FILLER MODELS FOR AUTOMATIC SPEECH RECOGNITION CREATED FROM HIDDEN MARKOV MODELS USING THE K-MEANS ALGORITHM

*Matthew E. Dunnachie†, Paul W. Shields‡, David H. Crawford‡, and Mike Davies**

† Institute for System Level Integration, Alba Centre, The Alba Campus, Livingston, EH54 7EG, United Kingdom
‡ Epson Scotland Design Centre, Integration House, The Alba Campus, Livingston, EH54 7EG, United Kingdom
*School of Engineering and Electronics, University of Edinburgh, The King's Buildings, Mayfield Road, Edinburgh, EH9 3JL, United Kingdom
email: ed.dunnachie@sli-institute.ac.uk web: www.sli-institute.ac.uk

ABSTRACT

In Automatic Speech Recognition (ASR), the presence of Out Of Vocabulary (OOV) words or sounds, within the speech signal, can have a detrimental effect on recognition performance. One common method of solving this problem is to use filler models to absorb the unwanted OOV utterances. A balance between accepting In Vocabulary (IV) words and rejecting OOV words can be achieved by manipulating the values of Word Insertion Penalty and Filler Insertion Penalty. This paper investigates the ability of three different classes of HMM filler models, K-Means, Mean and Baum-Welch, to discriminate between IV and OOV words. The results show that using the Baum-Welch trained HMMs 97.0% accuracy is possible for keyword IV acceptance and OOV rejection. The K-Means filler models provide the highest IV acceptance score of 97.3% but lower overall accuracy. However, the computational complexity of the K-Means algorithm is significantly lower and requires no additional speech training data.

1. INTRODUCTION

Automatic Speech Recognition (ASR) is an enabling technology that facilitates a speech interface to electronic devices or systems, permitting speech to be used as the primary mode of communication for an electronic device. One challenge that ASR systems must overcome is identifying command words embedded within speech. When interacting with an ASR system users tend to surround commands with extra words or sounds that are not part of the system's vocabulary. The presence of OOV words within the user's speech has a detrimental effect on recognition performance. To maintain suitable levels of recognition accuracy and allow users to interact with the system in a natural manner, ASR systems need to model both IV words and OOVs and manage them in an appropriate way.

Wilpon et al. [1] [2] described this tendency to add extra words to commands when presenting the results of research into ASR across the telephone network. Rose and Paul [3] discuss different ways to solve this problem. ASR systems use language models to define the set of allowable words and phrases. One solution to the OOV problem is to build a language model that contains all possible words; however, it is not possible to include all words in a finite grammar. This method creates a very large language model, which takes significant effort to create, is expensive in terms of system resources with a large portion of the system's vocabulary never used. The size of such a language model also precludes its use in smaller, embedded ASR systems. Another approach is to attempt to simplify the language model by restricting the words or phrases to a small closed grammar, but this can often feel unnatural to the user. The solution presented in this paper involves using filler models, also known as OOV models or garbage models, to absorb any extraneous words or sounds in the user's speech. This approach, known as keyword spotting, allows the user to speak in a natural way, while the ASR system ignores those words that are not part of the desired language model.

This paper presents a novel way of creating filler models using the K-Means algorithm. The recognition accuracy of an ASR system is measured using the K-means and alternative filler models. The performance of the filler models created using the K-Means algorithm is similar to those created using the Baum-Welch method.

The cost of false positives, classifying OOVs as IVs, and false negatives, classifying IVs as OOVs, is different. Creating a system that has both a low False Positive Rate (FPR) and a low False Negative Rate (FNR) can be difficult. Bou-Ghazale and Asadi [4] have presented a system with a low false alarm rate, 0.55%, but a higher false rejection rate, 15%. The filler models presented here give a balanced approach to dealing with IV and OOV utterances, providing a low false positive rate and a low false negative rate.

