ISCA Archive
http://www.isca-speech.org/archive

Speech and Language Technology in Education (SLaTE2009)
Wroxall Abbey Estate, Warwickshire, UK
September 3-5, 2009

Improving Phone Verification Using State-level Posterior Features and Support Vector Machine for Automatic Mispronunciation Detection

Khe Chai SIM

School of Computing, Department of Computer Science
National University of Singapore, Singapore.
simkc@comp.nus.edu.sg

Abstract

An important aspect of a Computer-Assisted Language Learning (CALL) system for pronunciation acquisition is the automatic detection of mispronunciations. This problem can be formulated as a phone verification task. For each phone to be verified, the system generates a verification score and a decision threshold is applied to accept or reject the pronunciation of that phone. Most verification systems use the HMM phone acoustic models to compute the log posterior probabilities (LPPs) as the verification score. A discriminative back-end using the Support Vector Machine (SVM) can also be applied to the vector of LPPs to further improve the verification performance. This paper investigates the use of a NN/HMM hybrid phone recogniser to obtain the LPP scores. The NN/HMM hybrid framework has been shown to yield superior phone recognition performance over the conventional GMM/HMM based systems. In addition, this paper also examines the use of frame-level phone or state posterior features directly with SVM. Experimental results reported on the TIMIT database show that state-level average posterior features with SVM yielded 9.5% relative Equal Error Rate (EER) improvement over the NN/HMM system.

1. Introduction

Computer-Assisted Language Learning (CALL) is a computer-based teaching software that is used to facilitate language acquisition. A CALL system for pronunciation acquisition consists of two major components: 1) the detection of mispronunciations at the phone level [1, 2]; 2) the overall assessment of the pronunciation quality at the sentence or speaker levels [3, 4]. The former provides a simple binary feedback to the users to identify the locations at which a pronunciation error has been made. The latter provides an overall assessment score which can be used for comparison between sentences or speakers. The scores can also be used for automatic assessment for oral examination which can be costly and time consuming if conducted by human examiners.

Mispronunciation can be defined in terms of the sound of the phonemes, the tone and intonation, duration as well as other prosodic attributes. This paper investigates only the mispronunciation in terms of the phoneme sound itself and its confusability with the sound of other phonemes. The detection of this kind of mispronunciation can be formulated as a phone verification task [2]. For each phone to be verified, a verification score is generated by the system and a decision threshold is applied to the verification scores, above which the pronunciation is accepted and *vice versa*. Typically the verification scores are obtained as the log posterior probabilities (LPPs) computed using a phone recogniser. The phone verification performance depends on the quality of the phone recogniser used to compute the LPPs. Hidden Markov Models (HMMs) are commonly used to model the phonemes based on the Mel Frequency Cepstral Coefficients (MFCCs) or Perceptual Linear Prediction (PLP) coefficients. To improve the quality of phone recognition, discriminative training paradigms such as Maximum Mutual Information (MMI) [5] can be used, instead of the conventional Maximum Likelihood (ML) approach.

Recently, the hybrid of neural network (NN) and HMM system based on the Temporal Patterns (TRAPS) features [6] with long temporal contexts has been shown to yield high quality phone recognition performance. This system, referred to as NN/HMM, uses a cascade of multiple neural networks to generate frame-level state posterior probabilities. This probabilities are used as the HMM state emission probabilities to perform the Viterbi decoding. While the NN/HMM phone recognisers can be used to generate the LPPs, the state posterior probabilities generated by the neural networks can also be used with a SVM back-end to produce the verification scores.

The remaining of this paper is organised as follows: Section 2 describes the phone verification mispronunciation detection system. Section 3 describes the NN/HMM hybrid phone recognition system. This is followed by the discussion of discriminative back-end using support vector machine in Section 4. The effect of speaker adaptation using CMLLR is described in Section 5. Experimental results are reported on the TIMIT database in Section 6. Finally, conclusions are drawn in Section 7.

