



# Phone Mismatch Penalty Matrices for Two-Stage Keyword Spotting Via Multi-Pass Phone Recognizer

Chang Woo Han, Shin Jae Kang, Chul Min Lee, and Nam Soo Kim

School of Electrical Engineering and the INMC,  
Seoul National University, Seoul, Korea.

{cwhan, sjkang, cmlee}@hi.snu.ac.kr, nkim@snu.ac.kr

## Abstract

In this paper, we propose a novel approach to estimate three types of phone mismatch penalty matrices for two-stage keyword spotting. When the output of a phone recognizer is given, text matching with the phone sequences provided by the specified keyword using the proposed phone mismatch penalty matrices is carried out to detect a specific keyword. The penalty matrices which is estimated from the training data through deliberate error generation are accounting for substitution, insertion and deletion errors. In comparative experiments on a Korean continuous speech recognition task, the proposed approach has shown a significant improvement.

**Index Terms:** phone mismatch penalty matrices, two-stage keyword spotting, multi-pass phone recognition outputs

## 1. Introduction

Currently, there is a growing interest in multistage approaches to automatic speech recognition (ASR) and keyword spotting [1]-[9]. In a multistage system,  $N$ -best phone sequences, phone lattices, or confusion networks are obtained at the first stage followed by a lexical search which applies specialized decoding steps, or uses more detailed information, e.g., morphological and domain-dependent knowledge.

Compared with a conventional integrated keyword spotting system, there are several advantages in multistage techniques. First, the multistage approach is more flexible in changing keyword list. When the keyword list is changed, we have to modify only the lexical decoding part because the phone recognition module is independent of the specified keywords. In addition, it is more suitable to build a vocabulary- or language-independent system. The second advantage is that this approach shows a relatively small computation load particularly when the vocabulary size is huge. Recently, the keyword spotting algorithm should be implemented on low computing power devices such as the mobile phones. Even though large vocabulary speech recognition technologies have already been well established, they are not sufficient to be successfully deployed in small hand-held devices. Since the basic phone recognizer at the first stage is mainly responsible for the overall computation load, the amount of computation load can be kept to a low level even though the vocabulary size increases.

The performance of the decoding module at the second stage is mainly dependent on the phone mismatch penalties imposed to the substitution, insertion and deletion errors. There have been several studies on determining these penalties for multistage keyword spotting [2]-[7]. In [4] and [5], the penalties for substitution are decided on the basis of some rules defined over the broad acoustic-phonetic classes, and the penalties for

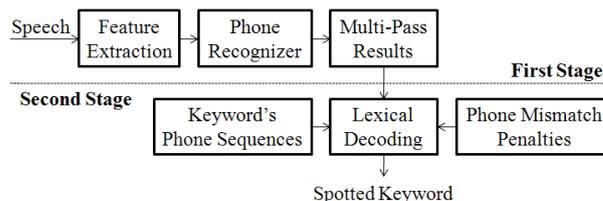


Figure 1: Overall block diagram of the implemented two-stage keyword spotting system.

insertion and deletion are fixed to constant values. In [6], the substitution penalties are automatically derived from the phone confusion matrix of the recognizer, while the insertion and deletion penalties are still set to fixed constants.

In this paper, we propose a new technique to estimate the phone mismatch penalty matrices for two-stage keyword spotting. In the proposed approach, the penalties corresponding to all types of errors are estimated from the training data. In order to evaluate our proposed method, we use a two-stage keyword spotting system based on the multi-pass phone recognition results, e.g.  $N$ -best phone sequences or phone lattices. The keyword spotting system applying the proposed penalty matrices shows better performance than that using other penalties when evaluated on a Korean continuous speech recognition task.

## 2. Implemented Two-Stage Keyword Spotting System

The overall block diagram of the implemented two-stage keyword spotting system is shown in Figure 1. The first stage, which is the conventional phone recognition module, generates multi-pass phone recognition outputs, such as the  $N$ -best phone sequences or phone lattices. In our system, we extract the mel-frequency cepstral coefficients (MFCCs) as the basic feature vectors and phone recognition is performed through the hidden Markov model (HMM) technique. The phone recognition module outputs  $N$  phone sequences, which correspond to the  $N$  highest likelihood hypotheses, or phone lattices.