The remaining parts of this paper are arranged as follows. In Section 2 an overview of Hidden Markov Models (HMMs) is provided, while in Sections 3 and 4 filler models and the K-Means algorithm are discussed. The simulation environment is described in Section 5 and the simulation results are provided in Section 6. Analysis of the simulation results are discussed in Section 7. Conclusions and further work are summarised in Section 8.

2. HIDDEN MARKOV MODELS

Hidden Markov Models [5] [6] may be used to represent the sequence of sounds within a section of speech. Each elemental speech sound, known as a phoneme, can be modelled by an individual HMM. The probability of the input speech feature vector matching the HMM is used to identify the words spoken. HMMs are stochastic state machines where the current state is not directly observable; an HMM emits an observable symbol per state. The probability of an HMM emitting a symbol is modelled by a mixture of Gaussian distributions, as described in equation (1).

$$b_j(x) = \sum_{m=1}^{M} C_{mj} N\left[x, \mu_{mj}, U_{mj}\right] \qquad (1)$$

Where x is the feature extracted from the speech e.g. mel frequency cepstral coefficient, C_{mj} , μ_{mj} and U_{mj} are the coefficient, mean vector and covariance for mixture component m in state j.

HMMs are typically created using an iterative training method called the Baum-Welch algorithm, which uses a set of training data to estimate the HMM model parameters. Starting with a prototype HMM, the Baum-Welch algorithm adjusts these parameters to maximise the likelihood of observing the data. The HMMs presented in this paper were trained using the Hidden Markov Training Kit (HTK) [7] and the training data was extracted from the SpeeCon UK-English Database [8].

3. FILLER MODELS

Filler models are used to represent OOV words. They allow an ASR system to classify incoming speech as either IV or OOV without having to define explicitly an OOV word. In the system being considered for this paper, each IV phoneme is modelled by a single HMM; however the filler model HMMs represent multiple sounds and therefore are more general than the IV phoneme HMMs. Filler models can represent the entire set of speech sounds or subsets. The performance of four different classes of filler model is evaluated in this paper. The simplest class of filler model contains one HMM to represent all of the speech phonemes; this is the “single” filler model. The next class of filler model uses two separate HMMs. The “VNC” filler model has one HMM to represent vowels and another for consonants. The “VUV” filler model uses an HMM for voiced and one for unvoiced phonemes. The final class of filler model uses three HMMs to represent vowels, voiced consonants and unvoiced consonants. This set of filler models was called “VCVUV”, vowel consonant voiced and unvoiced. When multiple filler models are available: VNC, VUV and VCVUV, they are used in parallel. This means that the system can select between the IV phonemes and multiple filler models simultaneously.

The filler models have the same format as the IV phoneme HMMs; this allows the ASR system to process them both in the same way. Three methods were used to create the filler models. The first method was the Baum-Welch algorithm which was used to create the filler models labelled “Trained HMM”. The other two methods were the Mean method and the K-Means method. The Baum-Welch algorithm operates on features extracted from the speech contained within the training database. The Mean and K-Means algorithms utilise the IV phoneme HMM Gaussian mixture component means and covariances. The Mean method calculates the m-th component j-th state mean vector as the mean of all the m-th component j-th state mean vectors for the individual IV phonemes models. The filler model component coefficients and covariances are calculated using the same methodology. The K-Means method uses K-Means clustering to create the filler models; a detailed description of this method is provided in the next section.

By combining the different creation methods and different numbers of HMMs used in the filler models, 12 different filler models were available for simulation. The 12 different filler models are listed in Table 1.

Filler Model	Nº HMMs	Creation Method
Single Mean	1	Mean
Single K-Means	1	K-Means
Single Trained HMM	1	Baum-Welch
VNC Mean	2	Mean
VNC K-Means	2	K-Means
VNC Trained HMM	2	Baum-Welch
VUV Mean	2	Mean
VUV K-Means	2	K-Means
VUV Trained HMM	2	Baum-Welch
VCVUV Mean	3	Mean
VCVUV K-Means	3	K-Means
VCVUV Trained HMM	3	Baum-Welch

Table 1 : List of Filler Models Created and Simulated

4. K-MEANS ALGORITHM

The K-Means algorithm is an iterative clustering algorithm. Cluster membership is based on a measure of the data points’ similarity. A measure commonly used is the Euclidean distance from a data point to a cluster’s mean. Using this measure, a data point is associated with the cluster closest to it. The cluster’s mean is then recalculated and the process continues until a predefined stop criterion is met.

