Thai Text-Dependent Speaker Identification by ANN with Two Different Time Normalization Techniques

Chai Wutiwiwatchai, Sutat Sae-tang, and Chularat Tanprasert
Software and Language Engineering Laboratory,
National Electronics and Computer Technology Center,
National Science and Technology Development Agency,
Ministry of Science, Technology, and Environment, THAILAND
539/2 Gypsum Metropolitan Building, 22nd floor, Sri Ayudhya Rd., Rachethewi, Bangkok 10400
Email: chai@nectec.or.th, sutat@notes.nectec.or.th, and chulak@nectec.or.th

Abstract

A text-dependent speaker identification system for Thai language was proposed. Thai isolated digits 0-9 and their concatenations were used for speaking text. Well-known artificial neural network (ANN) called multilayer perceptron (MLP) with backpropagation learning algorithm was conducted for recognition engine due to its simplicity and less processing time spending. Because of fix number of input neurons of MLP, time normalization algorithms must be applied to speech signal in order to obtain a unique number of input speech features. Two different time normalization algorithms, which are linear interpolation and synchronized overlap and add (SOLA) were implemented and compared. Experimental results showed that different algorithms of time normalization clearly effected system performance. SOLA, which can carry more original sound than linear interpolation, gave better identification rate in all speaking digits.

1. Introduction

Many recognition engines have been proposed for a task of speaker identification [1][2]. Some efficient engines are Dynamic Time Warping (DTW), Vector Quantization (VQ), Hidden Markov Model (HMM), and Artificial Neural Networks (ANN). Three speaker recognition systems using DTW, VQ, and continuous density HMM (CHMM) have been compared in [2]. Competitive performances of DTW, VQ with DTW, discrete HMM (DHMM), and CHMM applied to isolated word recognition has also been studied in [3]. Both can roughly guide that DTW is the most efficient approach for text-dependent task including speaker recognition and speech recognition. This was, again, obviously indicated in our previous works [4] that DTW is a good matching machine. We have deeply research on the use of DTW for speaker identification system with Thai concatenated-digit spoken text [5]. Our previous works were to calculate matching distances from interacted speech signals and use K-nearest neighbor (KNN) to improve the decision.

Although our previous experimented system has some advantages, a significant problem of very long time spending during recognition has obstructed the system in the practical implementation. There are several ways to overcome this problem. One is to use fewer references, which certainly effect system performance especially with large number of speakers to be recognized. What we expect is to use another recognition engine that provides optimal recognition rate and less processing time. Artificial neural network (ANN) is one of our expectations due to its very fast processing. There were some researches proposing to use ANN in speaker recognition tasks [6][7]. The speaker identification system for Thai language using ANN [8] was implemented and found its advantages on both identification rate and fast processing. However, due to our trials on the use of a well-known ANN called multilayer perceptron (MLP) with backpropagation learning algorithm, the obtained identification rate was not higher than the one obtained from DTW. A limitation of MLP is its fixed number of input neurons, which causes us to generate a unique number of features to represent each speech. This problem can be fixed by using time normalization algorithms, alternative designs of feeding methods, or other ANN structures. Here, we first investigated on several time normalization algorithms, which should very much effect to system performance.

Some algorithms of time normalization have been proposed [9][10] such as linear interpolation, sampling rate changing, and synchronized overlap-and-add (SOLA). Prior comparison of these three time normalization approaches regarding their complexity, processing time, and linguistic characteristics were studied in [9] which concludes that linear interpolation method may cause aliasing normalized speech. It will probably corrupt the original speech. In contrast, the other two approaches tried to preserve the characteristics of the original speech, hence the normalized speech still can be identified by human. Although these time normalization algorithms were proposed for other tasks such as speech recognition [9] and audio

time-scale modification [10], they should directly effect to speaker identification task.

In this paper, we have implemented a text-dependent speaker identification system for Thai language. Recognition engine is MLP with backpropagation learning algorithm. Thai digits 0-9 were used as speaking text. Two time normalization algorithms, linear interpolation and SOLA, were developed and compared. We didn't use sampling rate changing method because of its complexity, which consequently cause too much processing time. Additional experiment was done on concatenated digits in order to improve identification rate.

