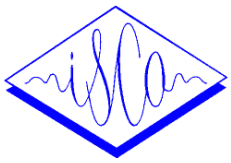

2nd European Conference on
Speech Communication and Technology
EUROSPEECH '91
Genova, Italy, September 24-26, 1991

A LOOK-AHEAD SEARCH TECHNIQUE FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Reinhold Haeb-Umbach, Hermann Ney

Philips Research Laboratory Aachen, P.O. Box 1980, D-5100 Aachen, Germany

ABSTRACT

In a large vocabulary continuous speech recognition task the search for the "best" (in the maximum-a-posteriori sense) word sequence is the most (computing) time consuming part of the system. End-of-word hypotheses are created almost every time frame. With a stochastic language model every lexicon entry is an admissible successor candidate. By using a "fast match" module which scores the word candidates according to their acoustic feasibility ahead of the current time frame, the search cost can be considerably reduced. Only the fraction of the words with favourable fast match scores will be further processed in the detailed match, where the likelihood of a segment of acoustics given the word model is computed. We derive a novel word selection strategy which is "consistent" in the sense that it introduces no additional decoding errors and which still reduces the search space by a factor of 2 - 3 compared to standard Viterbi beam search. Giving up the consistency requirement, pruning strategies can be deduced which further reduce the search effort significantly: the size of the word startup list is reduced to 2% - 4% of its original size with a modest increase in error rate by 1% - 2%.

1. INTRODUCTION

In a large vocabulary speech recognition system using Hidden Markov models the output word is chosen to be the one that has the maximum a-posteriori probability given the input acoustic observation. This involves calculating the likelihood of the observed acoustic events given the models for each word in the vocabulary. When the vocabulary is very large this results in large decoding times which are far from real time on a modest amount of hardware.

"Fast match" methods are aimed at speeding up the recognition process by curtailing the list of word candidates to a fraction of the total lexicon size - while, at the same time, ideally introducing no additional decoding errors. These word candidates must be considered each time an end-of-word hypothesis has been created. This is accomplished by scoring word hypotheses based on their acoustic similarity with the signal portion ahead of the current time frame. Various implementations based on different principles have been reported in the literature, e.g. [1-5]. They mainly differ in the way the fast match scores are obtained. Some methods rely on dynamic programming [3,4] whereas others avoid it [1,2]. Most implementations employ some sort of "coarse" acoustic models which allow a faster calculation of the rapid match scores than the complete models used in the detailed match.

A "consistent" fast match is described in [5]. Consistent means that the reduction of the size of the word startup list will not introduce additional recognition errors. This is achieved in a two-step method where first an upper bound on the likelihood is computed for each lexicon entry and second the detailed match is evaluated for the word candidate with the best upper bound. All words whose upper bound on the likelihood is smaller than this likelihood obtained by the detailed match can safely be discarded.

In our approach we use the detailed phoneme models for the rapid match. The computational overhead introduced by the look-ahead is still small since we use a tree organization of the lexicon, employ a "streamlined" dynamic programming algorithm including beam pruning, and reuse the calculated distances for the detailed match and successive rapid matches. Since exact look-ahead scores ("score" corresponds to negative log-likelihood, i.e. the smaller the score the higher the likelihood) are available a consistent preselection process is feasible. In Section 2 we describe the data base and the system environment for the recognition experiments. Section 3 contains details of the fast match technique, and Section 4 presents experimental results.

2. THE RECOGNITION SYSTEM ENVIRONMENT

The system under consideration is a phoneme-based speaker-dependent continuous speech recognizer with a 12,300-word (German) recognition vocabulary [6].

Training and test data are read speech recorded in an office environment. The training data consists of 300 sentences (200 short phonetically balanced sentences and 100 long sentences obtained from business correspondence) comprising 2734 words.

The test corpus consists of 50 sentences of business correspondence amounting to 1099 words.

We use a pronounciation dictionary with 43 context-independent phonemes and a silence model. They are represented by Hidden Markov Models of typically 6 states per phoneme and continuous mixture densities. Details of the preprocessing and the HMM's can be found elsewhere [6,7].

The system employs the Viterbi approximation (most likely state sequence) both in training and recognition. The search is a data-driven one-pass dynamic programming search with a potential search space of roughly 650,000 states. To achieve manageable decoding times beam pruning is used and the partial sentence hypotheses are list-organized such that only a (varying) fraction of the total number of states has to be considered every 10ms time frame.