The K-Means algorithm uses the IV phoneme HMM Gaussian mixture component mean vectors, μ_{mj}, as the data points for the clustering process. The final cluster means are used as the mean vectors for the mixture components in the filler models. The covariances were calculated using the same methodology as those for the Mean filler models. The component coefficients were calculated to be one over the number of components within a mixture i.e. 1/8 for 8 mixture components. The algorithm is run separately for each emitting state in the HMM and creates a filler model that has the same format as the IV phoneme HMMs. The number of clusters used by the K-Means algorithm matches the number of Gaussian mixture components used for each IV phoneme HMM state. The Gaussian mixture component mean vectors for groups of IV phonemes are used to create different filler model classes, i.e. vowels, consonants, voiced, unvoiced, etc.

The K-Means algorithm has two main deficiencies: a local minimum, not necessarily a global minimum, is often found, and results are very dependent upon the choice of initial cluster means. A common method of initialising the cluster means is to select data points randomly. The algorithm is then run multiple times to ensure that it converges. Two alternatives to this random method were investigated. It was found that the filler models produced by these methods gave higher recognition accuracies. The first method involved the use of Principal Component Analysis (PCA) to reduce the dimensionality of the data points from 39 down to 3 and then select data points that were evenly distributed across this 3-D space as the initial cluster means. The second method calculated the Euclidean distance of each data point from the origin, ordered the data points in terms of this distance, and then selected every n^{th} data point as an initial cluster mean, where n is the number of speech phonemes used to provide the data points. The second method resulted in the filler models with the highest recognition accuracies and was the preferred method of choosing the initial cluster means when the K-Means algorithm was used.

5. SIMULATION METHODOLOGY

5.1 Speech File Selection

The speech files used to test the filler models were selected from the SpeeCon database. This database contains a wide variety of words: objects, people's names, place names, commands, digits and internet addresses. The recordings were made in several different acoustic environments and with a wide range of Signal to Noise Ratios. Using this database ensured that the filler models were tested under real-world conditions. The simulations used 8 different keywords with approximately 270 instances of each, resulting in 2217 different speech files being used. The number of instances, mean, min and max Signal to Noise Ratio (SNR) for each keyword are listed in Table 2 .

Keyword	**Nº of Instances**	**SNR (dB)**		
		Mean	Min	Max
Assistant	285	20.3	4.2	38.0
Battery	274	22.5	3.8	35.8
Camera	294	23.3	4.0	37.7
Camcorder	283	22.6	5.6	35.0
Clock	289	23.1	5.9	36.5
Computer	268	21.7	3.2	45.8
Microphone	244	22.4	4.8	35.9
Radio	280	24.7	4.9	36.6
Total Nº of Files	2217			

Table 2 : Number of Speech Files per Keyword

5.2 Filler Model Simulations

The performance of each filler model was measured by determining its ability to discriminate between IV and OOV words. For each simulation, one keyword was selected as IV and the remaining keywords were classed as OOV, the system's ability to discriminate between IVs and OOVs was measured and the process repeated for all of the keywords. The language model used for these simulations is provided in Figure 1. In this language model the speech sequence starts and ends with silence, (Sil); the recogniser can select between the keyword, filler model, silence and short pause, (SP). Short pause is the small period of silence between words. The keyword can only be spoken once but silence, short pause or the filler model can be repeated any number of times.

In order to maximise the performance of each filler model simulations were performed with a range of Word Insertion Penalty (WIP) and Filler Insertion Penalty (FIP) [9]. The values of WIP and FIP that gave the highest combined keyword and OOV accuracy, for that particular filler model, were identified and used in the filler model performance comparisons.