2. Speaker Identification System

Conclusive model of speaker identification system derived from proposals in [11] is shown in figure 1.

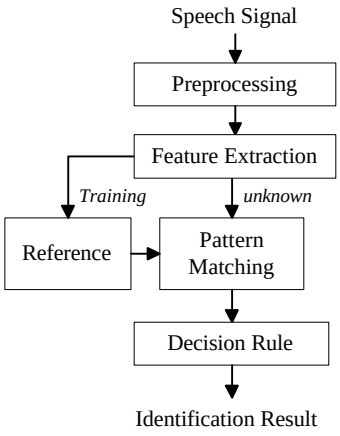

Figure 1. Speaker identification model

Speech signal – Analog speech signal is digitized using any A/D converter to be digital form. Some parameters must be set optimally e.g., sampling rate and quantization bits for digital storage. To avoid aliasing problem occurred during sampling and to achieve precise values of digitized samples, 8-20 kHz sampling rate and 12-16 bit resolution are considered.

Preprocessing – This step is to adjust an input speech signal to be in a better quality or to has appropriate characteristic for the next processing step. Preprocessing covers digital filtering, endpoint detection, and time normalization. Filtering is to filter out any surrounding noise using several algorithms of digital filter. Endpoint detection is a process of clamping only a desired speech interval. A lot of endpoint detection algorithms have been proposed for speech processing tasks (e.g. energy-based, zero-crossing, and fundamental frequency). We used energy-based endpoint detection in this research due to its simplicity. Although this approach probably causes error with too small loudness of speech, protection of this error can be done during speech recording. Time normalization is to stretch or press original speech to normalized speech with desired time duration. This sub-procedure is an option and is only applied to some recognition strategies. Several algorithms of time normalization are presented and deeply focused later.

Feature extraction – Aim of this important step is to extract a set of essential characteristics that can identify or represent whole speech signal. Both linguistic knowledge and speech coding strategies were conducted to achieve this problem. A lot of speech features have been proposed for speech and speaker recognition tasks. Effective features has been classified into two groups [13]. One is called "high level" features such as dialect, context, speaking style, etc. The other is called "low level" features which can be roughly grouped to spectral envelop-based features and prosodic features [12] [14]. Although prosodic parameters (e.g. pitch, formant frequencies, and energy profile) should be considered to be effective features, however, the measurement of these values is quite difficult due to their non-robustness. This was insisted by [12][13], which suggested using spectral envelop-based parameters. Among several spectral-envelop measurements, family of cepstrum [16][17] seemed to be the best. An easy approach to calculate cepstral coefficients is to derive from linear prediction coefficients (LPC) using simple linear predictive coding [15], so called linear predictive coding derived cepstrum (LPCC). It was chosen to use in this paper.

To compute LPCC, preprocessed speech signal is passed through preemphasis, which stresses high frequency component of speech using first order filter, frame blocking and windowing by a window function e.g. Hamming window, LPC extraction using autocorrelation analysis [15]. LPC is simply converted to LPCC [16][17] with appropriate coefficient order. Hence, speech signal is represented by a set of feature vectors, in which consisting of a number of cepstral coefficients.

Recognition engine - As described in the previous section, we decided to use a well-known type of ANN, MLP with backpropagation learning algorithm due to its simplicity and especially less time processing. With ANN, the training set is required in order to obtain the set of optimal weights which will be used to compute the output for the unknown patterns in the testing process. Therefore, both reference part and pattern matching part shown in figure 1 is together when ANN is selected as the recognition engine. Figure 2

illustrates a brief backporpagation learning algorithm and also a MLP structure.

Input training vector is fed into an input layer with number of input nodes equal to number of input features. Input values are fed forward and passed to a hidden layer to get output values at an output layer. Error (E) between output values and desired output values is sent back to adjust weights (W_{ij}) in the network. These processes are recursive until an optimal error (E_T) is reached. More details of MLP structure and algorithm of training can be viewed in [18][19]. In the testing process, an unknown pattern is fed into the network to find out the output values which uses to identify the speaker.

