UNIVERSITY OF CAMBRIDGE

**MPhil in Machine Learning, Speech and Language Technology**
**2015-2016**

# MODULE COURSEWORK FEEDBACK

Student Name: Riashat Islam

Module Title: Speech Recognition

CRSiD: ri258

Module Code: MLSALT2

College: St John's

Coursework Number: 1

***I confirm that this piece of work is my own unaided effort and conforms with the Department of Engineering guidelines on plagiarism***

I declare the word count for this piece is: 4995

Student's Signature: Riashat Islam

---

Date Marked:

Marker's Name:

**This piece of work has been completed to a standard which is** *(Please give mark as appropriate)*:

**Marker's Comments:**

# TIMIT Speech Recognition with GMM-HMMs

Riashat Islam
Department of Engineering
University of Cambridge
Trumpington Street, Cambridge, CB2 1PZ, England
ri258@cam.ac.uk

***Abstract*—This paper demonstrates the effectiveness of context dependent phone modeling and examines the influence of language models for phone recognition on a limited size training corpus using CUED HTK toolkit. In this study, we draw comparison of using both context dependent and context independent acoustic phone models applied to unigram and bigram language models. We examine the performance differences, compare the models and comment on an overall best system for phone recognition using GMM-HMMs. Our results show the improvements in recognition performance when context information is modeled. Finally, we present a review of how the approach can be scaled up for large vocabulary speech recognition maintaining a balance between modeling accuracy, design, complexity and run-time.**

## I. Introduction

Automatic speech recognition is the process of mapping a speech signal to the corresponding sequence of words that it represents. Any of the general purpose speech recognition systems are based on Hidden Markov Models (HMMs). In this work, we consider building a speech recognition system based on GMM-HMMs, considering phones. Our work is based on the Cambridge HTK toolkit. Section II gives an overview of basic concepts of training and decoding the systm, along with brief overview of Gaussian Mixture Model (GMM) based HMMs and the language model. Section III then considers acoustic modelling with monophone models. Section IV considers using context dependent triphone models and section V considers biphones for acoustic modelling. We then evaluate the performance of the system based on using a Bigram Language Model instead of using a Unigram model in section VI. Each section summarizes the method used, experimental results and a discussion of results. Finally, based on modelling accuracy, estimated parameters and generalisation, we comment on the overall best system suitable for phone recognition based on available training corpus.

## II. Background

HMMs are used as they are effective for modelling the time varying sequences of the speech spectrum. In a speech recognition system, the input audio waveform from a speaker is converted into a sequence of acoustic vectors in a feature extraction step $Y_{1:T} = y_1, ...y_T$ and the decoder then attempts to find the sequence of words $w_{1:L} = w_1, w_2, ...w_L$ which is likely to have generated the speech waveforms. The goal of speech decoder is therefore to find:

$$\hat{w} = \underset{w}{\operatorname{argmax}}\, P(w|Y) = \underset{w}{\operatorname{argmax}}\, p(Y|w)p(w) \qquad (1)$$

where the likelihood $p(Y|w)$ can be determined from the acoustic model and the prior $p(w)$ determined from the language model. The basic unit of speech is represented by a phone. To estimate the parameters of these phone models we use the training data that consists of speech waveforms and their corresponding transcriptions. The training data consists of phonetic transcriptions (TIMIT) which contains recordings of phonetically balanced English speech. The langauge model, given by the prior $p(w)$ is an N-gram model where the probability of each word is conditioned on the $N-1$ previous words. The basic task of a speech decoder is to search through all possible speech sequences and to output the most likely word sequence at the end of the speech using an efficient search across all possible word sequences for the speech [1].

### A. Gaussian Mixture Model based HMMs

Hidden Markov Models are used as the acoustic models for speech recognition, where the acoustic model provides the likelihood for a set of acoustic vectors given a word sequence [1]. The outputs are described as continuous density probability functions where we consider output distributions as mixture of Gaussians. In this work, we consider training the GMM-HMM models where the use of multiple mixture components allows for the modelling of abstract distributions. In the acoustic model, during training the word string is mapped to the relevant set of HMM models M, and we search the observed data over $p(Y_T|M)$ such as to maximize the likelihood of the data given the model.

### B. Training HMMs

The HMM model parameters (transition and emission probabilities), state and mixture means, covariances and mixture weights of the Gaussian mixture models (model parameters for GMM-HMM systems) are all estimated during training to match the training data well given a training criterion (commonly Maximum Likelihood). The ML training scheme is used to maximise the likelihood of the training data.

### C. Language Models

In our work, we will consider both unigram and bigram language models.The probability of a given word sequence

is obtained from the language model. In language models, the N-gram probabilities are estimated by counting N-gram occurrences to form Maximum Likelihood parameter estimates since we want to associate probabilities with given word strings.A modification of the language model scoring is to use a word insertion penalty. The word insertion penalty can penalise the addition of words into the hypothesised word string as word errors are frequently caused by the insertion of short words with wide contexts. When we subtract a word insertion penalty in the log level, then this is equivalent to scaling or discounting the word probabilities by a fixed amount.

### *D. Decoding*

During the decoding step, the aim is to find the most likely utterance over all possible word sequences. We calculate $p(Y_T|M)$ over each word sequence. We search for the optimal path with highest likelihood using a *token passing* method where for a given time step and feature vector, each state is assigned a single token and the token contains a word-end link and a value of the likelihood. The token with the highest log probability is then traced back to give the most likely sequence of words.

$$PercentCorrect = \frac{N - D - S}{N} * 100\% \tag{2}$$

$$PercentAccuracy = \frac{N - D - S - I}{N} * 100\% \tag{3}$$

where N is the total number of words, D is the number of deletions, S number of substitutions and I insertions. Note that the accuracy measure includes the insertion errors compared to the correction measure, and hence is a more representative figure of the recogniser performance.

### *E. Context Dependent Phone Models*

One major problem of concatenating phone models is that decomposing each voabulary word into a sequence of context-independent base phones fails to capture the large degree of context-dependent variations that exist in real speech waveforms. In our work, we will consider both context-independent phone models (monophones), and two different approaches to context-dependent phone models (biphones and triphones).

## III. MONOPHONE MODEL

### *A. Flat Start Monophones*

We consider training GMM-HMMs based on labelled and unlabelled data, and investigate whether labelled data is better suited for training the systems. The initial estimates of the HMM parameters are important, since good initial estimates can ensure that the local maximum is as close as possible to the global maximum of the likelihood function. We first use the Flat Start (FS) initialization. If no information about the boundaries of the phones are available, FS is usually used to set up HMMs with the same data where such initialisation does not require human intervention. In flat start initialisation, we manually segment the training data and assign all models the global mean and variance. When labelled data is available to us, HInit and HRes can be used for initial parameterisations of HMM parameters, followed by re-estimation of parameters using HRest.

*1) **Experimental Results**:* Using the step-mono script, with 8 Gaussian mixture components and FBK feature parameterizations, the following results were obtained for environment Z provided.

| Differences between FS and Non-FS | | |
|---|---|---|
| Initialisation | Flat Start | Without Flat Start |
| Correction | 55.60 | 55.41 |
| Accuracy | 30.80 | 30.90 |

The results above shows that using flat start initialisations, $30.80\%$ of phones were recognised correctly, compared to $30.90\%$ without flatstart initialisations, although the accuracy measures are slightly lower for flat start. Below we evaluate the choice of best initialisation.

*2) **Discussion**:* In the above procedure, HInit uses the Viterbi algorithm for labeled training data, whereas flat start can always be used when we do not have labels of the training data available. The requirement of labelled training data is a limitation of HInit with sub-word models. Flat Start initialisation is the alternative when no initialisation strategy is available and we can move directly to the embedded training using HERest. This ensures that the first iteration of the training sceheme will rely on a uniform segementation of the data. We use HInit only when a labelled training data is available. Since the accuracy is slightly lower for Flat Start, but almost similar, we conclude that using Flat Start is better suited as it avoids the need to manually label phone boundaries, which is more convenient for training large vocabulary speech recognition systems. Therefore, for future experimentation, we use unlabelled training data, without phone boundaries for training.