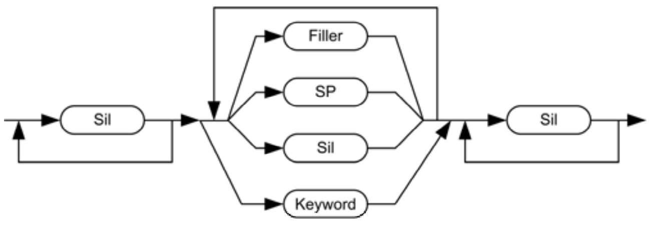

Figure 1 : Language Model for Filler Model Simulations

5.3 Calculation of Accuracy

The HResults function, from HTK, was used to compare a reference file with the output of the ASR system and determine if the keyword or OOV was correctly identified. The recognition accuracy of correctly recognised keywords and rejected OOVs were then calculated separately. A Receiver Operating Characteristics (ROC) curve was used to compare the performance of the different filler model types by plotting False Positive Rate (FPR) against True Positive Rate (TPR).

6. SIMULATION RESULTS

The percentages of correctly identified keywords and rejected OOVs are listed in Table 3 and displayed in Figure 2. The false positive rate and true positive rate for each of the filler models are listed in Table 4 and used to create the ROC curve plotted in Figure 3.

Filler Model	**Correctly Identified Keywords**	**Correctly Rejected OOVs**	**Combined Accuracy**
Single Mean	76.5%	89.3%	82.9%
Single K-Means	91.8%	94.4%	93.1%
Single Trained HMM	94.3%	97.6%	96.0%
VNC Mean	86.9%	94.7%	90.8%
VNC K-Means	97.3%	94.0%	95.7%
VNC Trained HMM	97.0%	96.9%	97.0%
VUV Mean	85.0%	94.5%	89.8%
VUV K-Means	92.8%	95.5%	94.2%
VUV Trained HMM	95.2%	97.5%	96.4%
VCVUV Mean	93.2%	94.8%	94.0%
VCVUV K-Means	96.8%	95.6%	96.2%
VCVUV Trained HMM	97.0%	97.0%	97.0%

Table 3 : Filler Model Recognition Accuracy

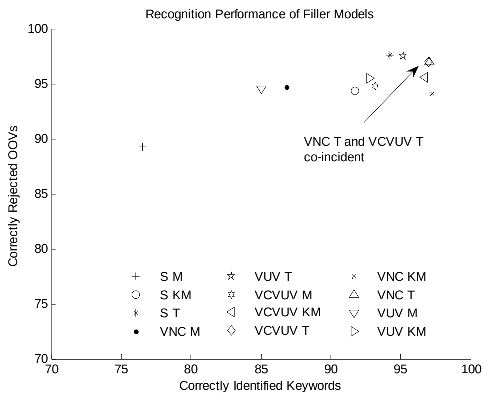

Figure 2 : Keyword Acceptance vs OOV Rejection

Filler Model	FPR	TPR
Single Mean	0.11	0.76
Single K-Means	0.06	0.92
Single Trained HMM	0.02	0.94
VNC Mean	0.05	0.87
VNC K-Means	0.06	0.97
VNC Trained HMM	0.03	0.97
VUV Mean	0.05	0.85
VUV K-Means	0.04	0.93
VUV Trained HMM	0.02	0.95
VCVUV Mean	0.05	0.93
VCVUV K-Means	0.04	0.97
VCVUV Trained HMM	0.03	0.97

