Lattice Parsing for Speech Recognition

J.-C. Chappelier, M. Rajman, R. Aragües and A. Rozenknop

EPFL, DI-LIA, INR (Ecublens), CH-1015 LAUSANNE (SUISSE)

Abstract

A lot of work remains to be done in the domain of a better integration of speech recognition and language processing systems. This paper gives an overview of several strategies for integrating linguistic models into speech understanding systems and investigates several ways of producing sets of hypotheses that include more "semantic" variability than usual language models. The main goal is to present and demonstrate by actual experiments that sequential coupling may be efficiently achieved by word-lattice syntactic analyzers, efficiently parsing the huge number of hypothesis (i.e. possible sentences) contained in the lattice produced by the speech recognizer.

1. Motivations

The past decade has seen significant progress in speech recognition technology: word (recognition) error rates continue to drop by a factor of 2 every two years (Rabiner *et al.*, 1996) and high performance systems are now becoming available. Several factors have contributed to this rapid progress:

- Generalisation and continuous improvements of the powerful Hidden Markov Model (HMM);
- Better language models and powerful algorithms allowing their integration in speech recognition systems;
- Production of large speech corpora allowing researchers to optimize parameters of the recognizers in a statistically meaningful way;
- Establishment of standards for performance evaluation and advances in computer technology.

However, speech recognition remains a difficult problem, due to the large variability associated with the input signal it considers. In particular a lot of work is to be done towards a better integration of linguistic models into continuous speech recognition systems, although several ways have already been studied: statistical models (bigrams and trigrams), stochastic or deterministic finite state automata (FSA), and context-free grammars (CFGs). These approaches are indeed mainly designed to reduce the overall word-error-rate and are not necessarily appropriate for higher level knowledge post-processing.

The current paper investigates several ways of producing sets of hypotheses[1] that include more "semantic" variability, becoming therefore more appropriate for higher-level linguistic post-processing.

[1] i.e. possible transcriptions for the speech input utterance.

The main goal is to present and overview several strategies for integrating linguistic[2] context-free models into speech understanding systems.

2. Linguistic Models and Speech Recognition Coupling: State of the Art

Word recognition is not sufficiently reliable to enable multi-word utterances to be recognized with acceptable accuracy without the additional use of a language model (Wright *et al.*, 1992). Two main classes of language models exists: the N-gram based statistical models (Baker, 1989; Russell *et al.*, 1990), flexible and adaptable but which suffer from considerable over-generation (diminishing accuracy), and the stochastic phrase-structure grammar model (Murveit & Moore, 1990; Kita *et al.*, 1990), relatively inflexible and difficult to adapt (hence prone to under-generation) but providing better performance for those sentences that are recognized by the grammar (and also providing with a structure that can be used by further linguistic post-processing, e.g. semantic modules).

The standard N-gram-based statistical language models suffer from two main shortcomings: the huge number of parameters that need to be estimated[3] (data sparseness); and limited scope dependencies.

Several partial solutions to data sparseness have been proposed:

- smoothing techniques, which have led to the backing-off model (Katz, 1987) and to the combined model (Jelinek & Mercer, 1980).
- enhancement of the N-gram estimates with estimates of class-sequence probabilities (such as sequence of lemmas or sequence of Parts-of-Speech tags). The number of distinct N-classes is actually smaller than the one of N-grams (Dumouchel *et al.*, 1988; Elbeze & Derouault, 1990; Cerf-Danon & Elbeze, 1991).

Concerning the limited scope of probabilistic dependencies, the *coupling* with higher level linguistic information is required in the speech recognition process. Several coupling modalities have already been investigated (Rayner *et al.*, 1994; Roussel & Halber, 1997) and the resulting models are often called hybrid models (Jones, 1992). Three main architectures have been proposed: tight-coupling (Wright *et al.*, 1993; Jurafksy *et al.*, 1995); sequential coupling; and mixture of experts (Bourlard, 1995). Since they are the most often used, only the first two architectures are now described further.

2.1. Tight-Coupling

In the framework of HMM recognition systems, the search algorithm should, in principle, consider all possible hypotheses, evaluate their posterior probability using all available knowledge sources, and then choose the hypotheses with the highest probability. Such cases, where the constraints expressed by the linguistic model(s) are directly integrated in time synchronous recognition algorithms, corresponds to *tight-coupling*.