### *B. Front-End Parameterisations*

This section considers front-end parameterizations, to obtain final acoustic feature vectors required for a speech recognition system. We consider the effect of using different parameterized speech feature vectors. The two parameterizations considered are log mel-filter bank channel outputs (FBANK) and mel-frequency cepstral coefficients (MFCCs). The provided $Z$, $Z - D$ and $Z - D - A$ environment files contains varying amounts of appended differential coefficients. These are the delta coefficients which are the derivatives of the static parameters. These coefficients are added to FBANK (which itself is obtained by taking logarithm of mel-scale coefficients) and to MFCC (taking the Discrete cosine transform of log-mel-filter bank coefficients). In the HTK toolkit, these are specified by attaching qualifiers to the basic parameter kind - the qualifier *D* indicates that first order regression coefficients (delta coefficients) are appended. The qualifier $A$ indicates that second order regression coefficients (acceleration coefficients) are appended. We

add both these coefficients to the Filter Bank and MFCC features. Delta cepstral and Double-Delta cepstral features are typically used to add dynamic information to the static cepstral features. They help towards improving accuracy by adding a characterization of temporal dependencies to the HMM frames which are normally assumed to be statistically independent of one another.

| Coefficients in Feature Vector | | | |
|---|---|---|---|
| Environments | Z | Z-D | Z-D-A |
| FBK | 24 | 48 | 72 |
| MFCC | 13 | 29 | 39 |

Since MFCCs use Discrete Cosine Transforms, this compresses the spectral information into lower order coefficients, and hence the feature vectors with MFCCs are lower in dimensions compared to FBK.

*1)* ***Experimental Results****:* The results are for experiments using different environments, each containing different number of differential coefficients for the different parameterisations listed in above table.

| FBK Parameterization | | | |
|---|---|---|---|
| Environments | Z | Z-D | Z-D-A |
| Correction | 55.60 | 60.83 | 63.69 |
| Accuracy | 30.80 | 40.70 | 40.59 |
| **MFCC Parameterization** | | | |
| Environments | Z | Z-D | Z-D-A |
| Correction | 57.78 | 66.63 | 69.31 |
| Accuracy | 37.55 | 51.82 | 52.90 |

The results below shows that with MFCC parameterizations using Flat Start initializations for training, the correction values are higher, while the accuracy is significantly better (comparing 52.90 for MFCC with 40.59 FBK for Z-D-A environments). Additionally, as delta coefficients are first added (from Z to Z-D), there is a sharp increase in accuracy values (for both parameterizations). However, when second order coefficients are also added (Z-D-A), there is a fall in corrections for MFCC, and a fall in accuracy values for FBK. As we add more delta coefficients, we are adding robustness to the additive noise and hence the original transcription labels are more likely to be matched by the generated phones, leading to less insertion penalty. Adding dynamic information therefore means the word insertion penalty decreases. Figure 1 shows how the correction values increase as we add more delta coefficients to the feature vectors.

*2)* ***Experimental Results: Grammar Scale and Insertion Penalty for Monophones****:* The Gaussian Mixture Models (GMMs) are used to model the output distributions such as to accurately model distributions of real-time speech. The mixture components can be increased to take account of the complex real distribution of the speech spectra. GMM output distributions can model the emitting state distributions accurately, especially if we increase the number of mixture components. However, increasing the number of Gaussian mixture components also increases the complexity of the models that we need to train, given that we have limited

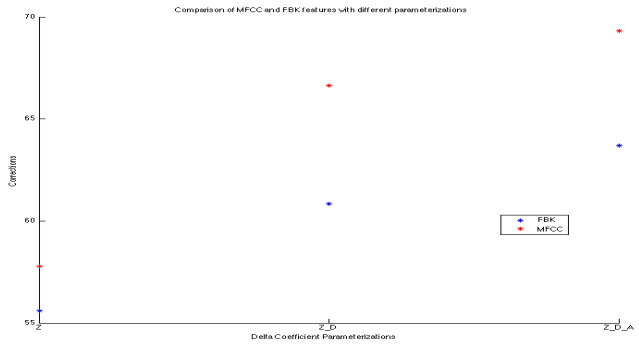


Figure 1. Performance changes with differential coeffcients

training data. We may run into issues like overfitting if there are too many parameters that need to be estimated with a relatively smaller training data. Hence, we need to maintain a balance between accuracy and increasing the number of Gaussian mixture components. Given the training data, we need to balance complexity and the accuracy achieved by the recogniser. The table below shows the total number of Gaussian components in the system containing 48 HMMs each containing 3 emitting states. NUMMIXES shows the number of mixture components used per state. NUMMIXES therefore determines the total number of Gaussians in the system. $1NM = 144Gaussians$, $2NM = 288Gaussians$ and $32NM = 4608Gaussians$.

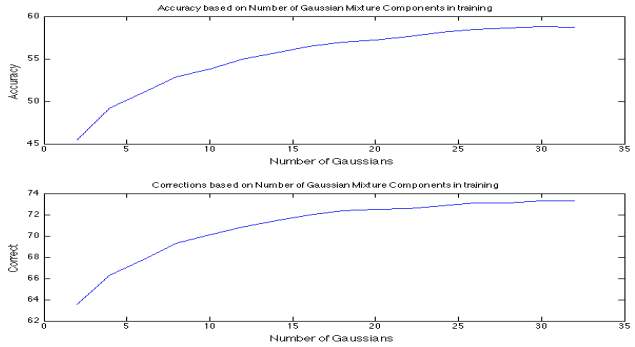


Figure 2. Accuracy and Corrections measure as a function of Gaussian Mixture Components for training

Figure 2 shows that as we increase the number of mixture components (increasing total number of Gaussians), then accuracy and correctness increases. However, it can be notably observed that beyond 10 mixture components, the accuracy does not increase significantly. Additionally, increasing mixture components increases the run-time of the training and decoding processes, introducing a tradeoff between achieving very high accuracy and computational complexity (run-time). If mixture components are too high (around 32), as discussed earlier, it also increases the total number of parameters (Gaussians) increasing model complexity. The experiments done here are for monophones.

We then evaluated how the performance of the recognizer depends on the word insertion penalty. The word insertion

penalty values determines the log transition probabilities in each token, such that the model penalises every insertion and is not willing to include phones in transcriptions in cases when the insertion penalty value is high. We changed the insertion penalty across a range of values from -40 to 20, summarized in figure 3. The grammar scale, for unigram langauge models does not have any effect and hence not been changed for the experiment below.

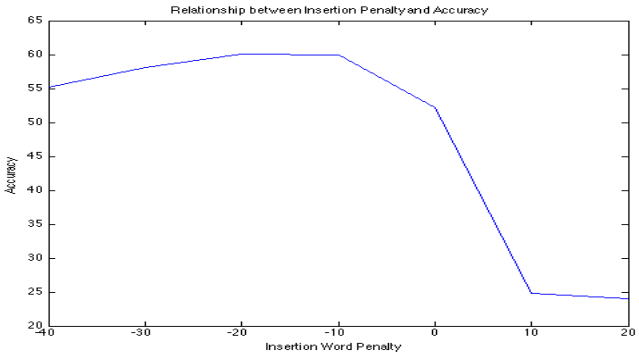


Figure 3. Accuracy vs Insertion Word Penalty

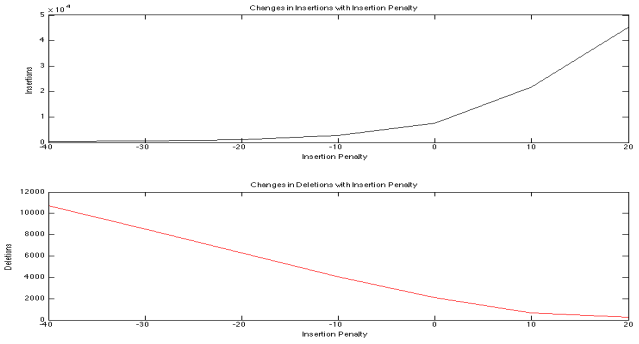


Figure 4. Changes in Insertions and Deletions with Insertion Penalty

The accuracy decreases as word insertion penalty is increased from -40 to 20 as shown in figure 4. This is because the model is more reluctant to accept phones. The figure 3 above shows that as insertion penalty is increased from -40 to 20, then accuracy further falls. Therefore, we need to balance the number of insertions and deletions, and hence the accuracy which is dependent on the insertion penalty.