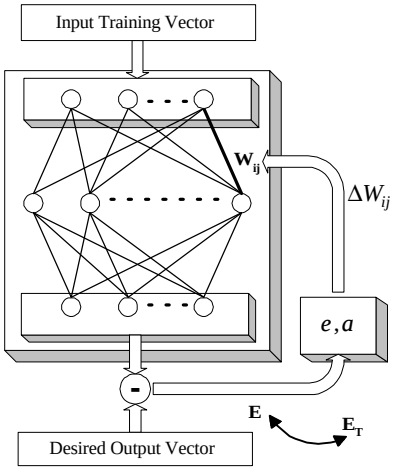

Figure 2. MLP structure with backpropagation learning algorithm

3. Time Normalization Approaches

As described above, MLP receives a unique number of input values. Decision on using MLP and backpropagation training forces us to perform a process to obtain unique amount of speech features from each speech signal. Simple solution is to use a time normalization algorithm, which can adjust a speech duration to be in the desired length. Normalized speech signals with equal length are then passed to a feature extraction step to obtain an equal number of features subsequently.

Time normalization is applied to increase or decrease an amount of speech samples within whole digital speech signal. This task can probably cause distortion or aliasing in normalized speech. And this is the first significant problem to be considered. A few methods of time normalization have been proposed such as sampling rate changing, linear interpolation, and synchronized overlap-and-add. Details of three algorithms are given as following.

3.1 Sampling rate changing

Realize that to change speech duration is to change amount of speech samples. Hence, to change number of speech samples is to change sampling rate during A/D conversion. However, we already have speech signal in form of digital, which force us to applied time normalization in digital domain. Brief steps of sampling rate changing are shown in figure 3 [9].

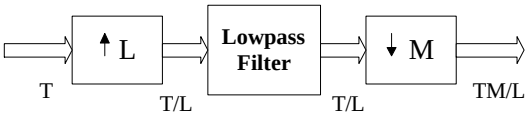

Figure 3. Sampling rate changing procedure

Original speech signal with digital period of T (T = 1/f, where f denotes an original sampling rate) is L-time sampling up with zero values adding between original samples. Then lowpass filter is applied to perform smooth signal before M-time sampling down to be TM/L period (new sampling rate is fL/M Hz). This method may cause a little distortion from an original speech depended on how much sampling rate scaling is.

3.2 Linear interpolation

A very simple time normalization method called linear interpolation was proposed in [9]. This method has even been used with our previous work in [8] and obtained a good performance. Key of this method is the same as ever, which tries to increase or decrease amount of speech samples. New speech samples are generated using linear interpolation from two neighbor samples of original speech. Figure 4 demonstrates this algorithm.

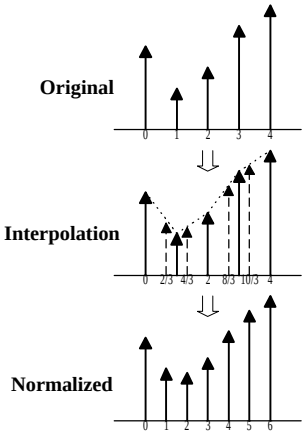

Figure 4. Linear interpolation procedure

This method can highly make a distorted speech with aliasing problem. Although, the new normalized speech cannot be understood by human, but the recognition system is still able to learn and identify the normalized speech pattern.

3.3 Synchronized overlap-and-add

An efficient algorithm abbreviated as SOLA has been proposed to the task of audio time-scale modification in [10]. With this previous task, similarity of time scaling speech and original speech is very important. The basic idea of this algorithm is to cut signal into overlapped frames, modify the distance between adjacent frames according to desired time-scale, weight them and add them up. Figure 5 shows how the signal is cut into frames and put them back together.