2. System Description

The typical architecture of a phone verification mispronunciation detection system is shown in Figure 1. Given the speech waveform and its corresponding word-level orthographic transcription, phone segmentation can be achieved by forced-aligning the speech waveform with the text transcription using a phone recogniser and a pronunciation dictionary. For each phone segment, a verification score is generated using phone recogniser. The log posterior probabilities (LPP) can be used as the verification scores. It is computed as follows:

$$\log P(m^*|\mathcal{O}) = \log \left(\frac{p(\mathcal{O}|m^*)P(m^*)}{\sum_{m \in \mathcal{M}} p(\mathcal{O}|m)P(m)} \right) \quad (1)$$

where m^* is the identity of the actual phone to be uttered. $\mathcal{O}$ is the sequence of observations aligned to m^*. $p(\mathcal{O}|m)$ is the likelihood of m generating $\mathcal{O}$ and $P(m)$ is the prior probability of m. Typically, uniform prior distribution is assumed and the term $P(m)$ can be eliminated from the above equation. $\mathcal{M}$ is the set of phones in the system.

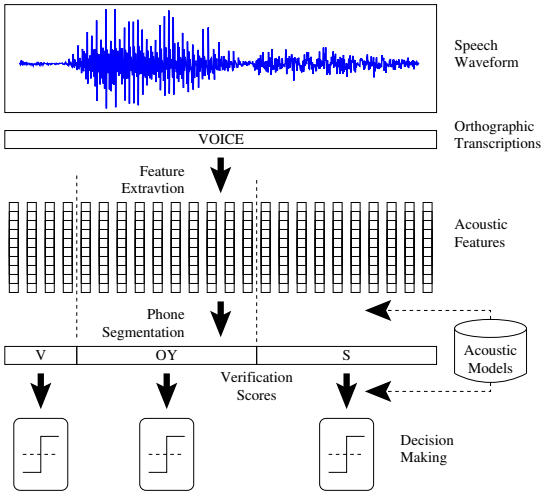

Figure 1: Architecture of a phone verification mispronunciation detection system.

A decision threshold is applied to the verification scores to determine if the pronunciation should be accepted or rejected. The decision threshold can be adjusted to compromise the trade-off between false acceptance and false rejection rates. Typically, equal costs are assigned to both false acceptance and false rejection errors. Hence, the decision threshold is adjusted so that these errors are equal. The error at this operating point is known as the Equal Error Rate (EER). EER metric is commonly used for verification tasks such as speaker and language verification. It will be used to evaluate the phone verification performance in this paper. There are two ways of computing the overall EER of the system:

- **Pooled EER:** This approach pools all the true and false scores from all the phone segments and compute the EER based on a *global* decision threshold.
- **Average EER:** This approach computes the EER for each reference phone and then compute the overall EER by taking the average. This effectively applies a different decision threshold to different phones.

3. The NN/HMM Hybrid Phone Recogniser

The HMM acoustic models used to model the phonemes for speech recognition typically adopt the 3-state left-to-right topology with self-transitions. The state emission probabilities are commonly represented as a Gaussian Mixture Model (GMM). The HMM parameters can be estimated efficiently using the Baum-Welch algorithm in the case of Maximum Likelihood (ML) training. Alternatively, discriminative training criterion such as Maximum Mutual Information (MMI) [5] has been found to yield better recognition performance.

Recently, NN/HMM hybrid phone recognition systems have been shown to outperform conventional GMM/HMM phone recognisers [7]. In this paper, the use of the NN/HMM system as described in [7] for phone verification is investigated. This system uses a cascade of neural networks to estimate the posterior probabilities for each HMM state in the system. Firstly, the TRAPS features [6] are computed for each critical band and then split into the left and right contexts. The TRAPS features are generated at every 10ms frame shift. A neural network is trained on each of these TRAPS features to predict the state posteriors. Then, a final neural network is trained to combine the output from the previous neural networks to yield the final state posterior probabilities. These posterior probabilities are used directly as the HMM state emission probabilities to perform the standard Viterbi decoding to obtain the best phone sequence.

