USING PROBABILISTIC SHIFT-REDUCE PARSING IN SPEECH RECOGNITION SYSTEMS[1]

David Goddeau

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139, U.S.A.

ABSTRACT

This paper describes the application of probabilistic shift-reduce parsing to the problem of continuous speech recognition. In previous work, a probabilistic LR parser was applied to the task of speech understanding in the MIT VOYAGER system. The performance metric for that task was based on the fraction of utterances for which correct semantics were produced. For the speech recognition task, word and sentence recognition accuracy are the important performance criteria. In this work, the probabilistic LR language model is extended with robust parsing techniques to achieve 100% coverage, even though the underlying grammar has incomplete coverage. The resulting model provides additional constraints over a word bigram, but retains the trainability and efficiency of the simpler model.

Recognition experiments were performed in the DARPA ATIS task domain using a version of the MIT SUMMIT speech recognizer with context-independent phoneme models. Integrating the new language model into the SUMMIT N-best search algorithm decreased the word error rate from 24.1% to 21.5% and the sentence error rate from 72.9% to 65.2%, when compared to a word bigram model.

INTRODUCTION

Language models can improve the performance of speech understanding systems by providing constraints on the space of word strings that needs to be considered. Many language models provide these constraints by restricting the space of word strings to a subset of "grammatical" sentences. Probabilistic models, on the other hand, provide constraints by estimating the probability distribution for the space of word strings. This paper describes a trainable grammar-based probabilistic language model incorporating a deterministic shift-reduce parser, in particular a modified LR parser. The motivation for using a grammar-based model is two-fold: to provide long distance linguistic constraints, such as number agreement and verb-object semantic restrictions, and to produce a trainable model which more effectively uses the available data.

The LR parsing algorithm [1] is a deterministic table-driven shift-reduce parser applicable to a subset of context-free grammars. It is an appealing technology for use in spoken language system because it is easy to implement, analytically tractable, very fast, and detects syntactic errors as early as possible. A small number of speech recognition systems now use some form of LR parser as their language model, either as a filter on hypotheses [2][3], or to produce a sentence score [4][5].

[1]This research was supported by DARPA under Contract N00014-89-J-1332, monitored through the Office of Naval Research.

An earlier paper by Goddeau and Zue [6] presented an LR parser-based probabilistic language model which was applied to the task of speech understanding in the MIT VOYAGER system. In that system, the LR parser was used both to filter out ungrammatical sentences and to compute sentence probabilities. The probabilistic LR language model was able to substantially increase the VOYAGER system's performance, as measured by the fraction of utterances for which correct semantics were produced minus the fraction which produced incorrect semantics. However, the fact that this language model rejects ungrammatical sentences limits its usefulness for speech recognition tasks, since in any realistic domain it is impossible to produce a grammar which anticipates 100% of the linguistic constructions that users will employ.

This paper describes the application of probabilistic LR parsing to the speech recognition task. The next section introduces a generalization of the probabilistic LR language model used in the VOYAGER domain and describes a particular implementation of that model for use in speech recognition. The modifications made to the basic LR parser to guarantee 100% coverage are discussed in the next section along with features added to smooth the next-word probability estimates. Finally, recognition experiments are presented evaluating the benefits and cost of integrating the language model into the SUMMIT speech recognition system.

PROBABILISTIC LANGUAGE MODEL

A probabilistic language model assigns to each sentence $S = w_0...w_n$, a probability $P(S)$. This probability can be written as the product of the probabilities of each word conditioned on the previous words,

$$P(w_0...w_n) = \prod P(w_i|w_0...w_{i-1}).$$

In the VOYAGER PLR model [6], these conditional word probabilities were approximated as

$$P(w_i|w_0...w_{i-1}) \approx P(w_i|Q_i)$$

where Q_i is the LR parser state which resulted after processing the string $w_0...w_{i-1}$.

