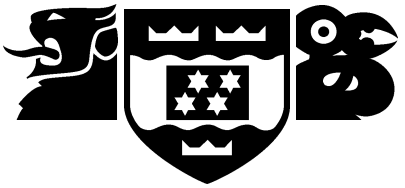
VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

Learning Models for English Speech Recognition

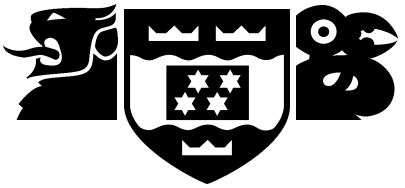
Huayang Xie, Peter Andreae, Mengjie Zhang, Paul Warren

Technical Report CS-TR-03/12
September 2003

School of Mathematical and Computing Sciences
Victoria University
PO Box 600, Wellington
New Zealand

Tel: +64 4 463 5341
Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

VICTORIA UNIVERSITY OF WELLINGTON
Te Whare Wananga o te Upoko o te Ika a Maui



School of Mathematical and Computing Sciences
Computer Science

PO Box 600
Wellington
New Zealand

Tel: +64 4 463 5341, Fax: +64 4 463 5045
Email: Tech.Reports@mcs.vuw.ac.nz
http://www.mcs.vuw.ac.nz/research

Learning Models for English Speech Recognition

Huayang Xie, Peter Andreae, Mengjie Zhang, Paul Warren

Technical Report CS-TR-03/12
September 2003

Abstract

This paper reports on an experiment to determine the optimal parameters for a speech recogniser that is part of a computer aided instruction system for assisting learners of English as a Second Language. The recogniser uses Hidden Markov Model (HMM) technology. To find the best choice of parameters for the recogniser, an exhaustive experiment with 2370 combinations of parameters was performed on a data set of 1119 different English utterances produced by 6 female adults. A server-client computer network was used to carry out the experiment. The experimental results give a clear preference for certain sets of parameters. An analysis of the results also identified some of the causes of errors and the paper proposes two approaches to reduce these errors.

Keywords Signal processing, HMM design, speech encoding, frame period, window size

Author Information

All the authors are at Victoria University of Wellington, New Zealand. Huayang Xie is a Masters student in Computer Science, Peter Andreae and Mengjie Zhang are academic staff members in computer science in the School of Mathematical and Computing Sciences, and Paul Warren is an academic staff member in the School of Linguistics and Applied Language Study.

Learning Models for English Speech Recognition

Huayang Xie*, Peter Andreae*, Mengjie Zhang*, Paul Warren+

*School of Mathematical and Computing Sciences
+School of Linguistics and Applied Language Studies
Victoria University of Wellington,
P. O. Box 600, Wellington, New Zealand,
Email: {Huayang.Xie, Peter.Andreae, Mengjie.Zhang, Paul.Warren}@vuw.ac.nz

Abstract

This paper reports on an experiment to determine the optimal parameters for a speech recogniser that is part of a computer aided instruction system for assisting learners of English as a Second Language. The recogniser uses Hidden Markov Model (HMM) technology. To find the best choice of parameters for the recogniser, an exhaustive experiment with 2370 combinations of parameters was performed on a data set of 1119 different English utterances produced by 6 female adults. A server-client computer network was used to carry out the experiment. The experimental results give a clear preference for certain sets of parameters. An analysis of the results also identified some of the causes of errors and the paper proposes two approaches to reduce these errors.

Keywords: Signal processing, HMM design, speech encoding, frame period, window size

1 Introduction

As English becomes more and more important as a communication tool for people from all countries, there is an ever increasing demand for good quality teaching of English as a Second Language (ESL). New Zealand is one of the destinations for foreign students wanting to learn English from English speaking teachers, and for political reasons is often perceived as a desirable destination. Learning English well requires lots of practice and a great deal of individualised feedback to identify and correct errors in students' use of English. Providing this individualised feedback from ESL teachers is very expensive, and the shortage of ESL teachers means that there is increasing demand for computer software that can provide useful individualised feedback to students on all aspects of their English.

The ESL Software Tools research group at Victoria University of Wellington is developing a software system to provide individualised feedback to ESL students on prosodic aspects of their speech production, focusing particularly on the stress and rhythm of the speech. The overall design of the system involves a pedagogic component that engages in simple dialogues with the student, and a speech analyser that analyses the student's speech, identifying the stress pattern in the speech and comparing it with a target pattern in order to provide useful feedback on stress and rhythm errors.