4. Discriminative Back-end Using SVM

Instead of using the LPP scores corresponding to the target phone alone, it is possible to use the entire vector of LPPs computed for all the phones and perform a second stage of classification. This vector of LPPs is referred to as the posterior features:

$$PF = \left[\begin{array}{cccc} P(m_1|\mathcal{O}) & P(m_2|\mathcal{O}) & \ldots & P(m_M|\mathcal{O}) \end{array} \right]^T \quad (2)$$

where M, the dimension of the feature vector, is the total number of phones in the system. This can be performed efficiently using support vector machine to serve as a discriminative back-end. In this paper, the SVM back-end uses the Radial Basis Function (RBF) kernels for classification. One SVM classifier is trained for each phone. The LPP vectors corresponding to the target phone are used as the positive samples while the remaining vectors are used as the negative samples. This is a *one-versus-rest* classification strategy.

While the LPP vectors are a natural choice of features for HMM based systems, the frame-level state posterior probabilities generated by the neural network of the NN/HMM system can also be fed directly into an SVM classifier for the back-end. Since a posterior probability vector is generated every 10ms, there will be a sequence of posterior feature vectors for each phone segment. To represent each phone segment by a single feature vector, the *Average Posterior Feature* (APF) is computed for each phone segment. Therefore, if there were N frames in the phone segment, the APF is computed as:

$$APF = \frac{1}{N} \sum_{i=1}^{N} PF_i \quad (3)$$

where PF_i denotes the posterior feature vector of the ith frame. APF can be computed based on the state or phone posterior probabilities. They are referred to as State-level APF (SAPF) and Phone-level APF (PAPF) respectively. For SAPF, the posterior features are generated by the cascade of neural networks from the NN/HMM system. To obtain PAPF, all the state posterior probabilities of each phone are summed together to yield the phone posterior probabilities.

5. Speaker Adaptation

Adaptation techniques are commonly applied to speech recognition systems to compensate for channel and/or speaker variability. Maximum Likelihood Linear Regression (MLLR) [8] is widely used to perform adaptation of HMM-based acoustic models. This paper considers the Constrained MLLR (CMLLR) [8] technique which applies a global affine transformation to the acoustic feature vectors. Adaptation can be performed in two modes: *supervised* and *unsupervised*. In the former case, the transcription labels for the adaptation data is available. In the latter case, the transcription labels are generated by performing an initial phone recognition. For pronunciation learning, the transcription labels are available since the users are prompted with text to speak. However, it is important to note that the users may make pronunciation mistakes. In this paper, phone

verification is performed on native speech. Hence, the performance of supervised adaptation is clearly superior, as shown in Section 6. For the NN/HMM system, adaptation of the neural networks cannot be performed easily. Adaptation has always been as issue for neural networks. Adaptation of neural networks is not explored in this paper and will be investigated for future research.

6. Experimental Results

This section presents the experimental results of phone verification on the TIMIT database. A summary of the training and testing data sets used in this paper is tabulated in Table 1. All the

Table 1: Summary of the TIMIT training and testing data sets used in this paper.

Data Set	# of Speakers	# of Utterances	Amount of Data (hours)
train	462	3696	3.12
dev	88	640	0.60
eval	80	740	0.54

HMM-based acoustic models used in this paper are trained on the 3696 training utterances provided with the TIMIT database, which amounts to about 3.12 hours of speech data. The TIMIT test data is divided into two sets: `dev` and `eval`. The `dev` set will be used later to train the SVM back-end. The odd and even numbered sessions were chosen as the `dev` set and `eval` set respectively. This yields a total of 0.60 and 0.54 hours of speech data for the two sets.

All the phone recognisers used in this paper are context-independent monophone systems. The reduced TIMIT phone set with 40 phones (including the silence model) was used. First of all, two GMM/HMM systems were trained on the 39 dimensional feature vectors. This consists of 13 static coefficients (12 MFCC plus the C0 energy term) and the Δ and $\Delta\Delta$ parameters. Each monophone is modelled by a 3-state left-to-right HMM. The observation probability distribution of each HMM state is modelled by a 32-component GMM. The parameters of the first GMM/HMM system were estimated using the ML criterion. The second GMM/HMM system was trained using the MMI criterion. These two systems are referred to as HMM-ML and HMM-MMI respectively. In addition, a hybrid NN/HMM phone recogniser is also used for the experiments. This phone recogniser is available for download from the BUT website[1]. The neural networks of this phone recogniser were trained on the same training data as the GMM/HMM systems. Hence, the results will be comparable.