### C. *Discussion*

**MFCC Parameterizations with Z-D-A environment**: The results in III-B1 shows that MFCC parameterizations with first and second order differential coefficients achieves the best results. The delta coefficients adds more dynamic information to the static cepstral features and hence the recognizer becomes more robust to surrounding noise. Also, the MFCC vector with 39 coefficients (environment $Z-D-A$) achieves the best result as it contains both first and second order delta coefficients. We will use these acoustic vectors for further experiments.

**Insertion Penalty of -10**: The insertion penalty is the value added to each token when it transmits from the end of one word to the start of the next word. The grammar scale being only a measure of how the language model probability should be scaled does not affect in case of unigram language models as there is no measure of transition probabilities in unigram models. The results show that using an insertion penalty of -10, the decoder achieves the best accuracy, while corrections always increase as more penalty is added (since it does not take account of insertion errors). Based on results above, we therefore can conclude that an insertion penalty of -10 is ideally suited for acoustic modelling using monophones, with a unigram language model, for evaluating the performance of the recognizer on the trained HMMs.

**Dependence on Number of Gaussians**: GMMs are known for their ability to represent arbitrary complex distributions with multiple modes. The results in figure 2 above shows that balancing run-time of the algorithm, accuracy and complexity with the given training data, 8 mixture components (NUMMIXES) with 1152 Gaussians in the entire system is ideally suited to achieve a good recognition performance. Even though higher Gaussian mixtures may increase accuracy further, but this may not be suitable for generalisation during decoding since we are using a limited size training corpus.

## IV. Triphone Models

In this section, we build triphone models based on decision tree clustering using monophones, with the best performing parameterizations from the above section. Throughout the experiments in this section, we used MFCC front-end parameterizations using 8 Gaussian mixture components with flat start initializations.

To achieve good performance in a continuous density HMM system, we need to use output distributions of mixture of Gaussians, together with context dependent phone models. When a cross word context dependency is used, a large number of possible cross-word triphones are created, which are context dependent [2]. Hence we need to create generalised triphones that can share models across different contexts. Phonetic decision trees are therefore used to determine contextually equivalent sets of HMM states and to accomodate unseen triphones. By using decision tree state tying, in other words, parameter tying, we can reduce the number of parameters for acoustic modelling with triphones [3]. A separate decision tree is grown for every sub state of the HMM. Since there are 47 phonemes in total, a total of 141 separate trees are built. Each tree starts with all possible phonetic contexts represented in the root node. Then a binary question is chosen that can split the logical states represented by the node into two child nodes. The question that creates two new clustered states that can maximally increase the log likelihood of the training data is chosen at every node. This process is applied repeatedly until the log likelihood increase is less than a threshold at which point a final clustering step is performed. The choice of the threshold is important because

it directly affects the depth of the tree and therefore the final size of the acoustic model. Even though in a decision tree state tying approach, we create a tied state system using a phonetic decision tree, we however, still need to maintain balance between complexity and the available training data. For each triphone, during training we therefore accumulate sufficient statistics to train a single Gaussian per HMM state. Decision tree based approach is based on asking questions about the left and right contexts of each triphone, and we attempt to find the contexts which make the largest difference to the acoustics that can distinguish clusters. Decision tree tying based acoustic modelling is therefore used for modelling speech variations in large vocabulary speech recognition [3].

**Procedure**: With the given step-xwrd script, we create context-dependent triphone HMMs. The monophone transcriptions are converted to triphone transcriptions and a set of triphone models are created. The HTK command HLEd converts the monophone transcriptions. We created a script to perform triphone HMM training with a range of ROVAL and TBVAL values. ROVAL is used to determine the outlier threshold that determines the minimum number of occupancies in each cluster and can prevent a single outlier state from forming a singleton cluster just because it is acoustically very different from all other states. In comparison, the TBVAL determines the threshold for splitting, which is required each time the node is split based on the log likelihood. The decrease in log likelihood is calculated for merging terminal nodes with different parents, and any pair of nodes for which the log likelihood decrease is less than the threshold are therefore used to stop splitting and merge.

### A. *Experimental Results*

In this section, we trained the HMM models with different ROVAL and TBVAL values for the decision tree clustering. We then test the trained HMMs and examine the variations in the performance. Based on this, we can therefore the optimum values of the ROVAL and TBVAl for decision tree state clustering. For example, choosing an appropriate ROVAL value is based on trying to minimize the number of outliers and merge the outliers with the nearest neighbours such that we have less clusters, leading to less number of parameters to be estimated for the triphone models. Using decision tree clustering and models with shared parameters, we examine the number of clusters formed (hence the number of Gaussians) and examine how performance depends on the number of clusters.

Figure 5 shows how the number of clusters, and hence the number of Gaussians in the entire model depends on the threshold for outlier states. Figure shows that as we increase the threshold for outliers, the number of clusters decreases due to more state tying. This is because increasing ROVAL (RO in HTK) means that we are clustering more of the outlier states - even if the outlier states are acoustically quite different from other states. Hence high RO value corresponds to more clustering, so 'Number of Clusters' falls, as more states are clustered together. However, figure 5 also shows that as we decrease the TB threshold values with changing RO, there are more clusters (less tying). This is because TB affects the depth of the tree. Higher TB values means that the minimum log likelihood values are higher. Hence, smaller TB values also leads to more clusters, as shown in figure 6, since increased threshold values corresponds to less parameter tying (more clusters to form). When TB values are increased, we therefore have fewer parameters to estimate, as high TB influences more state tying. The values of TB and RO are also dependent on the number of training data available.

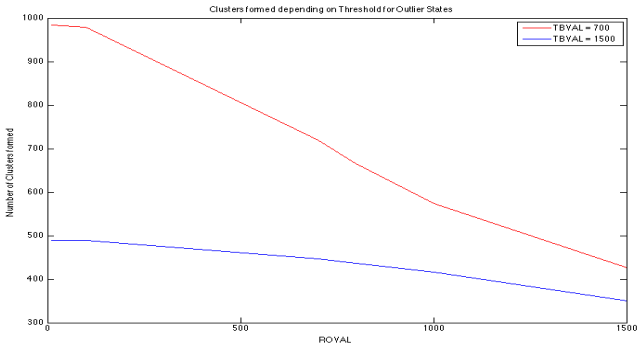


Figure 5. Number of Clusters as function of ROVAL

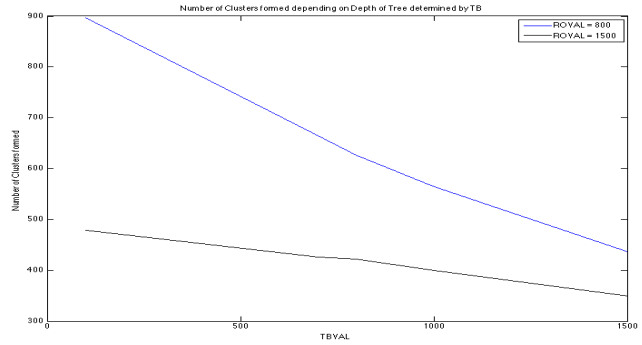


Figure 6. Number of Clusters as function of TBVAL

We then examine how the performance depends on the number of clusters formed. The number of final clusters formed after decision tree tying can be found from the log files looking at the below command where 12782 shows final clusters, and 60201 shows the initial nummber of states.