At the second stage, the lexical decoding block detects keywords by comparing the recognized multi-pass results with the hypothesized lexical phone sequences of the keywords. This operation can be considered as a simple string match algorithm. To measure the similarity between each pair of phone strings, we define the phone mismatch penalty matrices accounting for each type of errors. In the final step, the multi-pass phone recognition outputs are decoded by applying a dynamic programming approach.

Let  $Q^{(n)} = (q_1^{(n)}, q_2^{(n)}, \dots, q_{N_{Q^{(n)}}}^{(n)})$  be one of the phone sequences obtained from the multi-pass phone recognizer, and  $P = (p_1, p_2, \dots, p_{N_P})$  be the lexical phone sequence of a hypothesized keyword where  $N_{Q^{(n)}}$  and  $N_P$  denote the number of phones of  $Q^{(n)}$  and  $P$ , respectively. Then, we can compute the sequence mismatch distance  $\mathfrak{D}(P, Q^{(n)})$  by applying a dynamic programming technique similar to the dynamic time warping (DTW) algorithm [3]. Finally, keyword spotting is accomplished according to the following decision rule:

$$\min_{1 \leq n \leq N} \mathfrak{D}(P, Q^{(n)}) \begin{matrix} \leq \\ \geq \end{matrix} \begin{matrix} H_1 \\ H_0 \end{matrix} \gamma \quad (1)$$

where  $N$  is the number of possible phone sequences from the multi-pass outputs,  $\gamma$  is a prespecified threshold and the two hypotheses  $H_1$  and  $H_0$  respectively indicate the presence and absence of the target keyword.

In order to carry out dynamic programming, we need a set of penalties that measure a degree of phone sequence mismatch. For this, we introduce three penalty matrices for the three types of phone errors: substitution, insertion and deletion. Let  $\Psi = \{\phi_1, \phi_2, \dots, \phi_{N_\phi}\}$  be the set of all phone identities, where  $N_\phi$  is the total number of phones. The penalty matrix for each type of phone errors is a  $N_\phi \times N_\phi$  matrix. Let  $PM_{sub}(\phi_i, \phi_j)$  be the  $(i, j)$ th element of the penalty matrix for substitution.  $PM_{sub}(\phi_i, \phi_j)$  represents a penalty imposed when the actual phone identity is  $\phi_j$  but misrecognized as  $\phi_i$ . In a similar manner,  $PM_{ins}(\phi_i, \phi_j)$ , the  $(i, j)$ th element of the penalty matrix for insertion, is defined as the penalty for the case when  $\phi_i$  is inserted after a spoken phone  $\phi_j$ . Finally,  $PM_{del}(\phi_i, \phi_j)$ , the  $(i, j)$ th component of the penalty matrix for deletion, indicates the penalty required when the spoken phone  $\phi_j$  is missed after  $\phi_i$ .

There are a number of string matching methods taken into consideration of substitution, insertion and deletion errors [1]-[9]. They usually implement the decoder by applying a dynamic programming algorithm to Markov chains or finite state machines. Since our purpose in this work is to propose and evaluate a new method to estimate the three different kinds of phone mismatch penalty matrices, we apply the conventional dynamic programming method similar to that used in [3].

Let  $C_{i,j}^{(n)}$  be the accumulated penalty of the best path upto  $(q_i^{(n)}, p_j)$ . Then, it is updated as follows:

$$C_{i,j}^{(n)} = \begin{cases} PM_{sub}(q_i^{(n)}, p_j), & i = j = 1 \\ C_{i,j-1}^{(n)} + PM_{del}(p_{j-1}, p_j), & i = 1, j \neq 1 \\ \min [PM_{sub}(q_i^{(n)}, p_j), \\ C_{i-1,j}^{(n)} + PM_{ins}(q_i^{(n)}, p_j)], & i \neq 1, j = 1 \\ \min [C_{i-1,j-1}^{(n)} + PM_{sub}(q_i^{(n)}, p_j), \\ C_{i-1,j}^{(n)} + PM_{ins}(q_i^{(n)}, p_j), \\ C_{i,j-1}^{(n)} + PM_{del}(p_{j-1}, p_j)], & \text{otherwise,} \end{cases} \quad (2)$$

in which  $1 \leq i \leq N_{Q^{(n)}}$  and  $1 \leq j \leq N_P$ .