6.1. Phone Recognition

The quality of the phone recognisers were compared by evaluating the Phone Error Rate (PER) performance on the `dev` and `eval` data sets. The PER results are shown in Table 2. The HMM-ML system achieved PER of 41.0% and 40.7% on the `dev` and `eval` data sets respectively. With discriminative training of the HMM parameters, the HMM-MMI system gave a consistent absolute PER improvement of 3.1% and 3.9% over the HMM-ML system. Finally, the NN/HMM system yields further absolute PER reduction of 4.7% and 3.8% over the HMM-MMI system, giving the best phone recognition performance of 33.2% and 33.0% on the two data sets.

[1] `http://speech.fit.vutbr.cz/en/software/phoneme-recognizer-based-long-temporal-context`

Table 2: Comparison of PER (%) performance of HMM-ML, HMM-MMI and NN/HMM systems on the `dev` and `eval` data sets.

System	dev	eval
HMM-ML	41.0	40.7
HMM-MMI	37.9	36.8
NN/HMM	33.2	33.0

6.2. Pooled *versus* Average EER

Next, phone verification performance is evaluated for these systems. To perform verification, phone segmentation is performed by applying Viterbi forced-alignment. For each phone segment, the log posterior probability (LPP) of the phone given the observation sequence associated with the segment is computed using Equation (1). This is used as the positive scores for that phone and negative scores for the rest of the phones in the system. Using the positive and negative scores, a decision threshold can be determined to yield the Equal Error Rate (EER).

Table 3: Comparison of pooled and average EER performance for various systems without SVM back-end on the `dev` and `eval` data sets.

System	Pooled EER		Average EER	
	dev	eval	dev	eval
HMM-ML	6.72	6.67	6.19	6.06
HMM-MMI	5.68	5.65	5.44	5.35
NN/HMM	**5.54**	**5.33**	**4.64**	**4.45**

The comparison of pooled and average EER performance is given in Table 3. The average EER is consistently lower than the pooled EER for all the systems on both test sets. This suggests that it is better to use phone specific decision thresholds. Hence, for subsequent results, only the average EERs will be reported. As shown in the table, the average EERs of the HMM-ML system are 6.19% and 6.06% on the `dev` and `eval` data sets respectively. The HMM-MMI system gave absolute EER reduction of 0.71 – 0.75% over the HMM-ML system. Finally, the hybrid NN/HMM system yields the best performance of 4.64% and 4.45% on the two test sets, which are 0.80 – 0.90% absolute better than the HMM-MMI system.

6.3. Phone Verification With SVM Back-end

Table 4: Comparison of EER (%) performance of various phone verification systems with and without SVM back-end

System	Without SVM		With SVM	
	dev	eval	dev	eval
HMM-ML	6.19	6.06	4.84	5.81
HMM-MMI	5.44	5.35	4.38	5.34
NN/HMM	4.64	4.45	4.10	**4.32**

The effect of using a SVM back-end on phone verification performance is shown in Table 4. One SVM classifier is trained for each phone. The input to each SVM classifier is a 40-dimensional LPP vector. Radial Basis Function (RBF) kernel was used and the classifiers were trained on the `dev` data

set. Hence, the improvement from using the SVM back-end is greater on the dev set. On the eval set, the absolute EER improvements from using a SVM back-end were 0.25%, 0.01% and 0.13% respectively for the HMM-ML, HMM-MMI and NN/HMM systems. Hence, the NN/HMM system with SVM back-end yields the lowest EER of 4.32%.

6.4. Average Posterior Features

For the NN/HMM system, instead of using the state-level posteriors as the HMM state emission probabilities to generate the LPP scores, it is also possible to use the posterior vectors as input features to the SVM back-end to generate the final scores for verification. The SAPF and PAPF features as described in Section 4 will be investigated.

Table 5: Comparison of Average EER (%) performance of systems with SVM back-end using different posterior features.