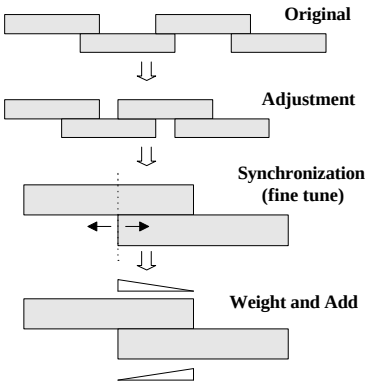

Figure 5. Synchronized overlap-and-add

An important algorithm applied before adding frames up is to fine tune for obtaining the best matching points between the interacted frames (called synchronization). This is to avoid phase shifting which causes a speech to be broken. Synchronization algorithm has been originally proposed in [10]. The algorithm is to maximize the cross-correlation coefficient between the preceding frame and the next adding frame with respect to fine sliding point. Cross-correlation R(k) is defined as

$$R(k) = \frac{\sum_{j=0}^{L-1} x(j).y(j+k)}{\sqrt{\sum_{j=0}^{L-1} x^2(j) \sum_{j=0}^{L-1} y^2(j+k)}} \quad , -K_m \le k \le K_m$$

where x(j) and y(j) denotes speech samples in first frame and next frame respectively. L is an amount of samples in each frame (assume equal length). Index k is varied between allowable tuning interval $[-K_m, K_m]$ and K_m is set manually.

This algorithm can produce new normalized speech with non-aliasing problem, but capability of stretching or pressing is limited by overlapped length of frames. Furthermore, synchronizing adjustment will cause normalized speech to have unequal length as we desire. Cutting off exceeding sample or zero samples adding must be done to obtain the desired duration.

We have a prior expectation that the recognition system used to be same as human reception. Hence, feeding of features from clear speech should be better than from distorted speech. SOLA which provides more likelihood to original speech should give better performance than other methods. By the way, experiment on sampling rate changing is not performed due to its much computation compared to others.

4. Experiments and Results

We proposed two experiments of text-dependent speaker identification. One is performed on Thai isolated digits 0-9. The other is tested on the concatenated top-score isolated digits which forms a new longer input speech. The second experiment is corresponding to our final target, which is text-prompt speaker identification system using concatenation of Thai isolated digits. Therefore, this task can help us in the selection of some efficient digits to be used in the complete system.

4.1 Parameter setting

Input speech signals were collected in crossing a computer microphone with 11.025 kHz sampling rate and 16-bit resolution. In the preprocessing step, a highpass filter with 200 Hz cut off frequency was applied to filter out some low frequency noise generated by a power supply. Detection of speech interval was done using energy-based approach. Speech signal, in SOLA normalization step, was blocking into 600-sample frame with 150-sample overlapping. K_m described in section 3.3 was set to 50. No parameter is set for linear interpolation method. Normalized speech was obtained from an average duration of training set, which was 6000-sample duration for Thai isolated digit 0-9. In the feature extraction procedure, preemphasis with first order filter defined as $H(z) = 1\text{-}0.95z^{-1}$ is applied. Preemphsized speech was blocked into 20-ms frame (220 samples for sampling rate of 11.025 Hz) with a quarter of frame overlapping (55 samples). 15-order autocorrelation, LPC, and LPCC analysis were conducted to obtain 15-order cepstral vector per each frame. The MLP network consisted of four layers; one input layer, two hidden layers and one output layer. For isolated digit experiment, the input layer contains 555 neurons (6000-speech sample giving 37 frames and 15-order LPCC per frame), which are fully connected to the first hidden layer. Number of input nodes was increased proportional to number of concatenated digits e.g., 555×3 input neurons for 3-concatenated digit. The two next hidden layers consisted of 20 neurons per layer. The output layer consisted of 20 neurons, one neuron for representing one speaker. The ANN

simulator software named SNNS [19] was used in our experiments.

4.2 Experiment on isolated digits

In this trial, the Thai speaker identification systems of two time normalization techniques have been evaluated with 20 speakers (11 male and 9 female) by pronouncing each digit ten times per week for five consecutive weeks. Therefore, the total utterances of each digit were 1000 utterances (20 x 5 x 10). The data were divided into two sets, with 600 utterances in the training set (week 1^{st} - 3^{rd}) and 400 utterances in the test set (week 4^{th} - 5^{th}). Experimental results were shown in table 1.

