ISSN 1018-4074 (Volume 3)
ESCA

EUROSPEECH '93

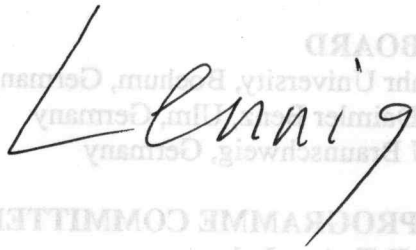
3rd EUROPEAN CONFERENCE ON SPEECH COMMUNICATION AND TECHNOLOGY

Berlin, Germany
21 - 23 September 1993

PROCEEDINGS

VOLUME 3

EUROSPEECH '93



PROCEEDINGS

VOLUME 3

A VERY FAST METHOD FOR SCORING PHONETIC TRANSCRIPTIONS

P. Kenny, P. Labute, Z. Li, R. Hollan, M. Lennig and D. O'Shaughnessy

INRS-Télécommunications, Université du Québec
16, place du Commerce, Verdun, Québec H3E 1H6 Canada

ABSTRACT

We present a new fast match for very large vocabulary continuous speech recognition in which phonetic transcriptions are scored at a cost of one floating-point operation per phone. The algorithm uses a table of estimates of phone scores and durations derived by searching a relatively small phonetic graph in a pre-processing step. The pre-processor runs in faster than real time on a mid-range work station.

1. INTRODUCTION

The role of a fast match in continuous speech recognition is to perform a rapid but not necessarily very accurate search of a large lexicon and return a list of hypothetical word transcriptions to be scored exactly. Words in the lexicon may be transcribed in terms of phonemes or other phonetic units (we use the word 'phone' to embrace both possibilities) and the transcriptions compiled into a lexical tree. For large vocabulary applications with multiple transcriptions per word the lexical tree may be very large and, since it has to be searched every time a word boundary is hypothesized, very fast algorithms are needed. In [1] we showed how a lexical tree can be searched efficiently using a heuristic obtained from a prior search of a relatively small phonetic graph G^* which imposes much weaker phonotactic constraints on phone strings than the lexical tree. In this paper we will show how additional information extracted from searching such a graph can be used as a basis for a fast match search of a lexical tree with the following properties:

(i) The inclusion rate (that is, the percentage of the time that the word to be recognized is included in the list returned by the fast match) is essentially the same as the inclusion rate for a Viterbi search of the lexicon using context-independent phoneme models (at least in the clean-speech, speaker-dependent case).

(ii) *A phonetic transcription can be scored at a cost of one floating point operation per phone.*

As in [1], an A^* search of the lexical tree can be carried out using backward probabilities at the nodes of the graph G^* to compute a heuristic. What is new in this paper is that the backward probabilities are also used to draw up a table of estimates of durations (and scores) for each phone, indexed by the starting time of the phone and the identity of the immediately following phone. These estimates are not always exact but they are sufficiently accurate that they can be used effectively as a fast match scoring mechanism. A similar table has been used by Ljolje and Riley [4] as an exact match.

2. THE FAST MATCH SCORING MECHANISM

In [1] we experimented with several phonetic graphs, each of which can be used to impose phonotactic constraints on phone strings which are weaker than those imposed by the lexicon and can therefore serve as a basis for calculating an A^*-admissible heuristic for searching a large lexical tree. In that paper we showed that by constructing the phonetic graph appropriately it is possible to devise an accurate method of segmenting partial transcriptions *as they are hypothesized* in the course of an A^* search which leads to a large reduction in the effective size of the search space.

Let us say that the phone inventory has the k-phone lookahead property (for some positive integer k) if, in matching speech data with an arbitrary phone string $F_1F_2\ldots$ the endpoint of the partial transcription $F_1\ldots F_n$ is independent of the phones $F_{n+k+1}F_{n+k+2}\ldots$ for each $n = 1, 2\ldots$. We have found empirically that in speaker-dependent recognition of clean speech with (context-independent) phoneme models the 2-phone lookahead property obtains. We now outline how this fact can be used to construct an admissible heuristic for continuous speech recognition (a more detailed treatment and experimental results are given in [2, 3]).