The first stage of the speech analyser must perform phoneme level speech recognition on the student's speech to identify the start and end times of all the segmental units. Later stages must further analyse the speech to identify which elements would be perceived as stressed, and to match the rhythm pattern to the target pattern. These later stages depend critically on the accuracy of the phoneme-level speech recognition, particularly on the reliable recognition of the vowel phonemes and the times of their boundaries.

Our system uses a Hidden Markov Model (HMM) speech recogniser with HMMs for each phoneme, trained from a hand-annotated speech data set. The use of the speech recogniser within our system is rather different from the normal speech recognition context — we know the sentence that the student is trying to say, and the goal is to identify errors in the stress and rhythm patterns. The system uses a dictionary to translate the sentence into the expected phoneme sequences. (There are multiple sequences because the dictionary includes alternative pronunciations.) The recognition system uses the phoneme level HMMs to align the best of the expected phoneme sequences with the speech signal. This is referred to as "forced alignment". Because of this different goal, it is not necessarily the case that the HMM design and training parameters commonly used for speech recognition are appropriate for our task.

This paper reports on a set of experiments for identifying the optimal choice of parameters for the design and training of these HMMs, given the context that the speech recogniser will be used in. The experiments sought to identify the appropriate size of the statistical models in the HMM states and the best

This research is supported by the New Economy Research Funds, New Zealand.

combinations of features for encoding the speech input to the HMMs. An analysis of the results and the recognition errors also suggests some techniques for improving the performance in the future.

The paper is organised as follows: section 2 describes the design of the HMMs; section 3 describes the speech encoding process and the choice of important parameters and features; section 4 describes the exhaustive experiment design, method and configurations; section 5 presents the results; section 6 presents two techniques for further development; section 7 concludes the paper.

2 HMM Design

A phoneme-level HMM is a description of segments of input speech signals that correspond to a particular phoneme. The HMM consists of a network of states where each state describes subsegments of input speech signals that correspond to a different section of a phoneme (for example, the initial component of the phoneme).

Before the HMMs for each phoneme can be trained, their architecture needs to be specified. This involves three key decisions: the number of states needed in each phoneme level HMM, the connection mode of the phoneme states, and the size of the mixture-of-Gaussian models in each state of the HMMs.

2.1 Architecture of the HMMs

The dynamic nature of speech entails that with fewer states an HMM will be a less precise model because each state must describe a larger section of a phoneme. It will also be less accurate at identifying the start and end times of the phoneme. With more states, the HMM may have greater accuracy, but will require a higher computation cost when recognising the speech input. It will also require more training data in order to determine all the values in the state descriptions.

To balance the need for accuracy against the computation time and size of training data, we chose to follow standard practice, using a three state model for each phoneme HMM. The first state describes the transition from the previous phoneme into the current phoneme, the second state describes the middle section of the phoneme, and the third state describes the transition out of the current phoneme to the next phoneme.

The states can be connected in many ways. We chose the commonly used connection mode of a chained connection (Young, Evermann, Kershaw, Moore, Odell, Ollason, Valtchev & Woodland 2002), where each state is connected to itself (since each state may cover a number of samples from the input signal) and to the next state. This mode does not allow connections that skip over a state or connect back to a previous state. An example of this connection mode is shown in figure 1.

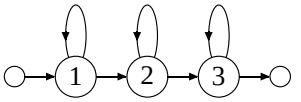


Figure 1: A chained connection mode HMM. (Adapted from (Young et al. 2002))

In addition to the phonemes, there are also a number of silences and short pauses in each speech signal. `/sil/` indicates a measurable silent pause in the speech file before or after the utterance itself, while `/sp/` indicates a short pause within the utterance. Because silences and pauses do not have the same regular structure as phonemes, we allowed more flexible structures for the silence and short pause HMMs: we used a modified three-state HMM with backward and forward skipping connections to model the silences and a tied one-state connection to model the short pauses, as shown in figure 2.

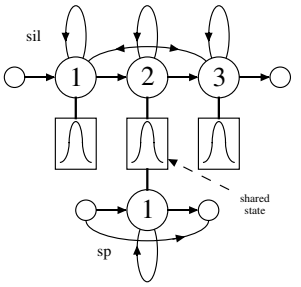


Figure 2: HMMs for silences and short pauses. (Adapted from (Young et al. 2002))

2.2 Stochastic Models: Mixture-of-Gaussians

An HMM state describes segments of speech signals using Gaussian models of each of the features used to encode the signal. If all speakers always pronounced a given phoneme in very similar ways, then there would be little variability in the signal and simple Gaussian models (mean and variance) would be sufficient. However, there is considerable variability, and a mixture-of-Gaussians model may provide a more accurate description. The design issue is to determine an appropriate number of Gaussians in the mixture-of-Gaussian models. The greater the size of the mixture-of-Gaussian model, the more training data is required to learn the parameters of the model. We explored a range of possible sizes of the models from 1 to 16.