Table 4 : FPR and TPR for Filler Models

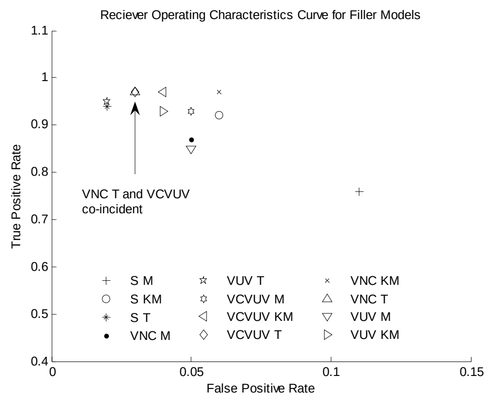

Figure 3 : ROC Curve for Filler Models

The highest scoring keyword for each filler model and corresponding keyword accuracy score are listed in Table 5. In Table 6 the number of phonemes, the number of voiced to unvoiced or unvoiced to voiced phoneme transitions and the mean keyword accuracy for each keyword are tabulated. Figure 4 plots keyword accuracy against the number of phonemes in the keyword, while Figure 5 plots the keyword accuracy against the number of voiced unvoiced transitions in the keyword.

Filler Model	Highest Scoring Keyword	Highest Keyword Accuracy
Single Mean	Assistant	95.1%
Single K-Means	Assistant	99.6%
Single Trained HMM	Assistant	99.0%
VNC Mean	Assistant	99.3%
VNC K-Means	Assistant	99.3%
VNC Trained HMM	Assistant	98.2%
VUV Mean	Microphone	95.9%
VUV K-Means	Microphone	98.0%
VUV Trained HMM	Camcorder	98.6%
VCVUV Mean	Assistant	98.2%
VCVUV K-Means	Assistant	99.0%
VCVUV Trained HMM	Battery	98.5%

Table 5 : Highest Performing Keyword for Filler Model

Keyword	Nº Phonemes	Nº V/UV Transitions	Mean Keyword Accuracy
Assistant	8	5	97.7%
Battery	6	2	94.4%
Camcorder	7	3	95.6%
Camera	6	1	79.0%
Clock	4	2	81.6%
Computer	8	5	94.4%
Microphone	8	4	97.2%
Radio	5	0	96.0%

