluation.

The book would also have been greatly strengthened by a summary chapter at the end. Many additional conclusions might have been drawn had there been a comparison across the participating systems. There were some very interesting differences across systems, such as dramatic differences in processing times, that are left with no explanation. For example, the average parsing time for the DESPAR system was 10.3 seconds and was 0.3 seconds for the SEXTANT system. Some of this difference is undoubtedly due to hardware differences, but the chapters also vary in the completeness of the descriptions of their hardware.

It is not clear that this approach would be very helpful for someone who was actually hoping to compare parsers in order to select one for a particular application. Although that was not the immediate goal of this evaluation exercise, it is to be hoped that this is at least one goal of evaluation activities in general. Beyond ruling out some parsers as unacceptably slow, this comparison doesn’t address a number of practical questions that need to be asked in a working application, such as ease of adapting the parser to a particular domain and extending the general capabilities of the parser, available hardware platforms, hardware requirements, interoperability with other system components, what programming language is it written in, and scalability. On the other hand, the approach described in this book might be very sensible for developers who are trying to measure incremental progress during the development of a specific system by comparing older and newer versions of the same system. This in effect was done in the three evaluation phases of this evaluation. This may ultimately be the most practical benefit to parsing evaluations, since many of the problematic aspects of cross-system comparisons can be controlled for (e.g. system goals, implementation language, formalism, hardware, etc.).

To sum up, this book represents a helpful step in the direction of improving the parser evaluation process, but it also clear that there is a long way to go until the natural language processing field has valid, standard, and practical ways of comparing different parsers.

References


DEBORAH A. DAHL
Unisys Corporation e-mail: deborah.dahl@unisys.com

The ‘traditional’ approach to processing spoken language makes use of the model of language as a set of independent layers: first phonetics, which extracts features from the acoustic stream, then phonology, in which these features are combined into candidate phonemes. Next, the lexical/morphological stage results in a set of words which might make up the utterance, while syntactic analysis will weed out anything that doesn’t comply to the expected standards of grammaticality. Finally, the semantic layer will yield the meaning of the original utterance, which has been filtered appropriately by all those consecutive layers of analysis.

The problem with this approach is that it does not work very well. Even though it sounds very logical and straightforward, actual spoken language does not at all lend itself to this kind of analysis. It is full of false starts, repetitions, corrections, and other constructions which would make it very hard for a full-blown syntactical analysis to get a grip on the correct reading. This is because grammatical formalisms are usually aimed at describing language in its entirety in a strictly formalised way. In processing spoken language, however, this is not at all what is required. Instead, using a performance-oriented formalism will yield much better results. One such formalism is case grammar, and case grammar is what the systems described in this book are using.

Case grammar uses templates to describe the structure of a sentence. These templates have a certain number of slots, which describe properties and arguments, and these slots need to be filled during the processing of the utterance. The exact types of slots are determined by the application’s domain: a flight, for example, has a place of departure, a destination, a number, and a time. A query to an air travel information system about flight schedules will require at least a certain number of these slots to be filled from the query so that a database query expression can be generated.

There are several approaches to extracting these slot-fillers from the input data, with the major ones being rule-based or stochastic. In the rule-based approach, an expert hand-crafts a number of rules describing for example how the destination of a flight could be expressed, whereas a stochastic parser learns the associations between words and their respective semantic labels from an annotated training corpus.

This book deals with travel information systems, first on air travel, then on train travel, which is the more complex of the two domains. Its aim is to evaluate how well a stochastic approach to semantic analysis does compared to a rule-based one. For this comparison it is not only performance which is taken into account, but also portability to both other languages and other domains. The starting point is an existing rule-based system, and it is compared with a newly-developed stochastic component, which is slotted into the place of the rule-based parser.

The first three chapters set the scene with an overview of grammar theories, related projects in the area of speech processing, and the the domain of travel information retrieval. They are fairly general, and provide an introduction to some of the problems faced by speech processing systems, as well as a discussion of different evaluation methods.

Despite the fact that all the utterances are spoken queries, the phonetics aspect of the process is not covered. Instead, the input consists of almost perfectly transcribed sentences, which only contain a few characteristics of speech, namely false starts and repetitions. This might seem a little bit idealised, but in such systems the vocabulary is usually fairly limited and thus easier to recognise. One amusing error is in the French train travel system, where “d’accord” is misinterpreted as “Dunkerque”. However, there are hardly any errors of this kind in the examples given.

After the initial introductions, chapter four then describes the rule-based case frame parser and its portability from French to English. There are few problems on the language side, as the two languages are similar and require not too many changes. The fact that the rules
are specified in a declarative way means that they can easily be changed manually. More problems occur when the size of the domain is considered: here the rule-based approach shows weaknesses, as the interaction between semantic concepts increases, and with it the complexity of the set of rules.

In the following chapter the stochastic parser is developed. This includes a brief outline of Hidden Markov Models and how they are used in speech processing. For the system described here the rules of the previous parser are replaced by a stochastic model, while later processing modules are shared with the rule-based parser. The stochastic case frame parser achieves about the same accuracy as the rule-based one, but a qualitative analysis shows that it is a lot more flexible and also more robust. This in itself would be something one would expect from a stochastic system, but it is still a bit surprising when the effort put into the systems is considered: the stochastic component does not require any rules to be written, instead it only needs a training corpus from which the relevant facts are extracted.

With the performance having been evaluated, the next step is to see how well the stochastic parser does when it comes to portability. Here the system is ported from English air travel to French train enquiries. As the stochastic component does not make use of any linguistic knowledge, porting it means basically creating a training corpus and applying it to the system. A further port was undertaken to a different domain, namely that of human appointment scheduling, where two humans agree on an appointment by way of a speech-to-speech translation system. This is considerably more complex than human-machine interaction, and consequently involves a more powerful semantic representation.

The final result of this experiment is that the stochastic component itself could be ported very straightforwardly, much more easily than the rule-based implementation. This applies both to human-machine interactions and the human-to-human interaction task. In both cases, it is robust and flexible, and its performance is comparable to the rule-based approach. The latter, however, is also influenced by the training data used; an evaluation of this influence has not been done. An important point that is briefly mentioned in the conclusions is that of representativity of the training data.

While the stochastic components do not outperform manually tuned rule-based systems, they have advantages in adaptability to other languages and domains. This is likely to become an important factor with a more widespread adaptation of speech processing systems, as it makes systems more feasible from a commercial point of view. The authors suggest several further avenues of research in which their findings could prove useful.

Altogether this book is well worth reading. The only criticism with it is the sloppy handling of references: quite a few references are omitted in the bibliography, which makes it hard to follow up some of the mentioned points in more detail. However, the research reported in this book is very interesting, and the authors managed to almost find the right level of detail in their descriptions, enabling the reader to understand how a system works without losing the overall picture.

OLIVER MASON
Birmingham University
e-mail: O.Mason@bham.ac.uk