Large Vocabulary Continuous Speech Recognition: a Review

Steve Young
Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ
email: sjy@eng.cam.ac.uk

April 8, 1996

1 Introduction

Considerable progress has been made in speech recognition technology over the last few years and nowhere has this progress been more evident than in the area of Large Vocabulary Recognition (LVR). Current laboratory systems are capable of transcribing continuous speech from any speaker with average word error rates of between 5% and 10%. If speaker adaptation is allowed, then after 2 or 3 minutes of speech, the error rate will drop well below 5% for most speakers.

Hitherto, LVR systems have been limited to dictation applications since they were speaker dependent and they required words to be spoken with a short pause between them. The capability to recognise natural continuous speech input from any speaker, however, opens up many more applications and as a result LVR technology appears to be on the brink of widespread deployment across a range of Information Technology (IT) systems.

This article will discuss the principles and architecture of current LVR systems and identify the key issues affecting their future deployment. To illustrate the various points raised, the Cambridge University HTK system will be described. This is a modern design giving state-of-the-art performance and it is typical of the current generation of recognition systems.

2 System Overview

Current LVR systems are firmly based on the principles of statistical pattern recognition. The basic methods of applying these principles to the problem of speech recognition were pioneered by Baker, Jelinek and their colleagues from IBM in the 1970's and little has changed since[13, 54]. Figure 1 illustrates the main components of an LVR system.

An unknown speech waveform is converted by a front-end signal processor into a sequence of acoustic vectors, $\boldsymbol{Y} = \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T$. Each of these vectors is a compact representation of the short-time speech spectrum covering a period of typically 10 msecs. Thus, a typical ten word utterance might have a duration of around 3 seconds and would be represented by a sequence of $T = 300$ acoustic vectors.

The utterance consists of a sequence of words $W = w_1, w_2, ..w_n$ and it is the job of the LVR system to determine the most probable word sequence $\hat{W}$ given the observed acoustic signal $\boldsymbol{Y}$. To do this, Bayes' rule is used to decompose the required probability $P(W|\boldsymbol{Y})$ into two components, that is,

$$\hat{W} = \arg\max_{W} P(W|\boldsymbol{Y}) = \arg\max_{W} \frac{P(W)P(\boldsymbol{Y}|W)}{P(\boldsymbol{Y})} \tag{1}$$

This equation indicates that to find the most likely word sequence W, the word sequence which maximises the product of $P(W)$ and $P(\boldsymbol{Y}|W)$ must be found. The first of these terms represents

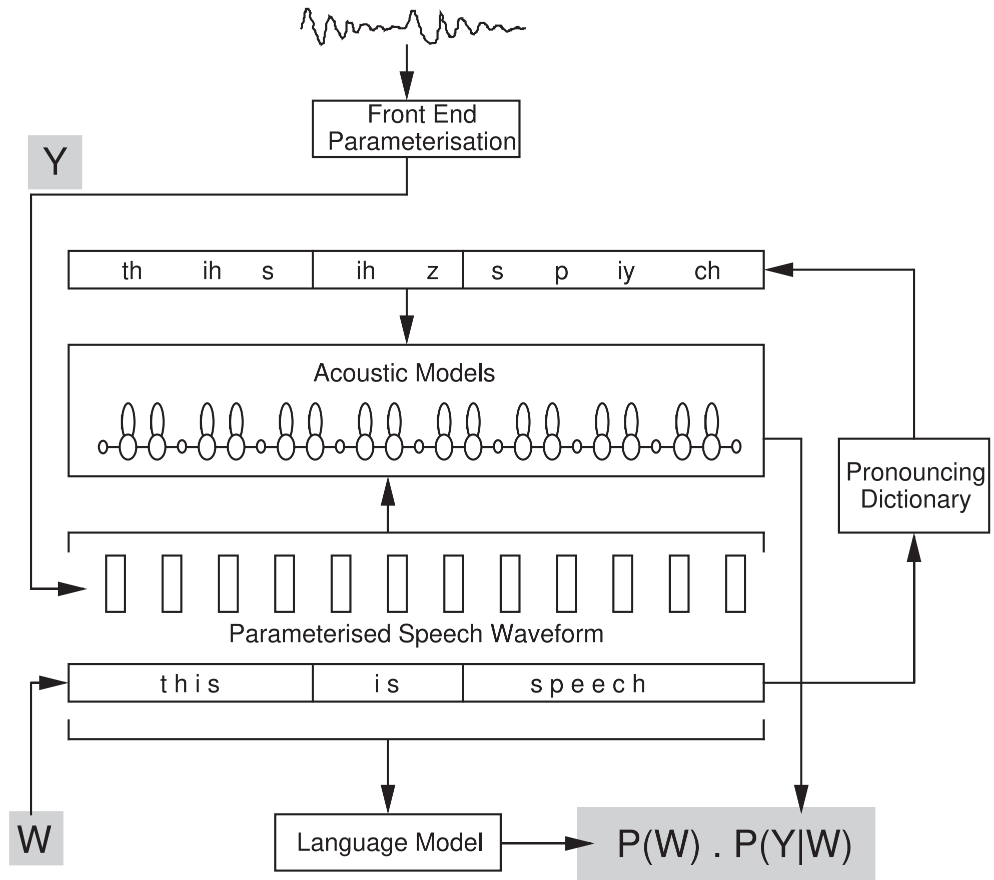

Figure 1: **Overview of Statistical Speech Recognition.** This diagram shows the computation of the probability $P(W|\boldsymbol{Y})$ of word sequence W given the parameterised acoustic signal $\boldsymbol{Y}$. The prior probability $P(W)$ is determined directly from a language model. The likelihood of the acoustic data $P(\boldsymbol{Y}|W)$ is computed using a composite hidden Markov model representing W constructed from simple HMM phone models joined in sequence according to word pronunciations stored in a dictionary.

the *a priori* probability of observing W independent of the observed signal and this probability is determined by a *language model*. The second term represents the probability of observing the vector sequence $\boldsymbol{Y}$ given some specified word sequence W and this probability is determined by an *acoustic model*.

Figure 1 shows how these relationships might be computed. A word sequence $W =$"This is speech" is postulated and the language model computes its probability $P(W)$. Each word is then converted into a sequence of basic sounds or *phones* using a pronouncing dictionary. For each phone there is a corresponding statistical model called a hidden Markov model (HMM). The sequence of HMMs needed to represent the postulated utterance are concatenated to form a single composite model and the probability of that model generating the observed sequence $\boldsymbol{Y}$ is calculated. This is the required probability $P(\boldsymbol{Y}|W)$. In principle, this process can be repeated for all possible word sequences and the most likely sequence selected as the recogniser output.

To convert the above design philosophy into a practical system requires the solution of a number of challenging problems. Firstly, a front-end parameterisation is needed which can extract from the speech waveform all of the necessary acoustic information in a compact form compatible with the HMM based acoustic models. Secondly, the HMM models themselves must accurately represent the distributions of each sound in each of the many contexts in which it may occur. Furthermore, the HMM parameters must be estimated from data and it will never be possible to obtain sufficient data to cover all possible contexts. Thirdly, the language model must be designed to give accurate word predictions based on the preceding history. However, as for the HMMs, data sparsity is an ever-present problem and the language model must be able to deal with word sequences for which no examples occur in the training data. Finally, the process outlined above for finding $\hat{W}$ by enumerating all possible word sequences is clearly impractical. Instead, potential word sequences are explored in parallel, discarding hypotheses as soon as they become improbable. This process is called *decoding* and the design of efficient decoders is crucial to the realisation of practical LVR systems capable of fast and accurate operation on today's computing platforms. The next four sections of this paper deal with each of these issues in more detail.

3 Front-End Parameterisation

A key assumption made in the design of current recognisers is that the speech signal can be regarded as stationary (i.e. the spectral characteristics are relatively constant) over an interval of a few milliseconds. Thus, the prime function of the front-end parameterisation stage is to divide the input speech into blocks and from each block derive a smoothed spectral estimate. The spacing between blocks is typically 10 msecs and blocks are normally overlapped to give a longer analysis window, typically 25 msecs. As with all processing of this type, it is usual to apply a tapered window function (e.g. Hamming) to each block. Also the speech signal is often pre-emphasised by applying high frequency amplification to compensate for the attenuation caused by the radiation from the lips.

The required spectral estimates may be computed via Linear Prediction or Fourier analysis[89] and there are a number of additional transformations that can be applied in order to generate the final acoustic vectors. To illustrate one typical arrangement, Fig 2 shows the front-end of the HTK Recogniser which generates Mel-Frequency Cepstral Coefficients (MFCCs)[24].

