ENHANCEMENT AND OPTIMISATION OF A SPEECH RECOGNITION FRONT END
BASED ON HIDDEN MARKOV MODELS

F.R. McInnes, Y. Ariki* and A.A. Wrench

Centre for Speech Technology Research, University of Edinburgh, and
*Department of Information Science, Faculty of Engineering, Kyoto University

ABSTRACT
A method for performance evaluation of the acoustic-phonetic front end of a continuous speech recognition system, using the entropy of its output, is described. Results are given for a front end based on phonemic hidden Markov models, with various optional enhancements which have been optimised using the entropy criterion.

INTRODUCTION
This paper has two main parts. In the first, the problem of evaluating the performance of the acoustic-phonetic front end of a continuous speech recognition system is considered, and an evaluation technique using an entropy measure is described. In the second part, results of the application of this technique to a particular front end, which incorporates phoneme modelling based on the hidden Markov model paradigm, are given. The entropy evaluation technique allows various parameters of the front end to be optimised.

FRONT-END EVALUATION USING ENTROPY
The problem of how to evaluate the performance of the front end or acoustic-phonetic component of a continuous speech recognition system is one which has often proved difficult. Evaluation of the performance of a complete speech recognition system is comparatively straightforward: for most applications, the measure which is of most interest is the utterance recognition accuracy. For any input utterance, the system succeeds if it selects the correct recognition, and fails if it does not. (There are, of course, more detailed characteristics which it may be desirable to measure for some applications, such as the distribution of errors across the set of possible utterances, and the rate at which the correct recognition occurs among the top n candidates for some n > 1.) It is less obvious how to evaluate a front end, which produces as its output not a single recognition of the utterance - or of each word or phoneme within it - but a lattice of scored hypotheses from which the recognition is to be selected by the back end of the system using lexical and other linguistic information. The front end's output may contain not only ambiguities as to which speech unit (e.g. phoneme) occurs in each interval of the input, but also errors or uncertainties as to the segmentation of the utterance into a sequence of such intervals. It is desirable to have a performance measure which takes into account the omission and interposition of segments, as well as the identities of the hypotheses within each segment and the scores assigned to them.

It will be assumed for the discussion here that the front end generates as its output, for a given input utterance, a lattice of phoneme hypotheses, each of which has a start time, an end time and a probability score. The lattice will be assumed to be well structured in the sense that each hypothesis has abutting left and right neighbours (preceding and following hypotheses), so that no gaps or overlaps occur between successive phonemes in a path through the lattice corresponding to a possible recognition of the utterance. Moreover, for simplicity, it will be assumed that all the phonemes in the language are hypothesised in every segment (with varying probability scores), and so no phoneme substitutions are required in the extraction of a recognition from the lattice. Multiple segmentations of the input timescale may occur, however, to accommodate uncertainties as to the number and positions of phoneme boundaries. A small portion of a phoneme lattice is shown in figure 1. (Only the best-scoring one or two hypotheses are shown in each segment.)

Figure 1: part of a phoneme lattice
(for an utterance beginning "Some of...")

negative log probability score

0.4

0.3

0.2

0.1

-0.1

-0.2

-0.3

-0.4

time
To allow for possible segmentation errors in the lattice, insertions and deletions are permitted (for which probabilities must be estimated). The consequence is that for any recognition of the utterance (defined as a sequence of phonemes) there is a set of possible alignments to the lattice (depending on which segments are used and where insertions and deletions are performed); each combination of a recognition and an alignment defines a path through the lattice. The probability for a path is the product of the probability contributions for its individual steps - i.e. the hypotheses adopted from the lattice and any deletions and insertions which occur.

The measure adopted for the evaluation of the front end’s performance is the entropy (per phoneme) associated with the task which the back end must perform in selecting the correct recognition of an utterance from the set of possible recognitions left open by the front end.

In understanding the concept of entropy as it applies to the selection of a recognition from a lattice, it may be helpful to consider first a simple case, in which the segmentation is perfect, so that no insertions or deletions need be permitted (or, equivalently, the insertion and deletion probabilities can be set to 0), and in each segment there are k hypotheses with equal probability 1/k, one of which is correct (and all the other phonemes have probability 0). In this case, the number of possible recognitions of an N-phoneme utterance permitted by the acoustic evidence, as processed by the front end, is \( k^N \) (each one corresponding to one path through the lattice).

The back end has the task of selecting one of these \( k^N \) acoustically equiprobable recognitions, using the lexical and syntactic constraints which eliminate phoneme sequences not composing permissible sequences of words. The amount of information required to do this is equivalent to that contained in a \( \log_2(k^N) \)-bit number - i.e. \( N \log_2 k \) bits. That is, the uncertainty inherent in the front end’s output is \( \log_2 k \) bits per phoneme.