First we specify the phonetic graph G^*. Let us say that a phone triple FGH is a legitimate triphone if it can be obtained by concatenating phonetic transcriptions of words in the dictionary and define the notion of a legitimate diphone similarly. Then the nodes, branches and branch labels in G^* are as follows:

(i) *Nodes:* There is one node for every legitimate diphone FG

(ii) *Branches:* For every legitimate triphone FGH there is a branch from the node FG to the node GH

(iii) *Branch Labels:* If FGH is a legitimate triphone then the branch from FG to GH carries the label F

(It is easy to see that this graph imposes triphone phonotactic constraints on phone strings, that is if $F_1F_2\ldots$ is a phone string obtained by tracing any path through G^* then $F_iF_{i+1}F_{i+2}$ is a legitimate triphone for each $i = 1, 2, \ldots$)

Now suppose that we are given speech data in the form of a string of feature vectors $Y_1, Y_2 \ldots$. For each node FG in G^* and each time t, let $\beta_t^*(FG)$ denote the Viterbi score of the data $Y_{t+1}Y_{t+2}\ldots$ on the optimal path through G^* that leaves node FG at time t: these backward probabilities can be calculated by performing a Viterbi search of G^* in the reverse time direction (see Section 3). For each phone string $F_1 \ldots F_n$, let $\alpha_t(F_1 \ldots F_n)$ denote the Viterbi score of the data $Y_1 \ldots Y_t$ against the partial transcription $F_1 \ldots F_n$. For each $n \geq 2$ define the heuristic score of the partial transcription $F_1 \ldots F_n$ to be

$$\max_t \alpha_t(F_1 \ldots F_{n-2})\beta_t^*(F_{n-1}F_n). \quad (1)$$

The heuristic is admissible (irrespective of whether or not the 2-phone lookahead property holds) since the graph G^* allows greater freedom in extending $F_1 \ldots F_n$ then the lexicon does.

In order to take advantage of the 2-phone lookahead property, note that, since the first two phone labels on any path in the graph G^* that starts at the node $F_{n-1}F_n$ must be F_{n-1} and F_n, the endtime of the phone F_{n-2} is

$$\operatorname{argmax}_t \alpha_t(F_1 \ldots F_{n-2})\beta_t^*(F_{n-1}F_n).$$

Thus it is only necessary to associate one endtime hypothesis with each partial transcription (namely the endtime of the third-to-last phone); this minimizes the amount of work needed in expanding paths in the A^* search.

The point we wish to make in this paper is that, if the stronger 1-phone lookahead property holds, then the heuristic score of $F_1 \ldots F_n$ can be evaluated at a cost of $n-1$ floating point operations (once the β^*'s have been calculated). This statement is based on the following observation.

Suppose that a phone F is hypothesized to start at time t in a context where it is followed by a phone G. Consider the optimal path in the graph G^* that leaves node FG at time t. Suppose that the next node that the path visits is GH and that the time of the visit is t'. Then we can estimate the end-time of the F segment as t' and if the 1-phone lookahead property holds this estimate will be exact. (Note also that the score of the F segment can be evaluated as $\beta_t^*(FG)/\beta_{t'}^*(GH)$.) Thus in the course of the β^* calculation, we can draw up a table from which we can read off the duration and score of a phone whenever the starting time and immediate right context of the phone are given. Thus if we are given a partial phonetic transcription $F_1 \ldots F_n$, then each of the phones $F_1, \ldots, F_{n-2}$ can be scored and endpointed using this table and the last two phones F_{n-1} and F_n can be scored using the backward probability $\beta_t^*(F_{n-1}F_n)$ (where t is the endtime of F_{n-2}) for a total of $n-1$ floating point operations.

In fact we have found that the 1-phone lookahead property does *not* hold on our applications so the scoring mechanism that we have just outlined is not always exact. However we conducted an experiment which shows that for fast match purposes it is just as effective as our original scoring mechanism (1).