In this approach, the real-problem is to effectively integrate all the linguistic informations that could be used to reduce the size of the hypotheses space by determining which word can come next to any hypothesised end of word, and with what probability. This requires for the linguistic model to be formulated in a predictive, left-to-right manner[4], which may impose strong restrictions on the type of grammar that can be used.

---

[2] Although not really linguistic in the wide sense, we will use the term "*linguistic models*" as opposed to the usual term of "language models", here designing only N-grams of words.

[3] V^N, where V is the size of the vocabulary used

[4] i.e. prefix probabilities

Several solutions have been studied for tight-coupling:

1. to use a non-probabilistic linguistic model to generate word-transition list and assign the probabilities by the usual N-gram language methods (Goodine *et al.*, 1991; Hauenstein & Weber, 1994);
2. to use the linguistic model to smooth/estimate N-gram probabilities of words, providing a direct transfer to usual language model. The advantage of such an approach is that it requires almost no modification of the acoustic decoder. However, the integration of long distance dependencies still remains very limited. Various degrees of approximation of the linguistic model can be used for the N-gram probabilities approximation: theoretical values may be more (Stolcke & Segal, 1994) or less (Zue *et al.*, 1991; Corazza *et al.*, 1992) directly derived from the model.
3. to construct a finite state representation of the linguistic model (Pereira & Wright, 1991; Evans, 1997) for which efficient decoding algorithms are available (Bridle *et al.*, 1982; Lee & Rabiner, 1988);
4. to use the linguistic model as a (probabilistic) predictive model for word transitions (Goddeau, 1992; Jurafksy *et al.*, 1995). Notice, however, that for stochastic context-free grammars (SCFGs), this kind of approach leads to suboptimal solutions since the parser's probabilities have to be locally approximated (Jurafksy *et al.*, 1995).
5. to directly use a SCFG as a language model (without approximations) by, for instance, directly parsing the phone lattice with a phone-level grammar. However, such an approach has in principle the drawback of being slow since the parsing algorithms have a cubic time worst-case dependency on the input length[5].

2.2. Sequential Coupling

In the sequential approach, the basic assumption is that, while all available knowledge sources (acoustic, lexical, syntactic, semantic) contribute to improve overall recognition accuracy, the influence and the computational cost of each vary greatly. For example, a first-order statistical language model can reduce perplexity[6] by at least a factor 10 with little extra computation, while applying a complete natural language model of syntax and semantics to all partial hypotheses typically requires heavy computation for less perplexity reduction (Murveit & Moore, 1990; Schwartz & Austin, 1991).

It is therefore advantageous to apply the knowledge sources one or two at a time, in a sequential order proper to progressively constrain the search : the most powerful knowledge sources are used first to produce an intermediate result which is then filtered and reordered on the basis of the information provided by the remaining knowledge sources. Such an approach is called the sequential "N-best paradigm" (Schwartz & Chow, 1990; Young, 1984) or the word-lattice approach (Su *et al.*, 1992).

A usual implementation of the sequential approach is to use a linguistic analyzer (e.g. a syntactic parser) operating on the N-best hypotheses of the recognizer, compactly represented as a word lattice. The advantage of such an approach is that the techniques developed for language processing can be used with almost no modification. The standard polynomial-time context-free parsing algorithms that are most frequently used are essentially variations of the Earley top-down parsing algorithms (Earley, 1986; Stolcke, 1995; Nederhof & Satta, 1997) or of the CYK bottom-up parsing algorithm (Graham *et al.*, 1980; Nederhof, 1994) with more or less improvement for the coupling with a speech recognizer (Thomason, 1986; Fred & Leito, 1993; Jurafksy *et al.*, 1995).

[5] which in that case would be the number of frames, one per 10 ms.

[6] perplexity is value quite well inversely correlated with recognition accuracy.

However, the two main drawbacks of sequential implementations are that:

1. it loses the advantage of jointly considering speech and language information, which usually greatly reduces the perplexity and therefore the number of the states to be explored.
2. the syntactic module is often still insufficient for selecting a unique sentence. Additional semantic information has then to be integrated so as to further limit ambiguity. Several extensions concerning either the probabilistic model or the linguistic formalism have been considered (tri-gram models for the stochastic modelling of the syntactic rules (Maruyama, 1990), optimized first-order dependence models (Wright *et al.*, 1992), Spoken Language Constraint Networks corresponding to extensions of Constraint Dependency Grammars (Shieber, 1985; Kita *et al.*, 1990), more complex syntactic formalisms, such as unification-based grammars).