To compute MFCC coefficients, the Fourier spectrum is smoothed by integrating the spectral coefficients within triangular frequency bins arranged on a non-linear scale called the *Mel-scale*. For 8kHz bandwidth speech, the HTK Recogniser uses 24 of these triangular frequency bins. The Mel-scale is designed to approximate the frequency resolution of the human ear being linear upto 1000Hz and logarithmic thereafter. More importantly, its use has been shown empirically to improve recognition accuracy[91]. In order to make the statistics of the estimated speech power spectrum approximately Gaussian, log compression is applied to the filter-bank output.

The final processing stage is to apply the Discrete Cosine Transform (DCT) to the log filter-

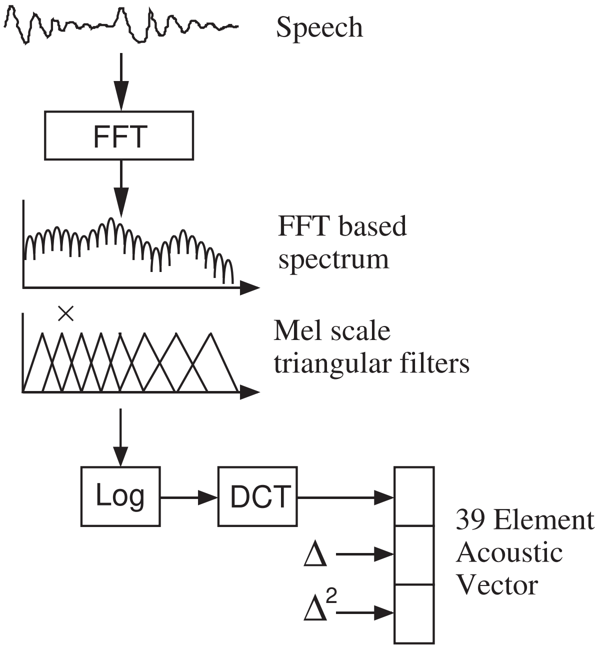

Figure 2: **MFCC-based Front-End Processor.** To perform pattern-matching, the speech waveform must be converted to a sequence of acoustic vectors representing a smoothed log spectrum computed every 10 msecs. Performance is improved by using a non-linear Mel-frequency scale followed by a Discrete Cosine Transform (DCT). The latter has the effect of decorrelating the signal thereby improving assumptions of statistical independence. Finally, first and second differentials are appended to incorporate dynamical information about the signal.

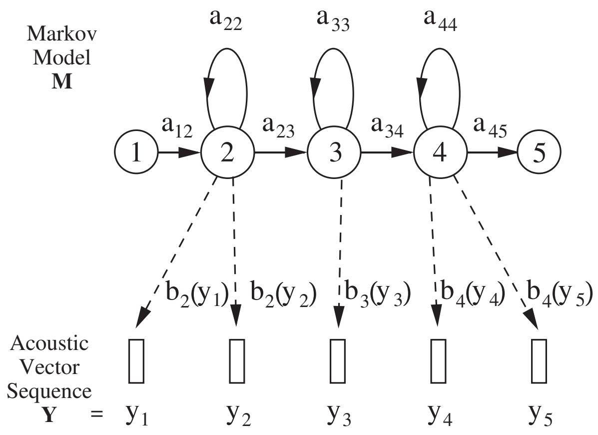

Figure 3: **HMM-based Phone Model.** A HMM can be regarded as a random generator of acoustic vectors. It consists of a sequence of states connected by probabilistic transitions. It changes to a new state each time period generating a new acoustic vector according to the output distribution of that state. The transition probabilities therefore model the durational variability in real speech and the output probabilities model the spectral variability.

bank coefficients. This has the effect of compressing the spectral information into the lower order coefficients and it also decorrelates them allowing the subsequent statistical modelling to use diagonal covariance matrices. In the HTK Recogniser, the signal energy plus the first 12 cepstral coefficients are retained to form a basic 13-element acoustic vector. Cepstral coefficients can also be derived from LP coefficients where they achieve a similar decorrelating effect[4, 30]. Good results have also been reported using LP coefficients to derive a smoothed spectrum which is then perceptually weighted to give Perceptually weighted Linear Prediction (PLP) coefficients[44].

As will be discussed in the next section, the acoustic modelling assumes that each acoustic vector is uncorrelated with its neighbours. This is a rather poor assumption since the physical constraints of the human vocal apparatus ensure that there is continuity between successive spectral estimates. However, appending the first and second order differentials to the basic static coefficients will greatly reduce the problem[31, 3]. In the HTK Recogniser, these are approximated by fitting a linear regression over a window covering the two preceding and two following vectors. When this is done, the final acoustic vector has 39 components.

Although the above description is specific to one particular recognition system, it is typical of most modern LVR systems. An important point to emphasise is the degree to which the front-end of modern recognisers has evolved to optimise the subsequent pattern-matching. For example, in the above, the log compression, DCT transform and delta coefficients are all introduced primarily to satisfy the assumptions made by the acoustic modelling component.

4 Acoustic Modelling

The purpose of the acoustic models is to provide a method of calculating the likelihood of any vector sequence $\boldsymbol{Y}$ given a word w. In principle, the required probability distribution could be found by finding many examples of each w and collecting the statistics of the corresponding vector sequences. However, this is impractical for large vocabulary systems and instead, word sequences are decomposed into basic sounds called *phones*.

Each individual phone is represented by a hidden Markov model (HMM). A HMM has a number of states connected by arcs. HMM phone models typically have three *emitting* states and a simple left-right topology as illustrated by Fig 3. The entry and exit states are provided to make it easy to join models together. The exit state of one phone model can be merged with the entry state of another to form a composite HMM. This allows phone models to be joined together to form words and words to be joined together to cover complete utterances.

A HMM is most easily understood as a generator of vector sequences. It is a finite state machine which changes state once every time unit and each time t that a state j is entered, an acoustic speech vector $\boldsymbol{y}_t$ is generated with probability density $b_j(\boldsymbol{y}_t)$. Furthermore, the transition from state i to state j is also probabilistic and is governed by the discrete probability a_{ij}. Figure 3 shows an example of this process where the model moves through the state sequence $X = 1, 2, 2, 3, 4, 4, 5$ in order to generate the sequence $\boldsymbol{y}_1$ to $\boldsymbol{y}_5$.

The joint probability of a vector sequence $\boldsymbol{Y}$ and state sequence X given some model M is calculated simply as the product of the transition probabilities and the output probabilities. So for the state sequence X in Figure 3

$$P(\boldsymbol{Y}, X|M) = a_{12}b_2(\boldsymbol{o}_1)a_{22}b_2(\boldsymbol{o}_2)a_{23}b_3(\boldsymbol{o}_3)\ldots \tag{2}$$

More formally, the joint probability of an acoustic vector sequence $\boldsymbol{Y}$ and some state sequence $X = x(1), x(2), x(3), \ldots, x(T)$ is

$$P(\boldsymbol{Y}, X|M) = a_{x(0)x(1)} \prod_{t=1}^{T} b_{x(t)}(\boldsymbol{y}_t) a_{x(t)x(t+1)} \tag{3}$$

where $x(0)$ is constrained to be the model entry state and $x(T+1)$ is constrained to be the model exit state.

In practice, of course, only the observation sequence $\boldsymbol{Y}$ is known and the underlying state sequence X is hidden. This is why it is called a *Hidden Markov Model.* However, the required probability $P(\boldsymbol{Y}|M)$ is easily found by summing equation 3 over all possible state sequences. An efficient recursive method of performing this calculation is available called the *Forward-Backward* algorithm. A crucial feature of this algorithm is that it also allows the probability of being in a specific model state at a specific time to be calculated. This leads to a very simple and efficient procedure called the *Baum-Welch* algorithm for finding Maximum-Likelihood estimates of both the a and b HMM parameter sets. Parameter estimation is beyond the scope of this article but it is important to note that the existence of Baum-Welch has been a key factor in making HMMs the dominant technology in acoustic modelling.

Alternatively, $P(\boldsymbol{Y}|M)$ can be approximated by finding the state sequence which maximises equation 3. Again a simple algorithm exists for computing this efficiently. It is called the *Viterbi* algorithm and, as will be discussed later, this algorithm is important in decoding where determination of the most likely state sequence is the key to recognising an unknown word sequence.

The above brief outline of HMMs is textbook material that has been well understood for many years. However, it is only recently that methods have been developed which allow HMM-based phone models to provide the acoustic discrimination necessary for large vocabulary speaker independent speech recognition.

It is instructive to rewrite equation 3 in logarithmic form and separate out the a and b terms

$$\log P(\boldsymbol{Y}, X|M) = \sum_{t=0}^{T} \log a_{x(t)x(t+1)} + \sum_{t=1}^{T} \log b_{x(t)}(\boldsymbol{y}_t) \tag{4}$$

The transition probabilities $a_{x(t)x(t+1)}$ model the temporal structure of the data. Regarding each log probability in equation 4 as a *score*, each transition term can be viewed as the cost of moving from one state to another. This actually provides a very poor model for the duration of real speech but this is not crucial since in practice the above expression is dominated by the output probabilities $b_{x(t)}(\boldsymbol{y}_t)$. Each HMM state provides a prototype acoustic vector and the log output probability function provides a distance metric to allow the actual acoustic vectors to be compared with the prototype.