$TB : Stats 78 -> 10[12.8\%] 60201 -> 12782[21.2\%] total$

Figure 7 shows that as we increase the number of clusters, the accuracy for the triphone models falls. Higher number of clusters means that there is less parameter tying leading to more parameters to be estimated. This therefore means that with available training data, we are more likely to overfit. High number of parameter estimation with limited size training data corresponds to being more prone to overfitting. Therefore, the generalisation performance of the recogniser decreases. Hence, there is a tradeoff in achieving the high accuracy and the number of clusters formed. If too few clusters are formed,

then it also worsens performance, as further discussed below and shown in figure 9. Therefore, the threshold values must be balanced to achieve a desired level of accuracy, while also estimating desired parameters and avoiding overfitting.

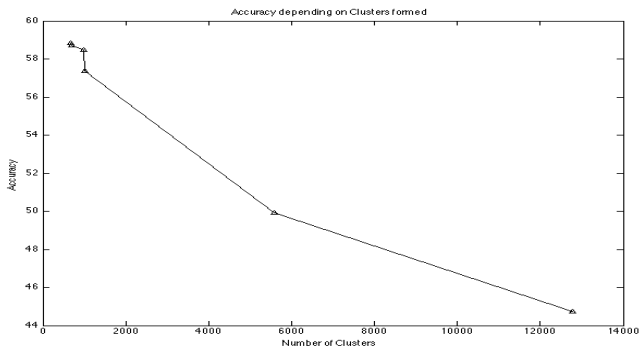


Figure 7. Accuracy falls with higher clusters

The relationship between thresholds and accuracy can be better explained as follows: For very small TB values, high number of clusters are formed due to less clustering of states (less state tying), whereas very high TB values corresponds to only one cluster due to more clustering of states (corresponds similar to a monophone). In figure 9 below, we show how accuracy depends on the number of mixture components and the threshold values. The plot shows that accuracy increases as we have more mixture components. However, it is also affected by the threshold values. $TBVAL = 800$ corresponds to less clusters compared to $TBVAL = 100$, as discussed before, which increases the accuracy of the recognizer. However, if TBVAL is too high to 40K, there is effectively only one cluster (same as a monophone model), that leads to a fall in accuracy as there is virtually no effectiveness of the decision tree. Very low TBVAL values corresponds to untied triphones (more separate clusters, indicating less tying). Figure 9 therefore proves that the TBVAL threshold values must be balanced to achieve high accuracy.

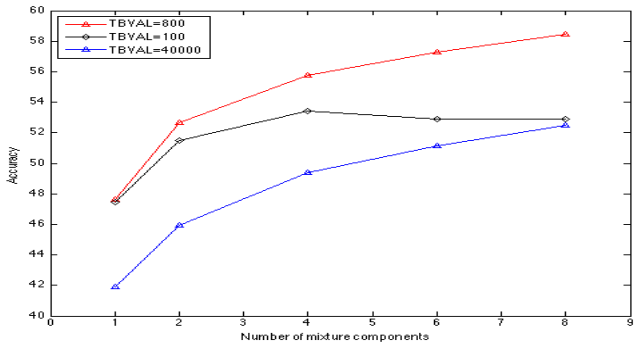


Figure 8. Accuracy increases for Mixture Components varying TBVAL

Finally, we present results of how accuracy depends on the number of mixture components. The results are carried out for $TBVAL = 0$ when there is no tying of triphones. This corresponds to more parameters that need to be estimated since there is no sharing of parameters with TBVAL 0 and ROVAL 0. Therefore, if we increase mixture components, overfitting leads to poor generalization in performance.

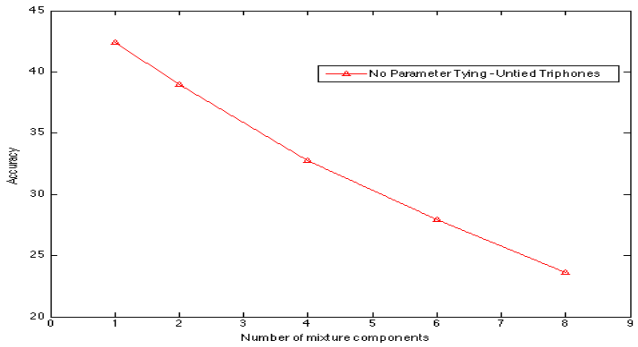


Figure 9. Untied Triphones - Overfitting with Higher Mixture Components

Furthermore, for decoding when using triphone models, we again need to determine the insertion word penalty for the unigram language model. Figure 10 again shows how the accuracy varies with the insertion penalty. This again suggests that for triphone acoustic models, we also need to optimize the insertion penalty. For this, an insertion penalty of $-30$ is desirable.

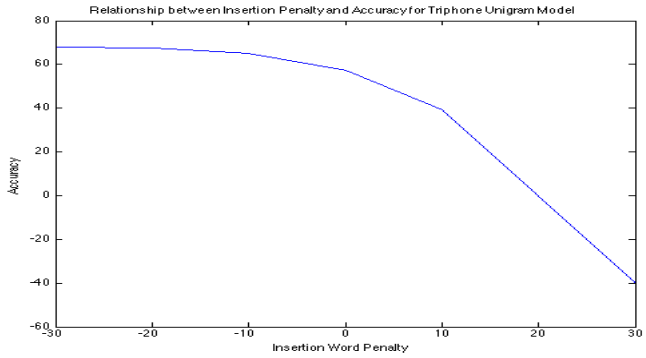


Figure 10. Accuracy vs Insertion Word Penalty for Triphone Models

Figure 11 shows an example of a decision tree based clustering of each monophone to tied triphones.

```
aa[3]
{
    0              'R_r'          -1          -5
   -1       'R_UnFortLenis'       -3          -2
   -2              'R_l'     "aa_s3_4"   "aa_s3_5"
   -3        'L_C-Front'          -4          -6
   -4          'L_Liquid'     "aa_s3_1"   "aa_s3_3"
   -5          'L_C-Back'     "aa_s3_6"   "aa_s3_7"
   -6      'L_Class-Stop'     "aa_s3_2"   "aa_s3_3"
}

aa[4]
{
    0          'R_Liquid'          -1      "aa_s4_5"
   -1           'R_Nasal'          -2      "aa_s4_4"
   -2        'R_Anterior'          -3      "aa_s4_3"
   -3      'R_NonAnterior'     "aa_s4_1"   "aa_s4_2"
}

ao[2]
{
    0          'L_C-Front'          -1      "ao_s2_4"
   -1              'R_r'            -2      "ao_s2_3"
   -2      'L_Unvoiced-cons'     "ao_s2_1"   "ao_s2_2"
}
```

Figure 11. Clustering in Decision Tree

### B. *Discussion of Results*

Based on the experimental results in IV-A, it can be seen that the clustering thresholds affects the performance of the system

when using context dependent triphone acoustic models. This is because too low a value of the thresholds means that there exists very high number of clusters (less parameter tying), which increases the number of GMM parameters to be estimated. On the other hand, if the threshold values are too high (both TB and RO), then even though we can cluster many states together to be in the same context (and reduce number of parameters), it significantly worsens the accuracy of the system during decoding. Hence, during acoustic model training, we need to maintain a balance between the depth of the decision tree (and number of shared parameters) and the accuracy. If there is less parameter tying, then with a small available training data, overfitting is more likely to occur, leading to poor generalization during decoding as more triphones have rarely been observed during training.