3 Input Representation: Speech Encoding

The input speech signal consists of a time sequence of values of the raw analog speech data, sampled at

16KHz. The HMM speech recogniser requires these values to be encoded into a sequence of feature vectors, where each feature vector encodes the essential features of a short "frame" or "window" of the input signal. The encoding requires several parameters to be determined: the size of each window, the interval ("frame period") between each two adjacent frames, and the set of features to be extracted from each frame. The encoding process is shown in figure 3.

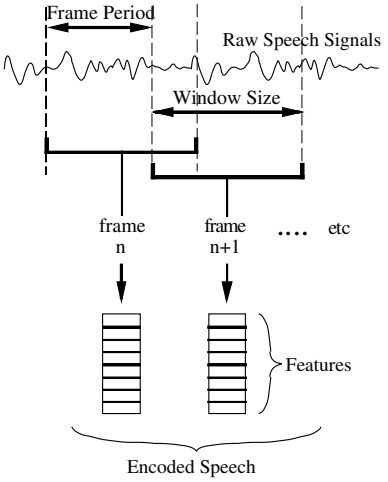


Figure 3: Speech encoding process. (Adapted from (Young et al. 2002))

3.1 Window Size and Frame Period

The window size and frame period are important parameters for the speech recogniser.

If the window size is too small, the window will not contain enough of the signal to be able to measure the desired features; if the window size is too large, the feature values will be averaged over too much of the input signal, and will lose precision. We explored a range of window sizes, from 10ms to 30ms, with the lower limit being chosen to be large enough to include at least two complete cycles of the fundamental frequency of the speech signal for the female speakers being analysed, and the upper limit chosen to ensure that a window seldom spanned more than a single phoneme.

If the frame period is too long, there will be insufficient feature vectors for each phoneme, which will prevent the HMMs from recognising the speech. If the frame period is longer than the window size, then some parts of the speech signal will not be encoded at all. If the frame period is too short, then there will be too many feature vectors, which will increase the computational cost. The absolute lower limit on the frame period is governed by the sample rate of the raw speech signal. We explored a range of frame periods, from 4ms to 12ms, subject to the constraint that the frame period was not larger than the window size.

3.2 Feature Extraction and Selection

There are many possible classes of features that could be used to encode a speech signal. We have followed common practice in using Mel-Frequency Cepstrum Coefficients (MFCCs) on a Hamming window. MFCCs use a mathematical transformation called the cepstrum which computes the inverse Fourier transform of the log-spectrum of the speech signal (Young et al. 2002). Within this class of features, there is still considerable choice about which MFCC features to use. In addition to the 12 basic MFCC transformation coefficients (1st–12th), there are also the energy (E), the 0th cepstral coefficient (0), and the first and second order derivatives (D and A) of those coefficients. Not all combinations of these features are sensible. For example, it makes little sense to use the second order derivatives without the first order derivatives, and the HTK toolkit (Young et al. 2002) will not use both the energy and the 0th cepstral coefficient simultaneously. We have identified six combinations to explore in our experiments, as shown in table 1.

Table 1: Six feature combinations and the number of features in each set.

No	Combination	No. of Features
1	MFCC-E	13
2	MFCC-E-D	26
3	MFCC-E-D-A	39
4	MFCC-0	13
5	MFCC-0-D	26
6	MFCC-0-D-A	39

4 Experiment Design and Method

For our experiments, we trained a collection of phoneme level HMM models on a training set of annotated speech samples with each combination of parameters. We then evaluated the quality of the HMM models by using them to recognise and label speech samples in a separate test set. This section describes the data sets used in the experiment, the parameter combinations we explored, the training and testing process, the experiment configuration, and the performance evaluation.

4.1 Data Set

The experiments used a speech data set collected by the School of Linguistics and Applied Language Studies at Victoria University, as part of the New Zealand Spoken English Database (www.vuw.ac.nz/lals/nzsed). This data set contains 1119 utterances of 200 distinct English sentences produced by six female native speakers. The utterances

were recorded at a 16kHz sampling rate, which allows accurate analysis of all frequencies up to 8kHz (the Nyqvist frequency for this sampling rate). The range to 8kHz includes all perceptually relevant information in human speech.