This is one example of a general language model architecture pictured in Figure 1. In the general paradigm, a deterministic shift-reduce parser is run on the partial sentence $w_0...w_{i-1}$. As part of its operation, the parser stores both grammar symbols and parse states alternately on the stack. A *Shift* action pushes the input word on the stack followed by a parser state. A *Reduce* action pops one or more symbols and states from the stack, then pushes a non-terminal symbol and the next parser state. When the input string, $w_0...w_{i-1}$, has been consumed by the parser, the resulting parse stack configuration, $S_0...S_M$ represents a condensation of the information in the partial sentence. This string of stack symbols is then used to construct a conditional model for the next-word probabilities.

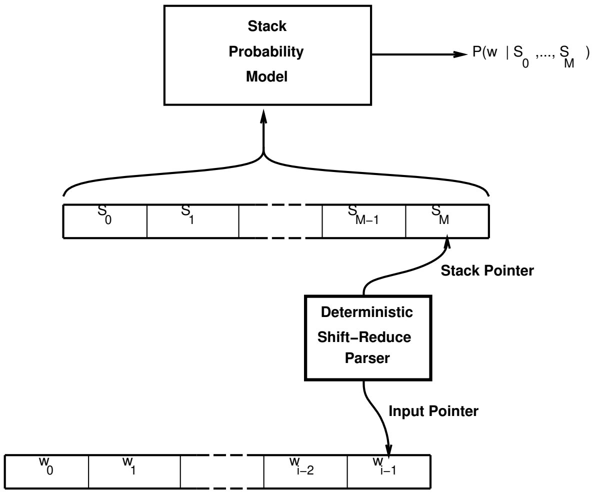

Figure 1: Shift-reduce parser based language model

This model combines the trainable parametric approach in the computational model for $P(w|S_0...S_M)$ with the application of linguistic knowledge contained in the parser. It also generalizes the word n-gram and part-of-speech n-gram models in that a version of this model can be constructed which precisely emulates either of them.

Implementation for Speech Recognition

The current system is also an instance of this general model specialized to provide constraints for speech recognition. In this implementation, the shift-reduce parser is realized with a modified version of the LR algorithm. The stack-based probability model uses only the grammar symbols stored on the stack, the state entries are ignored[2]. When no further parser moves are possible from the available input, the string of grammar symbols found on the stack, $G_0...G_M$, is truncated to length $k+1$ and used to estimate the conditional next-word probabilities;

$$P(w|w_0...w_{i-1}) \approx P(w|G_{N-k}...G_N).$$

The estimates for $P(w|G_{N-k}...G_N)$ are developed by training the system on a corpus of domain specific sentences, and accumulating the counts $C(G_{N-k}...G_N, w)$ and $C(G_{N-k}...G_N)$. An interpolation technique is then used to smooth the estimates.

PARSER MODIFICATIONS

This section describes two modifications to the standard LR parser used in this system. The first augments the parse table produced by the input grammar to achieve 100% coverage. The second helps to smooth the probability estimates by operating on the parse stack.

[2]This is in contrast to the VOYAGER PLR system which used the current parser state, found at the top of the stack.

Coverage Extension

In constructing a grammar for a speech recognition application, there is a tradeoff between the coverage of the grammar and its constraining power. In practice, a grammar will only cover some subset of the input sentences, however, rejecting entire utterances is inappropriate in a recognition task. It is desirable, therefore, to have a language model which can deal with ungrammatical sentences while retaining as much constraining power as possible. The probability model described above depends on the parser to push appropriate grammar symbols onto the stack as it processes the input. Extending the model to guarantee 100% coverage is accomplished by modifying the basic LR parser to recover from parsing errors and continue processing its input.

Parsing errors occur when a word is encountered in the input string which cannot lead to any grammatical sentence. The standard LR algorithm halts further processing and returns, indicating that the sentence was ungrammatical. Since this is not appropriate for the current task, the problem is to determine how to continue processing. The most straightforward approach is to have the parser enter an error state which shifts the input onto the stack. The parser would remain in the error state while processing the rest of the input. This approach makes the probability model default to a word n-gram once a parsing error has occurred.