We use a unigram or bigram language model. In this report we only apply a unigram model with test set perplexity of 1831.

3. DESCRIPTION OF THE FAST SELECTION ALGORITHM

3.1 A Consistent Word Selection Algorithm

In previous approaches [1-5] the fast match module computes approximate scores which gauge the acoustic probability of each word in the anticipatory interval. A word is considered a valid candidate for startup if its rapid match score is below some threshold. Since we use the exact phoneme models in the fast match stage the look-ahead (LA) score contains more information than would be exploited if it were only used to be compared to a threshold. When combined with the detailed match score a consistent pruning strategy is attainable.

Let $S_w(I+1:I+\Delta I)$ denote the acoustic LA score of the word w for the anticipatory interval $[I+1,I+\Delta I]$. Let further $S^E_{v1...vn}(1:I)$ be the score of the partial sentence hypothesis containing the words $v_1,...,v_n$ starting in time frame 1 and ending in I with the end of word v_n (the superscript "E" denotes word end). Then

$$S_{v1..vn,w}(1:I+\Delta I) = S^E_{v1...vn}(1:I) + S_w(I+1:I+\Delta I) + S_{LM}(w|v_1,..,v_n) \quad (1)$$

is the score of the partial sentence hypothesis $v_1,...,v_n,w$ at time frame $(I+\Delta I)$ which assumes a word boundary from word v_n to word w at frame I. $S_{LM}(w|v_1,..,v_n)$ denotes the language model score (assuming an n-gram language model). With

$$S_{min}(I) = \min S(1:I)$$

$$S_{min,LA}(I+1:I+\Delta I) = \min \{S_w(I+1:I+\Delta I)+S_{LM}(w|v_1,..,v_n)\}$$

- where the first minimum is taken over all partial sentence hypotheses at frame I (time-synchronous minimum of all hypotheses at time frame I) and the second over all words w and preceding word sequences of arbitrary length n - the following inequality holds

$$S_{min}(I) + S_{min,LA}(I+1:I+\Delta I) \geq S_{min}(I+\Delta I) \quad (2)$$

where $S_{min}(I+\Delta I)$ is the minimum score of all hypotheses at time frame $(I+\Delta I)$. Partial sentence hypotheses not indicating a word boundary at frame I may or may not result in a better (= smaller) score at frame $(I+\Delta I)$.

Now it is evident how to obtain a consistent pruning strategy: being at time frame I, add word w to the word startup list if

$$S^E_{v1...vn}(1:I) + S_w(I+1:I+\Delta I) + S_{LM}(w|v_1,..,v_n) - (S_{min}(I) + S_{min,LA}(I+1:I+\Delta I)) \leq T \quad (3)$$

where T is the pruning threshold which is used for the conventional beam pruning in the detailed match.

This pruning is consistent in the sense that a word that will not be placed on the word startup list could have never survived the beam pruning in the detailed match if it had been placed on the startup list.

Note that the LA pruning and the beam pruning of the detailed match are not independent of each other. The stronger the beam pruning the stronger will be the LA pruning. Thus the size of the word startup list is automatically adjusted to a change in the beam pruning threshold. No manual adjustment is required.

Further note that the number of word startups depends on the likelihood of the word boundary at frame I. If $S_{min}(I)$ is smaller than the score of the hypothesis that signals a word boundary at frame I then fewer words will be included in the word startup list than if the two scores were equal.

Language model recombination will be performed before word startup. In case of an unigram language model this is particularly simple. Since then $S_{LM}(w|v_1,..,v_n) = S_{LM}(w)$ the starting score for word w will be $\min(S^E_{v1...vn}(1:I) + S_{LM}(w))$ where the minimum is taken over all hypotheses that indicate a word boundary at frame I.

The larger the anticipatory interval length ΔI is chosen the more word candidates will be pruned and the higher will be the computational cost of the LA calculation (see Section 4). The second conclusion is obvious. The first is readily seen when Eq. (3) is rearranged:

$$\{S^E_{v1...vn}(1:I) + S_{LM}(w|v_1,..,v_n) - S_{min}(I)\} + \{S_w(I+1:I+\Delta I) - S_{min,LA}(I+1:I+\Delta I)\} \leq T$$

Whereas the terms in the first pair of braces are independent of the anticipatory interval length ΔI, the terms in the second pair will increase with increasing ΔI. Due to the Markov Model assumptions the local contributions to the overall score that are added per time frame are independent and random. Therefore the variance of S_w increases with increasing ΔI ("random

walk"). Thus, for a given ΔI, the probability that S_W falls within a fixed range whose width is independent of ΔI decreases with increasing ΔI.