The utterances were labelled at the phoneme level, specifying the name and the start and end times of each phoneme. The labelling was at a coarse level, with 44 different English phonemes and two labels for silences and short pauses. The first half of the 1119 utterances were hand-labelled at the phoneme level by trained linguists; the second half were automatically labelled by a computer program (based on the labelling of the first half) then checked by a trained linguist. Each utterance was also identified with the text of the sentence it represented.

For our experiments, the labelled utterances were split into a training set with 544 utterances and their labels and a test set of the remaining 575 utterances. The split preserved an even distribution of speakers in both sets, but was otherwise random.

4.2 Design of Case Combinations

The goal of the experiment was to explore the performance of different choices of feature sets, window sizes, frame periods, and sizes of the mixture-of-Gaussian models. As described in the previous section, we have 6 combinations of feature sets to explore.

Since the average fundamental frequency of female speech is around 220 HZ, the period of the voiced sounds is approximately 4.5 ms. We therefore chose a set of nine frame periods from 4ms to 12ms in steps of 1ms. The smallest frame period was chosen to be only marginally smaller than the average period of the fundamental frequency. The largest frame period was chosen to be above the default frame period suggested by the HTK toolkit.

We chose a set of nine possible window sizes from 10ms to 30ms in steps of 2.5ms. With the constraint that the window size should be at least as long as the frame period, there are just 79 possible combinations of window size and frame period and therefore 474 combinations of the speech encoding parameters.

We also explored five different sizes for the mixture-of-Gaussian models: 1, 2, 4, 8, and 16. There were therefore a total of 2370 different possible combinations of parameters to evaluate.

We refer to the different sets of parameters by hyphenated codes such as "9-10-E-D-A-4" where the first number is the frame period, the second number is the window size, the letters specify which of the MFCC features were used, and the final number is the size of the mixture-of-Gaussian models.

4.3 Training and Testing Process

For each combination of parameters, a set of phoneme level HMMs was trained on the utterances (and their labels) in the training set. During the training process, each utterance was encoded and the relevant features were extracted based on the choice of features, window size, and frame period. Each HMM state was modeled initially by a mixture-of-Gaussians of size 1 and trained using four cycles of the Baum-Welch training algorithm (Young et al. 2002). The maximum size of the mixture-of-Gaussians was then doubled and two cycles of the Baum-Welch re-estimation were applied. This was repeated until the maximum size of 16 was reached.

After obtaining the phoneme level HMMs, the testing process was conducted by applying these HMMs to the utterances in the test set using forced alignment and the Viterbi algorithm (Young et al. 2002). The testing process generated a set of auto-labelled phonemes (phoneme name, start, and end time) for each utterance. These auto-labelled phonemes were then be compared to the hand-labelled phonemes to measure the accuracy of the HMMs.

4.4 Experiment Configuration

Since training and testing a set of HMM models is a computationally intensive task, it would not have been feasible to run this exhaustive experiment on a single computer. Instead, we constructed a server-client computer network to run the experiment. The system consisted of one Sun Fire 280R Server and 22 1.8GHZ Pentium 4 workstations with 128MB RAM running NetBSD, as shown in figure 4. Even with these 22 computers, the experiment took more than 48 hours to complete.

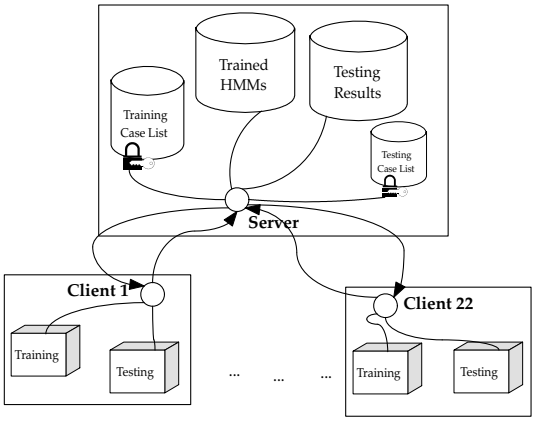


Figure 4: Experiment configuration.

There are two central synchronised lists on the server, one containing all training cases (one case for each combination of speech encoding parameters) and the other for testing cases. To reduce the network traffic, the utterances in the training and test data

sets were pre-distributed to each client (workstation). The training time varies with different training cases. Therefore, instead of pre-assigning a fixed number of training cases to each client, we created connections between the server and the 22 clients so that the clients can train or test any case. Clients repeatedly request a training case (a combination of parameters) from the server, perform the training process, then send the trained HMMs back to the server to be added to the list of testing cases. Once the list of training cases is empty, each client starts requesting testing cases from the server, performing the testing and sending the resulting set of auto-labelled utterances back to the server. The effect of this distributed process is that no clients are idle until the very end of the experiment.