The scheme actually used in the system is slightly more complex. It specifies an error recovery action for each word in the lexicon. When a word is encountered during processing which leads to a parsing error, the recovery action for that word is invoked, the configuration of the parser is modified and normal processing continues. The recovery actions, like the parse table, are generated automatically from the collection of LR sets of items. The algorithm essentially looks for the largest phrasal unit which the unexpected word could begin, and generates an error action which starts the parsing of that unit. This scheme has the probability model degrade to a phrase n-gram model when the sentence grammar fails.

Smoothing

Another modification to the basic LR parser helps to smooth the next-word probability distributions. The basic LR parser operates by shifting the input words onto the stack until it is able to reduce the symbols on top of the stack with some grammar rule. The choice of action, shift or reduce, is specified by a parse table indexed by the current state and current input word. When the parser processes an incomplete sentence, $w_0...w_{i-1}$, to provide the distribution for w_i, the last move the parser is able to make is to shift w_{i-1} into the stack. Since there are no more words in the input, the parser is unable to perform the table lookup to get the next move. As a consequence, the stack contents used to estimate the next-word probabilities has w_{i-1} at the top of the stack.

To provide some additional smoothing for the next-word probabilities, it is useful to be able to perform grammar reductions after the input string has been exhausted. One way to accomplish this would be to add a special End-of-Input word, and specify the desired reductions by writing grammar rules. The alternative used here allows non-terminals to be specially tagged as reduction targets and generates the needed reduce actions automatically. In practice, use of this feature provides a significant reduction in model perplexity.

INTEGRATION WITH SPEECH RECOGNITION

This work investigates two methods for integrating the language model with the acoustic scoring component of a speech recognition system. The first approach is to use the PLR language model to re-score the N-best sentences produced by a recognizer incorporating a first-order language model, such as a bigram. The second approach is to combine the language model with speech recognition using an A^*-like stack search algorithm.

The stack search algorithm maintains a priority queue of partial sentence hypotheses. At each step, the highest scoring hypothesis is popped from the queue. If it covers the entire acoustic input, it is added to the N-best list. Otherwise, it is sent to the parser, which scans the partial sentence, and computes the probabilities for each of the next-word candidates. The current hypothesis is extended by each word, scored against the acoustic input, and the lengthened hypotheses are placed on the priority queue. This process can be continued to produce the desired number of transcription hypotheses.

During the search, incomplete sentences are scored as in an A^* search, using the sum of a *forward score*, the computed score for the section of the utterance already seen by the recognizer, and a *backward score*, an estimate of the score for the rest of the utterance. Unlike an A^* search, however, these backward estimates are not necessarily upper bounds on the score. The backward estimates are computed using a Viterbi algorithm incorporating the MIT ATIS bigram model which has a perplexity of 20. The forward score for a hypothesis is computed using a linear combination of the acoustic score and the logarithm of the language model probability, which is normalized by subtracting a constant times the hypothesis length.

The two integration schemes are compared experimentally since it cannot be predicted from theory which one is more efficient. The N-best re-scoring approach requires the production of more sentence hypotheses. On the other hand, fully integrating the higher-level into the N-best search destroys its admissibility, which could lead to an inefficient search.

EXPERIMENTS

Recognition experiments were performed using data from the DARPA ATIS domain [7]. This is a spontaneous-speech interactive task domain which centers around queries about airline flights between several US cities. The utterances include simple queries, fragments, comments, and multiple-sentence requests. The corpus contains mispronunciations, false starts, filled and unfilled pauses, among other spontaneous speech phenomena.

The language model was integrated with a version of the MIT ATIS system [8] which has a lexicon of 841 words and uses context-*independent* Gaussian-mixture phoneme models[3]. The system uses a word bigram as its default language model, with serves as the baseline for performance comparison.

All language models used in experiments were trained on the same set of 9711 training sentences. Evaluation was performed on the DARPA October '91 Test set [9] and on a 1595 sentence development set covering all collection sites.

[3]The results presented here depend only on relative performance, and the context-independent system was used for convenience. Lower error rates have been achieved with the context dependent system (see [8]).