We took a 350-word extract from a recording of the novel *White Fang* and used the Viterbi algorithm to identify the word boundaries. Using a 60,000-word dictionary containing an average of 2.2 phonemic transcriptions per word and no language model, we attempted to recognize each of the words (given the starting time of the word but not the endtime) with a set of (context-independent) phoneme models, using both the fast match and the original mechanism (1) to score all of the transcriptions in the dictionary. The backward probabilities at the nodes of the graph G^* were calculated for all times up to 100 frames beyond the end of the word (1 frame = 10 ms).

The results of the experiment are shown in Figure 1 (at the end of the paper). To interpret the figure, a point (x, y) on the continuous curve indicates that, when the exact scoring mechanism is used, a transcription of the word to be recognized is found among the top x transcriptions y% of the time (and similarly for the fast match scoring mechanism). It is evident that the inclusion rates for the two scoring mechanisms are almost identical.

3. THE PRE-PROCESSOR

The practical value of the fast match method that we have proposed depends on how rapidly the table of phone scores and durations can be drawn up. Exhaustively searching the triphone graph using the algorithm described in [7] takes 5 times real time on a Sparc 10. The reason why this takes so long is that the triphone graph, while much smaller than the lexical tree, is still a big graph. (Recall that it contains one branch for every possible triphone, including triphones that can only appear in cross-word contexts. Our 60,000 word dictionary, which contains more than 130,000 transcriptions, gives rise to 40,000 triphones.)

In fact, although we did not realize this when we performed the experiment described in the previous section, searching the triphone graph is not the most efficient way to draw up the table of phone scores and durations. The primary reason for using the triphone graph is that this is what is needed to take advantage of the 2-phone lookahead property which was the basis for our earlier work on searching [1,2,3,6]. However, the new fast match is based on the 1-phone lookahead property and for this it turns out to be sufficient to search a graph which imposes *diphone* constraints on phone strings rather than triphone constraints.

This graph is constructed as follows.

(i) *Nodes:* There is one node for every phone F

(ii) *Branches:* For every legitimate diphone FG there is a branch from the node F to the node G

(iii) *Branch Labels:* If FG is a legitimate diphone then the branch from F to G carries the label F

The table of phone scores and durations can be derived from a reverse-time search of this graph using a slight modification of the Viterbi algorithm. Recall that the table is indexed by triples (t, F, G) where F is a phone hypothesized to start at time $t+1$ in a context

where it is followed by G. Suppose that on the best path in the graph which leaves the node F at time t *and which begins with the branch joining F to G*, the time at which the node G is visited is t'. Then we can conclude that, if the 1-phone lookahead property holds, the F segment ends at time t' and the score of the F segment is just the score of the data in the interval $[t+1, t']$ on this branch. Thus the information needed to draw up the table of phone scores and durations can obtained by maintaining a collection of traceback pointers for each node (one pointer for each branch originating in the given node).

The diphone graph is sufficiently small that it can easily be searched in real time. Note also that right context models can be used for this purpose. Suppose we are scoring a given branch, say joining the node F to the node G. According to the labelling scheme (iii), this branch has to be scored with an F model. However, for the same reason, we know that if this branch is taken then the next phone encountered is going to be G. Thus we can use a model for F in the context of G (instead of a context-indepdendent model for F).

Our dictionary uses an inventory of 40 phonemes, giving rise to about 1300 diphone pairs. Thus the diphone graph contains 1300 branches. The time taken to search the diphone graph with right context VQ discrete HMMs and draw up an exhaustive table of phone scores and durations is about 0.7 times real time on a HP 720.

4. CONCLUSION

We have presented a new fast match scoring procedure for very large vocabulary continuous speech recognition. A pre-processing step constructs a table of phone durations and scores which can be used to score phonetic transcriptions at a cost of one floating-point operation per phone. The pre-processor, whose computational cost is essentially independent of the size of the vocabulary, runs in faster than real time on a HP 720.

An interesting consequevnce of the new fast match is that, since all of the time alignment is done in the pre-processor, it completely changes the character of the search problem, that is, the problem of finding the N best word histories for a given stream of acoustic data. We are now dealing with this problem using the graph search techniques described in [5]. These have the advantage that the search space is not opened up into a tree as in stack decoding. We will be reporting the results of this work in the near future.