System	dev	eval
NN/HMM	4.10	4.32
PAPF	3.42	4.62
SAPF	1.49	**3.91**

The EER performance of these systems are summarised in Table 5. When the PAPF were used as input to the SVM back-end classifier, there was an absolute EER improvement of 0.68% on the dev set that was used to train the SVM classifier. However, on the eval set, the EER performance deteriorated by 0.30% absolute compared to the NN/HMM system where the LPP vectors generated by the HMMs were used instead. On the other hand, using the SAPF gave a huge improvement of 2.61% on the dev set and an absolute improvement of 0.41% on the eval data set. This translates to approximately 9.5% relative improvement. The results show that there is a clear advantage of generating posterior probabilities at a granularity higher than phone level, such as state level.

6.5. CMLLR Speaker Adaptation

Table 6: Comparison of EER (%) performance of various phone verification systems using no, unsupervised and supervised CMLLR speaker adaptation with SVM back-end.

System	Adaptation	dev	eval
HMM-ML	none	4.84	5.81
	unsup	4.55	5.62
	sup	4.09	4.99
HMM-MMI	none	4.38	5.34
	unsup	4.28	5.22
	sup	3.80	4.62

Finally, the effect of CMLLR speaker adaptation on the EER performance of the HMM-ML and HMM-MMI systems is summarised in Table 6. Two speaker adaptation modes were considers: unsup and sup refer to unsupervised and supervised speaker adaptation respectively. Speaker adaptation is performed using a global CMLLR affine transformation with a bias vector. none denotes no speaker adaptation were performed. Unsupervised adaptation yields 0.19–0.29% and 0.10–0.12% absolute EER improvements for the HMM-ML and HMM-MMI systems respectively. As expected, supervised adaptation achieves much greater performance improvements of 0.75–0.82% and 0.58–0.72% for the two systems. Note that the best performance of the adapted system is 4.62 on the eval data set. This is still not as good as the unadapted performance of the NN/HMM and SAPF systems. As for future work, we will explore adaptation techniques for neural networks to further improve the NN/HMM and SAPF systems.

7. Conclusions

This paper has presented a comparative study of different approaches for phone verification and proposed a new approach based on the state average posterior features with SVM to improve verification performance. The results reported in this paper showed that the log posterior probability scores produced by an HMM system yielded better phone verification performance if the HMM parameters were discriminatively trained using the MMI criterion compared to the conventional ML training paradigm. Furthermore, the hybrid NN/HMM which is based on long temporal TRAPS features outperformed both ML and MMI trained HMM systems. In addition, the posterior probabilities of all the phones can be used as input features to the SVM back-end to further improve the verification performance. Finally, instead of generating the phone posterior scores using a set of HMM models, the average state-level posterior features generated by a cascade of neural networks can also be used as input features to the SVM back-end. This was shown to yield approximately 9.5% relative improvement over the NN/HMM system on the TIMIT database.

8. Acknowledgement

This research is done for CSIDM Project No. CSIDM-200806 partially funded by a grant from the National Research Foundation (NRF) administered by the Media Development Authority (MDA) of Singapore.

9. References

[1] H. Franco, L. Neumeyer, Y. Kim, and H. Ronen, O. abd Bratt, "Automatic detection of phone-level mispronunciation for language learning," in *Proc. Eurospeech*, vol. 2, 1999, pp. 851–854.

[2] J. Jiang and B. Xu, "Exploring the automatic mispronunciation detection of confusable phones for mandarin," in *Proc. ICASSP*, 2009, pp. 4833–4836.

[3] L. Neumeyer, H. Franco, V. Digalakis, and M. Weintraub, "Automatic scoring of pronunciation quality," *Speech Communication*, vol. 30, pp. 83–93, 2000.

[4] S. Witt and S. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech Communication*, vol. 30, no. (2/3), pp. 95–108, 2000.

[5] L. Bahl, P. Brown, P. deSouza, and L. Mercer, "Maximum mutual information estimation of hidden markov model parameters for speech recognition," in *Proc. ICASSP*, 1986, pp. 49–52.

[6] H. Hermansky and S. Sharma, "Temporal patterns (TRAPS) in ASR of noisy speech," in *Proc. ICASSP*, 1999.

[7] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, "Phonotactic language identification using high quality phoneme recognition," in *Proc. Eurospeech*, September 2005.

[8] M. J. F. Gales and P. C. Woodland, "Mean and variance adaptation within the MLLR framework," *Computer Speech and Languages*, vol. 10, pp. 249–264, 1996.