Based on figure 7 we therefore suggest that using threshold values that can lead to $0 - 2000$ clusters is usually ideal to maintain an accuracy of around $58\%$. Using very low RO values of 10 or 100 worsens performance. RO values of around 800 is ideal, since it leads to around 437 clusters to be formed. Similarly, using very low TB values of 100 leads to around 12000 and 5000 clusters to be formed which is worse for decoding. TB values should be chosen between 1000 to 1500 as it leads to forming around 700 or 500 clusters which in turn can maintain high accuracy of around $58\%$ during decoding with trained context dependent acoustic models. If there is less clustering in the system (hence more number of clusters), then effectively there is less tying and hence context dependency is ignored which in turns lowers the accuracy.

The performance also depends on optimizing the insertion penalty for the unigram language model. Based on figure 10 we therefore suggest that using a very low insertion penalty of around $-20$ or $-30$ for unigram language modelling is ideal.

## V. Biphone Models

In this section, we consider training and testing biphone models (for both right and left context). We use the same appraoch of parameter sharing for clustering biphone models to reduce the number of parameters. Similar to triphone models, we again consider analysing how performance of the recognizer depends on acoustic modelling of context-dependent biphones. The threshold values used for clustering are again changed, and the decision tree clustering is formed using suitable clustering questions. For example, for creating left biphones, only the questions related to left context are considered, and we ignored the right set of questions for each phone. Same approach is made for right biphones. Finally, we consider using an ideal set of threshold values for clustering, and optimize the insertion penalty again for the unigram language model. We evaluate the performance of the recognizer, and compare performance achieved using biphone and triphone models.

### A. *Experimental Results for Left Biphone Models*

Figure 12 further shows how the number of left biphone clusters are dependent on the RO values, keeping TB values constant. Again, figure 13 again shows that as we change TB values, keeping RO constant, the threshold again affects the depth of the decision tree and the number of shared parameters.

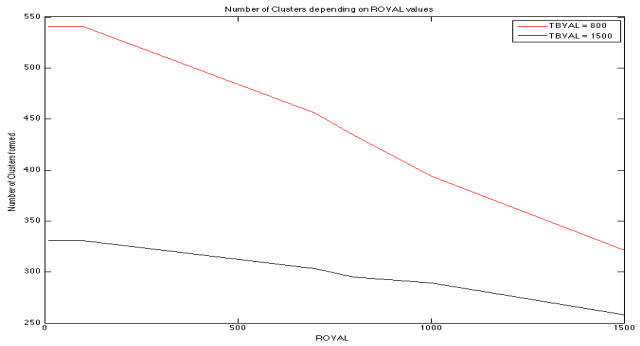


Figure 12. Number of Clusters for different ROVAL thresholds (Left Biphones)

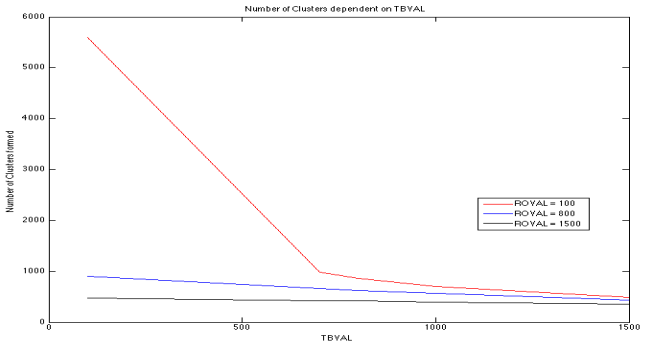


Figure 13. Number of Clusters for different TBVAL thresholds (Left Biphones)

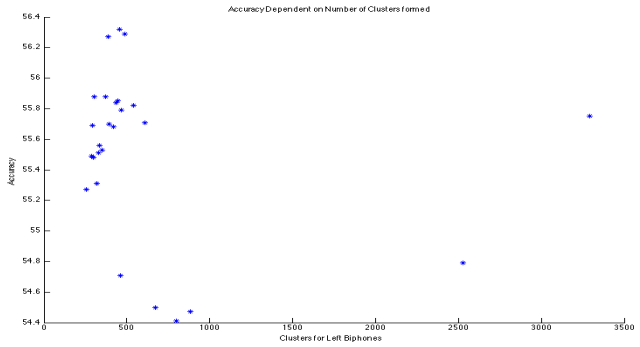


Figure 14. Accuracy dependent on Number of Clusters for Left Biphones

Figure 14 shows how the accuracy values depend on the number of clusters, and hence the number of Gaussians in the model. Figure 14 shows that the number of clusters which is dependent on the threshold values affects the performance of the recognizer. The same analysis of how the depth of the decision tree is determined by the threshold values applies here (similar to that of triphones).

### B. *Experimental Results for Right Biphone Models*

Similar to left biphones, we again present the results achieved with right biphone models for acoustic modelling, showing the effect of the theshold values on number of clusters, and how this in turn affects performance of the recognizer.

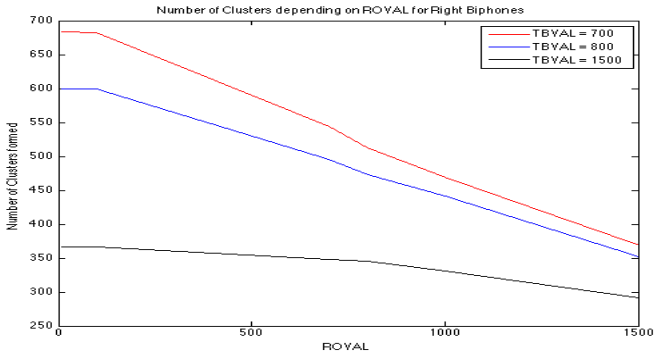


Figure 15. Number of Clusters for different ROVAL thresholds (Right Biphones)

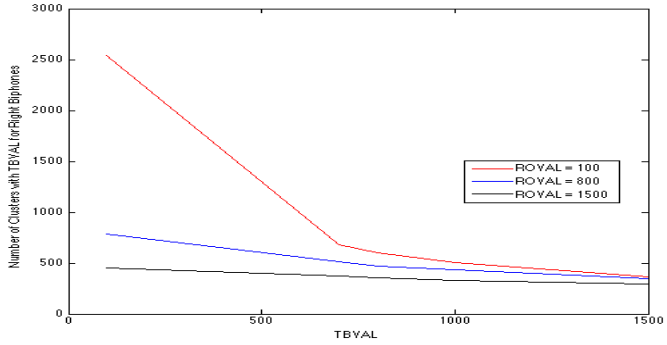


Figure 16. Number of Clusters for different TBVAL thresholds (Right Biphones)

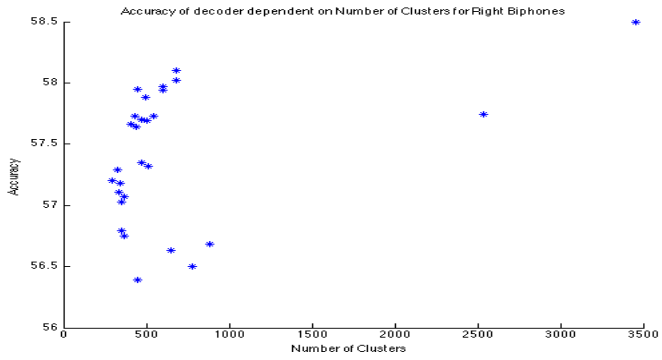


Figure 17. Accuracy dependent on Clusters for Right Biphones

**Insertion Penalty Biphone Acoustic Models**

The figures 18 below shows how the insertion penalty influences the accuracy values during unigram language modelling when using right biphone acoustic models (same effect for left biphones). Analysis of results is given in section V-C.