4.5 Performance evaluation

To measure the recognition performance of a case, the system compares the auto-labelled phonemes in each utterance of the test set against the hand-labelled phonemes.

In the context of our speech analyser system, the most important requirement on the recognition system is the accuracy of the time boundaries of the auto-labelled phonemes. The simplest way of measuring this accuracy would be to measure the average error, where the error is the time difference between the boundary of the auto-labelled phoneme and the hand-labelled phoneme. However, we suspect that large errors will be much more significant for the rest of the speech analyser than small errors. Also, the nature of continuous speech is such that determining hand-labelled phoneme boundaries necessarily involves a subjective judgement. This means that small errors should not affect the accuracy measure. We therefore set a threshold and counted the percentage of phonemes for which the difference between the auto-labelled boundary and the hand-labelled boundary is less than the threshold.

Obviously, the actual accuracy measure will vary with different thresholds — with a sufficiently high threshold, all the cases would have a 100% accuracy. However, the purpose of the accuracy measure is to compare the performance of different cases, so only the relative value of the accuracy measure is important, and the threshold value is not too critical. According to the literature (Boeffard, Miclet & White 1992, Grayden & Scordilis 1994, Ljolje, Hirschberg & van Santen 1997, Vonwiller, Cleirigh, Garsden, Kumpf, Mountstephens & Rogers 1997, Wightman & Talkin 1997), a match distance between an auto-labelled phoneme and its corresponding hand-labelled phonemes within 2-4 cycles of the fundamental frequency would be considered a very close match. Since the average fundamental frequency for adult female speakers is about 220HZ, we chose 16 milliseconds for the threshold in our measure. We also looked at the effect on the results of changing this threshold in either direction.

We were also interested in the sources of boundary time errors. We hypothesised that some of the errors might be due to the recogniser misclassifying phonemes during the recognition process. We therefore performed a more standard recognition accuracy evaluation on the best performing HMM, measuring the percentage of phonemes in the test set that were misclassified by the HMM.

5 Results and Discussion

This section presents the relative recognition performance of the cases and gives some further analysis of the results.

5.1 Best Parameter Combinations

Using the measure described in the previous section, we calculated the relative phoneme boundary accuracy of the HMM speech recogniser with 2370 different combinations of parameters. Since the vowels play a more important role in the later stages of the speech analyser, we also calculated the relative accuracy results for vowels only (measuring just the end boundary of the vowel). The best ten and the worst results are given in tables 2 and 3.

Table 2: Best and worst parameter choices by boundary timing accuracy.

rank	Case	Boundary Accuracy
1	9-15-0-D-A-8	81.47%
2	9-15-0-D-A-4	81.38%
3	10-12.5-0-D-A-1	81.24%
4	10-15-E-D-A-4	81.23%
5	10-12.5-0-D-A-4	81.23%
6	9-17.5-0-D-A-4	81.23%
7	9-17.5-E-D-A-4	81.21%
8	11-15-0-D-A-4	81.21%
9	10-17.5-0-D-A-4	81.20%
10	9-12.5-0-D-A-4	81.19%
⋮	⋮	⋮
2370	4-30-0-16	57.66%

There are several observations that can be made from the results.

- The best performance (81.47% accuracy for 9-15-0-D-A-8) is considerably better than the worst performance (58% accuracy for 4-30-0-16), so that the choice of parameters is important.
- There is a clear advantage in using the derivative (D) and acceleration (A) features. All of the

Table 3: Best and worst parameter choices by boundary timing accuracy of vowels only.

rank	Case	Boundary Accuracy
1	11-15-0-D-A-4	82.49%
2	12-17.5-0-D-A-4	82.45%
3	12-15-0-D-A-4	82.42%
4	11-17.5-0-D-A-4	82.36%
5	12-20-0-D-A-2	82.22%
6	11-12.5-0-D-A-4	82.21%
7	12-12.5-0-D-A-4	82.19%
8	12-20-0-D-A-4	82.17%
9	11-17.5-0-D-A-8	82.16%
10	12-15-0-D-A-2	82.13%
⋮	⋮	⋮
2370	4-30-0-16	61.17%

top 13% of the cases have the acceleration features, and all of the top 46% of the cases have the derivative features.