Sentence Grammar

The ATIS sentence grammar used in these experiments consists of 1007 context-free rules. The terminal symbols of the grammar are the words of the recognizer lexicon. The grammar was constructed by hand using a set of the training sentence as a guide and captures both syntactic and semantic category information. The goals for the sentence grammar were to maximize coverage on the training set and to produce parses which could serve as a basis for semantic interpretation. The reduction of model perplexity was not considered explicitly during grammar creation.

The coverage of the basic grammar on the 9711 training sentences is 62%. The modified LR parse table contains 1614 states. The stack history was truncated to length 2 for building the probability model and the interpolated stack trigram uses just over 35000 stored parameters. The trained sentence grammar has a test-set perplexity of 16 on the October '91 Test set.

Table 1 gives recognition results for the ATIS sentence grammar on a 371 sentence development set and the October '91 Test set. These results were obtained by incorporating the probabilistic LR language model into the N-best search as described above. Since its search is not admissible, the PLR system was run until 8 sentence hypotheses were produced or a threshold number of hypothesis extensions was reached. The highest scoring sentence hypothesis was chosen as the answer.

The data show lower error rates for both word and sentence recognition for the PLR model compared to the bigram model. The magnitude of the improvement is greater for sentence recognition than for word recognition, however, the percentage decrease in error rate is about the same.

Data Set	Model	Sent. Err.	Word Err.
Development	Bigram	62.8	20.5
	PLR	55.0	17.0
Oct. '91 Test	Bigram	72.9	24.1
	PLR	65.2	21.5

Table 1: Recognition results on Development and Oct. '91 Test sets for PLR Sentence Grammar

Table 2 compares the integrated system with a scheme that uses the same language model to re-order the 50 best hypotheses produced by the MIT ATIS system with only the bigram model. The error rates are comparable to the integrated system's, indicating that the non-admissible search is successful in finding the correct transcription, a conclusion also reached in the VOYAGER system.

System	Sent. Err.	Word Err.	Av. Ext.
Bigram (N=1)	72.9	24.1	13
Re-scored 50	65.5	21.1	542
Integrated	65.2	21.5	161

Table 2: Comparison of integration methods for ATIS sentence grammar on October '91 Test set

The last column gives the average number of hypothesis extensions needed to produce the N-best sentences for each utterance. Extending hypotheses during the search involves scoring next-word candidates against the acoustic signal and accounts for the majority of the processing time for an utterance[4]. While very

[4]The amount of this processing devoted to parsing and computing next-word probabilities is a few percent of the total.

few extensions are required to produce the initial bigram output (first row in table), producing the 50 top scoring hypotheses requires significant additional computation. The fully integrating the language model into the search reduces the number of hypotheses extended by a factor of 3, representing a substantial reduction in recognition time.

Phrase Grammar

One of the features of the parse-table generator described above is that it allows the incorporation of different amounts of linguistic knowledge. A complete grammar covering the application domain is not necessary to increase performance. Figure 2 plots the test-set perplexity of the language model, measured on the October 1991 Test set (solid line) and development set (broken line), as the number of rules in the grammar is increased.

With no rules, the parser simply pushes its input string onto the stack; the resulting probability model is equivalent to a word trigram. The first 270 rules all have right-hand sides of length one and the PLR model emulates a class trigram. The remainder of the rules reduce multi-word sequences to single non-terminals. The graph shows a steady decrease in perplexity as the number of rules is increased. However, further experiments are needed to determine whether the decrease in perplexity produces a corresponding improvement in recognition performance.

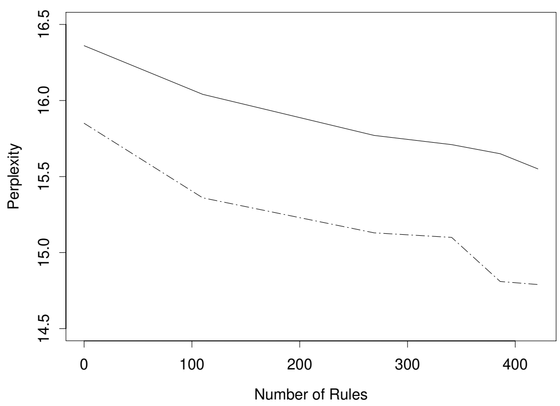

Figure 2: Perplexity vs. Number of rules

SUMMARY AND FUTURE WORK

This paper describes a probabilistic language model for speech recognition based on a shift-reduce parser. The model can effectively incorporate grammatical knowledge, yet retains the desirable properties of purely data-driven models: trainability, computational efficiency, and complete coverage. Recognition experiments demonstrate that incorporating grammatical knowledge can improve recognition performance over a word n-gram model, reducing both word and sentence error rates. Furthermore, integrating the grammar-based model into the search is significantly more efficient than re-scoring the N-best bigram output.

There are several areas which may further increase the performance of the PLR model. One issue which needs to be explored concerns the backward score estimates used in the N-best search. At present, these are computed by a Viterbi search incorporating a bigram language model. While this yields satisfactory performance in most cases, other alternatives are worth exploring. One possibility is to use a first-order model derived from the PLR parameters in place of the bigram. Another alternative is to explore stack search schemes which do not require scanning the entire utterance to produce these estimates. This would reduce overall computation and improve pipelining of the recognition process. An interesting approach to this problem is described in [10].

A second area of future work involves further improvements to the parsing model. One appealing idea is to incorporate a feature-based formalism into the stack-based prediction scheme. It may be that certain syntactic properties of the next word, such as gender, number, or verb form, can be predicted based on a vector of features constructed by the parser. The underlying root or "semantic content" of the next word may be best predicted from a different condensation of the input history string. A language model which could separate these aspects would perhaps be more efficient than one which must estimate each combination.

A final area of future work involves automatic grammar learning. This construction of grammars by hand is tedious and difficult. It is not always the case that one's intuitive knowledge of a language, or one's favorite linguistic theory, provides the proper insights to improve recognition performance in a particular domain. The fact that the PLR model can be implemented very efficiently makes it an interesting platform for implementing iterative or hill-climbing learning algorithms.

REFERENCES

[1] Aho, A. and Ullman, J., *The Theory of Compiling, Translation, and Compiling* Prentice-Hall, Englewood Cliffs, N.J., 1973.

[2] Kita, K., Kawabata, T. and Saito, H., "HMM Continuous Speech Recognition Using Predictive LR Parsing", *Proc. ICASSP 89*, pp. 703-706, 1989.

[3] Kita, K. and Ward W., "Incorporating LR parsing into SPHINX", *Proc. ICASSP 91*, pp. 269-272, 1991.

[4] Wright, J.H., Jones, G.J.F., and Wrigley, E.N., "Hybrid Grammar-Bigram Speech REcognition System With First-Order Dependence Model", *Proc. ICASSP 92*, I169-I171, March 1991.

[5] Su, K.-Y., Chiang, T.-H., and Lin, Y.-C., "A Unified Framework to Incorporate Speech and Language Information in Spoken Language Systems", *Proc. ICASSP 92*, I185-I187, March 1991.

[6] Goddeau, D. and Zue, V., "Integrating Probabilistic LR Parsing into Speech Understand Systems", *Proc. ICASSP 92*, pp. I181-I184, March, 1992.

[7] Price, P., "Evaluation of Spoken Language Systems: The ATIS Domain", *Proc. DARPA Speech and Natural Language Workshop*, pp. 91-95, June, 1990.

[8] Zue, V., Glass, J., Goddeau, D., Goodine, D., Hirschman, L. Phillips, M., Polifroni, J., and Seneff, S., "The MIT ATIS System: February 1992 Progress Report", *Proc. DARPA Speech and Natural Language Workshop*, February, 1992.

[9] MADCOW, "Multi-Site Data Collection for a Spoken Language Corpus", *Proc. DARPA Speech and Natural Language Workshop*, February, 1992.

[10] Paul, D., "An Efficient A* Stack Decoder Algorithm for Continuous Speech Recognition with a Stochastic Language Model", *Proc. ICASSP 92*, pp. I25-I28, March, 1992.