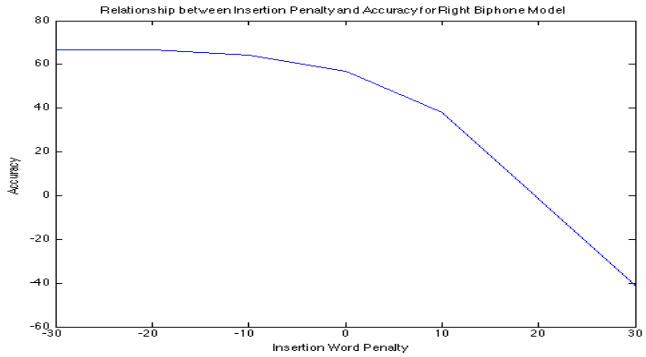


Figure 18. Insertion Penalty with Right Biphone Acoustic Models

### C. *Discussion*

Experimental results in above sections V-A and V-B again confirms that, even with biphone decision tree parameter tying, the number of clusters formed decreases as we increase the threshold values TB and RO (similar to results obtained for Triphone models). However, unlike triphone models, figures 14 and 17 now shows that even with very high number of clusters (hence higher number of Guassians in the model leading to more parameters), the accuracy when trained with biphone acoustic models does not fall significantly. This means that the number of parameters after tied biphones does not significantly influence the performance. This may be because we already have enough training data for the biphone models, and hence most of the biphones during decoding have already been observed from training. Therefore, even if we have more number of clusters (less biphone tying), the generalisation performance is maintained and the system avoids overfitting. For the available training data based on which biphone models have been created, this means that the tradeoff between accuracy and number of parameters is avoided.

*1)* ***Comparison of Results between Triphone and Biphone Acoustic Modelling:*** Notice that using triphone acoustic modelling, the maximum accuracy achieved was $58.5\%$ as shown in figure 7. Using biphone models, almost same level of accuracy can be obtained as shown in figure 17. However, for triphones, the accuracy was significantly influenced by number of parameters to be estimated (more dependency of number of clusters). This is not influential for biphone models, as irrespective of the number of clusters (influenced by threshold values), we can achieve similar levels of recogniser performance. This provides an advantage in using biphone models since it is less dominated by run-time and complexity unlike triphones. However, it is worth mentioning that the triphone models takes more context dependent information into account. Triphone models maybe more preferred in cases when large training data is available, such that even with less parameter tying, it can generalize better during decoding. In this case, biphones are less influened by parameters perhaps due to sufficient data available for modelling only one sided context information. This may not be the case for large vocabulary systems. In large systems, triphones maintaining similar run-times and accuracy as biphones will be more

preferred since it models better context dependent information. Overall, for similar accuracy values, triphone acoustic models maybe more preferred than biphones since they capture better acoustic information.

## VI. Bigram Language Models

The language model is further used to reorder these probabilities based on their likelihood of being a complete sentence. In this section, we consider using the bigram language model instead of the unigram model for evaluating the performance of the recognizer, with trained context dependent and context independent acoustic models. At first, we show our results achieved using monophone context independent models and evaluate how using a bigram language model the performance can be improved. Results then show evaluation with context-dependent acoustic models (both triphones and biphones). Finally, in the discussions section VI-D, we compare the effect of using bigram language models instead of unigram models, on both context dependent and context independent models.

### *A. Bigram Language with Context Independent Acoustic Models (Monophones*

In this section, we show the results of how the correction and accuracy values changes with the insertion word penalty and grammar scale. As discuccsed earlier, since we are using bigram language models, the grammar scale now scales the log transition probabilities of the language model, for the decoder to improve the phone recognition.

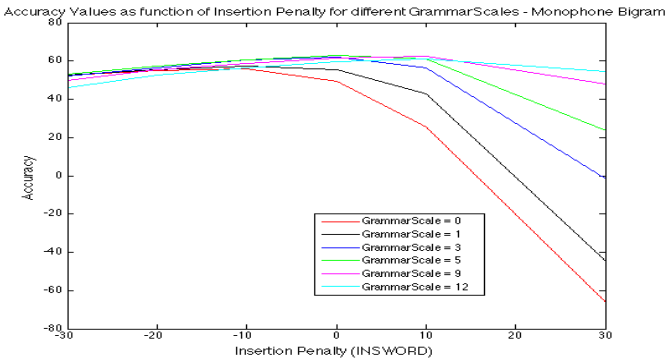


Figure 19. Accuracy dependent on insertion and grammar scale for Monophone with Bigram models

Figure 19 shows that as we increased the insertion word penalty, the accuracy values initially increases, becomes maximum and then starts decreasing. The accuracy values are usually high when there is an increase in insertions. However, this increase is affected by the grammar scale factor. As the grammar scale factor is increased from 0 to 12, we notice that there is an increase in accuracy values for the recognizer. Hence it can be suggested that a high grammar scale is favorable to maximise the test set accuracy when using bigram language models. Based on results above, an insertion penalty of $-10$ and a grammar scale of approximately 9 is favorable to maximize test set accuracy.

**Comparision betwee Unigram and Bigram Language Models for Context Independent Phone Models**: The language model does not significantly affect performance when combined with context independent acoustic models. A maximum accuracy of $58\%$ to $60\%$ is achieved for both unigram and bigram, as shown in figures 3 and 19. Similarly, the percentage of phones that are recognized correctly also remains relatively similar for both unigram and bigram models (both achieving a maximum corrections of $75\%$. **Explanation**: This may be because, the monophone models ignoring context dependency does not take into account the transitions between words or phones during training. Therefore, even if we assign transition probabilities to the langauge model, having the acoustic model trained with context independent phones plays a significant impact.

### *B. Bigram Language with Context Dependent Acoustic Models (Triphones)*

We show the results of how accuracy and corrections vary with the insertion penalty and the grammar scale factor, with using context dependent information.

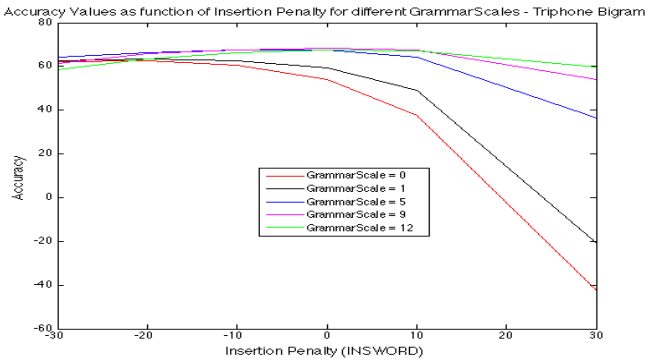


Figure 20. Accuracy dependent on insertion and grammar scale for Triphone with Bigram models

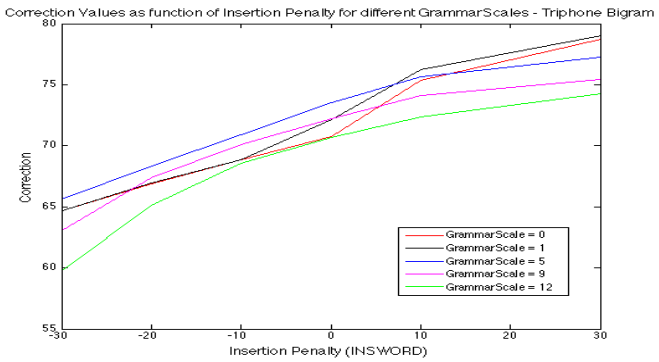


Figure 21. Correction dependent on insertion and grammar scale for Triphone with Bigram models

Similar to before, the results show that the accuracy increases as we fine tune the insertion word penalty. Again, an insertion word penalty of approxiamately 10 is suitable to maximize the test set performance of the recognizer. The results show that,

the recognizer performs best with a high grammar scale factor of approximately 9 or 12 with an insertion word penalty of 10.

**Comparison between Unigram and Bigram Langauge Models with Context Dependent Triphone Acoustic Models**: Using unigram with triphones, a maximum accuracy of $58.5\%$ was achieved. However, when using bigram models, the maximum accuracy is above $61\%$ as shown in figure 20. This suggests an improvement in performance when transition probabilities are assigned to langauge models, combined with modelling context dependent information. **Explanation**: The two models takes better account of transitions between words and also transitions between phones during utterance. The contex-dependent acoustic model takes phone variations and contexts into account, while bigrams can better model the transitions between phones due to which the overall system achieves better generalisation performance. The results suggest that using a bigram language model, combined with triphone acoustic models, relatively higher values of accuracy can be achieved compared to using unigram models.