Table 1. Identification result using isolated digits

Digit	Phonetic	Identification rate (%)	
		Linear interpolation	SOLA
0	/su:n4/	77.00	76.00
1	/nvng1/	61.75	76.25
2	/s@:ng4/	66.75	72.00
3	/sa:m4/	67.00	77.50
4	/si:1/	71.75	78.75
5	/ha:2/	70.00	79.25
6	/hok1/	65.75	77.00
7	/cet1/	64.50	81.75
8	/pa:t1/	58.25	64.25
9	/ka:o2/	70.00	70.50
Average		67.28	75.33

Notes that, in phonetic symbols, ":" means long vowel utterances, digits at last indicates Thai tone, which consists of 0-4 (middle tone, low tone, falling tone, high tone, and rising tone, respectively [8]). Comparison can be prominent in figure 6. Linear interpolation time normalization produced the best accuracy with digit 0 at 77% and worst one with digit 8 at only 58.25. Meanwhile, the SOLA technique gives the best performance with digit 7 at 81.75% and the worst one with digit 8 at 64.25%.

SOLA gives clearly better performance over the linear interpolation in almost isolated digits as expected. However, these two techniques can give approximately equal performance in digit 0 and 9. There were many factors to explain the phenomenon such as the original characteristics of these two digits, speaking skill of speakers, hardness of pronouncing the digit, and appropriated parameters used in this system. One interesting observation is that SOLA may largely improve the performance over the linear interpolation when using the technique with a short-vowel digit e.g. digit 1, 6, and 7. The experiment confirms our observation that a short-vowel utterance gives a low identification rate which may due to the less samples of speech due to its short duration. This might be because of the interpolated normalization increasingly pulls down the speech characteristic, whereas SOLA still tries to maintain original speech characteristics.

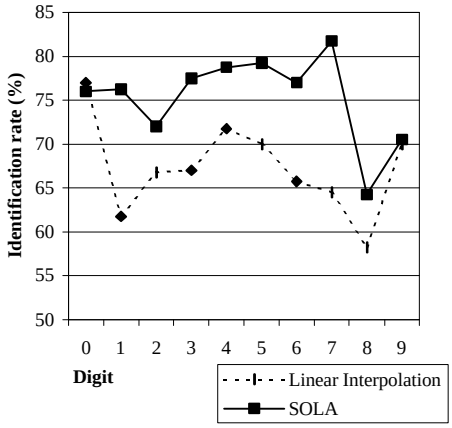

Figure 6. Comparison of identification results in each isolated digit

4.3 Experiment on concatenated digit

Further experiment was performed with a longer speech by selecting the best 3, 5, and 7 digits from table 1 and concatenated them as one sentence. Therefore, the concatenated digits of the linear interpolation time normalization were "0-4-5", "0-4-5-9-3", and "0-4-5-9-3-2-6" and of SOLA were "7-5-4", "7-5-4-3-6", and "7-5-4-3-6-1-0". The results are shown in table 2.

Table 2. Identification result using 3-, 5-, and 7-concatenated digits

Digit	Identification rate (%)	
	Linear interpolation	SOLA
Top-1	77.00	81.75
Top-3	81.00	82.50
Top-5	77.50	88.75
Top-7	75.00	87.50

The linear interpolation gives the best identification accuracy with Top-3 at 81.00% and the highest identification result of SOLA is 88.75% with Top-5 sentence. It can be concluded that longer speech gives a better identification rate than a speech of a single digit even when using a concatenation of short-vowel digits.

5. Conclusion

Time normalization technique is a very significant process in preparing data for the backpropagation neural network, which has a limitation to have a fixed amount of input neurons. Linear interpolation and synchronized overlap-and-add (SOLA) were

experimented and compared the results. SOLA technique gives a better identification rate than the linear interpolation on both isolated and concatenated digits data. Furthermore, the research team tried to identify the normalized speech. The speech from linear interpolation is hardly identified by human what digit it is. In contrast, human easily identifies the speech from SOLA technique. This can be used as a reason to explain the identification rate obtaining from the two neural networks. Since the backpropagation network is created to emulate the human behavior and SOLA can tolerated the characteristics of each utterance for human's listening, that is why SOLA gives a better performance in the proposed speaker identification system.

Further works on the use of ANN in speaker identification task is to try other ways of time normalization methods, other designs of input feeding, or use of other ANN models that avoid time normalization, which causes highly effect to identification performance.