3.2 Details of Implementation

3.2.1 Tree Organization of the Lexicon

For the detailed match the recognition vocabulary is stored in a simple table which, for each word, contains its phonetic transcription. In our case this means storing 110,000 phonemes for the 12,300 words of the vocabulary. However, more efficient storage and decoding can be achieved if the phonetic transcriptions for all the words are arranged into a tree [3,4] since many words in a large vocabulary will begin with the same initial sequence of phonemes. In the tree structure each arc corresponds to a phoneme, and each leaf to a word. Our lexical tree comprises 43,000 arcs which means a reduction factor of 2.6 versus the table organization. Table 1 gives the number of arcs in the first 5 layers (generations) of the tree

layer	1	2	3	4	5
#arcs	28	331	1511	3116	4380

Table 1: Number of tree arcs for the first 5 layers of the tree-organized lexicon.

If the anticipatory interval length corresponds to the first two layers of the tree then (28+331)*6 states have to be considered per time frame in the tree organized search versus 12300*12 states for the simple table organization!

3.2.2 Scoring for Look-Ahead Calculation

For each word w the look-ahead score $S_W(I+1:I+\Delta I)$, which is required at time frame I, is obtained by time-aligning the succeeding ΔI input frames with the Markov Models of the phonemes. We chose the maximum number of states, NJ, that can be traversed during the anticipatory interval of length ΔI to be

$$NJ = 2/3\ \Delta I$$

since preliminary experiments showed that for our phoneme models there are on the average 1.5 frames assigned to a state. Each word is given a look-ahead score as the minimum score of its initial phoneme sequence using a dynamic programming (DP) algorithm working on the tree-organized lexicon. The look-ahead score is defined as the minimum (= best) score of all path hypotheses which may either end in state NJ or in frame $(I+\Delta I)$ - where hypotheses of different length have to be first normalized (divided by their length in frames) before they can be compared. The minimum LA score $S_{min,LA}(I+1:I+\Delta I)$ is, however, chosen to be the minimum score of all hypotheses that end in frame $(I+\Delta I)$ (time-synchronous minimum).

To reduce the amount of computation for the rapid match beam pruning with a carefully chosen pruning threshold is also applied to the fast match computation. Note that there is no need for trace-keeping and traceback as in the detailed search since the only objective is to obtain the look-ahead score for each word. To further reduce the computational effort the fast match routine is called at most every other time frame. If an end-of-word hypothesis is created in the intermediate frame the most recent look-ahead scores are used.

Due to these simplifications we actually do not compute $S_W(I+1:I+\Delta I)$ but rather a (very good) estimate of it.

3.2.3 Detailed Phoneme Models

The DP performed during the fast match incorporates the detailed phoneme models. Therefore the distances calculated can later be used in the detailed match (and in successive fast match calls). Apart from some storage instructions, there are no additional costs by distance calculations because the distances calculated during fast match would have to be computed anyway even in the absence of a fast match.

4. EXPERIMENTAL RESULTS

4.1 Consistent Look-Ahead

We have run experiments to check the validity of the approximations of Section 3.2.2 to the consistent word startup pruning strategy and to assess the effectiveness of the pruning. The "correctness" of the pruning is measured by the recognition error rate and the effectiveness or speed-up is measured by the average number of gridpoints (= states) to be evaluated per frame and by the average size of the word startup list.

Table 2 presents the results (for one speaker) for a recognition experiment with and without look-ahead. The anticipatory interal length was chosen to ΔI = 13 frames and NJ = 9 states which corresponds to approximately 1.5 phonemes. The recognition error rate remained unchanged when the look-ahead was employed. However we observed that for 3 of the total of 50 sentences there were differences in the detected word boundaries and different errors than in the case of no LA. The minimum score of these sentences was slightly larger.

	without LA	with LA
gridpoints	45000	18000
gridpoints (LA)	–	1000
word startups	10650	1900
errors: de/ins/sub	27/24/161	25/25/162
error rate	19.3%	19.3%

Table 2: Recognition results with and without "consistent" LA

If no fast match is applied the average number of word startups is 10650 which is somewhat less than the vocabulary size because end-of-word hypotheses and therefore word startups do not occur in every time frame. If the fast match module is used the word startup list size is reduced by a factor of 5.6 and the number of gridpoints by 2.5 and 2.4, respectively, when the number of gridpoints required for LA is included. From this and similar results for other speakers we concluded that the approximations of Section 3.2.2 to the consistent pruning are legitimate.