If the segmentation is still perfect but the number of hypotheses per segment varies and the hypotheses in each segment are not equiprobable, then the asymptotic equipartition theorem [1] states that as the utterance length N tends to infinity the log (to base 2) probability per phoneme for the true recognition tends to \( l \) with probability 1, where \( l \) is the expected (mean) log probability assigned to a correct phoneme hypothesis. That is, the correct utterance recognition is effectively, for large enough N, one of approximately \( 2^{-IN} \) recognitions each with acoustic probability close to \( 2^{-IN} \). Thus the amount of information per phoneme required to select the correct recognition is \(-l\) bits. This quantity \(-l\), the expected negative log probability per phoneme, is known as the entropy of the front end’s output, and denoted by \( H \).

In the more general, and more realistic, case in which the segmentation is not perfect, and so non-zero insertion and deletion probabilities are required, the idea of entropy can still be applied; but there is a complication in that different paths through the lattice no longer necessarily correspond to different recognitions of the utterance: multiple alignments of the same recognition are possible. This becomes even more so when the lattice contains multiple segmentations. Given that the back end scores any recognition according to its best-scoring path, an appropriate normalised entropy estimate is

\[
H_{\text{best}} = E[(\log_2 \sum \max \frac{P(\text{path})}{\text{recognition}}) / N_{\text{utt}}],
\]

where the expectation is taken over the distribution of utterances and \( N_{\text{utt}} \) is the number of phonemes in the correct recognition of a particular utterance. In practice the expectation will be estimated by an average over a set of evaluation utterances. Unfortunately, the denominator of the expression occurring in (1) for each utterance is difficult to compute. To avoid this problem, an approximation to \( H_{\text{best}} \) can be defined, using the sum over all paths for each recognition instead of the maximum:

\[
H_{\text{sum}} = E[(\log_2 \sum P(\text{path}) / N_{\text{utt}})].
\]

The numerator and denominator for each utterance can be computed using recursions which proceed through the lattice obtaining sums of partial-path probabilities.

In order to compute \( H_{\text{sum}} \) it is necessary to have not only the lattice of hypotheses with their probability scores, but also estimates of the insertion and deletion probabilities. These can be obtained by counting occurrences of insertion and deletion in alignments of utterances with their (correct) recognitions; the alignment can be found by a dynamic programming recursion.

**THE HMM-BASED FRONT END**

The front end to which the entropy evaluation technique has been applied is one based on an extension of the well-established hidden Markov model (HMM) technique [2], applied at the phoneme level. The extension lies in the incorporation of state duration probability distributions. Strictly speaking, with duration probabilities incorporated, the models are no longer HMMs but hidden semi-Markov models (HSMMs) [3].

The input speech is lowpass filtered at 7.5kHz and digitised at 16kHz. Various forms of signal processing can be applied to generate vectors of acoustic feature values. Here the signal processing is a 20th-order LPC analysis.
(after preemphasis with factor 0.97) in a 20ms Hamming window every 5ms, yielding a vector of 20 cepstral coefficients for each frame [4]. The acoustic feature vectors are converted to integers from 0 to 255 by vector quantisation [2]. This sequence of integers (VQ indices) is input to the HMM recognition module, which incorporates a three-state discrete-output HMM (or HSMM) for each phoneme.

There are two stages to the HMM-based recognition procedure. The first is the determination of the acoustically optimal sequence of phonemes together with the corresponding sequence of segment boundaries in the timescale of the input. This is accomplished by a one-stage Viterbi connected phoneme recognition algorithm [5]. The second stage involves tracing back through the segmentation and computing probability scores for all the phonemes in each segment. Optionally, during this second stage, additional segments may be defined, by an algorithm which, for a given segment end time \( t \), finds the best few local optima of the overall probability for the interval from 0 to \( t \) as a function of the previous segment boundary time; in this case, probabilities for all the phonemes are found in each segment so defined.

The output of the HMM processing for a given utterance to be recognised is a set of pairs (start time, end time) - one per segment - accompanied by vectors of phoneme probabilities. The probabilities are normalised for segment duration, by division of each raw segment log-probability by the number of frames in the segment.

PROBABILITY POSTPROCESSING

In order to obtain optimal phoneme probability estimates, it is necessary to apply some postprocessing to the probability scores generated by the HMMs.

Firstly, the normalisation by the number of frames in a segment may not be optimal. Secondly, the differing amounts of training data introduce a bias in the probabilities: phonemes whose models are trained on larger numbers of tokens tend to have higher probabilities assigned to them in the recognition than phonemes with smaller numbers of training tokens. To correct for these effects, the quantity \( p_i \) output by the HMM for phoneme \( i \) is replaced by

\[
q_i = p_i^{1/\alpha} P_i^{\beta},
\]

where \( P_i \) is the frequency of occurrence of phoneme \( i \) in the training data and \( \alpha \) and \( \beta \) are constants. (If \( \alpha < 1 \), the contrasts among the probabilities within a segment are enhanced; if \( \alpha > 1 \), they are moderated. The value of \( \beta \) corresponds to the degree of compensation for amounts of training data.) The quantities \( q_i \) are then scaled to values \( q_{\min} \) which sum to 1 within the segment.