Since we do not know the exact starting point of the hypothesized keyword, we use  $(\min [PM_{sub}(q_i^{(n)}, p_j), C_{i-1,j}^{(n)} + PM_{ins}(q_i^{(n)}, p_j)])$  when  $i \neq 1$  and  $j = 1$  instead of  $(C_{i-1,j}^{(n)} + PM_{ins}(q_i^{(n)}, p_j))$ . This modification enables us to spot the hypothesized keyword regardless of the exact starting time.

After calculating  $C_{i,j}^{(n)}$  for all the possible  $(i, j)$  grids,  $\mathfrak{D}(P, Q^{(n)})$  is obtained as follows:

$$\mathfrak{D}(P, Q^{(n)}) = \min_{1 \leq i \leq N_{Q^{(n)}}} (C_{i, N_P}^{(n)} / l_{i, N_P}^{(n)}) \quad (3)$$

where  $l_{i, N_P}^{(n)}$  is the length of the best path upto  $(q_i^{(n)}, p_{N_P})$  which is available through the backtracking technique.

### 3. Estimation of Phone Mismatch Penalty Matrices

It has been reported that the accuracies of the state-of-the-art HMM-based phone recognizers are around 70% [10], [11]. The performance deteriorates in the presence of background noise. In many cases, the phone sequences of spoken keywords may not be found in the phone recognition results because there exist frequent occurrences of the three types of errors: substitution, insertion and deletion. For that reason, an appropriate penalty to each type of error should be taken into consideration.

A simple way may be assigning the same penalty to all the possible errors. This is called the Levenshtein metric which counts the number of corrections required for converting a sequence to the target sequence. In a number of preliminary experiments, we could observe that some specific error patterns occur more frequently than others. In that case, for a better performance, it is desired to assign a different penalty for each error pattern depending on its possibility of occurrence. One of the successful previous approaches is the phone confusion matrix, in which the penalties are estimated based on the recognition error patterns obtained from the training data [6]. However, the amount of recognition errors observed in the training data is usually considered to be insufficient to reflect all the possible phone error patterns. Here we propose a novel technique to determine each penalty matrix from a set of training data.

Let  $X = (x_1, x_2, \dots, x_{N_P})$  be an acoustic feature vector corresponding to a spoken phone sequence  $P = (p_1, p_2, \dots, p_{N_P})$ , in which  $N_P$  is the number of spoken phones and  $x_i$  represents the feature vector segment associated with the  $i$ -th phone,  $p_i$ . Given  $P$ ,  $X$  can be segmented into each phone region  $\{x_i\}$  by applying forced alignment such as the Viterbi decoding approach. Once  $X$  is segmented into  $(x_1, x_2, \dots, x_{N_P})$ , the log-likelihood for  $P$  can be factorized as follows:

$$\log \mathcal{P}(X|P) = \sum_{k=1}^{N_P} \log \mathcal{P}(x_k|p_k) \quad (4)$$

where  $\mathcal{P}(\cdot)$  denotes the likelihood computed in the HMM framework.