ACKNOWLEDGEMENTS

This work was supported by the Natural Sciences and Engineering Research Council of Canada and by Alex Parallel Computers.

REFERENCES

[1] P. Kenny, R. Hollan, V. Gupta, M. Lennig, P. Mermelstein and D. O'Shaughnessy, "A* - admissible heuristics for rapid lexical access", to appear in *IEEE Transactions on Speech and Audio Processing*, January 1993.

[2] P. Kenny, R. Hollan, G. Boulianne, H. Garadudri, Y.-M. Cheng, M. Lennig and D. O'Shaughnessy, "Experiments in continuous speech recognition with a 60,000 word vocabulary", *Proceedings ICSLP 92*, pp. 225–228, October 1992.

[3] P. Kenny, R. Hollan, G. Boulianne, H. Garudadri, M. Lennig, and D. O'Shaughnessy, "An A* algorithm for very large vocabulary continuous speech recognition", *Proc. DARPA Speech and Natural Language Workshop*, pp. 333–338 February 1992.

[4] A. Ljolje and M.D. Riley, "Optimal Speech Recognition Using Phone Recognition and Lexical Access", *Proceedings ICSLP 92*, pp. 313–316. October 1992.

[5] N. Nilsson, "Principles of Artificial Intelligence", Tioga Publishing Company, 1982.

[6] P. Kenny, G. Boulianne, H. Garudadri, S. Trudelle, R. Hollan, M. Lennig and D. O'Shaughnessy, "Experiments in Continuous Speech Recognition Using Books on Tape", submitted to *Speech Communication*, January 1993.

[7] P. Kenny, P. Labute, Z. Li, R. Hollan, M. Lennig and D. O'Shaughnessy "A New Fast Match for Very Large Vocabulary Continuous Speech Recognition", *Proceedings ICASSP 93*, April 1993.

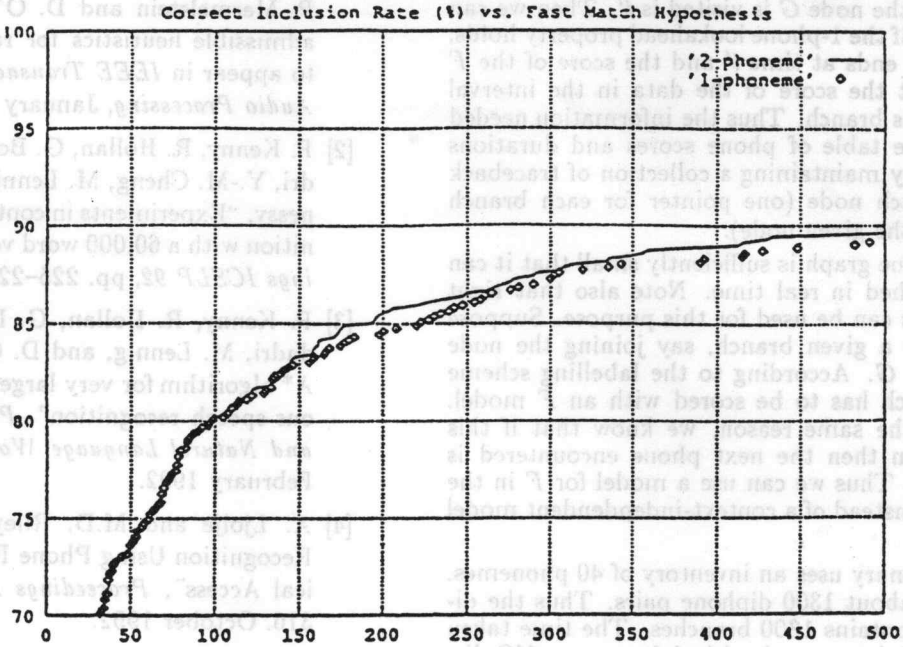


Fig. 1. Comparison of fast-match and exact scoring procedures