### *C. Bigram Language with Context Dependent Acoustic Models (Biphones)*

We present results for right context biphone models with bigram langauge models (similar results not presented here is achieved for left context biphones).

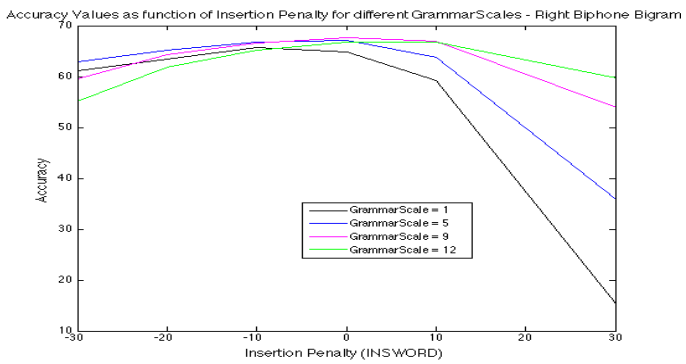


Figure 22. Accuracy dependent on insertion and grammar scale for Right Biphone with Bigram models

**Comparison between Unigram and Bigram Language Models with Biphone Acoustic Models**: As expected, with right biphone acoustic models, combined with bigram language, even better accuracy of almost $68\%$ can be achieved with an insertion penalty of 10. Like before, it suggests that using right biphone contexts, with the amount of available training data, overfitting is avoided with a balanced parameter tying. Hence, biphones achieve significantly better results. Notice that unlike using unigram models when maximum accuracy achieved was $58.5\%$, the bigram model offers significantly more advantages, improving the generalization performance by almost $10\%$ to $68\%$. Hence, bigram language models play a significant role when combined with context dependent models.

### *D. Discussion*

**Effect of Bigram Language on Context Dependent and Context Independent Models**: Bigram langauge models can improve accuracy by almost $10\%$ when used with biphones, and by almost $4\%$ when used with triphones compared to unigram models. Additionally, there is no increase in generalization performance when used with context independent models like monophones. The much higher increase in performance when used with biphones can be explained as follows. Bigram language models, assigning transiton probabilities conditioned only on the previous phone, also partially models context information. This context modelling is similar to that of using biphone contexts. For triphone contexts, perhaps, if we use da trigram langauge model, where transition probabilites are assigned to both previous and next phones similar to triphone context structure, better results may have been obtained. The bigram langauge model captures context information more similarly to biphone acoustic models. However, effect of it is still observed with triphones.

**Higher order n-gram language models**

Using trigram or n-gram language models, based on results above, it can be expected that higher order language models can perform even better with optimized grammar scale and word insertion penalties. This is because higher order language models can further capture the global dependencies between words or phones. Hence, with trained context dependent acoustic models, it is expected that the recognizer will do better alignment of words. Both the training models and the language model can now efficiently capture phone dependencies along long sequences (sentences instead of words). Bigram or trigram language models may only represent local constraints within few successive words. Compared to that, even higher order language models would have better ability to capture global or long distance dependencies between words. However, this will be computationally impractical. This is because dependencies are often independent of content and length of the word string. As suggested earlier, a trigram language model may offer more significant advantages when used with triphones compared to with biphones, since both captures context dependency in similar manner. However, this also further means that we will have a much larger network of the language model which might increase computational expense.

## VII. Overall Best System

The results above have evaluated and drawn differneces in performance when combining different models together. We can conclude that using triphone acoustic modeling combined with bigram language model, an overall performance accuracy of $63\%$ is satisfactory. It sufficiently captures context information, and also balances the number of parameters to be estimated by efficient state tying with decision tree based approach. Parameter estimation balancing is required since

using Gaussian mixture models, we want to sufficiently capture the output distributions of each state, while also ensuring that we do not use too many mixtures that can increase the overall number of parameters to be estimated during training. Given the small training data, this is a satisfactory level of performance. Results could have been even better with larger training data, as this system generalizes quite well having been trained with insufficient data. Even though the right biphone acoustic model with bigram language achieved a better result of $68\%$, our hypothesis is that this system performs only better since trained with small training set, and it would not generalize well. The system with biphone acoustic models seems less affected by parameter tying, which would not be true for large vocabulary systems. The biphone acoustic model only achieves slighly better results, since the accuracy is influenced by the bigram langauge model as well. Hence, to maintain a tradeoff between a system that estimates balanced number of parameters, achieves sufficient tying with optimized thresholds, insertions and grammar scale, while also achieving a good performance of $64\%$, the triphone acoustic model combined with bigram language model is the overall satisfactory system that can be found, given the amount of training data available.

## VIII. **Write Up Section 2**: Context Dependent Acoustic Models in Large Vocabulary Speech Recognition

In this section, we examine other ways to structure context dependent models. In particular, we focus on other approaches of forming tied triphones or biphones using a different decision tree approach compared to the conventional top-down tying approach. We examine how different methods for structing context information may affect the performance of the speech recogniser. Context dependent models such as triphone models are often limited using just the neighbouring phones for triphone HMMs. If the decoder only considers linear sequence of words or phones, then the context-dependent cross word triphones does not make any impact on the search space of the decoder. This means, the search space for the decoder during testing remains somewhat indifferent to the context-dependent phone models.

[4] takes the approach of forming a global decision tree for parameter state tying. This can be achieved by withdrawing the restriction that logical states from different phones or different states cannot share the same clustered state. For example, in the conventional decision tree approach, a clustered state associated with state 1 of one HMM can never be associated with a different state of any other HMM. [4] takes the approach of allowing cross-center phone and cross-state clustering, such that the acoustic model can better describe the acoustics of the training data. Here, a different decision tree is not built for every context independent phone state and uses a two step look ahead for decision tree questions, such that each question is evaluated by looking at the likelihood increase induced by its direct children. **Design, Complexity and Run-Time**: The global decision tree approach can reduce the model size significantly. This means, by reducing the number of parameteres required to be estimated, we can achieve advantages during large vocabulary speech recognition. Furthermore, this approach might allow using higher Gaussian mixture components for each state to further improve accuracy. Our approach before discussed that too many mixture components increases the required estimation of parameters, which becomes difficult with small training data. A global decision tree approach discussed by [4] can reduce number of model parameters by almost $50\%$.

[5] discusses how context dependent models can be formed by pooling all vowels in one cluster, and all consonants in another cluster. This provides the advantage that some parameters are shared by HMMs that are of different phones. The parameter sharing model is done by tying HMM states across different target phones. This is the approach of forming context dependent models to word specific models. This is done so that it can work for function of words which are very frequent and often unstressed. A related idea is to use lexical stress information for acoustic modelling that offers the advantages of using language information being language independent. **Accuracy**: [5] shows that using lexical stress information, it can improve phone recognition accuracy even with very few mixture components. However, such a method may only be effective when a large amount of training data is available. One obvious drawback of this approach for large systems is that using lexical stress information may significantly increase the number of parameters to be estimated, and lead to overfitting during training, further lowering generalization performance.

Another method to improve the robustness and accuracy of acoustic modelling was considered in [6]. It discusses combining the conventional decision tree clustering approach with agglomerative clustering of rare acoustic phonetic events, such that it can incorporate both phonetic and nonphonetic features. Using this approach, rarely seen triphones in the training data are clustered into generalized triphones such as to improve the coverage of the acoustic modelling stage. This method allows training data sharing across various conditions in contrast to conventional approach that faces training data depletion. Therefore, it allows tying generalized phonetic and nonphonetic features (such as gender and position). Hence, due to this condition dependent acoustic modelling, the robustness of a condition and context dependent model can be increased.