To achieve larger speed-up we tried out more stringent pruning strategies derived from this consistent method, which, however, no longer guarantee consistency.

4.2 Alternative Pruning Strategies

In Section 3.2.2 we noted that the minimum LA score $S_{min,LA}(I+1:I+\Delta I)$ is defined as the time-synchronous minimum of all hypotheses at the end of the anticipatory interval $(I+\Delta I)$. On the other hand the look-ahead score of some word w is calculated as the best score of all path hypotheses for that word that end either in state NJ or at the end of the anticipatory interval $(I+\Delta I)$. If, in the same manner, $S_{min,LA}(I+1:I+\Delta I)$ is chosen as the minimum of these LA scores then the resulting value will be equal or smaller than before resulting in more stringent pruning when used in (3). Note, however, that the pruning is no longer consistent! Table 3 presents the experimental results for different anticipatory interval lengths.

	NJ = 9 ΔI = 13	NJ = 15 ΔI = 23	NJ = 21 ΔI = 30	NJ = 27 ΔI = 40	NJ = 33 ΔI = 50
gridpts	11300	7400	6100	5000	4400
gridpts (LA)	1000	2600	3500	4500	5000
total	12300	10000	9600	9500	9400
word startups	700	350	260	190	150
error rate	20.3%	20.4%	20.4%	21.0%	22.1%

Table 3: Performance of LA for different anticipatory interval lengths

The total number of gridpoints to be evaluated is a good measure for the overall search cost since it includes fast and detailed match. Note that the average number of gridpoints to be evaluated per frame, and in particular the number of word startups could be considerably reduced - at the expense of an increase of the error rate by 1% - 3%, though. Larger anticipatory interval lengths lead to greater reductions in the search effort but also to a larger increase of the error rate.

Table 4 summarizes results for 3 different speakers without LA and with LA with parameters NJ = 21, ΔI = 30 which we considered a good compromise between speed-up and correctness.

Speaker	M021	M022	M024
a)			
gridpts	45000	50000	60000
word startups	10650	10660	11250
error rate	19.3%	25.0%	31.9%
b)			
gridpts	6100	6700	13000
gridpts (LA)	3500	4200	6000
total	9600	10900	19000
word startups	260	270	450
error rate	20.4%	26.3%	33.3%

Table 4: Recognition results for 3 speakers:
a) without LA b) with LA

5. REFERENCES

[1] L.R. Bahl, R. Bakis, P.V. de Souza, R.L. Mercer, "Obtaining candidate words by polling in a large vocabulary speech recognition system", Proc. ICASSP'88, New York, pp 489 - 492, April 1988.

[2] L.S. Gillick, R. Roth, "A rapid match algorithm for continuous speech recognition", Darpa Workshop, June 1990.

[3] L.R. Bahl, S.V. De Gennaro, P.S. Gopalakrishnan, R.L. Mercer, "A fast approximate acoustic match for large vocabulary speech recognition", Proc. 1989 Eurospeech Conference, Paris, pp 156 - 158, Sep. 1989.

[4] X.L. Aubert, "A fast lexical selection strategy for large vocabulary continuous speech recognition", NATO ASI Series F, P. Laface (ed): Speech Recognition and Understanding: Recent Advances, Trends and Applications, Springer Verlag, to appear.

[5] L.R. Bahl, P.S. Gopalakrishnan, D. Kanevsky, D. Nahamoo, "Matrix fast match: a fast method for identifying a short list of candidate words for decoding", Proc. ICASSP'89, Glasgow,pp 345 - 348, April 1989.

[6] V. Steinbiss et al., "A 10000-word continuous-speech recognition system", Proc ICASSP'90, Albuquerque, NM, pp 57 - 60, April 1990.

[7] H. Ney, "Acoustic modelling of phoneme units for continuous speech recognition", Signal Processing V: Theories and Applications; L. Torres, E. Masgrau, M.A. Lagunas (eds). Elsevier Science Publishers, pp 65-72, 1990.