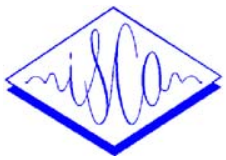

6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

REDUCING SEARCH COMPLEXITY IN LOW PERPLEXITY TASKS

Martin Franz, Miroslav Novak

IBM T. J. Watson Research Center P. O. Box 218,
Yorktown Heights, NY 10598
(franzm, miroslav)@us.ibm.com

10.21437/Eurospeech.1999-117

ABSTRACT

In this paper we present a new method for improving the throughput of an asynchronous stack search based speech recognition system in the low perplexity applications. The algorithm reduces the acoustic fast match use in the cases where the word context information represented by the language model is sufficient to provide a reliable list of word candidates for the detailed match processing. The proposed technique improves the throughput of the system by reducing the number of fast match calls and by shortening the list of candidate words to be processed by the detailed match. Tested on the set of 3400 sentences, the new method reduces the CPU requirements of the search part of the speech recognition system by 47.7%, increasing the throughput of the entire speech system by 30.8% without degrading the recognition accuracy.

1. INTRODUCTION

Systems for automatic speech recognition (ASR) are often applied in the tasks where the vocabulary size and overall text complexity are lower than in the general text entry applications, but the domain covered is larger than just a limited set of commands. Such applications may include complex command-and-control systems and the telephony dialog systems for retail, banking and other service domains. Applications of that kind typically require the use of ASR systems similar to the ones used for general dictation, but with the acoustic and language models trained on the specific domain data. Speed is often one of the critical parameters of such systems.

Most of the current ASR systems consist of the three major components: an acoustic fast match (FM), a language model (LM) and an acoustic detailed match (DM). The fast match is employed to construct a list of candidate words given the acoustic evidence at the current point in the data stream. The language model is used similarly to create a list of candidate words given the word context. The candidate words from the top of the combination of the FM and LM lists are then processed by the more detailed and accurate, but computationally expensive and thus time consuming detailed match.

The technique proposed in this paper utilizes the fact that in many ASR systems, used in low to medium perplexity applications, the word context information supplied by the language model is in some cases strong enough to be used in place of the acoustic fast match.

2. ALGORITHM DESCRIPTION

The technique for reducing the search complexity can be outlined in the following steps:

1. Based on the state of the current search hypothesis, as characterized by the distance from the beginning of the sentence and the word history, the decision is made whether to call the fast match immediately. If the answer is positive, a fast match call is performed to obtain a list of candidate words with their acoustic scores and language model scores are obtained for the word candidates proposed by the fast match.

2. In the other case the language model is used to determine the candidates for the further processing. Based on the characteristics of the language model score list, the fast match may be optionally called, but it is often possible to use the list of words with high language model scores as an input of the detailed match, avoiding the need of fast match computation.

3. Detailed match acoustic scoring is performed on the list of top ranking words from the step 1 or 2.

The corresponding flow chart is shown on Figure 1.

3. EXPERIMENTAL RESULTS

The above described technique was tested in the context of the IBM research ASR system, similar to the one described in [1], applied in a prototype of a natural language understanding system for the financial domain.

The system uses different acoustic models for subphonetic units in different contexts. These instances of context dependent classes are identified by growing

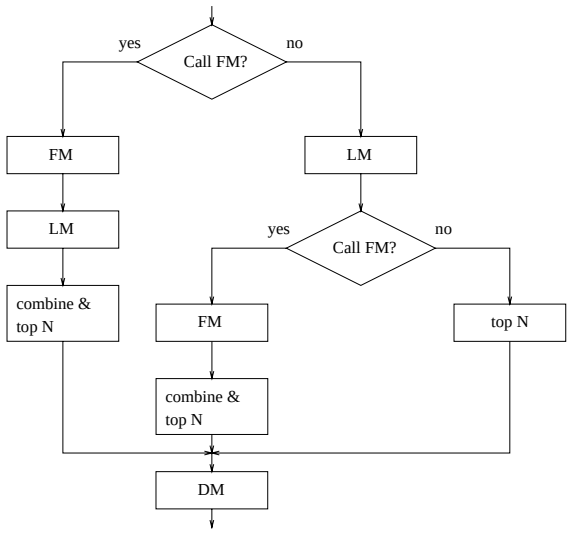

Figure 1: Reducing the search complexity by eliminating most of the fast match calls

a decision tree from the training data. The acoustic feature vectors that characterize the training data at the leaves are modeled by a mixture of Gaussians pdf's, with diagonal covariance matrices (a total of 30050 Gaussians were used). As far as the output distributions on the state transitions of the model are concerned, rather than expressing the output distribution directly in terms of the feature vector, the system expresses it in terms of the rank of the leaf [4]. The acoustic front end uses a FFT based filter bank followed by cepstral rotation. Frame energy and dynamic parameters ($\Delta + \Delta\Delta$) were added to each feature vector. Sentence based cepstra mean normalization was used.