For a good performance of string match, it is desirable to assign a heavy penalty to the error type that occurs rarely and light penalties to frequent error patterns. To estimate the relative frequency of each error pattern, we deliberately substitute or delete the spoken phones and insert non-spoken phones so as to create intended phone error patterns. The penalty matrix for substitution is computed as follows:

$$PM_{sub}(\phi_i, \phi_j) = \begin{cases} -\log \Pr [\mathcal{P}(x_k|\phi_j) < \mathcal{P}(x_k|\phi_i) | p_k = \phi_j] \\ +\alpha_{sub}, & \phi_i \neq \phi_j \\ 0, & \phi_i = \phi_j \end{cases} \quad (5)$$

with

$$\begin{aligned} & \Pr [\mathcal{P}(x_k|\phi_j) < \mathcal{P}(x_k|\phi_i) | p_k = \phi_j] \\ & \simeq \frac{\sum_{k=1}^{N_P} \mathbb{I}[p_k = \phi_j, \mathcal{P}(x_k|\phi_j) < \mathcal{P}(x_k|\phi_i)]}{\sum_{k=1}^{N_P} \mathbb{I}[p_k = \phi_j]} \end{aligned} \quad (6)$$

where  $\Pr[\cdot]$  denotes the probability of the enclosed event,  $\mathbb{I}[a]$  represents the indicator function which equals 1 when the condition  $a$  is satisfied and 0 otherwise, and  $\alpha_{sub}$  is a non-negative control parameter which is empirically determined depending on the phone insertion/deletion rates. In (5) and (6), it is noted that we make a deliberate substitution of the spoken phone  $\phi_j$  with another phone  $\phi_i$  and if the likelihood of the substituted phone becomes larger, then we treat the case as a possible phone substitution error. From that,  $PM_{sub}(\phi_i, \phi_j)$  can be trained based on the training data  $X$ .

In a similar manner,  $PM_{ins}(\phi_i, \phi_j)$  is obtained by

$$\begin{aligned} & PM_{ins}(\phi_i, \phi_j) \\ & = -\log \Pr [\mathcal{P}(x_k|\phi_j) < \mathcal{P}(x_k|\phi_j, \phi_i) | p_k = \phi_j] + \alpha_{ins} \end{aligned} \quad (7)$$

with

$$\begin{aligned} & \Pr [\mathcal{P}(x_k|\phi_j) < \mathcal{P}(x_k|\phi_j, \phi_i) | p_k = \phi_j] \\ & \simeq \frac{\sum_{k=1}^{N_P} \mathbb{I}[p_k = \phi_j, \mathcal{P}(x_k|\phi_j) < \mathcal{P}(x_k|\phi_j, \phi_i)]}{\sum_{k=1}^{N_P} \mathbb{I}[p_k = \phi_j]} \end{aligned} \quad (8)$$

In (7) and (8), we replace the original spoken phone  $\phi_j$  by the concatenated phones  $(\phi_i, \phi_j)$ , and  $\alpha_{ins}$  is an experimentally determined control parameter. The likelihood  $\mathcal{P}(x_k|\phi_j, \phi_i)$  can be easily calculated by constructing an HMM concatenating two phone models for  $\phi_j$  and  $\phi_i$ .

Finally, the penalty matrix for deletion is estimated as follows:

$$\begin{aligned} & PM_{del}(\phi_i, \phi_j) \\ & = -\log \Pr [\mathcal{P}(x_{k-1}, x_k|\phi_i, \phi_j) < \mathcal{P}(x_{k-1}, x_k|\phi_i) | \\ & \quad p_{k-1} = \phi_i, p_k = \phi_j] + \alpha_{del} \end{aligned} \quad (9)$$

with

$$\begin{aligned} & \Pr [\mathcal{P}(x_{k-1}, x_k|\phi_i, \phi_j) < \mathcal{P}(x_{k-1}, x_k|\phi_i) | p_{k-1} = \phi_i, p_k = \phi_j] \\ & \simeq \frac{\sum_{k=2}^{N_P} \mathbb{I}[p_{k-1} = \phi_i, p_k = \phi_j, \mathcal{P}(x_{k-1}, x_k|\phi_i, \phi_j) < \mathcal{P}(x_{k-1}, x_k|\phi_i)]}{\sum_{k=2}^{N_P} \mathbb{I}[p_{k-1} = \phi_i, p_k = \phi_j]} \end{aligned} \quad (10)$$

where  $(x_{k-1}, x_k)$  is the concatenation of the two feature vector segments,  $x_{k-1}$  and  $x_k$ , and  $\alpha_{del}$  is an empirically determined control parameter.