The choice of output probability function is crucial since it must model all of the intrinsic spectral variability in real speech, both within and across speakers. Early HMM systems used discrete output probability functions in conjunction with a vector quantiser. Each incoming acoustic vector was replaced by the index of the closest vector in a precomputed codebook and the output probability functions were just look-up tables containing the probabilities of each possible VQ index. This approach is computationally very efficient but the quantisation introduces noise which limits the precision that can be obtained. Hence, modern systems use parametric continuous density output distributions which model the acoustic vectors directly[67, 56, 10]. The most common choice of distribution is the multivariate mixture Gaussian

$$b_j(\boldsymbol{y}_t) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\boldsymbol{y}_t; \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) \tag{5}$$

where c_{jm} is the weight of mixture component m in state j and $\mathcal{N}(\boldsymbol{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a multivariate Gaussian of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

So far there has been an implicit assumption that only one HMM is required per phone, and since approximately 45 phones are needed for English, it may be thought that only 45 phone HMMs need be trained. In practice, however, contextual effects cause large variations in the way that different sounds are produced. Hence, to achieve good phonetic discrimination, different HMMs have to be trained for each different context. The simplest and most common approach is to use *triphones* where every phone has a distinct HMM model for every unique pair of left and right neighbours. For example, suppose that the notation `x-y+z` represents the phone `y` occurring after an `x` and before a `z`. The phrase, "Beat it!" would be represented by the phone sequence `sil b iy t ih t sil`, and if triphone HMMs were used the sequence would be modelled as

```
sil sil-b+iy b-iy+t iy-t+ih t-ih+t ih-t+sil sil
```

Notice that the triphone contexts span word boundaries and the two instances of the phone `t` are represented by different HMMs because their contexts are different. This use of so-called *cross-word triphones* gives the best modelling accuracy but leads to complications in the decoder as discussed later. Simpler systems result from the use of *word-internal triphones* where the above example would become

```
sil b+iy b-iy+t iy-t ih+t ih-t sil
```

Here far fewer distinct models are needed simplifying both the parameter estimation problem and decoder design. However, the cost is an inability to model contextual effects at word boundaries and in fluent speech these are considerable.

The use of Gaussian mixture output distributions allows each state distribution to be modelled very accurately. However, when triphones are used they result in a system which has too many parameters to train. For example, a large vocabulary cross-word triphone system will typically need around 60,000 triphones[1]. In practice, around 10 mixture components gives good performance in LVR systems. Assuming that the covariances are all diagonal, then the HTK recogniser with its 39 element acoustic vectors would require around 790 parameters per state. Hence, 60,000 3-state triphones would have a total of 142 million parameters.

The problem of too many parameters and too little training data is absolutely crucial in the design of a statistical speech recogniser. Early systems dealt with the problem by tying all Gaussian components together to form a pool which was then shared amongst all HMM states. In these so-called tied-mixture systems, only the mixture component weights were state-specific and these could be smoothed by interpolating with context independent models[48, 16, 86]. Comparisons between discrete, tied-mixture and continuous density HMMs showed that tied-mixture were

[1] With 45 phones, there are $45^3 = 91125$ possible triphones but not all can occur due to the phonotactic constraints of the language

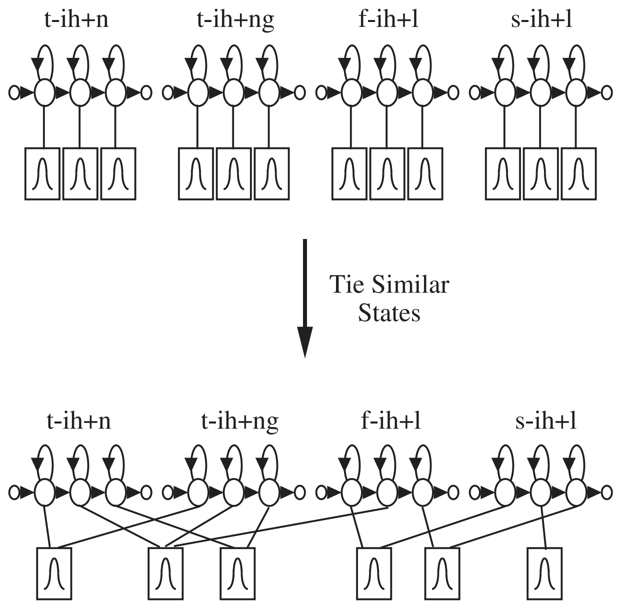

Figure 4: **State Tying.** In order to maximise the amount of data available to train each state whilst preserving discrimination ability, similar HMM states of the allophonic variants of each basic phone are tied together. The choice of which states to tie is made using a decision tree.

superior[47]. However, this followed mainly from the lack of good smoothing techniques for continous density systems. More recently smoothing based on parameter tying has become popular[104]. In particular, state-tying[49, 106] and phone-based component-tying [26] have been studied. Using these tying techniques with continuous density HMMs has led to substantial improvements in modelling accuracy.

The HTK recogniser uses state-tying. The idea is to tie together states which are acoustically indistinguishable. This allows all the data associated with each individual state to be pooled and thereby give more robust estimates for the parameters of the tied-state. This is illustrated in Fig 4. At the top of the figure, each triphone has its own private output distribution. After tying, several states share distributions.

In the HTK recogniser, the choice of which states to tie is made using *phonetic decision trees*[12, 57, 105]. This involves building a binary tree for each phone and state position. Each tree has a yes/no phonetic question such as "Is the left context a nasal?" at each node. Initially all states for a given phone state position are placed at the root node of a tree. Depending on each answer, the pool of states is successively split and this continues until the states have trickled down to leaf-nodes. All states in the same leaf node are then tied. For example, Fig 5 illustrates the case of tying the centre states of all triphones of the phone /aw/ (as in "out"). All of the states trickle down the tree and depending on the answer to the questions, they end up at one of the shaded terminal nodes. For example, in the illustrated case, the centre state of `s-aw+n` would join the second leaf node from the right since its right context is a central consonant, and its right context is a nasal but its left context is not a central stop.

The questions at each node are chosen to maximise the likelihood of the training data given the final set of state tyings. In practice, phonetic decision trees give compact good-quality state clusters which have sufficient associated data to robustly estimate mixture Gaussian output probablity functions. Furthermore, they can be used to synthesise a HMM for any possible context whether it appears in the training data or not, simply by descending the trees and using the state distributions associated with the terminating leaf nodes. Finally, phonetic decision trees can be used to include

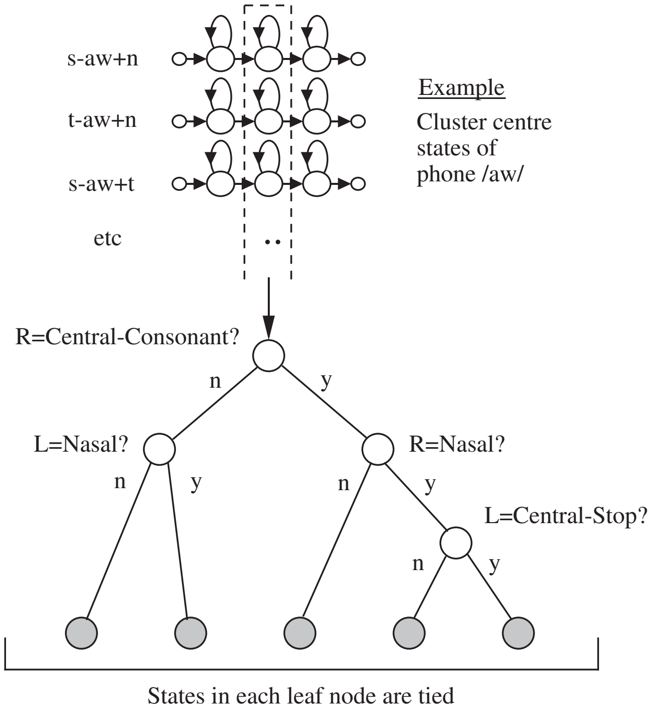

Figure 5: **Decision Tree Clustering.** Initially, all corresponding HMM states of all allophonic variants of each basic phone are tied to form a single pool. Phonetic questions are then used to partition the pool into subsets in a way which maximises the likelihood of the training data. The leaf nodes of each tree determine the sets of state tyings for each of the allophonic variants.

more than simple triphone contexts. For example, the HTK recogniser can use questions spanning ± 2 phones and can also take account of the presence of word boundaries.