In any segment, some phonemes will have very small probabilities \( q_i \), and it may be helpful to adjust these upward to some minimum value \( m \).

After all this processing, the acoustic probabilities are multiplied by the corresponding phoneme frequencies estimated for the language domain in view, and rescaled to sum to 1 again. This converts quantities proportional to \( P(\text{acoustic evidence} | \text{phoneme}) \) into a posteriori probabilities \( P(\text{phoneme} | \text{acoustic evidence}) \).

In the case with multiple segmentations, the differing overall unnormalised probabilities

\[
Q = \sum_i q_i
\]

in different segments can be used to adjust the final probability scores placed in the lattice: within a segment with a given value of \( Q \), all the phonemes' (weighted and rescaled) probabilities are multiplied by \( Q^d \), for some constant power \( x \). It may also be desirable to make the probability of deleting a segment depend on \( Q \). This has been implemented by making the deletion probability proportional to \( Q^d \) for a constant \( d \) which can be set independently of \( x \). If \( d < x \), deletion will be favoured more in a segment with a poor overall score.

EXPERIMENTS AND RESULTS

Experiments have been conducted, using the entropy evaluation criterion, to explore various aspects of the HMM-based front end. The experiments involved the construction of phoneme lattices for sentences spoken by three male speakers of RP English. The system operated in speaker-dependent mode: for each speaker, the HMMs for the 44 phonemes were trained on data from a set of 98 hand-labelled sentence utterances (from which the VQ codebook construction data were also derived), and then the lattice construction and entropy evaluation were performed on another set of 98 sentence utterances. 49 of the 98 test utterances' lattices were used to estimate insertion and deletion probabilities (by an iterative alignment and reestimation procedure), and the entropy was estimated on the other 49. The entropy results are presented in table 1. (Here \( S \) is the average number of segments per utterance, which is increased when multiple segmentations are allowed.)

First, the effects of the postprocessing were measured, for the default case of HMM processing with duration probabilities and the cepstral feature space. The postprocessing parameters \( \alpha, \beta, m, x \) and \( d \) were optimised (on separate data for the first of the three speakers) by the minimum-entropy criterion. Entropies for the three speakers are shown for the "no-postprocessing" values and for the optimised values. The introduction of the postprocessing yielded a highly significant entropy.
The largest contribution to this entropy reduction was evaluated. The results without duration modelling from the training-frequency compensation (using the individual-utterance terms) for each of the speakers. The effect of the state duration modelling was also evaluated. The results without duration modelling in table 1 show increases in entropy (in two cases highly significant) for all the speakers.

All the above results were with a single segmentation of each utterance. The introduction of multiple segmentations was found to allow improvements in the entropy provided that suitable values of \( x \) and \( d \) were employed. With the previously selected value of \( \alpha \) (1.13), the optimal value of \( x \) was found to be about 3.0 or 4.0 — corresponding to an exaggeration of the contrast in overall probability between different segments, relative to the contrast in probabilities within any one segment. Results for several values of \( x \) are given. \( x - d \) was held constant at 0.25. Marginally better results were obtained with \( \alpha \) reduced to 1.05 and \( x \) and \( d \) scaled down proportionately. (Less moderation of the probabilities in each segment is necessary when multiple segmentations are provided.) When a threshold was imposed on the log-probability difference from the best-scoring segmentation in the traceback through the HMM scores (reducing the number of segments per utterance), the entropy was seen to worsen gradually.

**DISCUSSION AND CONCLUSIONS**

A mathematically well-founded evaluation measure for a speech recognition front end has been implemented, namely the entropy of the lattice of hypotheses which it generates. This permits different segmentation and probability scoring options to be compared, and so allows the performance of the front end to be optimised.

Certain postprocessing operations have been found to yield improvements in the probability estimates obtained from HMMs. This should be true in other cases as well as in the phoneme-based system studied here.

An improvement in entropy has also been obtained by the inclusion of multiple segmentations in the lattice; and the importance of appropriate processing of the probability scores in this case has been demonstrated.

Plans for further work include consideration of other acoustic features besides the cepstral coefficients, and exploration of various possible ways of combining probabilities obtained from these. (Experiments have already been performed which showed improved recognition of hand-segmented phonemes when the probabilities for the vowels were estimated using mel formant parameters instead of cepstral coefficients [6]. The hierarchical scoring framework adopted in these experiments can be applied to any combination of acoustic feature spaces.)

**ACKNOWLEDGEMENTS**

This work was supported by the Information Engineering Directorate/Science and Engineering Research Council as part of the IED/SERC Large Scale Integrated Speech Technology Demonstrator Project (SERC grants D/29604, D/29611, D/29628, F/10309, F/10316) in collaboration with Marconi Speech and Information Systems and Loughborough University of Technology. The use of entropy as a front-end evaluation measure was proposed by Alan Crowe.

**REFERENCES**