Table 6 : Mean Accuracy for Keywords

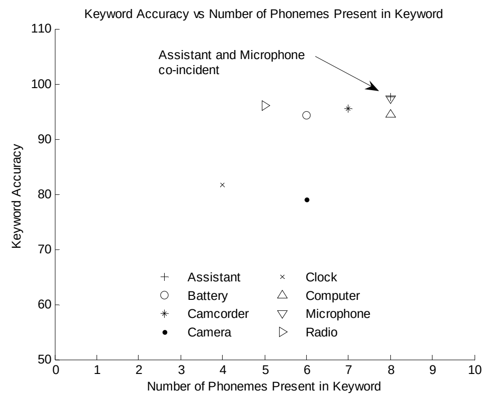

Figure 4 : Accuracy vs Number of Phonemes

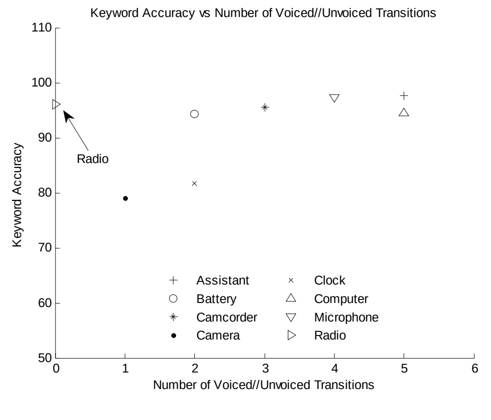

Figure 5 : Accuracy vs Number of Voiced/Unvoiced Transitions

7. DISCUSSION

The simulation results presented in Table 3 confirm that it is possible to create filler models with high recognition accuracy and balanced rates of IV acceptance and OOV rejection. The VNC Trained HMM and VCVUV Trained HMM models have similar IV acceptance and OOV rejection rates, approximately 97%. These two filler models also have the lowest difference between the two rates; 0.1% for VNC and 0.0% for VCVUV. This balance was achieved by manipulating the values of WIP and FIP to change the operating point of the ASR system. The highest keyword accuracy is achieved by the VNC K-Means filler, 97.3%, however this is off-set by a lower OOV rejection rate of 94%. The lowest keyword accuracy comes from the Single Mean model which also has the lowest OOV rejection rate, 76.5% and 89.3% respectively. Comparing the combined accuracy values of the different implementation methods for each filler model type it can be seen that the highest performance is achieved by the trained HMM models; higher IV acceptance and OOV rejection rates. Conversely the Mean method filler models have the lowest performance. The VNC K-Means filler model is the exception to this trend as it has higher IV acceptance than the VNC trained but lower OOV rejection than the VNC Mean. Using a filler model with more than one HMM can improve the recognition performance. The VNC and VUV models have higher accuracies than the single model for all implementation methods and the VCVUV filler model is the best performing model for Mean and K-Means but is marginally worse for the Trained HMM method.

The selection of an appropriate keyword has an impact on the recognition performance of the ASR system. Table 5 shows that the longest keywords generally have the highest recognition accuracy, i.e. Assistant and Microphone. When keyword recognition accuracy is plotted against the number of phonemes present in the keyword, Figure 4, keywords with increasing phoneme length exhibit higher recognition accuracy. When the keyword recognition accuracy is plotted against the number of voiced to unvoiced transitions within the keyword, Figure 5, there is a similar relationship. This would suggest that to maximise recognition accuracy a long keyword with 5 or more phonemes and 3 or more voiced to unvoiced phoneme changes should be selected.

8. CONCLUSIONS

This paper presents the results from a series of experiments evaluating the performance of a keyword speech recognizer using 12 different HMM based filler models. Three different methods of generating the filler model HMMs were evaluated: Mean, K-Means and Baum-Welch. Each of the three methods was used to create filler models with 1 or more HMMs: Single, VNC, VUV and VCVUV. The VNC and VCVUV filler models created using the Baum-Welch algorithm have superior overall performance compared to the filler models created using either the Mean or K-Means algorithms. However, the Baum-Welch trained HMMs performance advantage was only 0.8% to 2.9% over the K-Means generated HMMs. The VNC K-Means filler models offered the highest keyword detection score of 97.3%. The filler models created using the Mean method had the lowest performance, as much as 13.1% lower than the Baum-Welch trained HMMs. The K-Means algorithm is much less computationally intensive compared to the Baum-Welch algorithm, assuming the trained speech phonemes are available, as there is no additional training requirement. This provides the flexibility and simplicity of producing high quality OOV filler models when the original speech training data is not available. When considering the selection of a keyword, 5 or more phonemes and a minimum of 3 voiced to unvoiced phoneme transitions (and vice versa) were found to consistently offer superior performance.

REFERENCES

[1] Wilpon, J.G., D.M. DeMarco, and R.P. Mikkilineni. Isolated word recognition over the DDD telephone network. Results of two extensive field studies. in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. 1988.

[2] Wilpon, J.G., et al., Automatic recognition of keywords in unconstrained speech using hidden Markov models. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 1990. 38(11): p. 1870-1878.

[3] Rose, R.C. and D.B. Paul. A hidden Markov model based keyword recognition system. in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. 1990.

[4] Bou-Ghazale, S.E. and A.O. Asadi. Hands-free voice activation of personal communication devices. in *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*. 2000.

[5] Rabiner, L.R., A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 1989. 77(2): p. 257-286.

[6] Veeravalli, A.G., et al. A tutorial on using hidden Markov models for phoneme recognition. in *System Theory, 2005. SSST '05. Proceedings of the Thirty-Seventh Southeastern Symposium on*. 2005.

[7] Hidden Markov Model Toolkit. [Available from: http://htk.eng.cam.ac.uk.

[8] Speecon Database Homepage. [Available from: http://www.speechdat.org/speecon/index.html.

[9] Bazzi, I. and J.R. Glass, Modelling out-of-vocabulary words for robust speech recognition, in *ICSLP-2000*. 2000: Beijing China. p. 401-404.