5 Language Modelling

The purpose of the language model is to provide a mechanism for estimating the probability of some word w_k in an utterance given the preceding words $W_1^{k-1} = w_1 \ldots w_{k-1}$. A simple but effective way of doing this is to use N-grams in which it is assumed that w_k depends only on the preceding $n-1$ words, that is

$$P(w_k|W_1^{k-1}) = P(w_k|W_{k-n+1}^{k-1}) \tag{6}$$

N-grams simultaneously encode syntax, semantics and pragmatics and they concentrate on local dependencies. This makes them very effective for languages like English where word order is important and the strongest contextual effects tend to come from near neighbours. Furthermore, N-gram probability distributions can be computed directly from text data and hence there is no requirement to have explicit linguistic rules such as a formal grammar of the language.

In principle, N-grams can be estimated from simple frequency counts and stored in a look-up table. For example, for the case of *trigrams* ($N = 3$).

$$\hat{P}(w_k|w_{k-1}, w_{k-2}) = \frac{t(w_{k-2}, w_{k-1}, w_k)}{b(w_{k-2}, w_{k-1})} \tag{7}$$

where $t(a, b, c)$ is the number of times the trigram a,b,c appears in the training data and $b(a, b)$ is the number of times the bigram a,b appears. The problem, of course, is that for a vocabulary of V words, there are V^3 potential trigrams. Even for a very modest vocabulary of 10,000 words, this is a very large number. Thus, many trigrams will not appear in the training data and many others will only appear once or twice so that the estimate given by equation 7 will be very poor. In short, there is an acute data sparsity problem.

The solution to training data sparsity is to use a combination of *discounting* and *backing-off*[58, 77]. Discounting means that the trigram counts of the more frequently occurring trigrams are reduced and the resulting excess *probability mass* is redistributed amongst the less frequently occurring trigrams. Backing-off is applied when there are too few trigrams to form any estimate at all (e.g. just one or two occurrences in the training data). It involves replacing the trigram probability by a scaled bigram probability, that is

$$\hat{P}(w_k|w_{k-1}, w_{k-2}) = B(w_{k-1}, w_{k-2})P(w_k|w_{k-1}) \tag{8}$$

where B is a back-off function included to ensure that $\hat{P}(w_k|w_{k-1}, w_{k-2})$ is properly normalised.

Although robust estimation of trigram probabilities requires considerable care, the problems are soluble and good performance can be obtained. N-grams do have obvious deficiencies resulting from their inability to exploit long-range constraints such as subject-verb agreement[55]. As a consequence various alternatives have been studied such as tree-based models[11], trellis models[98], trigger models[63], history models[17] and variable N-grams [25]. However, in general, all of these attempts have yielded only small improvements at considerable computational cost. Thus to date, bigram and trigram language models dominate LVR systems.

6 Decoding

The preceding sections have described the main components of a large vocabulary system. In order to perform recognition using these components, the sequence of words $\hat{W}$ must be found which maximises equation 1. This is a search problem and its solution is the domain of the decoder.

As with all search problems, there are two main approaches: *depth-first* and *breadth-first*. In depth-first designs, the most promising hypothesis is pursued until the end of the speech is reached. Examples of depth-first decoders are *stack-decoders* and *A*-decoders*[53, 86, 59, 87]. More recently, a refinement of stack decoding based on an *envelope search* has been proposed[40].

In breadth-first designs, all hypotheses are pursued in parallel. Breadth-first decoding exploits Bellman's optimality principle and is often referred to as *Viterbi decoding*. LVR systems are complex and pruning of the search space is essential, this typically uses a process called *beam search*[96, 41]. The HTK decoder uses beam search and Viterbi decoding.

To understand the decoding problem, imagine that a branching tree network is constructed such that at the start there is a branch to every possible start word. All first words are then connected to all possible follow words and so on. This is illustrated in part (a) of Fig 6. Clearly this tree will be very large but if extended deep enough it would in principle represent all possible sequences. At first sight, this representation might seem very extravagant. In small vocabulary systems, it is usually sufficient to put all words in parallel and place a loop around them. This allows all possible word sequences to be represented in a compact way since every vocabulary word appears only once. Unfortunately, however, such an arrangement does not allow a trigram language model to be used since the available history is limited to 1 word. Furthermore, a single loop back prevents crossword triphones from being used. An explicit branching tree however allows both to be used in a straightforward manner[5].

Next let each word in this tree be replaced by the sequence of models representing its pronunciation. If there are multiple pronunciations then models can be joined in parallel within the word. Part (b) shows a fragment of the tree expanded into models. Finally, merge all identical phone models in identical contexts which have a common entry point as illustrated in part (c) of Fig 6. Notice here that the use of cross-word triphones significantly limits the amount of sharing of models possible.

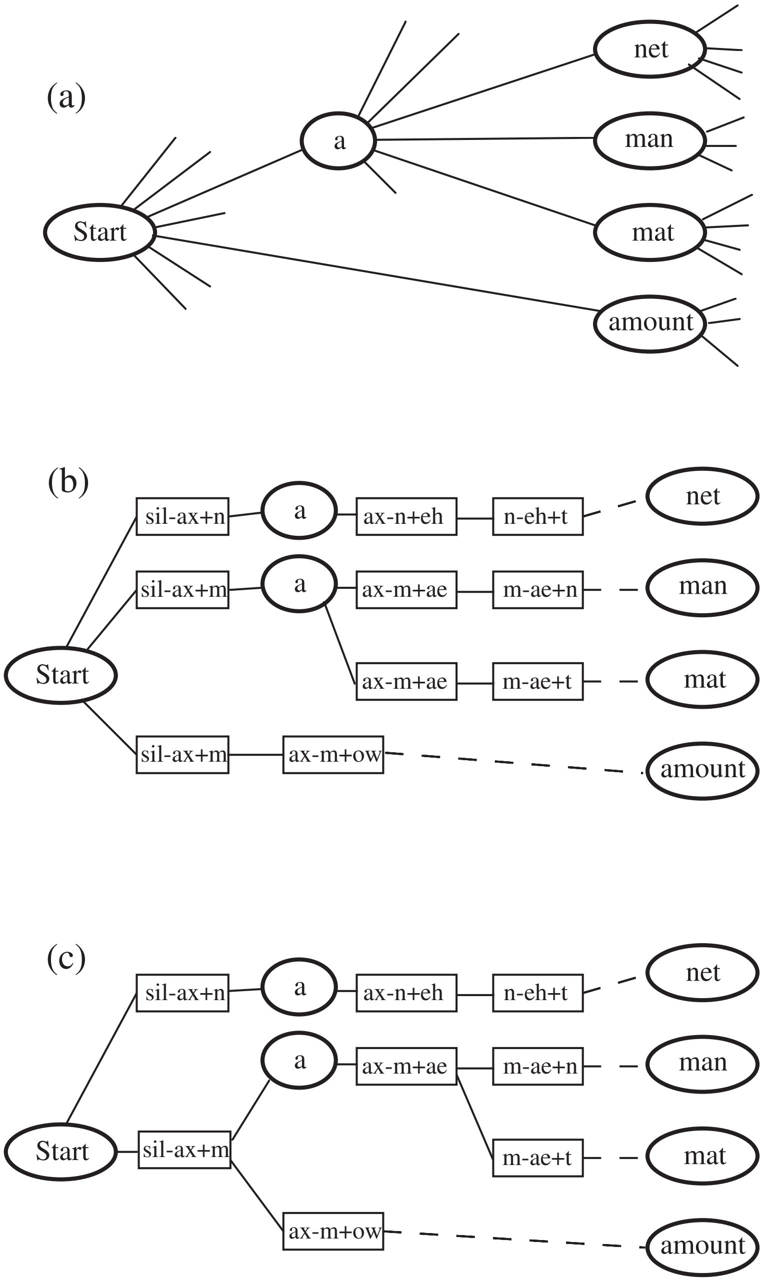

Figure 6: **Fragment of Decoder Network.** In principle, the decoder searches through a network representing all possible word sequences. In practice, only paths corresponding to the most likely word sequences are constructed. Part (a) shows the directed network of words which the recogniser is considering initially. Part (b) shows the same network decomposed into triphones. Note that in order to take account of cross-word context, the first /ax/ sound has to be replicated and the word a duplicated. Part (c) shows that tree-structuring the network can reduce the size of the network.