[7] also shows that better results can be obtained by a priori selecting a set of states that can be clustered, instead of solely relying on the acoustic similarity. This can have reductions in the run-time of the algorithm, especially in cases when a large amount of monophone transcriptions need to be clustered together to form triphones. [7] discusses a bottom-up approach in clustering for robust acoustic modelling, where a stopping criterion for the furthest neighbouring clustering procedure

is proposed that does not require any threshold (like the ROVAL values we tested with). Furthermore, a top-down approach can also be deployed that uses a selected impurity function for lookahead search during decision tree clustering that outperforms the classical decision tree growing algorithm.

In our work, we considered deciding the set of questions when creating biphone models from monophones. [8] discusses how can questions for decision tree based clustering can be automatically generated without manual labor. [8] discusses automatically defining a good set of phonetic questions for a phoneme set. Furthermore, [9] contains a study of the methods typically being used for obtaining context dependent models, and discusses issues of constructing decision trees, such as the choice of the partitioning method at each node, goodness of split criterion and the method for determining appropriate tree sizes. Other approaches to phonetic decision tree state tying for acoustic modelling were further discussed in [10], [11], [12], [13], [14] . All the related work considers efficient diverse ways of decision tree based clustering that can improve the run time and complexity of the algorithm during training of acoustic models in large vocabulary systems.

**Improving Language Models for Large Vocabulary Speech Recognition**

Language models also play an important role in speech recognition systems in addition to acoustic modelling. In our work we considered both unigram and bigram language models. Here, we include related work for improving the language models. [15] takes the approach of early incorporation of language model information such as to use all available language model information in one pass decoder. Using this, [15] shows that the search during decoding can be speeded up by almost a facotr of three without introducing additional search errors. Using early incorporation of language model information, it is possible to use tighter pruning thresholds that can lead to a more precise beam search and hence more efficient decoding. If the search space during decoding can be efficiently organized, then the recogniser performance can be improved by implementing a one-pass search strategy. [16] further considers a language model lookahead technique where it is possible to increase the number of tokens that can be pruned without loss of decoding precision. In cases when the language model is extremeley high, it might become computationally expensive to perform an exhaustive search where there are a large number of possible paths (hence large number of tokens) from which we need the optimal path. [17] considers a number of decoding strategies that can be used for large vocabulary systems from the viewpoint of search space representation. Other approaches for single pass decoding are discussed in [18], [19], [20], [21].

### A. Discussion

In the above related work, it was discussed that using variations in decision tree state tying for acoustic modelling, significant improvements can be made in recognizer performance. While some approaches are based on modelling better context information, others significantly take account of how state tying can be efficiently done, reducing number of parameters to be estimated, for large scale speech recognition systems where complexity and run-time is of significant importance. We further considered different approaches to language modelling, which may also influence the recogniser performance. Recent work has also shown large scale languge modelling in speech recognition where word error rates as low as $6\%$ depending on language model size, lattice scoring and amount of training data used [22]. Other work also considered lattice-based framework for maximum mutual information estimation (MMIE) of HMM parameters to train HMM systems. [23] considered lattice based framework for discriminative training large speech recognition systems for conversational telephone speech transcription.

## IX. Conclusion

In this work, we considered a GMM-HMM based phone recognition system, and analysed how performance of the recognizer depends on both the acoustic models and the language models. Our work showed that choosing appropriate initializations is important depending on training data available. We considered the effect of differential coefficients on phone error rates, and the impact of Gaussian mixture components and the number of parameters to be estimated on the recogniser performance. Our work considered optimizing threshold values that affect decision tree state tying approach, for both triphone and biphone acoustic models. We showed results of how including context dependent information can improve phone recognition performance. Finally we considered the significance of langauge modelling, and showed that using bigram language models, combined with context-dependent acoustic models, significantly better results can be achieved. Our work commented on an overall best system suitable for phone recognition based on a limited size training corpus.

## X. Acknowledgment

The author would like to thank the Cambridge University Engineering Department for providing access to the HTK toolkit. We would like to thank NVidia for providing the servers required for experiments. We thank P.C. Woodland for the enormous support through teaching, and C. Zhang for developing initial scripts to run the experiments. Finally, we thank the Cambridge University MPhil Machine Learning, Speech and Langauge Technology program for providing the practical experimentation opportunity.

## References

[1] Mark J. F. Gales and Steve J. Young. The application of hidden markov models in speech recognition. *Foundations and Trends in Signal Processing*, 1(3):195–304, 2007.

[2] Kai-Fu Lee. Context-dependent phonetic hidden markov models for speaker-independent continuous speech

recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 38(4):599–609, 1990.

[3] Steve J. Young, J. J. Odell, and Philip C. Woodland. Tree-based state tying for high accuracy modelling. 1994.

[4] Jasha Droppo and Alex Acero. Experimenting with a global decision tree for state clustering in automatic speech recognition systems. pages 4437–4440, 2009.

[5] J. Hogberg and Kåre Sjölander. Cross phone state clustering using lexical stress and context. 1996.

[6] Wolfgang Reichl and Wu Chou. Robust decision tree state tying for continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 8(5):555–566, 2000.

[7] Cristina Chesta, Pietro Laface, and Franco Ravera. Bottom-up and top-down state clustering for robust acoustic modeling. 1997.

[8] Klaus Beulen and Hermann Ney. Automatic question generation for decision tree based state tying. pages 805–808, 1998.

[9] Harriet J. Nock, Mark J. F. Gales, and Steve J. Young. A comparative study of methods for phonetic decision-tree state clustering. In *Fifth European Conference on Speech Communication and Technology, EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997*, 1997.

[10] Jacques Duchateau, Kris Demuynck, and Dirk Van Compernolle. A novel node splitting criterion in decision tree construction for semi-continuous hmms. 1997.

[11] Roland Kuhn, Ariane Lazaridès, Yves Normandin, and Julie Brousseau. Improved decision trees for phonetic modeling. pages 552–555, 1995.

[12] Ariane Lazaridès, Yves Normandin, and Roland Kuhn. Improving decision trees for acoustic modeling. 1996.

[13] Daniel Willett, Christoph Neukirchen, Jörg Rottland, and Gerhard Rigoll. Refining tree-based state clustering by means of formal concept analysis, balanced decision trees and automatically generated model-sets. pages 565–568, 1999.

[14] Douglas B. Paul. Extensions to phone-state decision-tree clustering: single tree and tagged clustering. pages 1487–1490, 1997.

[15] Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel. Efficient language model lookahead through polymorphic linguistic context assignment. pages 709–712, 2002.

[16] Marijn Huijbregts, Roeland Ordelman, and Franciska de Jong. Fast n-gram language model look-ahead for decoders with static pronunciation prefix trees. pages 1582–1585, 2008.

[17] Xavier L. Aubert. An overview of decoding techniques for large vocabulary continuous speech recognition. *Computer Speech & Language*, 16(1):89–114, 2002.

[18] Julian James Odell. The use of context in large vocabulary speech recognition, 1995.

[19] Xavier L. Aubert. One pass cross word decoding for large vocabularies based on a lexical tree search organization. 1999.

[20] Michael Finke, Jürgen Fritsch, Detlef Koll, and Alex Waibel. Modeling and efficient decoding of large vocabulary conversational speech. 1999.

[21] Janne Pylkkönen. An efficient one-pass decoder for finnish large vocabulary continuous speech recognition. In *in Proceedings of Second Baltic Conference on Human Language Technologies, 2005*, pages 167–172.

[22] Ciprian Chelba, Dan Bikel, Maria Shugrina, Patrick Nguyen, and Shankar Kumar. Large scale language modeling in automatic speech recognition. Technical report, Google, 2012.

[23] Philip C. Woodland and Daniel Povey. Large scale discriminative training of hidden markov models for speech recognition. *Computer Speech & Language*, 16(1):25–47, 2002.