- A frame period around 9ms to 10ms and a window size around 15ms appear to give the best performance over all phonemes, but a larger frame period around 11ms to 12ms and a larger window size around 17.5ms is better for performance on vowels alone.
- There appears to be just a slight advantage of the 0th Cepstral (0) feature over Energy (E) for the full set of phonemes, but the 0th Cepstral feature is clearly better on vowels alone. (Only three of the top 50 cases for vowels use Energy.)
- There seems to be a preference towards a size of 4 for the mixture-of-Gaussians, and none of the top 50 cases have a size of 16.
- The difference in relative accuracy over the top 5% of the cases is only 1%, so that the exact choice of E vs 0, frame period, window size and number of Gaussians, within the ranges above, does not appear to be critical.
- Changing the threshold to 8ms or 32ms makes no difference to the strong advantage of the Derivative and Acceleration features. However, the preferred frame period and window size are slightly smaller for the 8ms threshold, and slightly larger for the 32ms threshold. Also for the 8ms threshold, there is preference for the Energy feature rather than the 0th Cepstral feature.

The outcome of this experiment is a clear recommendation for the parameters we should use for the speech analyser system: using 0th Cepstral, Derivative and Acceleration features, along with a frame period of 11ms, a window size of 15ms, and a mixture of 4 Gaussians should minimise the boundary timing errors on the vowels. If it turns out in later work on the system that boundary timing differences of less than 16ms are significant, we would then need to use a smaller frame period and window size and the Energy feature.

5.2 Recognition Accuracy

The results above focused on the accuracy of the boundaries of the auto-labelled phonemes. The second evaluation attempted to identify some possible sources of the boundary errors by counting the kinds of phoneme recognition errors made by the recogniser using the best performing HMMs. There are three kinds of phoneme recognition errors:

- substitution, where the auto-labelled phoneme is different from the hand-labelled phoneme.
- insertion, where the auto-labelling includes an additional phoneme that is not present in the hand-labelling.
- deletion, where the auto-labelling does not contain a phoneme that is present in the hand-labelling.

Table 4 shows the percentage of each category of recognition errors for the highest ranked HMM (9-15-0-D-A-8) applied to the test set. Since insertion and deletion errors will almost certainly result in boundary time errors of phonemes adjacent to the error, in addition to the phoneme inserted or deleted, the nearly 5.8% of insertion or deletion errors is a non-trivial cause of the approximately 18% boundary timing error rate in table 2. The substitution errors may or may not result in boundary timing errors.

Table 4: Recognition errors for 9-15-0-D-A-8.

Kind of Error	Percentage of phonemes
Insertion errors	4.7% (1221 out of 25723)
Deletion errors	1.1% (286 out of 25723)
Substitution errors	4.4% (1134 out of 25723)

The recognition system uses forced alignment recognition in which the system knows the target sentence and uses a dictionary of alternative word pronunciations to determine the expected phonemes. Insertion and deletion errors will generally occur when the actual pronunciation by the speaker does not match any of the pronunciations in the dictionary: the speaker drops a phoneme or includes an extra phoneme, and the system is forced to align the dictionary pronunciation with the actual pronunciation. These errors are due primarily to inadequacies in the dictionary, rather than to the HMM models. Substitution errors will result from pronunciations by the speaker that are not in the dictionary, but also may

result from poor HMM phoneme models if the dictionary gives two alternative pronunciations, and the HMM for the wrong phoneme matches the speech signal better than the HMM for the correct phoneme.

6 Improving the Recognition Performance

Even with the best choice of parameters, there are clearly many phoneme boundaries that the recognition system is not able to identify accurately. Although we do not yet know exactly how much of a problem this would be for the overall speech analysis system, we do know that computer aided instruction systems need a very high accuracy when providing feedback to students, and we believe it will be important to improve the accuracy of the speech recognition component. We also expect the problem to be much greater with non-native speakers than with native speakers, since non-native speakers are much more likely to mispronounce words, and to delete or insert phonemes in their utterances.

Analysis of the results of our experiments, particularly a detailed analysis of some of the errors that the system makes, have enabled us to identify two approaches to deal with this problem.

The first approach is to augment the dictionary with a richer set of alternative pronunciations that capture more of the likely pronunciations and mispronunciations. The second is to modify the architecture of the phoneme-level HMMs to allow skipping of states. In particular, if the HMMs allow all three states to be skipped, then the recogniser, even with forced alignment, can handle missing phonemes by matching the phoneme against a zero length segment of the speech signal.