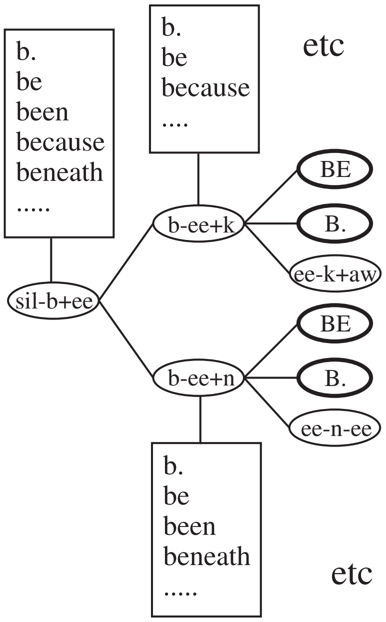

Figure 7: **Early Application of the Language Model.** In a tree-structured network, the language model probability cannot be applied until the end of the word is reached. However, this delayed application severely limits the effectiveness of the language model for pruning. To solve this problem, each phone model carries a list of all possible words that it can belong to. This allows the probability of the most likely word to be used as an estimate for the probability of the actual word.

The net result of the above is a branching tree of HMM state nodes connected by state transitions and word-end nodes connected by word transitions. Any path from the start node to some point in the tree can be evaluated by adding all the log state transition probabilities, all the log state output probabilities and the log language model probabilities. Such a path can be represented by a movable *token* placed in the node at the end of the path[103]. The token has a *score* which is the total log probability upto that point and a *history* which records the sequence of word-end nodes that the token has passed through. Any path can be extended by moving the token from its current node to an adjoining node and updating its score according to the current state transition probability, state output probability and the language model probability, if any.

The search problem can now be recast in the form of a token passing algorithm. Initially, a single token is placed in the start node of the tree. As each acoustic vector is input, every token is copied into all connecting nodes and the scores updated. If more than one token lands in a node only the best scoring token needs to be retained since, by Bellmans Optimality principle, all other tokens must lie on inferior paths. When all of the acoustic vectors have been processed, the word end nodes are scanned and the token with the highest score represents the most likely path and hence the most likely word sequence.

This basic token passing algorithm is guaranteed to find the best possible path but unfortunately, it would take too much time and space to compute. Hence, to make the algorithm tractable, pruning is employed. Every time frame, the best score in any token is noted and any token whose score lies more than a *beam-width* below this best score is destroyed. Since only the active tokens lying within the beam need to be kept in memory, only a fragment of the branching tree described above is ever needed at one time. As tokens move forward, new tree structure is created in front of them and old structure behind is destroyed. For this to work efficiently, it is crucial to prune tokens as soon as possible and here tree-structuring causes a problem since a side-effect of merging

Vocab Size	LM N-gram	Adapted	%Word Error
20k	3	No	10.5 †
65k	3	No	8.2
65k	4	No	7.9
65k	4	Yes	7.2 ‡

Table 1: HTK LVR Performance on ARPA CSR November 1994 Evaluation Data (†H1-C1 test, ‡H1-P0 test)

phone models is that the identity of each new word entered is not known until its end-node is reached. This is unfortunate since the language model provides a very powerful constraint which needs to be applied as soon as is practible in order to keep the number of active tokens as small as possible. The HTK decoder deals with this problem by associating a list of possible current words with every token (see Fig 7). As the token moves towards the end of the word, this list will shrink until eventually it contains just a single word. Tokens then receive a language model score equal to the most likely word in the current list. As this gets updated on every model transition, the tokens get pruned accordingly.

The dynamic network approach used in the HTK decoder results in a system which can exploit arbitrarily long span language models and HMM phone models that depend on both the previous and succeeding acoustic context. Furthermore, it can do this in a single-pass[81]. Most other Viterbi-based systems use a multiple-pass approach in which the first pass uses simple acoustic and language models and outputs a lattice of alternatives which are then rescored using more complex models in subsequent passes [6, 74, 90]. The problem with this is that an error in the first pass can never be recovered, hence large lattices must be used and any potential computational savings are lost. Multiple passes are useful for applying more complex language models but for maximum accuracy, the most accurate acoustic models available need to be applied as soon as possible.

7 Current State of LVR

The major benchmarks for assessing the performance of LVR systems are the US Advanced Research Project Agency (ARPA) CSR Evaluations. The last full evaluation of dictation-style large vocabulary recognition was the November 1994 evaluation[85, 61] in which the participating systems included AT&T Bell Laboratories [66, 69], BBN [78], Boston University [83], the CUED ABBOT group [46], the CUED HTK group [101, 100], IBM [8, 9], LIMSI-CNRS [38, 36], Philips [29], and SRI International [28].

The main focus of the evaluation was the so-called *hub test* H1 on which all participating sites evaluated their systems. This hub test was split into two main parts: H1-C1 in which the acoustic training data and a 20k word trigram language model trained on 227 million words of news text were fixed; and H1-P0 in which any acoustic or language model training data could be used. In H1-C1 each utterance had to be recognised independently whereas in H1-P0, each change of speaker was known so that unsupervised incremental adaptation could be used.

The HTK LV Recogniser had the lowest error rate of the systems tested in the November 1994 evaluation and it is therefore indicative of what can be achieved with current technology. Performance in terms of the percentage word error rate for a number of conditions (including H1-C1 and H1-P0) is shown in Table 7. As can be seen, the best performance achieved was 7.2%, that is, on average 7 words in every 100 were mis-transcribed. Although this figure is somewhat high, it is interesting to look at the error rates on a per speaker basis as shown in Figure 8 where the speakers have been ordered based on their performance. This figure suggests that, at least for part of the population, useable performance is achievable now. Conversely, it also shows that to

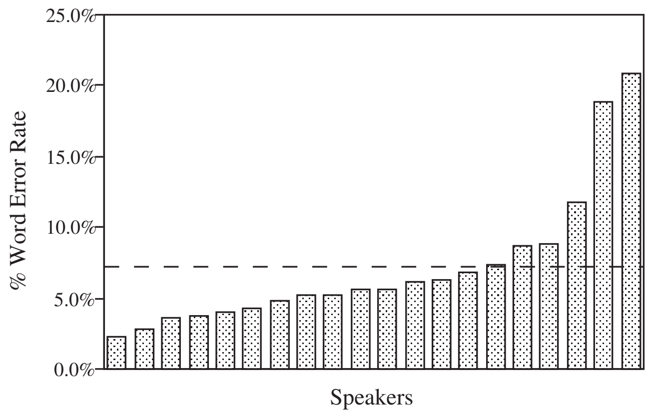

Figure 8: Word Error Rate plotted for each speaker using the HTK LVR (H1-P0) System on ARPA CSR November 1994 Evaluation Data (Dashed horizantal line shows the average 7.2% WER).

cover the majority of the population, better robustness and more effective adaptation is needed[2].

8 Current Issues

The performance levels described in the previous section were all obtained for read speech in quiet recording conditions with a known microphone and a known task domain. Furthermore, the majority of systems tested were operating at many times slower than real time. Thus, although large vocabulary continuous speech recognition appears to be feasible, the following issues will need to be resolved before widespread deployment of LVR technology is possible.

8.1 Speaker Adaptation

Speaker independence is highly desirable since it allows a system to be used *straight out of the box* and it allows systems to be built for which the speaker is not known in advance. However, in many applications, a speaker will either become a regular user or will input a reasonable quantity of speech at first use. In such cases it is natural to adapt the acoustic models to match that speaker. Adaptation can result in substantial reductions in error rate, particularly for atypical speakers, and its incorporation in future LVR systems will be essential.

Adaptation can be supervised or unsupervised, and it can be performed incrementally as the speaker is talking or off-line at the end of the session. Each of these different styles may be most appropriate for some particular application. However, unsupervised incremental adaptation is the least intrusive and most generally useful to the user. Current research is therefore particularly concerned with techniques which can yield worthwhile performance gains with very little adaptation data but which also asymptotically lead to speaker dependent performance once a large amount of data has been acquired.

The acoustic models in an LVR system comprise a very large number of parameters. Hence, adaptation involving a very small amount of new data must depend on some form of general transformation rather than a direct re-estimation of the parameters themselves. In VQ or tied-mixture systems, this can be achieved by code book mapping[95]. In CD systems, linear transformations can be applied to each Gaussian component. This can be a single global transform based on linear regression[52, 45] or canonical correlation[22]. As more data becomes available, Gaussians can be clustered and an individual transform determined for each cluster[65, 27]. Alternative approaches

[2] Some training of the speakers would also help!

include clustering[32], regression-based prediction[23], and data augmentation where the training data rather than the models is adapted[15]. Finally, when there is a reasonable amount of adaptation data classical MAP estimation can be used to update all of the acoustic parameters[64, 37].

8.2 Environmental Robustness

Robustness to background noise and channel variability is clearly an essential requirement for the widespread use of LVR technology. As explained above, LVR systems utilise acoustic feature vectors consisting of short term spectral estimates to which some form of amplitude compression has been applied such as a log operation.

The primary affect of additive noise is to shift the spectral means and shrink the variances. However, it is important to note that the addition of noise changes the whole distribution not just the means and variances[82]. Hence, a secondary effect of noise is to change the modelling assumptions. Channel variability is convolutive noise which after a log operation becomes a simple, but possibly time-varying, offset.