The proposed technique is similar to the phone confusion matrix particularly for the substitution error. Phone confusion matrix is derived from the recognition errors observed in the training data [6]. In contrast, the proposed method deliberately creates all the possible error patterns, which will be helpful for robust penalty estimation. Furthermore, more sophisticated treatment of the insertion and deletion errors is achieved compared to the phone confusion matrix technique.

Table 1: Database used in this study

	Number of sentences	Speakers (male/Female)	Duration (hh:mm:ss)
Training	30,399	298 (149/149)	64:08:36
Development	6,904	68 (34/34)	13:35:08
Testing	3,454	34 (17/17)	07:12:47
Total	40,757	400 (200/200)	84:56:31

## 4. Experimental Results

Performance of a keyword spotting algorithm with the proposed penalty matrices was evaluated on the Korean continuous speech Reading Sentence DB collected at Speech Information Technology & Industry Promotion Center (SiTEC) [12]. The SiTEC Reading Sentence DB contains 20,217 sentences consisting of about 30,000 different word tokens. It was collected by recording the speech from 200 male and 200 female speakers. The number of keywords was 1000.

The database was divided into three sets: training, development and testing sets. The training set was used for the estimation of HMM parameters, the development set was used to train the phone mismatch penalty matrices, and the testing set was used for the performance evaluation. A detailed information of each set of the DB is shown in Table 1.

In our keyword spotting system, the HMM-based phone recognizer was applied at the first stage. Context-dependent triphone models were used to construct this phone recognizer. The number of states for each phone was three, and the number of Gaussian mixtures for each HMM state was eight. As the output of the phone recognition, we generated  $N$ -best phone sequences or phone lattices where  $N$  varied from 1 to 100. The phone recognition accuracy calculated based on the 1-best phone sequences was 69.30%. The phone mismatch penalty matrices were estimated on the development set. To train the penalty matrices, we applied monophone HMMs instead of the triphone models used in the phone recognizer. The control parameters  $\alpha_{sub}$ ,  $\alpha_{ins}$  and  $\alpha_{del}$  were set to 0.5, 0.5 and 1.5, respectively, which showed a good performance in our experiments.

As reference systems with which we compared the performance, we also implemented two other keyword spotting systems which were similar to our approach but employed different penalties [6]. The first one employed the Levenshtein distance as the penalties. As we mentioned in the previous section, the goal of the Levenshtein distance is to find the minimum edit distance from the recognized phone sequences to keyword phone sequences. Let  $LD_{sub}(\phi_i, \phi_j)$ ,  $LD_{ins}(\phi_i, \phi_j)$  and  $LD_{del}(\phi_i, \phi_j)$  respectively denote the  $(i, j)$ th penalty value for substitution, insertion and deletion defined by the Levenshtein distance metric. Then,

$$\begin{aligned} LD_{sub}(\phi_i, \phi_j) &= \begin{cases} 1, & \phi_i \neq \phi_j \\ 0, & \phi_i = \phi_j \end{cases} \\ LD_{ins}(\phi_i, \phi_j) &= 1 \\ LD_{del}(\phi_i, \phi_j) &= 1. \end{aligned} \quad (11)$$

In the second system, the substitution penalties were calculated by applying the phone confusion matrix [6] technique. Let  $CM_{sub}(\phi_i, \phi_j)$ ,  $CM_{ins}(\phi_i, \phi_j)$  and  $CM_{del}(\phi_i, \phi_j)$  respectively be the  $(i, j)$ th penalty value for substitution, insertion and

Table 2: FOMs (%) for the two-stage keyword spotting with PMP-LD, PMP-CM and PMP-PM.