3. Lattice Parsing

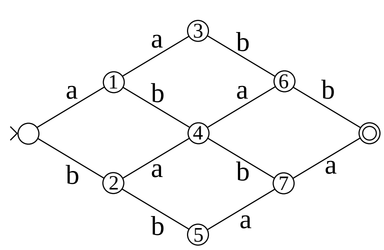
3.1. Adaptation of the Parser Input

For both tight and sequential coupling, lattice parsing (i.e. the parsing of a whole set of sequences compactly represented in the form of a lattice) is useful: for tight-coupling the parser takes the phoneme-lattice and a phoneme grammar as input, whereas in sequential coupling it parses (and filters out) the word lattice with some (word) phrase grammar.

Standard syntactic analysers however assume inputs in the form of a single sequence of words which is not compatible with the lattice-based output of the acoustic modules. The large lattices that are produced correspond to compact representations of the N-best hypotheses but, as N is very large in usual situations, it is not realistic to sequentially extract all the hypotheses and submit them one after the other to the analyser. Therefore, the parsing algorithm has to be changed in order to directly cope with an input in the form of a phone- or word-lattice.

We developed a parser (Chappelier & Rajman, 1998b; Chappelier & Rajman, 1998c) able to simultaneously take into account the specificities of the output produced by the acoustic modules (word- or phone-lattices) and to integrate probabilistic syntactic models such as simple SCFGs or more sophisticated stochastic tree-substitution grammars, such as Data-Oriented Parsing (Bod, 1995).

In order to illustrate how parsing takes place in the case of lattices, let us consider a simple example of a word lattice containing the following sentences: "a a b b", "a b a b", "b a a b", "b b a a", "b a b a" and "a b b a" (cf figure 1), the letters a and b beeing some words in the language.



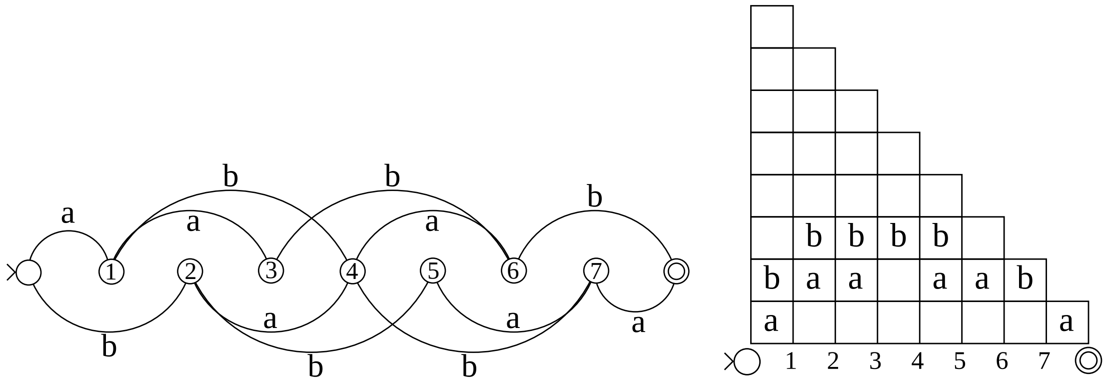
a a b b
a b a b
b a a b
b b a a
a b b a
b a b a

Figure 1: A toy word lattice containing 6 sentences.

Such a lattice may be easily converted in a chart as follows: all the nodes are ordered[7] by

[7] with random choice in case of equality.

increasing depth[8] (cf figure 2 (a)). Notice that such an order naturally occurs in the case of speech recognition lattices, in which nodes correspond to (chronologically ordered) times instants. This representation of the lattice is then directly mapped to a chart, by associating the node labels to the indices of the column of the chart table (cf figure 2).



(a) (b)

Figure 2: (a) Word lattice of fig 1 represented as a speech word-lattice (nodes are naturally ordered since they represent different time-instants). (b) The representation of the word lattice in (a) in the form of a chart table. Columns represent different time-instants, rows different sequence length.

As far as the parsing algorithm is concerned, not much needs to be changed but the initialisation step: rather than initializing the chart with POS tags only in the first row of the table as it is usually the case, initialization now occurs in all the cells corresponding to an arc in the lattice. More precisely, if in the lattice there is an arc (i, j) $(j > i)$ labelled by w, then the POS tag corresponding to w will be stored in the cell $(i, j - i + 1)$ of the chart table. The rest of the parsing algorithm remains unchanged.