6.1 Improved Forced Alignment System

The dictionary we use currently contains alternative pronunciations of many of the words in the dictionary. The alternative pronunciations reflect some of the pronunciations acceptable to a native speaker. Clearly, we need to continue augmenting the dictionary with other alternative pronunciations that are acceptable to native speakers. However, the dictionary does not include the common pronunciation mistakes of native speakers nor the mispronunciations of non-native speakers who are currently learning English. These mispronunciations cause auto-labelling errors in our system.

Figure 5 shows an example from a non-native speaker mispronouncing the word `pretend`. The dictionary pronunciation is /prɪˈtend/, but the speaker mispronounced it as /pˈtenə/ (she missed /rɪ/ and mispronounced the final phoneme /d/ as /ə/). The top viewport is the spectrogram of the sound of word `pretend` mispronounced by the speaker; the middle viewport shows the hand-labelling of the sound waveform, and the bottom viewport shows the auto-labelling of the same sound waveform. Clearly, the boundaries of the auto-labelled phonemes have been badly affected by missing and mispronounced phonemes.

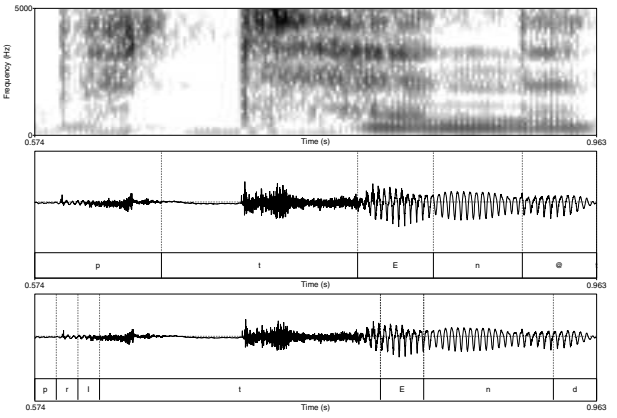


Figure 5: Auto-labelling errors of the forced alignment system.

Missing certain phonemes (such as /r/) and adding phonemes such as /ə/ at word boundaries and within consonant sequences are common mistakes among ESL students, and our system needs to be able to deal with them more effectively. However, adding all the possible alternative pronunciations directly to the dictionary would make the dictionary very large, and would also greatly increase the complexity of the HMMs built (on the fly) by the forced alignment speech recogniser. This increased complexity would have unacceptable consequences for the computation speed of the recogniser.

Instead, we intend to build a model of the kinds of deletions, insertions, and substitutions that are common in the speech of Chinese ESL students, and use this model to dynamically construct a better phoneme network that allows the recogniser to deal with insertion and deletion errors more gracefully.

6.2 A New HMM Model

The second approach for improving the recognition performance is to improve the HMM design. The constrained left-to-right connection mode between the three states of a phoneme HMM requires that every phoneme is allocated at least three frames of the speech signal. We have identified errors due to very short phonemes. The boundaries of the auto-labelled phonemes may be wrong because the system was forced to allocate frames to a short phoneme that should have been allocated to adjacent phonemes. Also, where the dictionary states that the phoneme should be present but the speaker has dropped it, the system is forced to "steal" at least three frames from neighbouring phonemes to allocate them to the expected phoneme.

We believe that these problems can be addressed by a more robust three state HMM in which some or all of the states can be skipped. This enhanced HMM is shown in figure 6.

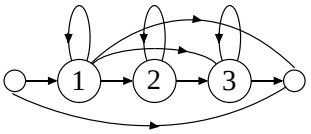


Figure 6: A new three-state HMM.

For long phonemes, particularly diphthongs, there may be more variation during the phoneme than can be captured well by just three states. We will consider HMMs with more than three states for such vowels.

In addition, the current HMM design deals with the short pause /sp/ and the silence /sil/ but does not model breathing properly since an audible breath has energy, and we do not have an HMM for breathing. Short periods of breathing currently confuse the recogniser, and cause it to label the phoneme boundaries badly.

Figure 7 shows an example of auto-labelling errors resulting from a short breath and an inserted phoneme. The first panel presents the spectrogram of a speech signal consisting of the word `but` preceded by a short breath and followed by an inserted /ə/, produced by a female ESL student. The second panel shows the sound waveform and the hand-labelling. The third panel presents the auto-labelling generated by the forced alignment system. As can be seen, the forced alignment system placed most of the boundaries quite inappropriately.