References

[1] J. P. Campbell, Jr., "Prolog to Speaker Recognition: A Tutorial", *Proceedings of IEEE*, Vol. 85, No. 9, p. 1436-1462, September 1997.

[2] K. Yu, J. Mason, and J. Oglesby, "Speaker Recognition uisng Hidden Markov Models, Dynamic Time Warping and Vector Quantisation", IEE Proc.-Vis. Image Signal Process, Vol. 142, No. 5, October 1995.

[3] L. R. Rabiner and B. -H. Juang, "Fundamentals of Speech Recognition", A. Oppenheim, Series Editor,Englewood Cliffs, NJ: Prentice-Hall, 1993.

[4] W. Sintupinyo, P. Dubey, S. Sae-tang, V. Acahriyakulporn, C. Wutiwiwatchai, and C. Tanprasert, "LPC-based Thai Speaker Identification using DTW", Proceedings of 1999 *NSTDA Annual Conference*, Thailand, p. 238-246, March-April 1999. (in Thai)

[5] C. Wutiwiwatchai, V. Achariyakulporn, and C. Tanprasert, "Text-dependent Speaker Identification using LPC and DTW for Thai Language", 1999 IEEE 10th Region Conference (TENCON'99), Vol. 1, September 1999.

[6] C. Zhongbao, Y. Zhenli, and Z. Lihe, "Automatic Speaker Verification using the Neural Network and Combined LPC Parameter", *1993 IEEE 10th Region Conference (TENCON'93)*, 1993.

[7] R. A. Finan, A.T. Sapeluk, R. I. Damper, "Comparison of Multilayer and Radial Basis Function Neural Netowks for Text-dependent Speaker Recognition", *1996 IEEE International Conference on Neural Networks (IJCNN'96)*, Vol. 4, pp. 1992-1997, 1996.

[8] C. Wutiwiwatchai, S. Sae-tang, and C. Tanprasert, " Text-dependent Speaker Identification Using Neural Network on Distinctive Thai Tone Marks", Proceedings of *International Joint Conference on Neural Networks*, July 1999.

[9] C. Wutiwiwatchai, "Thai Polysyllabic-word Recognition using Neural Network and Fuzzy technique", *Thesis of the Master Degree of Electrical Engineering, Chulalongkorn University*, Thailand, 1997.

[10] S. Roucos and A.M. Wilgus, "High Quality Time Scale Modification for Speech," *IEEE International Conference ASSP*, pp. 493-496, 1985.

[11] S. Furui, "Digital Speech Processing, Synthesis, and Recognition", New York and Basel: Marcel Dekker, Inc, 1989.

[12] G. R. Doddington, "Speaker Recognition-Identifying People by their Voices", *Proceedings of IEEE*, Vol. 73, No. 11, p.1651-1664, November 1985.

[13] J. M. Naik, "Speaker Verification: A Tutorial", *IEEE Communications Magazine*, pp. 42-48, January 1990.

[14] M. J. Carey, E. S. Parris, H. Lloyd-Thomas, S. Bennett, H. T. Bunnell, and W. Idsardi, "Robust Prosodic Features for Speaker Identification", *4th International Conference on Spoken Language Processing (ICSLP'96)*, Vol. 3, pp. 1800-1803, 1996.

[15] D. O'Shaughnessy, "Linear Predictive Coding", *IEEE Potentials*, p. 29-32, February 1988.

[16] S. Furui, "Cepstral Analysis Technique for Automatic Speaker Verification", *IEEE Transaction on Acoustic, Speech Signal Processing*, Vol. ASSP-29, pp.254-272, April 1981.

[17] R. J. Mammone, X. Zhang, and R. P. Ramachandran, "Robust Speaker Recognition, A Feature-based Approach", *IEEE Signal Processing Magazine*, p. 58-71, September 1996.

[18] L Fausette, "Fundamentals of Neural Networks–Architecture, Algorithm, and Applications", Prentice-Hall, 1994.

[19] SNNS (Stuttgart Neural Network Simulator) User Manual, Version 4.1, University of Stuttgart, Institute for Parallel and Distributed High Performance Systems (IPVR), Report No. 6/95.