3.2. Most-Probable Parse vs. Most-Probable Sentence

Usually, the aim of lattice parsing is to find the most probable sentence (according to the linguistic model) contained in the lattice. However, what a stochastic parser naturally provides however are the most-probable parses. It should be stressed that the sentence associated with the most-probable parse does not necessarily correspond to the most-probable sentence.

Let's illustrate this point on the toy example of figure 1 and the SCFG given in table 1. The

S -> AB BA	(0.22)	S -> BA BA	(0.19)
S -> BAB A	(0.19)	S -> X	(0.4)
X -> X X	(0.4)	X -> A	(0.3)
X -> B	(0.3)	AB -> A B	(1.0)
BA -> B A	(1.0)	BAB -> B AB	(1.0)
A -> a	(1.0)	B -> b	(1.0)

Table 1: A sample SCFG for parsing the toy lattice.

probability of the most probable parse as well as the sentence probability for all the sentences

[8]the minimal distance, in terms of number of arcs, from the initial node of the lattice.

contained in the considered word-graph are given in table 2. The sentence that has the most probable derivation is clearly not the most probable sentence.

sentence	# interp	MPP	sentence probability
a a b b	5	$p_1 \simeq 2.04$e-4	$5 \cdot p_1 \simeq 1.04$e-3
a b a b	5	$p_1 \simeq 2.04$e-4	$5 \cdot p_1 \simeq 1.04$e-3
b a a b	5	$p_1 \simeq 2.04$e-4	$5 \cdot p_1 \simeq 1.04$e-3
b b a a	5	$p_1 \simeq 2.04$e-4	$5 \cdot p_1 \simeq 1.04$e-3
a b b a	6	$p_3 = \mathbf{0.22}$	$5 \cdot p_1 + p_3 \simeq 2.21$e-1
b a b a	7	$p_2 = 0.19$	$5 \cdot p_1 + 2 \cdot p_2 \simeq \mathbf{3.81e\text{-}1}$

$p_1 = 0.3^4 \cdot 0.4^3 \cdot 0.4 \simeq 2.04$e-4

Table 2: Most-probable parse (MPP) and sentence probabilities for the toy example. The most-probable sentence is "baba" whereas the most-probable parse is associated with "abba".

Actually, finding the most probable sentence in a word lattice is a NP-hard problem[9] (Sima'an, 1996). However approximations may be found in reasonable time (i.e. with polynomial time complexity) by controlled sampling (Chappelier & Rajman, 1998a).

3.3. Experiments

We wanted to experimentally address two questions: does the sequential coupling significantly improve the recognition? And is it feasible in a reasonable time?

The first preliminary experiments we made on sequential coupling with lattice parsing were, for both aspects, promising. They were made on a set of ten spoken utterances with lots of ambiguities[10] as illustrated in table 3 (Aragues *et al.*, 1999). Experiments were made for several

sentence	1	2	3	4	5
(a)	$> 5 \cdot 10^{10}$	$> 5 \cdot 10^{10}$	$> 5 \cdot 10^{10}$	$> 5 \cdot 10^{10}$	3 261 227 208
(b)	521 243	862 334 592	4 594 041	3 326 421 643	984
sentence	**6**	**7**	**8**	**9**	**10**
(a)	$> 5 \cdot 10^{10}$	$> 5 \cdot 10^{10}$	$> 5 \cdot 10^{10}$	$> 5 \cdot 10^{10}$	6 009 538 238
(b)	290 658	5 873	92 322	26 740	273

Table 3: Ambiguities for the ten considered sentences. (a) number of word sequences in the word lattice produced by the speech recognizer ; (b) number of parses in the lattice.

different parameters of the speech recognizer[11]. On the average over all the experiments, in 35% of the cases the coupling with a SCFG strictly improved the results and in 67% of the cases it did at least as well as without SCFG. Furthermore, when restricted to the two parameter sets for which the speech-recognizer produced its best results, the former results improved to 50% and 80% respectively.

[9]The fact that this simple toy example only contains 33 derivations should not hide the fact that for such a grammar there is an exponential number (in the sentence size) of derivation. Mixing all these derivations together for several sentences at the same time makes the problem of finding the MPS NP-hard.

[10]and high recognition word-error rate

[11]Several values for the "acoustic factor" were tested.

It is worth emphasizing that, using the computationally efficient parser we developed (Chappelier & Rajman, 1998b), the overhead in time due to the adjunction of a SCFG filter is negligible with respect to the speech recognition time as shown in table 4.

	SR	average total coupling time			
		(a+b+c)	(a)	(b)	(c)
P1	219	0.72	0.08	0.54	0.1
P2	209	0.85	0.08	0.67	0.1
P3	81	0.81	0.08	0.63	0.1

Table 4: Coupling time vs. Recognition time for different parameter settings (P1, P2 and P3). SR represents the average speech recognition time using the STRUT recognizer (STRUT, 1996) and the NOWAY decoder (Renals, 1994). For the parser: (a) speech lattice to parser chart conversion, (b) actual parsing time, (c) parser input and output. Times are given in seconds of CPU time on a SPARC ULTRA 1.

These preliminary results are encouraging, demonstrating that sequential coupling with lattice parsing both improves the recognition and can be achieved with realistic computation times.

4. On the Use of Context-Free Grammars as Linguistic Models

4.1. Compromise between Linguistic Description and Computational Efficiency

Some may argue that context-free models are not very realistic from a linguistic perspective and that higher-level descriptions[12] should be used instead. Others may argue that context-free models are not realistic for real-time speech recognition since they may lead to important overheads in computational cost for real-world applications.

For real-world applications, only methods with low algorithmic complexity can be considered. Such a constraint imposes severe limitations on the sophistication of the linguistic models that can be used, and most often, only FSA models, which may be very efficiently parsed (linear time), can be considered. However, lower-level language models such as FSA do not provide the analysed sentence with useful syntactic structure that may be necessary for the subsequent processing modules (e.g. semantic).

We argue that CFG have both advantages to be efficient enough for implementation in real-time speech recognition applications and to incorporate at least some linguistic descriptions.

4.2. Context-Free Grammars: the "Machine Code" for Time-Sensitive Applications

We are nonetheless well aware of the fact that the writing of a grammar (mandatory for applications that require good performances) cannot reasonably be made within a context-free formalism. In our opinion, CF formalim is rather to be considered as a kind of "machine code" for syntactic descriptions, provided that some translation mechanisms from higher level description formalisms are available.

The applicative framework that we are currently using for the evaluation of such methods is the design and implementation of a dialogue-based information system for vocal information services

[12] HPSG for instance

and, in particular, spoken interaction with a phone-book inquiry system. In this framework, a unification grammar containing 92 rules and 25 non-terminals was written and then automatically converted into a CFG with 28 253 rules and 6 648 non-terminals. The probabilities of the CFG are to be learned in a second phase, either from a tagged corpus (simple estimation) or from an untagged corpus using the inside-outside algorithm. The 'huge' SCFG thus obtained is then used to parse, with our parsing toolbox, lattices produced by the speech recognizer.

The conversion from higher to lower level formalism was realized by introducing non-terminals for each possible attribute-value pair for each non-terminal of the former grammar. When applied naively, such a method leads to an explosion of the number of rules (due to the multiplication of possible cases): over than 700 000 rules for the 92 rules above mentioned grammar. We avoid this combinatorial explosion by introducing when necessary for each original rule new non-terminals taht represent attribute-value pairs that are actually not used in the rule. These new non-terminal absorb the useless associations and avoid their combinatorial propagation[13].

When such efficient automatic conversion mechanisms are available, it clearly appears that SCFG represent for practical use applications a reasonable compromise between a linguistic description useful to help speech recognition and dialogue management, and a good parsing efficiency.

5. Conclusion

After a rather detailed state of the art in the domain of the coupling of a speech recognizer with NLP modules, clearly stating what had and had not been done, we presented the major role played by lattice parsing, i.e. *efficient* parsing of the huge number of hypotheses contained in the lattice produced by a speech recognizer. Such an approach could be applied either to phone-lattices in the case of a tight-coupling or to word-lattices in the case of sequential coupling.

We then demonstrated by actual experiments that sequential coupling may be efficiently achieved by word-lattice parsers. This sequential coupling brings a 50% strict improvement of the speech recognition. Such very promizing preliminary results have now to be confirmed on a larger scale.

References

ARAGUES R., CHAPPELIER J.-C. & RAJMAN M. (1999). *Integration of syntactic constraints within a Speech Recognition System: Coupling a Speech Recognizer and a Stochastic Context-Free Parser.* Technical Report DI-98/309, EPFL, Lausanne (Switzerland).

BAKER J. M. (1989). Dragondictate-30K: Natural language speech recognition with 30 000 words. In *Proc. of Eurospeech'89*, p. 161–163.

BOD R. (1995). *Enriching Linguistics with Statistics: Performance Models of Natural Language*. Amsterdam (The Netherlands): Academische Pers.

BOURLARD H. (1995). Towards increasing speech recognition error rates. In *Proc. of EUROSPEECH'95*, p. 883–894, Madrid (Spain).

BRIDLE J. S., BROWN M. D. & CHAMBERLAIN R. M. (1982). An algorithm for connected word recognition. In *Proc. of ICASSP'82*, p. 899–902.

CERF-DANON H. & ELBEZE M. (1991). Three different probabilistic language models: Comparison and combination. In *Proc. of ICASSP'91*, volume 1, p. 297–300.

[13] such a method reduces the product of all useless combinations to their sum.

CHAPPELIER J.-C. & RAJMAN M. (1998a). Extraction stochastique d'arbres d'analyse pour le modèle dop. In *Proc. of 5ème conférence sur le Traitement Automatique du Langage Naturel (TALN98)*, p. 52–61, Paris (France).

CHAPPELIER J.-C. & RAJMAN M. (1998b). A generalized CYK algorithm for parsing stochastic CFG. In *TAPD'98 Workshop*, p. 133–137, Paris (France).

CHAPPELIER J.-C. & RAJMAN M. (1998c). *A Practical Bottom-Up Algorithm for On-Line Parsing with Stochastic Context-Free Grammars*. Technical Report DI-98/284, EPFL, Lausanne (Switzerland).

CORAZZA A. *et al.* (1992). Computation of upper-bounds for stochastic context-free languages. In *Proc of AAAI'92*, p. 344–349.

DUMOUCHEL P., GUPTA V., LENNIG M. & MERMELSTEIN P. (1988). Three probabilistic language models for a large-vocabulary speech recognizer. In *Proc. of ICASSP'88*, volume 1, p. 513–516.

EARLEY J. (1986). An efficient context-free parsing algorithm. In *Readings in Natural Language Processing*, p. 25–33. Morgan Kaufmann.

ELBEZE M. & DEROUAULT A.-M. (1990). A morphological model for large vocabulary speech recognition. In *Proc. of ICASSP'90*, volume 1, p. 577–580.

EVANS E. G. (1997). Approximating context-free grammars with a finite-state calculus. In *Proc. of ACL/EACL'97*, p. 452–459.

FRED A. L. N. & LEITO J. M. N. (1993). An heuristic-based context-free parsing algorithm. In *Proc. of ICASSP'93*, volume 2, p. 67–70.

GODDEAU D. (1992). Using probabilities shit-reduce parsing in speech recognition systems. In *ICSLP'92*, volume 1, p. 321–324, Banff (Canada).

GOODINE D., SENEFF S., HIRSCHMAN L. & PHILLIPS M. (1991). Full integration of speech and language understanding in the MIT spoken language system. In *Proc. of Eurospeech'91*, p. 24–26, Genova (Italy).

GRAHAM S. L., HARRISON M. A. & RUZZO W. L. (1980). An improved context-free recognizer. *ACM Transactions on Programming Languages and Systems*, **2**(3), 415–462.

HAUENSTEIN A. & WEBER H. H. (1994). An investigation of tightly-coupled time-synchronous speech language understanding using a unification grammar. In *Proc. of AAAI'94, Workshop on Integration of Natural Languages and Speech Processing*, p. 42–49.

JELINEK F. & MERCER R. L. (1980). Interpolated estimation of markov source parameters from sparse data. In E. L. GELSEMA & L. N. KANAL, Eds., *Pattern Recognition in Practice*, p. 381–397. North-Holland.

JONES G. J. F. (1992). The HMM interface with hybrid grammar-bigram language models for speech recognition. In *Proc. of ICSLP'92*, p. 253–256.

JURAFKSY D., WOOTERS C., SEGAL J., STOLCKE A., FOSLER E., TAJCHMAN G. & MORGAN N. (1995). Using a stochastic context-free grammar as a language model for speech recognition. In *Proc. of ICASSP'95*, p. 189–192.

KATZ S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. on ASSP*, **34**(3), 400–401.

KITA K., KAWABATA T. & HANAZAWA T. (1990). HMM continuous speech recognition using stochastic language models. In *Proc. of ICASSP'90*, p. 581–584.

LEE C. H. & RABINER L. R. (1988). A network-based frame synchronous level building algorithm for connected word recognition. In *Proc. of ICASSP'88*, p. 410–413.

MARUYAMA H. (1990). Structural disambiguation with constraint propagation. In *Proc. of ACL'90*.

MURVEIT H. & MOORE R. (1990). Integrating natural language constraints into HMM-based speech recognition. In *Proc. of ICASSP'90*, p. 573–576.

NEDERHOF M.-J. (1994). An optimal tabular parsing algorithm. In *Proc. of 32nd Annual Meeting of the Association for Computational Linguistics (ACL'94)*, p. 117–124, Las Cruces (New Mexico).

NEDERHOF M.-J. & SATTA G. (1997). A variant of earley parsing. In *AI*IA 97: Advances in Artificial Intelligence, 5th Congress of the Italian Association for Artificial Intelligence*, volume 1321 of *LNAI*, p. 84–95: Springer Verlag.

PEREIRA F. C. N. & WRIGHT R. N. (1991). Finite state approximation of phrase-structure grammars. In *Proc. of ACL'91*, p. 246–255.

RABINER L. R., JUANG B. H. & LEE C. H. (1996). An overview of automatic speech recognition. In C. H. LEE, F. K. SOONG & K. K. PALIWAL, Eds., *Automatic speech and speaker recognition*. Kluwer.

RAYNER M. *et al.* (1994). Combining knowledge sources to reorder n-best speech hypothesis lists. In *Proc. of the Human Language Technology Workshop*, p. 217–221.

RENALS S. (1994). Noway's manual page. University of Cambridge, http://www.clsp.jhu.edu/ws96/ris/man/noway.doc.

ROUSSEL D. & HALBER A. (1997). Filtering errors and repairing linguistic anomalies for spoken dialogue systems. In *Workshop on Spoken Dialogue Systems*, p. 74–81, Madrid (Spain): ACL/EACL.

RUSSELL J. *et al.* (1990). The ARM continuous speech recognition system. In *Proc. of ICASSP'90*, p. 69–72.

SCHWARTZ R. & AUSTIN S. (1991). A comparison of several algorithms for finding multiple (N-best) sentence hypotheses. In *Proc. of ICASSP'91*, volume 1, p. 701–704.

SCHWARTZ R. & CHOW Y.-L. (1990). The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses. In *Proc. of ICASSP'90*.

SHIEBER S. M. (1985). Using restriction to extend parsing algorithms for complex feature-based formalisms. In *Proc. of ACL'85*, p. 145–152.

SIMA'AN K. (1996). Computational complexity of probabilistic disambiguation by means of tree grammars. In *Proceedings of COLING'96*, Copenhagen (Denmark). cmp-lg/9606019.

STOLCKE A. (1995). An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistics*, **21**(2), 165–201.

STOLCKE A. & SEGAL J. (1994). Precise N-gram probabilities from stochastic context-free grammars. In *Proc. of ACL'94*.

STRUT (1996). Speech Training and Recognition Unified Tool. http://tcts.fpms.ac.be/speech/strut.html.

SU K.-Y. *et al.* (1992). A unified framework to incorporate speech and language information in spoken language processing. In *Proc. of ICASSP'92*, volume 1, p. 185–188.

THOMASON M. G. (1986). Syntactic pattern recognition: Stochastic languages. In *Handbook of Pattern Recognition and Image Processing*, chapter 5, p. 119–142. Academic Press.

WRIGHT J. H. *et al.* (1993). A consolidated language model for speech recognition. In *Proc. of Eurospeech'93*, volume 2, p. 977–980.

WRIGHT J. H., JONES G. J. F. & WRIGLEY E. N. (1992). Hybrid grammar-bigram speech recognition system with first order dependence model. In *Proc. of ICASSP'92*, volume 1, p. 169–172.

YOUNG S. (1984). Generating multiple solutions from connected word dp recognition algorithms. In *Proc. of the Institute of Acoustics*, p. 351–354.

ZUE V., GLASS J., GOODINE D., LEUNG H., PHILLIPS M., POLIFRONI J. & SENEFF S. (1991). Integration of speech recognition and natural language processing in the MIT VOYAGER system. In *Proc. of ICASSP'91*, volume 1, p. 713–716.