Audiable breathing has less of an impact on the native speaker data as these speakers were largely able to read the sentences fluently on a single intake of breath. For non-native speakers, we expect considerably more hesitation and therefore periods of breathing will be much more common. We will add an HMM model for breathing to our system, to remove these errors.

7 Conclusions

We have described an HMM based forced alignment speech recogniser that will be a component of a speech analyser system to assist ESL students practice and improve their English. The central requirement on the recogniser is that it can accurately identify the boundaries of the phonemes in a speech signal.

We have reported on an experiment that exhaustively explored a space of parameter values for the recogniser. This included the parameters of the encoding of the speech signal and the size of the statistical models in the states of the phoneme-level HMMs. The results of the experiment gave clear recommendations for the choice of frame period, window size,

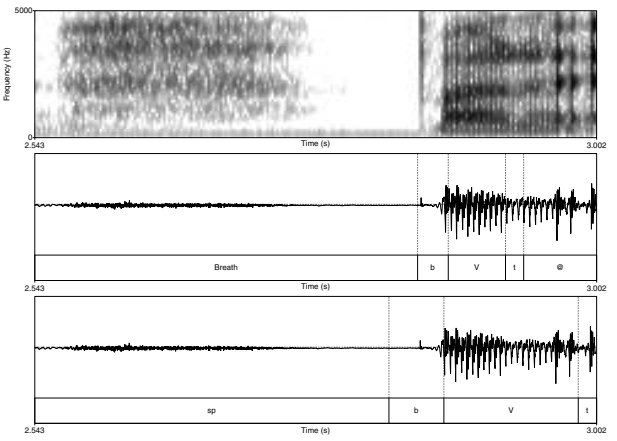


Figure 7: Auto-labelling errors with a short breath and an inserted phoneme.

MFCC features, and the statistical model in order to minimise the significant phoneme boundary errors.

The results of the experiment also identified several causes of phoneme boundary errors, including limitations of the dictionary used for forced alignment, the constraints of the HMM connection mode, and the need for a model of breathing. We also outlined the approaches that we will explore to address these limitations.

8 Acknowledgement

This project is part of a FRST funded NERF project. Other members of the project include David Crabbe of the School of Linguistics and Applied Language Study, Irina Elgort of the University Teaching Development Centre, Neil Leslie of Computer Science, and Mike Doig of VicLink.

References

Boeffard, O., Miclet, I. & White, S. (1992), Automatic generation of optimized unit dictionaries for text-to-speech synthesis, *in* 'Proceedings of the International Conference on Spoken Language Processing', Vol. 2, Banff, pp. 1211–1214.

Grayden, D. B. & Scordilis, M. (1994), Phonemic segmentation of fluent speech, *in* 'Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing', Adelaide, pp. 73–76.

Irino, T., Minami, Y., Nakatani, T., Tsuzaki, M. & Tagawa, H. (2002), Evaluation of a speech recognition/generation method based on hmm and straight, *in* 'Proceedings of the International Conference on Spoken Language Processing', Vol. 4, Denver, pp. 2545–2548.

Lindgren, A. C., Johnson, M. T. & Povinelli, R. J. (2003), Speech recognition using reconstructed phase space features, *in* 'Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing', pp. 60–63.

Livescu, K. & Glass, J. (2000), Lexical modeling of non-native speech for automatic speech recognition, *in* 'Proceedings of the International Conference on Acoustics, Speech and Signal Processing', pp. 1842–1845.

Ljolje, A., Hirschberg, J. & van Santen, J. P. (1997), Automatic speech segmentation for concatenative inventory selection, *in* J. P. van Santen, ed., 'Progress in Speech Synthesis', Springer-Verlag, New York, chapter 24, pp. 304–311.

Pellom, B. & Hansen, J. (1998), 'Automatic segmentation of speech recorded in unknown noisy channel characteristics', *Speech Communication* pp. 97–116.

Vonwiller, J., Cleirigh, C., Garsden, H., Kumpf, K., Mountstephens, R. & Rogers, I. (1997), 'Development and applicationof an accurate and flexible automatic aligner', *International Journal of Speech Technology* **1(2)**, 151–160.

Wightman, C. & Talkin, D. (1997), The aligner: Text-to-speech alignment using markov models, *in* J. P. van Santen, R. W. Sproat, J. P. Olive & J. Hirshberg, eds, 'Progress in Speech Synthesis', Springer-Verlag, New York, pp. 313–320.

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Valtchev, V. & Woodland, P. (2002), 'The HTK Book (for HTK Version 3.2)'. http://htk.eng.cam.ac.uk/prot-docs/htk_book.shtml.