There are a variety of approaches to dealing with noise[33, 42]. At the front-end, the noise can be removed from the speech[19], noise-robust features can be used[70, 73], the noise can be masked[71] or the features can be mapped[76]. The problem with these approaches is that at best they can only exploit knowledge of the global statistics of the speech in order to remove the noise. Given that the acoustic models within a recogniser encode very detailed information about the speech, this is an unnecessary handicap. Alternatively, an attempt can be made to make the pattern-matching process itself robust to noise[93]. However, in LVR systems, there are a very large number of over-lapping acoustic classes and maintaining maximal discrimination is essential.

Hence, for LVR systems, methods which adapt the recogniser to handle the corrupted speech signal directly are most attractive. These include code book mapping for discrete and tied-mixture systems[2], cepstral mean compensation[72], state-based filtering[14], and parallel model combination (PMC)[97, 34, 35]. Note also that many of the techniques used for speaker adaptation are also capable of adapting to different noise environments.

In the large vocabulary area, noise-robustness remains a substantially unsolved problem. As the results in [85] show, without any form of compensation the performance of an LVR system will drop dramatically in noise. Compensation can limit this effect but cannot yet give immunity.

8.3 Task Independence

Whereas the acoustic models in an LVR system are relatively task-independent, the language model is typically trained on a large corpus of task specific material. This leads to systems which are inherently task dependent. For example, a language model trained on office correspondence will not work well on legal documents. Moving domain typically results in a lowering of accuracy due to the language model mis-match and an increase in the incidence of out-of-vocabulary (OOV) words.

In the long term, task dependence will be reduced through a general increase in language modelling capability. A recent trend in computational linguistics has been an increased interest in statistical techniques[21] and this should eventually lead to the incorporation of explicit grammatical knowledge into LVR systems. In the nearer term, solutions being studied include multiple-domain modelling and adaptation. In the former, individual LMs are trained on a range of possible domains and then combined as a mixture model[50]. Adaptive systems typically combine a static LM trained off-line with a dynamic cache of N-grams which are being continuously updated[62, 92].

The OOV problem has many facets[102] but increased robustness and the ability to simply add new words to an existing LM suggest that N-grams should be class-based rather than word-based. Using word-classes allows compact robust language models to be developed which can support very large vocabularies[20, 51, 60]. However, there are a number of problems to solve in using class-based N-grams. Firstly, the method of choosing the classes is crucial since ideally words should only be grouped into a class if there is little or no resulting increase in perplexity. This implies that efficient data driven methods are needed. Secondly, since words can have multiple senses, it would be natural to allow the same word to appear in multiple classes. However, this adds considerable complexity to the decoding problem.

8.4 Spontaneous Speech

Much of the recent research effort in the LVR area has been directed at transcribing read speech. However, the ability to transcribe spontaneous casual speech is also an important requirement and this is now receiving increased attention.

Current experimental evidence, particularly using the Switchboard Database, suggests that the error rate of current systems increases substantially when applied to spontaneous speech[107]. The reasons for this are still unclear but probably stem from a number of factors including poor articulation, increased coarticulation, highly variable speaking rate, and various types of disfluency such as hesititations, false-starts and corrections.

Some of the linguistic aspects of spontaneous speech have been studied in some detail[84] and a number of specific issues have been addressed such as identifying repairs in speech[43], modeling non-speech interjections [94], detecting hesitations [79] and modelling the timing patterns of disfluent speech[80]. However much of this work is at a very early stage. Practical implementations to date have been focussed on small to medium vocabulary dialogue systems (eg [99]) and little has been done in the large vocabulary transcription area.

8.5 Real Time Operation

In addition to obtaining an acceptable level of recognition accuracy, computationally efficient implementation is needed in order to exploit LVR technology. As explained in section 6, recognition in LVR systems involves searching a very large space of hypotheses. In continuous density systems, the computational cost of this search will be divided between building and searching the network structure and evaluating Gaussian densities.

The search space space can be reduced by making approximations which allow alternative paths to be merged, for example, by approximating the language model[75] or by limiting the number of new words through some form of look-ahead based on a fast-match pre-selection of possible followers[39, 7]. The evaluation of state output probabilities can be reduced by using vector-quantisation to pre-select those Gaussians which will give sufficiently high likelihoods[18]. Also, careful coding can make a substantial difference on modern RISC-based processor architectures where efficient use of the cache can make a substantial difference to throughput[1].

9 Conclusions

This article has reviewed the main components of a speaker-independent continuous speech large vocabulary recognition system and briefly described the state-of-the-art. Whilst it is clear that much more needs to be done before robust, general-purpose LVR is ubiquitous, the technology is nevertheless on the threshold of usefulness for practical applications. Given a reasonably controlled environment and a well-defined task domain, the technology is useable now. By the end of 1996, off-the-shelf LVR systems will start to appear running in real-time on high-end PC class machines. LVR-based services will also appear in the form of remote servers in public telecom systems adding telephone-based transcription capability to information and personal management services. Finally, LVR systems will facilitate multi-media information retrieval by allowing video sound-tracks to be transcribed and then searched for key-words and phrases.

10 Acknowledgements

Many people have contributed to the HTK LVR System, but the contributions of Phil Woodland, Julian Odell and Valtcho Valtchev deserve particular mention. Also, the financial support of the UK Engineering and Physical Sciences Research Council (grant refs GR/J10204 and GR/K25380) is gratefully acknowledged.

References

[1] Alleva FA. *Search Organisation for Large Vocabulary Continuous Speech Recognition.* Proc NATO Workshop, 1990

[2] Anastasakos A, Kubala F, Makhoul J, Schwartz R. *Adaptation to New Microphones Using Tied-Mixture Normalisation.* Proc Human Language Technology Workshop, pp325-329, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[3] Applebaum T, Hanson B. *Regression Features for Recognition of Speech in Noise.* Proc ICASSP, S14.26, Toronto, 1991

[4] Atal BS. *Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification.* J Acoustical Soc Am, Vol 55, No 6, pp1304-1312, 1974

[5] Aubert X, Ney H. *Large Vocabulary Continuous Speech Recognition Using Word Graphs.* Proc ICASSP, Vol 1, pp49-52, Detroit, 1995

[6] Austin S, Schwartz R. *The Forward-Backward Search Algorithm for Continuous Speech Recognition.* Proc ICASSP, S10.3, Toronto, 1991

[7] Bahl L, Gopalakrishnan PS, Kanevsky DS, Nahamoo D. *A Fast Admissible Method for Identifying a Short List of Candidate Words.* Computer Speech and Language, Vol 6, No 3, pp215-224, 1992

[8] Bahl LR, Balakrishnan-Aiyer S, Bellegarda JR, Franz M, Gopalakrisnan PS, Nahamoo D, Novak M, Padmanabhan M, Picheny MA, Roukos S. *Performance of the IBM Large Vocabulary Continuous Speech Recognition System on the ARPA Wall Street Journal Task.* Proc ICASSP, Vol 1, pp41-44, Detroit, 1995

[9] Bahl LR, Balakrishnan-Aiyer S, Franz M, Gopalakrishnan PS, Gopinath R, Novak M, Padmanabhan M, Roukos S. *The IBM Large Vocabulary Continuous Speech Recognition System for the ARPA NAB News Task.* Proc Spoken Language Technology Workshop, pp121-126, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[10] Bahl LR, Brown PF, de Souza PV, Mercer RL. *Speech recognition with continuous parameters hidden Markov models.* Computer Speech and Language, Vol 2, No 3/4, pp219-234, 1987

[11] Bahl LR, Brown PF, de Souza PV, Mercer RL. *A Tree-Based Statistical Language Model for Natural Language Speech Recognition.* IEEE Trans ASSP, Vol 37, No 7, 1989

[12] Bahl LR, de Souza PV, Gopalakrishnan PS, Nahamoo D, Picheny MA. *Context Dependent Modeling of Phones in Continuous Speech Using Decision Trees.* Proc DARPA Speech and Natural Language Processing Workshop, pp264-270, Pacific Grove, Calif, Feb, 1991

[13] Baker JK. *The Dragon System - an Overview.* IEEE Trans ASSP, Vol 23, No 1, pp24-29, 1975

[14] Beattie VL, Young SJ. *Hidden Markov Model State-Based Cepstral Noise Compensation.* Proc ICSLP, pp519-522, Banff, Canada, Oct, 1992

[15] Bellegarda JR, de Souza PV, Nahamoo D, Padmanabhan M, Picheny MA, Bahl LR. *Experiments using Data Augmentation for Speaker Adaptation.* Proc ICASSP, Vol 1, pp692-695, Detroit, 1995

[16] Bellegarda JR, Nahamoo D. *Tied Mixture Continuous Parameter Modeling for Speech Recognition.* IEEE Trans ASSP Vol 38, No 12, pp2033-2045, 1990