Outputs of the 1st stage	Phone mismatch penalties of the 2nd stage		
	PMP-LD	PMP-CM	PMP-PM (proposed)
1-best	66.18	72.31	72.62
10-best	69.93	74.75	75.64
50-best	72.71	76.84	77.80
100-best	74.03	77.79	78.80
Lattice	77.79	80.01	81.64

deletion with the phone confusion matrix. Then,

$$\begin{aligned}
 CM_{sub}(\phi_i, \phi_j) &= \log\{S(i, i)/S(i, j)\} \\
 CM_{ins}(\phi_i, \phi_j) &= I \\
 CM_{del}(\phi_i, \phi_j) &= D.
 \end{aligned} \tag{12}$$

where  $S$  is the  $N_\phi \times N_\phi$  confusion matrix and  $S(i, j)$  indicates the number of times the recognizer substituted the  $\phi_i$  by the  $\phi_j$  as a fraction of the total number of recognized instances of the  $\phi_i$ .  $I$  and  $D$  are set to 3.5, which showed the best performance in our experiments.

We evaluated the performance of the two-stage keyword spotting systems with three different phone mismatch penalties: PMP-LD, PMP-CM and PMP-PM. For convenience, we denote the phone mismatch penalties derived from Levenshtein metric by PMP-LD, from phone confusion matrix by PMP-CM, and from the proposed penalty matrices by PMP-PM.

The performances of the keyword spotting systems based on the proposed and the reference penalties were compared in terms of figure of merit (FOM) and receiver operating characteristic (ROC) curves. The FOM implies the average detection probability when the number of false alarms per keyword per hour is kept between 0 and 10. First, we compared the FOMs of the three approaches as shown in Table 2. We can see that the technique using the proposed PMP-PM significantly outperformed those using the other approaches. The average relative improvement in FOM of the technique using the proposed penalties was 18.48 % compared to that using PMP-LD and 4.31 % compared to that using PMP-CM. Next, the ROC curves for these three systems were obtained as shown in Figure 2 when the first stage output was given by a phone lattice. The detection probability of the proposed algorithm was higher than that of the other approaches over the whole range of false alarms per keyword per hour. From the results, it can be concluded that the proposed approach produced better detection performance compared with the conventional approaches.

## 5. Conclusions

In this paper, we have presented a new technique to estimate the phone mismatch penalty matrices for two-stage keyword spotting. Proposed penalty matrices are employed to measure the similarity between the recognized multi-pass results and the hypothesized lexical phone sequences of the keywords. The phone mismatch penalties are estimated from the training data while considering all possible types of phone recognition errors. When the outputs of  $N$ -best phone sequences or phone lattices are given at the first stage, detection of a specific keyword is performed through dynamic programming based on the penalty matrices. From a number of comparative experiments, the proposed method has shown better performance than other approaches in two-stage keyword spotting.

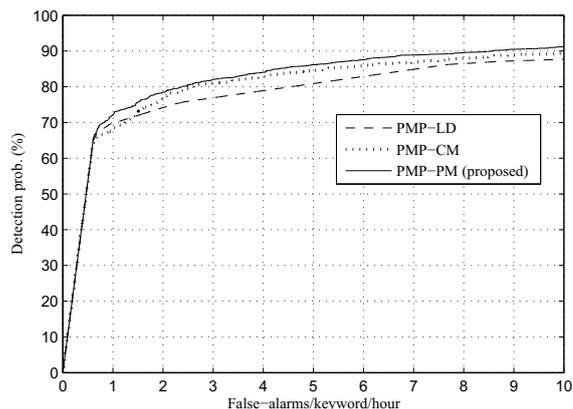


Figure 2: ROC curves for two-stage keyword spotting with PMP-LD, PMP-CM and PMP-PM (phone lattice).

## 6. Acknowledgements

This work was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0083044) and by the Seoul R&BD Program (10544).

## 7. References

- [1] S. Dharanipragada and S. Roukos, "A multistage algorithm for spotting new words in speech," *IEEE Trans. Speech, Audio Process.*, vol. 10, no. 8, pp. 542-550, Nov. 2002.
- [2] D. A. James and S. J. Young, "A fast lattice-based approach to vocabulary independent wordspotting," *Proc. ICASSP*, pp. 899-902, Apr. 1997.
- [3] K. M. