INTEGRATING PROBABILISTIC LR PARSING INTO SPEECH UNDERSTANDING SYSTEMS[1]

David Goddeau and Victor Zue

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 LR parsing to the problem of continuous speech understanding. It introduces a probabilistic model for LR parsing, discusses the integration of the parser into a speech understanding system, and presents recognition results. The goal of speech understanding is to process continuous speech input from users and provide appropriate responses. This requires a speech recognition system with a high sentence recognition rate, and a low sentence error rate (since rejection is allowed, these are not identical). The paper demonstrates that probabilistic LR parsing can help meet these requirements by applying linguistic and statistical constraints to speech recognition, and by providing effective rejection criteria. In experiments using the MIT VOYAGER spontaneous speech corpus, the use of probabilistic LR parser improved the percentage of utterances for which correct semantics were produced from 23% (using a perplexity 72 word-pair grammar) to 58%. System performance, as measured by the metric, $100 * (N_{correct} - N_{error})/N_{total}$, was 44.5 at a 32.7% rejection rate.

INTRODUCTION

Language models are an important component of speech understanding systems, providing both syntactic and semantic constraints for the speech recognition process, and semantic interpretation for the knowledge-based back-end of the system. The focus of the work described here is applying LR parsing technology in a speech understanding system to provide constraints, addressing the goal of improving sentence recognition accuracy.

The LR algorithm [2] is a table-driven shift-reduce left-to-right parser for context-free grammars. At each step, the action of the parser is determined by the current parser state and the current input word. It is an appealing algorithm for applying language constraints to speech recognition because it is simple, analytically tractable, very fast, and can prune ungrammatical hypotheses early. LR parsing has been used as a phoneme predictor in the ATR HMM-LR system [3],[4], and as the natural language component in the SPHINX-LR [5] speech recognition system. This work differs from these in several aspects, including the use of a probabilistic model based on the LR parser and the application of LR parsing to a spontaneous speech understanding task.

One of the main problems in integrating a probabilistic language model into a speech recognition system is in using the language probabilities effectively in the search phase of the recognition process. Models which impose only local constraints such as unigram, word-pair or bigram models, are easily integrated into a Viterbi search procedure. The introduction of non-local constraints, however, changes the structure of the search space, making dynamic programming techniques much more difficult to apply. This paper compares two search strategies for performance and efficiency.

The following section formulates a probabilistic language model based on the LR parsing algorithm. The model is easy to train, computationally efficient, and has a significantly lower perplexity than the non-probabilistic model (measured on test data). The integration of the parser into the speech recognition process is then discussed. Finally, results are presented for sentence recognition performance.

PROBABILISTIC LR PARSING

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,

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

If Q_j represents a state of the parser, and $P(Q_j|w_0...w_{i-1})$ is the probability that the parser is in state Q_j having just parsed the substring $w_0...w_{i-1}$ (without making any moves based on the value of w_i), then the conditional word probability can be re-written as

$$P(w_i|w_0...w_{i-1}) = \sum_j P(w_i|Q_j \wedge w_0...w_{i-1})P(Q_j|w_0...w_{i-1})$$

Making the assumption that the parser state captures much of the information in the substring $w_0...w_{i-1}$ relevant to the conditional probabilities, this can be approximated by

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

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

The set of Q_j for which $P(Q_j|w_0...w_{i-1})$ is non-zero is determined by the grammar. In particular, if the grammar is deterministic, then $P(Q_j|w_0...w_{i-1}) = 1$, for some $j = j_i$, and

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

The probabilities $P(w_i|Q_j)$ can be estimated from a corpus of training sentences using the ratio of the number of times w_i is the next word when the parser is in state Q_j to the number of times the parser is in state Q_j.

The model presented here differs from the standard Probabilistic CFG (PCFG), which attaches a probability to each production rule in the grammar. The PCFG model can be adapted to LR parsing [6], but was not used in this work due to its greater complexity and its inability to conveniently capture certain types of probabilistic relationships[2].

INTEGRATION WITH SPEECH RECOGNITION

One of the objects of this work was to fully integrate the LR parser into the speech understanding system so that the linguistic constraints are applied as part of the recognition process rather than as a filtering process on complete recognizer hypotheses. This accomplished by using a modified A^* search procedure (see [7]).

The system maintains a priority queue of partial sentence hypotheses. At each step, the highest scoring hypothesis is popped from the queue and sent to the parser, which scans the partial sentence, and produces a list of next-word candidates with associated word probabilities. A collection of new hypotheses are created by extending the selected partial sentence with each of the proposed words. Each of these extensions is scored against the acoustic input, and placed on the priority queue. The selected hypothesis is output if it is accepted by the parser as a complete sentence, and spans all of the acoustic input. This process can be continued to output the desired number of sentence hypotheses.

The partial 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 sentence. These backward estimates are computed using a Viterbi algorithm constrained by a perplexity 72 word-pair grammar, and represent only the acoustic contribution to the score.

This method of integrating LR parsing differs from that used in the SPHINX-LR system [5]. In that system, recognition proceeds via a beam search, with the parser functioning as a filter on proposed word transitions. As a result, that system can fail early in the recognition process if none of the proposed transitions pass the parser. Since the LR parser is used here as a predictor, all of the hypotheses in the queue represent valid sentence prefixes, and an A^* search will find the best-scoring grammatical sentence.

One of the main problems in integrating this language model is in effectively using the PLR probabilities in the search. Initial attempts in this work to implement an admissible A^* search by including upper bounds on the PLR probabilities in the backward estimates were unpromising; the resulting search was very inefficient because the upper bounds were too loose. This paper examines strategies which use only acoustic information in the backward estimates.

Two different search strategies, based on different forward score computations, were implemented. In the first strategy, only the acoustic-recognition score was used for the forward scores, during the search. The sentence probabilities were incorporated later by re-scoring the top N sentence candidates. In the second strategy, a combination of the acoustic score and the word-string probability was used during the search. This strategy has the advantage of eliminating the need to produce several sentence hypotheses. However, since the probabilities are not included in the backward estimate, it is no longer an upper bound on the combined score, and the search procedure is not admissible. Some recognition experiments, described below, were done to determine if this non-admissibility significantly reduced overall recognition performance.

Score Combination

Sentences are scored using a linear combination of the acoustic score, $AC(S)$, and the logarithm of the sentence probability, $P(S)$. The sentence log-probability is normalized by subtracting a constant γ, equal to the logarithm of the word perplexity, times N, the number of words in S (treating End-of-Sentence as a word).

$$Score(S) \;=\; AC(S) + \beta(\log P(S) - N\gamma)$$

Since the acoustic score for a hypothesis is the sum of the scores for each word, this expression can be rewritten as a sum over the combined word scores, allowing the scoring of partial hypotheses to be consistent with the scoring of complete sentences.

$$Score(S) \;=\; \sum_i (AC(w_i) + \beta(\log P(w_i|w_0, .., w_{i-1}) - \gamma))$$

Rejection

Rejection is important in any application in which an incorrect responses has a higher penalty than a "No Answer" response. The relative penalties of these responses determine the appropriate aggressiveness of the rejection criteria. In the experiments describe here, the rejection threshold was set to optimize the DARPA metric, $(N_{correct} - N_{error})/N_{total}$ [8]. This assigns penalties of $1, -1$, and 0 to correct, incorrect and rejection responses, respectively.

[2]For example, given a production A $\rightarrow$ BC, where B and C are preterminals, the PCFG model is unable to capture bigram probabilities between the words in these classes.

Two different rejection mechanisms were used in this system. The first was a limit on the number of hypotheses to be extended during the search. This was used to reduce computation time and was set at a value large enough to avoid rejecting any correct sentences. The second criterion (used in addition to the first) was a threshold on the sentence score (combined acoustic and parse score), which was set to optimize the DARPA metric.

EXPERIMENTS

Recognition experiments were performed using data from the VOYAGER spontaneous speech corpus [9]. This corpus consists of a 70 speaker training set, a 20 speaker development set, and a 10 speaker test set. Evaluation experiments were performed on a 12 speaker, 569 sentence subset of the development set and the 10 speaker, 497 sentence test set.

A context-free grammar for the VOYAGER domain was created by hand from a set of 2400 training sentences. The grammar contains 455 rules, and the corresponding LR parse table contains 788 states. The coverage of the grammar on this training set is 68%, with a perplexity (without probabilities) of 11. When the probabilities were trained on this training set, the perplexity dropped to 4. The trained grammar has a perplexity of 4 and 66% coverage on the development-test sentences. A subset of the training utterances was used to determine the value of β in the scoring formula and to set a rejection threshold. The speech recognition component was derived from the SUMMIT Speech Recognition System [10].

Since the focus of this work is on recognition for speech understanding systems, the definition of sentence "correctness" was based on semantic considerations. The recognizer's transcriptions were classified as correct if they were word for word identical to the orthography, or produced the same semantic expression as the orthography[3]. The semantic interpretations were computed using the TINA system [11].

Search Strategies

The first experiment compared the perplexity 72 word-pair grammar and two strategies for integrating PLR with speech recognition. In the first strategy, the LR parser applied only linguistic constraints; the parse probabilities were not incorporated during the A^* search. The 5 best sentences produced by the search were then re-scored incorporating the parse probabilities. In the second strategy, PLR search, the parse probabilities are used in the A^* search as part of the forward score. The backward score is still computed based on acoustics only. In both cases, the number of hypotheses examined in the search was limited to 2000, after which the utterance was rejected if no sentence was found. The recognition results for recognition with the word-pair grammar are presented for comparison.

The results on the development-test, shown in Table 1, show that the probabilistic LR parser provides a significant improvement in performance over the word-pair grammar, (as would be expected from the difference in perplexities). The word-pair system always produced an answer, while the LR systems rejected, based on search limits, a significant fraction of the utterances. However, the word-pair system produced far fewer correct answers. Comparing the performance of the two LR search strategies, it seems that using the probabilities directly in the search does not appear to degrade performance (compared to the N-best re-scoring), even though this search algorithm is not admissible. This was also found to be the case on the test set, and for recognition on a subset of the training data.

	Correct	Error	Rejected
Word-pair	23.2	76.8	0.0
Re-scoring	55.2	17.6	27.2
PLR search	58.2	20.5	21.3

Table 1: Recognition results on development-test set of search strategies for combining PLR and SR (Values are percent of sentences)

Rejection Criteria

The table below compares the system performance, with and without a rejection threshold on the combined score. This is used in addition to a threshold on the number of hypotheses extended. Results are given for both the test set and the development-test set, and show an improvement of about 15% from using the combined acoustic and language score as a rejection criterion. This improvement is obtained with a relatively small loss of correct sentences.

Development set	Score	Correct	Error	Rejected
Without PLR Rejection	37.7	58.2	20.5	21.3
With PLR Rejection	44.5	55.9	11.4	32.7
Test set	Score	Correct	Error	Rejected
Without PLR Rejection	36.7	57.4	20.7	21.9
With PLR Rejection	42.0	54.1	12.1	33.8

Table 2: Performance of PLR system on development and test sets with and without score-based rejection. *Score* represents the DAPRA metric expressed as a percentage.

To examine the source of the errors, results were tabulated for two subsets of the development set: "grammatical", those utterances with orthographies which are accepted by the parser, and "ungrammatical", those rejected by the parser[4]. The "ungrammatical" sentences include sentences with false starts, out-of-domain sentences, syntactic constructions not covered by the parser, and genuinely ill-formed utterances. The results show that the system is generally doing the appropriate thing; over 80% of the "grammatical" sentences produce correct semantics and over 75% of the "ungrammatical" sentences are rejected. Given this behavior, an

[3]In practice, about 85% of those scored "correct" were word for word identical with the orthography.

[4]The few "correct" answers found for ungrammatical orthographies are utterances for which the recognizer found a grammatical transcription which was semantically equivalent

obvious way to further improve performance would be to extend the grammar to cover more syntactic constructions and to handle some out-of-domain queries.

Subset	% sents	Correct	Error	Rejected
All	100	55.9	11.4	32.7
Gram.	67	80.9	7.9	11.3
Un-Gram.	33	4.8	18.7	76.5

Table 3: Performance on subsets of development-test set. The Correct, Error, and Rejected values in each row represent percentage of sentences in the designated subset.

SUMMARY AND FUTURE WORK

This paper has presented a model of probabilistic LR parsing developed for speech understanding systems. Integrating the language model into the recognition process resulted in a significant increase in recognition performance. In addition, the combination of acoustic and parse scores was shown to be a useful rejection criterion.

The LR parser is one of three language models present in this version of the VOYAGER system, along with the word-pair used to compute backward estimates and TINA, which generates the semantics. These other language models were retained for convenience, since the focus of this work was on providing linguistic constraints during recognition. However, the presence of multiple independent language models decreases the system's portability and maintainability. A covering word-pair grammar can be automatically generated from the LR grammar and used to constrain the backward estimates. The LR parser can also be used to generate semantics; this technology is at the heart of many compilers.

There are several ways in which this work can be extended. One of the major drawbacks of using a parser-based language model for speech understanding is that it assigns probability 0 to sentences outside of its syntactic domain. In practice, the input to a speech understanding system will be ungrammatical but nonetheless have a clear semantic interpretation. One important area of further work is to combine the LR parsing model with a less strict language model such as a bigram, without reducing its constraining power. One approach is to use a backoff mechanism like that used in the SPHINX-LR system [5].

Among the other interesting extensions to the parser are:

- Augmenting the CFG formalism with atom-valued feature restrictions on the productions, allowing a more concise representation of the grammar. These augmented rules can be expanded into CFG rules, or used directly by the LR parser.

- Condition the word probabilities on the stack contents as well as the LR state. This is relatively simple to implement but requires more training data to estimate the additional parameters.

- Use the Generalized LR parsing algorithm [12] to deal with non-deterministic grammars. This was not necessary in this work since a deterministic grammar gave adequate coverage of the VOYAGER domain.

REFERENCES

[1] Goodine, D., Seneff, S., Hirschman, L. and Phillips, M., "Full Integration of Speech and Language Understanding in the MIT Spoken Language System", *Proc. Eurospeech 91*, 24-26, September, 1991.

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

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

[4] Hanazawa, M., "ATR HMM-LR Continuous Speech Recognition System", *Proc. ICASSP 90*, 52-56, 1990.

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

[6] Wright, J.H., "LR Parsing of Probabilistic Grammars with Input Uncertainty", *Computer Speech and Language* **4**, 297-323, 1990.

[7] Barr, A. and Feigenbaum, E. em Eds., *The Handbook of Artificial Intelligence*, William Kaufmann Inc., Los Altos, CA, 1981.

[8] Pallett, D., "DARPA Resource Management and ATIS Benchmark Test Poster Session" *DARPA Speech and Natural Language Workshop*, 49-58, February, 1991.

[9] Soclof. M. and Zue, V., "The Collection and Analysis of Spontaneous and Read Corpora for Spoken Language System Development", *Proc. ICSLP 90*, 1105-1108, November, 1990.

[10] Zue, V., "The SUMMIT Speech Recognition System: Phonological Modelling and Lexical Access", *Proc. ICASSP 90*, 49-52, 1990.

[11] Seneff, S., "TINA: A Probabilistic Parser for Speech Understanding Systems", *DARPA Speech and Natural Language Workshop*, February, 1989.

[12] Tomita, M., *Efficient Parsing for Natural Language*, Kluwer, Boston, MA, 1986.