Review John Coleman, Introducing Speech and Language Processing Cambridge University Press, 2005

Daniel J. Hirst

John Coleman 2005. *Introducing Speech and Language Processing*. Cambridge University Press. ISBN 0-521-53069-5

Reviewed by Daniel Hirst CNRS Laboratoire Parole et Langage Université de Provence, Aix en Provence daniel.hirst@lpl.univ-aix.fr

It is unfortunate that there is still today an enormous gap between the community of linguists and phoneticians on the one hand and that of engineers and computer scientists on the other. Each community needs the other and, in an ideal world, linguists would provide theoretical frameworks and data which are useful to engineers, while engineers would provide tools which are useful to linguists. The exchange between the two communities, however, is in practice very slow. It often takes decades for ideas which are current in one community to be adopted by the other. The ideas of non-linear phonology which have dominated phonological analysis since the 1970's, for example, are only recently being integrated into speech synthesis and speech recognition systems. At the same time, linguists and phoneticians working on corpus analysis often use extremely primitive tools and software: there is still today, for example no widely available tool for robust automatic alignment of a waveform with a transcription. Although such tools do exist they are generally not available in a form which the average linguist or phonetician can easily use. The result is that linguists working on spoken corpora can spend hundreds of hours providing manual alignment of their data and, worse of all, that this manual alignment is very often not even fed back as a resource for evaluating automatic alignment algorithms. Any attempt to bridge this gap is consequently more than welcome. John Coleman's *Introducing Speech and Language Processing* is precisely such an attempt and is consequently to be highly recommended.

All in all, John Coleman does an excellent job of presenting many of the major algorithms in use today in both speech technology and natural language processing and he does so clearly and concisely, adopting much of the time a very laudable hands-on approach: readers are almost immediately confronted with a program written in C which the author then proceeds to explain in some detail. As an exercise, the readers are asked to compile and run the program and then to modify it in various ways. This is almost certainly the most efficient way to get students over the first hurdle of thinking they will never be able to understand anything about programming. Instead of writing little programs that display silly messages like "Hello world!" on their computer screen, they are shown how a list of understandable instructions can be used to produce a sound, which they can then hear, display and modify.

Another originality of this book is the fact that it attempts to deal at the same time with both signal processing and natural language and showing that the two areas have a lot in common. This again is a very welcome step and we can perhaps look forward to a new generation of

linguists with a solid background in both these areas. The two "strands" of the book, as Coleman calls them, 'Speech' and 'Language', are interwoven so that after an introduction and three chapters dealing with Speech: Sounds and numbers (Chapter 2), Digital filters and resonators (Chapter 3), Frequency analysis and linear predictive coding (Chapter 4), the readers are then introduced to the concepts of Finite-state machines (Chapter 5), this time using the language Prolog rather than C. This gives them enough background to move back to the 'Speech/C' strand and subsequently to be introduced to a selection of techniques used in speech recognition (Chapter 6), before shifting back once more to the 'Language/Prolog' strand, with the final three chapters on Probabilistic finite-state models (Chapter 7), Parsing (Chapter 8) and Using probabilistic grammars (Chapter 9).

Like other books in the Cambridge Introductions to Language and Linguistics series, this one is extremely well produced and attractively presented. The text is accompanied by numerous illustrations and includes a useful glossary of technical terms as well as an index and list of references. All the software described in the book, together with a C compiler and a Prolog interpreter are included on the accompanying CD-Rom and the reader is also referred to a companion website (www.islp.org.uk) with additional linksand updated information. So whether or not this is the perfect introduction to speech and language processing for students with no technical background, it's probably as close as we are likely to get for some time.

What follows are a few thoughts on what might possibly have made the book even better. As a Macintosh user, I was personally at first disappointed to see that the software, originally developed for use in a Unix environment, has been written for PC/Windows users only, although the author adds a note for Mac and Unix users saying that the software can be compiled and run under other operating systems, given appropriate technical support. The first program, *coswave.c* compiled successfully on my Mac OSX system, but others repeatedly gave a "Bus error". The same thing happened on a Linux system and my colleague, Robert Espesser, suggested modifying the source code, replacing all occurrences of "*length" by "length" and all occurrences of "length" by "&length". Once this was done, the other C programs all compiled successfully, so potential readers using Mac/Unix need not be put off by this.