The vocabulary of the system contains 1977 words. The test data consists of 3400 sentences - mutual fund names pronounced by 73 speakers. The average utterance length is 5.5 words. It is important to realize that the speakers were given a full freedom to pronounce the fund names in a natural way in which they would speak about the funds in their everyday lives and thus the utterances do not match the official fund names. This feature of the system makes the use of a language model covering only a predefined set of canonical sentences quite impractical. Our language model is a trigram model, trained on a set of sentences generated from the list of fund names. Examples of the test utterances are shown in Table 1.

Figure 3 shows the distributions of the language model probabilities of the first three words in the sentences of the test set. It is apparent that as the length of the hypothesized context path grows, the LM probability distribution sharpens, providing more information for the further search.

The results of out experiments are summarized in

kemper worldwide two thousand four fund
eaton vance municipals maryland fund traditional class
pioneer real estate shares
phoenix strategic
seneca midcap fund

Table 1: Examples of test utterances

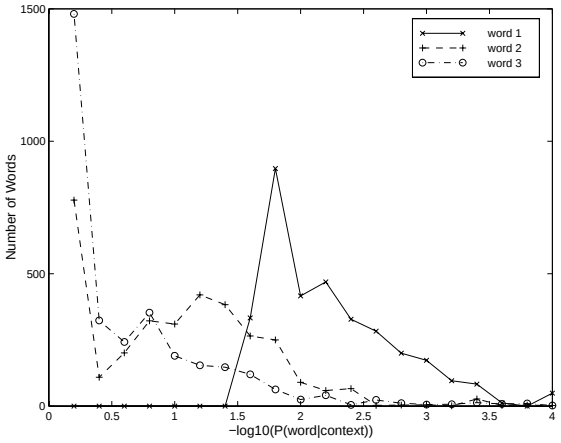

Table 2.

The first two columns of the table show the results of our baseline system.

We implemented the technique described in section 2 in a way where the FM call is always performed at the beginning of the utterance. In the rest of the utterance, the FM call is performed only if the list of candidate words created by the language model contains more than 50 words with probabilities greater than 0.31% of the probability of the highest ranking candidate in the LM list. In the cases where the FM is not used, the DM scoring is performed on the top 50 words hypothesized by the LM. The corresponding results are shown in the second two columns of Table 2.

We also experimented with forcing the top 10 candidate words from the LM list to be always included in the DM list, independently of their FM scores. The results obtained under this condition are tabulated in the last two columns of Table 2.

The lines of Table 2 show the absolute and percentage amounts of the CPU time spent by the whole decoding process (line 1), signal processing (line 2), detailed match, fast match and language model (lines 3 to 5), and the additional computations required to make the the fast match calls optional (line 6). The word and sentence error rates are listed on lines 7 and 8.

The CPU time spent by the fast match is reduced dramatically by avoiding most of the FM calls. The DM time requirements are also lowered by reducing the size of the list of candidate words processed by the DM. The overhead caused by making the FM

	baseline		complete		top 10	
	t[s]	t[%]	t[s]	t[%]	t[s]	t[%]
total	14202	0.63	11378	0.50	10015	0.44
SP	5034	0.22	5069	0.22	4696	0.21
FM	3281	0.14	1235	0.05	1533	0.07
DM	4195	0.19	3277	0.14	2239	0.10
LM	251	0.01	415	0.02	415	0.02
LM first	0.0	0.00	594	0.03	966	0.04
WER [%]		3.54		3.49		3.36
SER [%]		15.53		14.97		14.71

Table 2: Decoding Results

call optional is practically negligible and it is greatly overweighed by lowering the CPU requirements of the other components of the system. We also observed a modest improvement in recognition accuracy as measured by the word and sentence error rates, caused apparently by relying on the LM in the cases where it supplies stronger evidence than the acoustic models.

4. CONCLUSIONS

We have presented a novel technique to increase the throughput of an asynchronous stack search based speech recognition systems in low perplexity applications. Experiments show that using this method in our ASR prototype increases the throughput of the system by more than 30% with a modest improvement of the recognition accuracy.

5. REFERENCES

[1] Bahl,L.R., Balakrishnan-Aiyer, S., Bellegarda, J.R., Franz, M., Gopalakrishnan, P.S., Nahamoo, D., Novak, M., Padmanabhan, M., Picheny, M.A., Roukos, S., "Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task", in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, Detroit, Michigan, 1995.

[2] Bahl, L.R., De Gennaro, S.V., Gopalakrishnan, P.S., Mercer, R.L., "A Fast Approximate Acoustic Match for Large Vocabulary Speech Recognition", in *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 1, pp 59-67. January 1993.

[3] P.S. Gopalakrishnan, L.R. Bahl, R.L. Mercer, "A Tree Search Strategy for Large-Vocabulary Continuous Speech Recognition.", in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, Detroit, Michigan, 1995.

[4] Bahl,L.R., de Souza, P.V., Gopalakrishnan, P.S., Nahamoo,D., Picheny M.A., " Robust Methods for using Context-Dependent features and models in a continuous speech recognizer", in *Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing*, Adelaide, Australia, 1994