[17] Black E, Jelinek F, Lafferty J, Magerman DM, Mercer R, Roukos S. *Towards History-based Grammars: Using Richer Models for Probabilistic Parsing.* Proc DARPA, Spoken Language Systems Workshop, Feb, 1992

[18] Bocchieri E. *A Study of the Beam Search Algorithm for Large Vocabulary Continuous Speech Recognition and Methods for Improved Efficiency.* Proc Eurospeech, p1521-1524, Berlin, 1993

[19] Boll S. *Suppression of acoustic noise in speech using spectral subtraction.* IEEE Trans ASSP, Vol 27, pp113-120, 1979

[20] Brown PF, Della Pietra VJ, de Souza PV, Lai JC, Mercer RL. *Class-based n-gram Models of Natural Language.* Computational Linguistics, Vol 18, No 4, pp467-479, 1992

[21] Charniak E. *Statistical Language Learning.* Bradford MIT Press, 1993

[22] Choukri K, Chollet G. *Adaptation of automatic speech recognisers to new speakers using canonical correlation analysis techniques.* Computer Speech Language, Vol 1, No 2, pp95-107, 1986

[23] Cox S. *Predictive Speaker Adaptation in Speech Recognition.* Computer Speech Language, Vol 9, No 1, pp1-18, 1995

[24] Davis SB, Mermelstein P. *Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences.* IEEE Trans ASSP, Vol 28, No 4, pp357-366, 1980

[25] Deligne S, Bimbot F. *Language Modeling by Variable Length Sequences: Theoretical Formulation and Evaluation of Multigrams.* Proc ICASSP, Vol 1, pp169-172, Detroit, 1995

[26] Digalakis V, Monaco P, Murveit H. *Genones: Generalised Mixture Tying in Continuous Speech HMM-based Speech Recognisers.* IEEE Trans Speech and Audio Processing, to appear, 1995

[27] Digalakis V, Rtischev D, Neumeyer L. *Fast Speaker Adaptation using Constrained Estimation of Gaussian Mixtures.* IEEE Trans on Speech and Audio Processing, to appear, 1995

[28] Digilakis V, Weintraub M, Sankar Ananth, Franco H. *Continuous Speech Dictation on ARPA's North American Business News Domain.* Proc Spoken Language Technology Workshop, pp88-93, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[29] Dugast C, Kneser R, Aubert X, Ortmanns S, Beulen K, Ney H. *Continuous Speech Recognition Tests and Results for the NAB'94 Corpus.* Proc Spoken Language Technology Workshop, pp156-161, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[30] Furui S. *Cepstral Analysis Technique for Automatic Speaker Verification.* IEEE Trans ASSP, Vol 29, No 2, pp254-272, 1981

[31] Furui S. *Speaker-Independent Isolated Word Recognition using Dynamic Features of Speech Spectrum.* IEEE Trans ASSP, Vol 34, No 1, pp52-59, 1986

[32] Furui S. *Unsupervised Speaker Adaptation Method Based on Hierarchical Clustering.* Proc ICASSP, pp286-289, Glasgow, May, 1989

[33] Furui S. *Toward Robust Speech Recognition under Adverse Conditions.* Proc ESCA Workshop on Speech Processing in Adverse Conditions, pp31-42, Cannes-Mandelieu, November, 1992

[34] Gales MJF, Young SJ. *Cepstral Parameter Compensation for HMM Recognition in Noise.* Speech Communication, Vol 12, No 3, pp231-239, 1993

[35] Gales MJF, Young SJ. *Robust Speech Recognition in Additive and Convolutional Noise using Parallel Model Combination.* Computer Speech and Language, Vol 9, pp289-308, 1995

[36] Gauvain J-L, Lamel L, Adda-Decker M. *Developments in Large Vocabulary Dictation: The LIMSI Nov94 NAB System.* Proc Spoken Language Technology Workshop, pp131-138, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[37] Gauvain J-L, Lee C-H. *Maximum a Posteriori Estimation of Multivariate Gaussian Mixture Observations of Markov Chains.* IEEE Trans Speech and Audio Processing, Vol 2, No 2, pp291-298, April, 1994

[38] Gauvain JL, Lamel LF, Adda G, Adda-Decker M. *The LIMSI Continuous Speech Dictation System.* Proc Human Language Technology Workshop, pp319-324, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[39] Gillick L, Roth R. *A Rapid Match Algorithm for Continuous Speech Recognition.* Proc DARPA Speech and Natural Language Workshop, Hidden Valley, Pennsylvania, pp170-172, June, 1990

[40] Gopalakrisnan PS, Bahl LR, Mercer RL. *A Tree Search Strategy for Large Vocabulary Continuous Speech Recognition.* Proc ICASSP, Vol 1, pp572-575, Detroit, 1995

[41] Haeb-Umbach R, Ney H. *Improvements in Time-Synchronous Beam Search for 10000-Word Continuous Speech Recognition.* IEEE Trans Speech and Audio Processing, Vol 2, pp353-356, 1994

[42] Haton J-P. *Automatic Recognition of Speech in Adverse Conditions: a Review.* Reviewed for IEEE Speech and Audio, SAP 276, 1994

[43] Heeman PA, Allen J. *Tagging Speech Repairs.* Proc Human Language Technology Workshop, pp187-192, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[44] Hermansky H. *Perceptual Linear Predictive (PLP) Analysis of Speech.* J Acoustical Soc America, Vol 87, No 4, 1990

[45] Hewett AJ, Holmes G, Young SJ. *Dynamic Speaker Adaptation in Speaker-Independent Word Recognition.* Proc IOA Autumn Conf, Vol 8, Pt 7, pp 275-282, 1986

[46] Hochberg M, Renals S, Robinson A. *ABBOT: The CUED Hybrid Connectionist-HMM Large Vocabulary Recognition System.* Proc Spoken Language Technology Workshop, pp170-178, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[47] Huang XD, Hon HW, Hwang MY, Lee KF. *A Comparative Study of Discrete, Semi-Continuous and Continuous Hidden Markov Models.* Computer Speech and Language, Vol 7, No 4, pp 359-368, 1993

[48] Huang XD, Jack MA. *Semi-continuous hidden Markov models for Speech Signals.* Computer Speech and Language, Vol 3, No 3, pp239-252, 1989

[49] Hwang M-Y, Huang X. *Shared Distribution Hidden Markov Models for Speech Recognition.* IEEE Trans Speech and Audio Processing, Vol 1, No 4, pp414-420, 1993

[50] Iyer R, Ostendorf M, Rohlicek JR. *Language Modeling with Sentence Level Mixtures.* Proc Human Language Technology Workshop, pp82-87, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[51] Jardino M, Adda G. *Automatic Word Classification using Simulated Annealing Language Modelling.* Proc ICASSP'93, Vol 2, pp. 41-44, Minneapolis, 1993

[52] Jaschul J. *Speaker adaptation by a linear transformation with optimised parameters.* Proc ICASSP, pp1657-1660, Paris, 1982

[53] Jelinek F. *A Fast Sequential Decoding Algorithm Using a Stack.* IBM J Research and Dev, Vol 13, Nov, 1969

[54] Jelinek F. *Continuous Speech Recognition by Statistical Methods.* Proc IEEE, Vol 64, 4, pp532-556, 1976

[55] Jelinek F. *Up From Trigrams: the Struggle for Improved Language Models.* Proc Eurospeech, pp1037-1040, Genoa, 1991

[56] Juang B-H. *Maximum-Likelihood Estimation for Mixture Multivariate Stochastic Observations of Markov Chains.* ATT&T Technical J, Vol 64, No 6, pp1235-1249, 1985

[57] Kannan A, Ostendorf M, Rohlicek JR. *Maximum Likelihood Clustering of Gaussians for Speech Recognition.* IEEE Trans on Speech and Audio Processing, Vol 2, No 3, pp453-455, 1994

[58] Katz SM. *Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recogniser.* IEEE Trans ASSP, Vol 35, No 3, pp400-401, 1987

[59] Kenny P. *A* Admissible Heuristics for Rapid Lexical Access.* Proc ICASSP, S10.1, Toronto, 1991

[60] Kneser R, Ney H. *Improved Clustering Techniques for Class-based Statistical Language Modelling.* Proc Eurospeech, pp 973-976, Berlin, 1993

[61] Kubala F. *Design of the 1994 CSR Benchmark Tests.* Proc Spoken Language Technology Workshop, pp41-46, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[62] Kuhn R, De Mori R. *A Cache-Based Natural Language Model for Speech Recognition.* IEEE Trans Pattern Analysis and Machine Intelligence, Vol 12, No 6, pp570-583, 1990

[63] Lau R, Rosenfeld R, Roukos S. *Trigger-based Language Models: a Maximum Entropy Approach.* Proc ICASSP'93, Vol 2, pp45-48, Minneapolis, 1993