In chapter 5, (Finite state machines) the author says that the model NFSA1 (Figure 5.3) accepts "almost all the monosyllabic words in Mitton 1992 (…) apart from a few very unusual words, mostly foreign words." [p115]. There is one important group of words which are not accepted by this model, however, and that is all those which begin with a vowel or with /ju/ ("I" and "you" are hardly rare or foreign!). In fact this would have been a good opportunity to introduce the solution to this problem: two empty transitions (1, ", 4) and (1, ", 3). Alternatively, the

vowel/glide problem could have been solved by allowing more than one start state just as more than one end state is included.

The presentation of probabilistic finite state models in Chapter 7 is rather more abstract than a lot of the other material and it is a pity that the students could not have been given some practical applications using Hidden Markov Models. At the end of the chapter,there is a seven page criticism of Chomsky's objections to Markov models, as formulated in the late '50s and '60s which seems out of character with the introductory/didactic nature of the rest of the book.

Table 5.1 presents an Ascii coding of the IPA, credited to Mitton 1992. This is, in fact, an only very slightly modified version of the SAMPA alphabet developed in the 80's during the European Esprit Project SAM (Wells 1997). The only differences are the symbol /&/ for the vowel of "hat" instead of the SAMPA symbol /{/ and /0/ instead of the SAMPA symbol /Q/ for the vowel of 'hot'. In XSAMPA, /&/ is an open front rounded vowel and '0' is used as a diacritic for devoicing as in e.g. /b_0/). It is a pity that this attempt to define an International standard for Ascii coding is not given due credit here.

The use of Prolog for the "Language" strand made me wonder if readers would ever be able to produce anything useful for their research with this language. But on reflection, the same thing applies to C. The author's aim is clearly not to turn the readers into programmers but to get them to understand how the algorithms work. In fact students might usefully be warned about the dangers of amateur programming instead of using tried and tested freely available software. In his conclusion, Coleman suggests that readers that have reached this point should set their sights higher and "get to grips with some more technical works", recommending among others works on C and Prolog programming. In my opinion, once readers have acquired the level of understanding provided by this textbook they would probably do better to concentrate on learning to use some of the excellent software that is freely available today, rather than expecting to write such software themselves. They are much more likely to be able to produce useful applications using, for example, the scripting facility of *Praat* (Boersma & Weenink 2005) or with the *Cambridge HTK Language Modelling Toolkit* (Young et al. 2005), both of which are referenced on the companion website, rather than writing their own programs. Perhaps the most surprising omission in the book is a description of the mechanism of neural networks although Coleman does give references (p 7). The subject could, though, have been given a brief presentation so that the basic ideas would be familiar. For a hands-on approach to neural nets, student could be referred to David Weenink's tutorial in the online documentation of the Praat software (op. cit.). Another tool which they might usefully have been introduced to is that of Regular Expressions, using Perl (Wall et al.2000), which in my experience, like the

Praat scripting language, is one that students with a humanities background, can quickly learn to apply.

In conclusion, Coleman has produced an excellent textbook and one which will be extremely valuable to many students and teachers since it does a great deal to render accessible an area which is usually only covered by much more technical works. Even so, readers should be warned that they must not expect an easy ride through the lands of speech and language processing, although they can be guaranteed that it will be an exciting one.

References

Boersma, P. and Weenink, D. 2005. Praat: doing phonetics by computer. (Version 4.3.28 November 7, 2005) [computer program] http://www.praat.org

Young, S., Evermann, G. Gales, M., Hain, T., Kershaw, D., Moore, G., Odell, J., Ollason,D.,Povey, D.,Valtchev, V. andWoodland, P. 2005. *The HTK Book (for HTK 3.3)*. http://htk.eng.cam.ac.uk/

Wall, L., Christiansen, T. and Orwant, J. 2000. *Programming Perl* (3rd edition) O'Reilly.

Wells, J., Barry, W., Grice, M., Fourcin, A., and Gibbon, D., 1992. *Standard Computer-Compatible Transcription*. Esprit project 2589 (SAM), Doc. no. SAM-UCL-037. London: Phonetics and Linguistics Dept., UCL.

Wells, J.C., 1997. SAMPA computer readable phonetic alphabet. In Gibbon, D., Moore, R. and Winski, R. (eds.), 1997. *Handbook of Standards and Resources for Spoken Language Systems*. Berlin and New York: Mouton de Gruyter. Part IV, section B.http://www.phon.ucl.ac.uk/home/sampa) Wells, J.C. 2000 Computer-coding the IPA: a proposed extension of SAMPA. http://www.phon.ucl.ac.uk/home/sampa/x-sampa.htm.