Chart Parsing of Stochastic Spoken Language Models

Charles Hemphill and Joseph Picone

Texas Instruments
Speech and Image Understanding Laboratory
P.O. Box 655474, MS 238
Dallas, Texas 75265, USA

Abstract

Performance in speech recognition systems has progressed to the point where it is now realistic to begin integrating speech with natural language systems to produce spoken language systems. Two factors have contributed to the advances in speech: statistical modeling of the input signal and language constraints. To produce spoken language systems, then, the grammar formalisms used in natural language systems must incorporate statistical information and efficient parsers for these stochastic language models must be developed. In this paper we outline how chart parsing techniques provide advantages in both computation and accuracy for spoken language systems. We describe a system that models all levels of the spoken language system using stochastic language models and present experimental results.

Introduction

Speech technology has recently made tremendous progress toward speaker-independent large-vocabulary speech recognition (*e.g.*, Lee and Hon, 1988). These sorts of systems rely on Hidden Markov Models (HMMs) of the speech signal and language constraints of the application to achieve good performance. It is important to observe that the language model provides top-down information to reduce the search space during processing. Kubala *et al* (1988) have shown that the perplexity (roughly the number of choices) of the language model correlates with recognition performance. Because of the tendency for Finite State language models to drastically increase in perplexity as coverage increases, it is unlikely that these systems will extend to spoken language systems.

Natural language technology has steadily moved toward the unification grammar paradigm (Shieber, 1986). These formalisms allow various kinds of agreement by generalizing the notion of a grammar symbol to include features and variables. This approach offers advantages in elegantly integrating syntax, semantics and pragmatics while providing domain independence (Mani and Hemphill, 1988). The combination of these constraints during sentence processing can be used to greatly reduce perplexity, but unification grammars have not yet been integrated with speech systems. To do this, the parser must support stochastic grammars (grammars with rule probabilities), comprehend probabilistic hypotheses, and operate frame-synchronously to mesh with the speech system.

Various bottom-up approaches to combining speech and natural language have been tried (*e.g.*, Tomita, 1986). These systems suffer from many problems: top-down information is not available for lower levels, separate grammars must be used for both systems, missing words in the word lattice prove fatal, and probability is usually not available for pruning. Ney (1987) describes a combined top-down and bottom-up approach using the CYK algorithm, but this algorithm has bad average time complexity (the number of input frames cubed).

Our experience with unification grammars has shown us that word hypotheses may be efficiently produced during sentence processing (Hemphill *et al*, 1987). To facilitate integration with speech, we have developed a chart parser specialized to process N levels of stochastic regular grammars. This enables a conceptual shift from the paths and state probabilities found in current systems to symbols that best explain a segment of speech data. We have also constructed a probabilistic version of Earley's algorithm based on observations found in Aho and Peterson (1972). We are currently extending this latter algorithm to support the style of unification grammars found in our earlier work.

We demonstrate the feasibility of a completely symbol-based approach to speech processing by achieving the same performance with layers of stochastic regular grammars as our best FSA-based system (Picone *et al*, 1988). HMMs easily map to both stochastic RGs and FSAs, but the combined top-down and bottom-up parsing algorithm used in this system differs substantially from FSA processing techniques. Most importantly, the parsing algorithm offers computational advantages when hypothesis are needed more than once at the same time frame. This occurs frequently in large grammars and the proper treatment of this condition is essential for processing unification grammars appropriate for spoken language.

Stochastic Spoken Language Models and Parsing

Finite State Automata (FSA) have traditionally been used in speech processing, but they are clearly inappropriate for spoken language systems. In this section, we contrast unification grammars (UGs) with Context-Free grammars (CFGs) and discuss extensions needed for spoken language systems. These extensions involve both the formalism and the parser.

Natural language systems have benefited significantly from the advent of unification grammars. These grammars allow a significant reduction in the number of grammar rules required to represent an application. For example, the following unification grammar rule represents a simple modifying phrase for a relation (R) and attribute (A) in a database:

mod(R) → whose, attr(R, A), is, value(R, A).

This rule allows phrases such as "Find parts *whose color is red.*" If this rule were expanded for a CFG-based system, it would result in a number of rules equal to the number of all attributes for all relations. This problem becomes even larger in a complete grammar. For example, an interface using only 41 unification grammar rules has been constructed for the Force Requirements Expert System (FRESH), but when expanded to CFG rules, over 1000 rules result. Furthermore, processing the unification grammar rules requires less time and space and allows more flexibility in coverage.

The FRESH interface deals primarily with semantic agreement, but UGs also model syntactic and pragmatic phenomena well (Mani and Hemphill, 1988). These three sources of information may be combined in *on-line* parsing to reduce the search space during processing. On-line processing refers to left-to-right processing of the input as it becomes available. This allows the prediction of needed words at each point in processing, making the method appropriate for spoken language systems.

On-line parsing is normally achieved by chart parsing algorithms (Winograd 1983). These algorithms provide mechanisms for efficiently processing grammars by avoiding duplicate work when expanding symbols. Grammars associate symbols with both observations (terminal symbols) and alternate explanations (nonterminal symbols), allowing the elimination of duplicate work in re-hypothesizing the same observations and partial sentence hypotheses.

Spoken language systems, however, deal with probabilistic symbols and both the grammar formalism and the parsing algorithm must accommodate these (Hemphill and Picone, 1989). We have developed a chart parsing algorithm that allows on-line parsing and correctly operates with probabilities. Basically, it is similar to Earley's algorithm (Earley, 1970), augmented with unification (Pereira and Warren, 1983) and probability (Paeseler, 1987), but with a delayed commitment approach to scoring (Aho and Peterson, 1972). This algorithm operates from left to right in a combined bottom-up and top-down fashion, providing terminal hypotheses at each time frame to lower levels and accepting completed hypotheses that began at some time in the past. The algorithm has not yet been fully implemented for UGs, but the following section explores the ramifications of this type of approach.

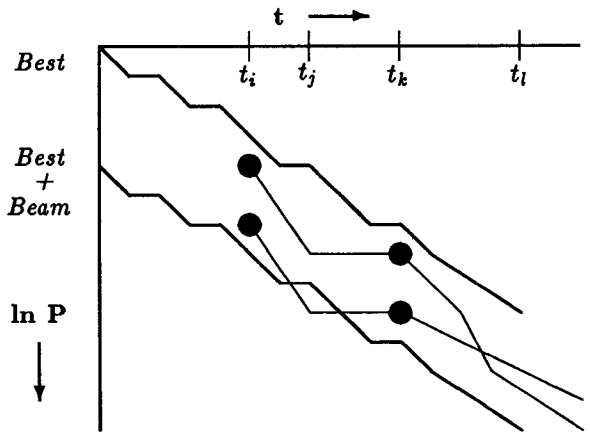

Figure 1: Effect of chart parsing on pruning.

Probabilistic Chart Parsing

In this section, we describe the chart parser as applied to stochastic regular grammars. This provides an indication that the ideas are appropriate for speech processing and calibrates the system with respect to existing FSA-based HMM systems. Three concepts found in speech systems are then applied to the chart parsing framework: pruning, garbage collection, training.

The chart parser can process N layers of stochastic regular grammars. The layers allow expansion of more than one symbol in a rule as in CFGs, but without recursive ability. Specifically, terminal symbols at one layer correspond to start symbols at the next layer. The layers correspond to such things as sentence, word, and phone-model grammars. The top level grammar dictates which hypotheses propagate to lower levels at each frame. Each grammar level in turn propagates hypotheses needed in order to successfully return complete observations. The last level includes a set of grammars that represent an HMM for each acoustic model. Appropriate reference data from this level is compared with the current input speech vector. The processor then incorporates the reference probabilities into the current state of the parse and any completed hypotheses pass to the next higher level as observations. Hypotheses and observations at each level propagate down and up at each frame until all of the speech data may be explained by the formation of a complete sentence.

As a practical consideration, treatment of symbols in this manner interacts with pruning. This is illustrated in Figure 1 for a standard beam search pruning strategy. In stochastic chart parsing, the same symbol may be needed for several different explanations of the speech signal, but only the most likely representative actually becomes hypothesized (t_i). The probability of the completed observation is then used in extending hypotheses awaiting that observation (t_k). This leads to a situation where a lower probability explanation of the symbol may not only survive where it otherwise would have been pruned (t_j), but the subsequent hypotheses using this symbol may actually give the more probable explanation (t_l). Furthermore, since the chart parser expands only the most likely symbol, the less likely hypotheses cause no additional computation during evaluation of the symbol.

Pruning helps reduce the amount of memory required during processing, but not sufficiently. Spoken language systems will contain large vocabularies and require processing of long sentences and sentences in discourse. To address this problem, we have embedded a time-stamp garbage collection scheme into the chart parser. This algorithm reduces memory requirements by an order of magnitude and adds only a small fraction to processing time. Because the algorithm operates on symbols, it applies to stochastic spoken language models in general.

Table 1: Grammars in the experiments.

Task	Level	Rules	Nonterms	Terminals
ULCD	0	14	1	13
	1	1994	1172	517
RM	0	21025	4744	578
	1	1017	578	1017
	2	26111	13017	12001
CKCD	0	83688	8781	21
	1	4026	1919	711

Finally, to become fully general, spoken language systems must support training. The layered grammar approach, although not strictly necessary with CFGs and UGs, allows training above the acoustic model level. For example, phone transition probabilities may be obtained using maximum likelihood training (Fu, 1982). This technique applies to symbols generalized with feature sets and logical variables and will be important as unification grammars find their way into the phonetic level.

Experimental Results

Three recognition experiments have been performed to calibrate the system with respect to an existing FSA system (Picone *et al*, 1988): unknown length continuous digit (ULCD) strings (Doddington, 1989), the 1000-word Resource Management (RM) task (Price *et al*, 1988), and fixed length continuous digit strings with a checksum encoded in the grammar (CKCD). The first two systems use 18-element reference vectors with a 20 msec frame period and pooled covariance. The third experiment uses only 16-element reference vectors. Because of the grammars involved, chart parsing offers no advantage in the first experiment, a small advantage in the second, and maximal advantage in the third. In all experiments, both systems obtain identical results when pruning is not a factor (*i.e.*, errors occur because of the models, not pruning). Table 1 indicates the relative size of the grammars.

The ULCD experiment consists of two levels of grammars. The first grammar allows zero or more occurrences of **oh**, **zero** - **nine**, a silence model, and a null-speech model. The second grammar contains the HMMs for each of these. Although multiple hypotheses of the same symbol at the same time occur in the HMM grammar, the hypotheses at this level correspond to reference vectors of one frame in duration and the FSA-based system evaluates these only once. Additionally, the FSA system evaluates hypotheses once per source node, making the one node sentence automata especially favorable for this system.

The RM experiment consists of three levels of grammars. The first contains a canonicalized grammar representing the various sentence patterns desired for the task, the second maps word types (*e.g.*, `'<ship-name>'`) to words, and the third defines the HMM models for the words. Only the first level in this experiment benefits from the chart parsing approach.

The CKCD experiment consists of two levels of grammars. The first grammar allows a fixed number of the digits **zero** - **nine** (with a male and female model for each) and a silence model. This grammar implements a checksum function with the last two digits serving as the checksum. The bottom level again contains HMM word models. The large branching factor in the sentence grammar creates a situation where chart parsing can help tremendously.

Table 2 summarizes the ratio of chart parsing time to FSA processing time for each experiment. The first experiment indicates that the overhead of chart parsing is at most a factor of two. The second experiment shows that even for small perplexity grammars (perplexity 9 for RM), the benefits of chart parsing begin to compensate for the overhead in the lower levels. The third experiment was conducted for three different normalized pruning values: 0.2 and 0.3, 0.35. The lower value greatly reduces the number of evaluated hypotheses and therefore favors the FSA processor. However, at this pruning level, the FSA processor

Table 2: Time ratios for RG and FSA processors.

Task	RG/FSA time
ULCD	1.9
RM	1.1
CKCD (0.20)	1.2†
CKCD (0.30)	0.7
CKCD (0.35)	0.6‡

† sub-optimal results
‡ memory overflows in FSA system

Table 3: Effects of pruning on accuracy for CKCD.

Beam	FSA Sent Corr	RG Sent Corr
0.20	90%	94%
0.30	98%	98%
0.35	96%‡	98%

Table 4: Garbage collection in chart parsing.

Task	Ave. Peak	Ave. Total
ULCD	2023	39074
RM	8461	120227
CKCD (0.20)	5692	254879
CKCD (0.30)	22241	668659
CKCD (0.35)	43576	1013254

correctly recognizes 90% of the sentences while the chart parsing technique recognizes 94% (Table 3). This provides evidence for the phenomena illustrated in Figure 1. At 0.3, both methods recognize 98% of the sentences, with a time advantage for the chart parsing approach. At 0.35, the time advantage increases for the chart parsing approach and, additionally, the FSA method exhausts memory when processing some sentences.

Table 4 summarizes the effect of garbage collection on space requirements for the chart parser in each of the experiments. The average peak memory column indicates the peak memory requirements over all frames for each sentence. The average total memory column indicates the amount of memory needed for each sentence without garbage collection. Note that the memory savings is in addition to pruning. As can be seen, there is a substantial savings, and in fact, the last two experiments could not be performed without this algorithm. The FSA system also uses this same method, but pursuing each hypothesis separately results in memory overflow in the last experiment.

Conclusions

We have demonstrated that chart parsing techniques successfully apply to stochastic language models. In the process, we have demonstrated an approach to speech recognition in which the entire recognition process, including acoustic processing, consists of a hierarchy of grammars. A shift from automata to grammars allows efficient processing of complex language models by hypothesizing symbols once per frame, no matter how many times they are needed. We have further shown that pruning, garbage collection, and training algorithms may be successfully incorporated into a chart parser for stochastic language models.

The layers of regular grammars used in the experiments are completely compatible with a unification grammar framework. We believe that the further development of efficient parsers for stochastic unification grammars is a required step toward spoken language systems. Future work will focus on development of these parsers, grammars that make effective use of contextual information, and more robust distance measures that comprehend this information.

Acknowledgments

We would like to thank George Doddington for much assistance with the FSA system, for invaluable insight on pruning and garbage collection, for sharing his expertise on HMMs, and for providing the digit experiments. We would also like to thank Jack Godfrey for sharing his knowledge on phonetics for many experiments.

References

[1] A. V. Aho and T. G. Peterson, "A Minimum Distance Error-Correcting Parser for Context-Free Languages," *SIAM Journal on Computing*, Vol. 1, No. 4, Dec. 1972.

[2] G. R. Doddington, "Phonetically Sensitive Discriminants for Improved Speech Recognition," *Proc. ICASSP*, Glasgow, Scotland, 1989.

[3] J. Earley, "An Efficient Context-Free Parsing Algorithm," *CACM*, Vol. 13, No. 2, 1970, pp. 94-102.

[4] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, 1982.

[5] C. T. Hemphill, I. Mani, and S. L. Bossie, "A Combined Free-Form and Menu-Mode Natural Language Interface," *Abridged Proc. of the 2nd Intl. Conf. on Human-Computer Interaction*, Honolulu, Hawaii, 1987.

[6] C. T. Hemphill and J. Picone, "Speech Recognition in a Unification Grammar Framework," *Proc. ICASSP*, Glasgow, Scotland, May 1989.

[7] F. Kubala, Y. Chow, A. Derr, M. Feng, O. Kimball, J. Makhoul, P. Price, J. Rohlicek, S. Roucos, R. Schwartz, and J. Vandegrift, "Continuous Speech Recognition Results of the BYBLOS System on the DARPA 1000-Word Resource Management Database," *Proc. ICASSP*, 1988, pp. 291-294.

[8] K. Lee and H. Hon, "Large-Vocabulary Speaker-Independent Continuous Speech Recognition using HMM," *Proc. ICASSP*, 1988, pp. 123-126.

[9] I. Mani and C. T. Hemphill, "A Natural Language Interface for Knowledge Based Systems," *Proc. of Third Annual User-System Interface Conf.*, Austin, TX, Feb., 1988.

[10] H. Ney, "Dynamic Programming Speech Recognition using a Context-Free Grammar," *Proc. ICASSP*, 1987, pp 3.2.1-4.

[11] A. Paeseler, "Modification of Earley's Algorithm for Speech Recognition," *Proc. of NATO ASI*, Bad Windsheim, 1987.

[12] F. C. N. Pereira, and D. H. D. Warren, "Parsing as Deduction," *Proc. of ACL*, Boston, MA, June, 1983.

[13] J. Picone, G.R. Doddington, and J.J. Godfrey, "A Layered Grammar Approach To Speaker Independent Speech Recognition," presented at the 1988 Speech Recognition Workshop, Harriman, NY, June 1988.

[14] P. Price, W. Fisher, J. Bernstein, and D. Pallett, "The DARPA 1000-Word Resource Management Database for Continuous Speech Recognition," *Proc. ICASSP*, NY, NY, April, 1988.

[15] S. M. Shieber, *An Introduction to Unification-Based Approaches to Grammar,* CSLI Lecture Notes, No. 4, 1986.

[16] Tomita, M., "An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition," *Proc. ICASSP*, Tokyo, 1986, pp. 1569-1572.

[17] T. Winograd, *Language as a Cognitive Process,* Addison-Wesley, 1983.