[64] Lee C-H, Lin C-H, Juang B-H. *A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models.* IEEE Trans ASSP, Vol 39, No 4, pp, 1991

[65] Leggetter CJ, Woodland PC. *Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models.* Computer Speech and Language, Vol 9, No 2, pp 171-185, 1995

[66] Levinson SE, Ljolje A, Miller LG. *Large Vocabulary Speech Recognition Using a Hidden Markov Model for Acoustic/Phonetic Classification.* Proc ICASSP, S11.5, p505-508, 1988

[67] Liporace LA. *Maximum-Likelihood Estimation for Multivariate Observations of Markov Sources.* IEEE Trans Information Th, Vol IT-28, No 5, pp729-734, 1982

[68] Ljolje A. *The Importance of Cepstral Parameter Correlations in Speech Recognition.* Computer Speech and Language, Vol 8, No 3, pp 223-232, 1994

[69] Lolje A, Riley M, Hindle D, Pereira F. *The AT&T 60,000 Word Speech-To-Text System.* Proc Spoken Language Technology Workshop, pp162-165, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[70] Mansour D, Juang B-H. *The Short-Time Modified Coherence Representation and Noisy Speech Recognition.* IEEE Trans ASSP, Vol 37, No 6, June, 1989

[71] Mellor BA, Varga AP. *Noise Masking in the MFCC Domain for the Recognition of Speech in Background Noise.* Proc Inst Acoustics Autumn Conf on Speech and Hearing, Vol 14, Part 6, pp361-368, 1992

[72] Moreno PJ, Raj B, Gouvea E, Stern RM. *Multivariate-Gaussian-Based Cepstral Normalisation for Robust Speech Recognition.* Proc ICASSP, Vol 1, pp137-140, Detroit, 1995

[73] Morgan N, Hermansky H. *RASTA Extensions: Robustness to Additive and Convolutional Noise.* Proc ESCA Workshop on Speech Processing in Adverse Conditions, pp115-118, Cannes-Mandelieu, November, 1992

[74] Murveit H, Butzberger J, Digalakis V, Weintraub M. *Large Vocabulary Dictation using SRI's Decipher Speech Recognition System: Progressive Search Techniques.* Proc ICASSP, pp319-322, Minneapolis, 1993

[75] Murveit H, Monaco P, Digilakis V, Butzberger J. *Techniques to Achieve an Accurate Real-Time Large Vocabulary Speech Recognition System.* Proc Human Language Technology Workshop, pp393-398, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[76] Neumeyer L, Weintraub M. *Robust Speech Recognition in Noise using Adaptation and Mapping Techniques.* Proc Spoken Language Technology Workshop, pp100-103, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[77] Ney H, Essen U, Kneser R. *On Structuring Probabilistic Dependences in Stochastic Language Modelling.* Computer Speech and Language, Vol 8, No 1, pp 1-38, 1994

[78] Nguyen L, Anastasakos Tasos, Kubala F, LaPre C, Makhoul J, Schwartz R, Yuan N, Zavaliagkos G, Zhao Y. *The 1994 BBN BYBLOS Speech Recognition System.* Proc Spoken Language Technology Workshop, pp77-81, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[79] O'Shaughnessy D. *Automatic Recognition of Hesitations in Spontaneous Speech.* Proc ICASSP, Vol 1, pp593-596, San Francisco, 1992

[80] O'Shaughnessy D. *Timing Patterns in Fluent and Disfluent Spontaneous Speech.* Proc ICASSP, Vol 1, pp593-596, Detroit, 1995

[81] Odell JJ, Valtchev V, Woodland PC, Young SJ,. *A One-Pass Decoder Design for Large Vocabulary Recognition.* Proc Human Language Technology Workshop, pp405-410, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[82] Openshaw JP, Mason JS. *On the Limitations of Cepstral Features in Noise.* Proc ICASSP, pp49-52, Adelaide, Australia, April, 1994

[83] Ostendorf M, Richardson F, Iyer R, Kannan A, Ronen O, Bates R. *The 1994 BU NAB News Benchmark System.* Proc Spoken Language Technology Workshop, pp139-142, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[84] Oviatt S. *Predicting Spoken Disfluencies during Human-Computer Interaction.* Computer Speech Language, Vol 9, No 1, pp19-36, 1995

[85] Pallett DS, Fiscus JG, Fisher WM, Garofolo JS, Lund BS, Martin A, Przybocki. *The 1994 Benchmark Tests for the ARPA Spoken Language Program.* Proc Spoken Language Technology Workshop, pp5-38, Morgan Kaufmann Publishers Inc, Austin, Texas, Jan, 1995

[86] Paul DB. *The Lincoln Tied Mixture HMM Continuous Speech Recogniser.* Proc DARPA Speech and Natural Language Workshop, pp332-336, Hidden Valley, Pennsylvania, June, 1990

[87] Paul DB. *Algorithms for an Optimal A* Search and Linearizing the Search in the Stack Decoder.* Proc ICASSP, pp693-996, Toronto, 1991

[88] Rabiner LR, Juang B-H, Levinson SE, Sondhi MM. *Recognition of Isolated Digits Using HMMs with Continuous Mixture Densities.* ATT&T Technical J, Vol 64, No 6, pp1211-1233, 1985

[89] Rabiner LR, Schafer RW. *Digital Processing of Speech Signals.* Prentice-Hall, Englewood Cliffs, NJ, 1978

[90] Richardson F, Ostendorf M, Rohlicek JR. *Lattice-Based Search Strategies for Large Vocabulary Recognition.* Proc ICASSP, Vol 1, pp576-579, Detroit, 1995

[91] Robinson AJ, Holdsworth J, Patterson R, Fallside F. *A Comparison of Preprocessors for the Cambridge Recurrent Error Propagation Network Speech Recognition System.* Proc International Conference on Spoken Language Processing, Kobe, Japan, November, 1990

[92] Rosenfeld R. *A Hybrid Approach to Adaptive Statistical Modeling.* Proc Human Language Technology Workshop, pp76-81, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[93] Sankar A, Lee C-H. *Robust Speech Recognition Based on Stochastic Matching.* Proc ICASSP, Vol 1, pp121-124, Detroit, 1995

[94] Schultz T, Rogina I. *Acoustic and Language Modeling of Human and Nonhuman Noises for Human-Human Spontaneous Speech Recognition.* Proc ICASSP, Vol 1, pp293-296, Detroit, 1995

[95] Shikano K, Lee K, Reddy R. *Speaker adaptation through vector quantisation.* Proc ICASSP, pp2643-2646, 1986

[96] Steinbiss V. *Sentence-Hypotheses Generation in a Continuous-Speech Recognition System.* Proc European Conf on Speech Communication and Technology, Vol 2, pp51-54, Paris, 1989

[97] Varga AP, Moore RK. *Hidden Markov Model Decomposition of Speech and Noise.* Proc ICASSP, Albuquerque, 1990

[98] Waegner NP, Young SJ. *A Trellis-based Language Model for Speech Recognition.* Proc ICSLP, pp245-248, Banff, Canada, Oct, 1992

[99] Ward W. *Understanding Spontaneous Speech: the Phoenix System.* Proc ICASSP 91, pp 365-368, Toronto, Canada, May, 1991

[100] Woodland PC, Leggetter CJ, Odell J, Valtchev V, Young SJ. *The 1994 HTK Large Vocabulary Speech Recognition System.* Proc ICASSP, Vol 1, pp73-76, Detroit, 1995

[101] Woodland PC, Young SJ. *The HTK Continuous Speech Recogniser.* Proc Eurospeech '93, pp2207-2219, Berlin, Sept, 1993

[102] Young SR. *Learning New Words from Spontaneous Speech: A Project Summary.* CMU Tech Report, CMU-CS-93-223, July, 1993

[103] Young SJ, Russell NH, Thornton JHS. *Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems.* Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Dept, 1989

[104] Young SJ. *The General Use of Tying in Phoneme-Based HMM Speech Recognisers.* Proc ICASSP, Vol 1, pp569-572, San Francisco, March, 1992

[105] Young SJ, Odell JJ, Woodland PC. *Tree-Based State Tying for High Accuracy Acoustic Modelling.* Proc Human Language Technology Workshop, pp307-312, Plainsboro NJ, Morgan Kaufman Publishers Inc, March, 1994

[106] Young SJ, Woodland PC. *State Clustering in HMM-based Continuous Speech Recognition.* Computer Speech and Language, Vol 8, No 4, pp369-384, 1994

[107] Young SJ, Woodland PC, Byrne WJ. *Spontaneous Speech Recognition for the Credit Card Corpus using the HTK Toolkit.* IEEE Trans Audio and Speech Processing, Vol 2, No 4, pp615-621, 1994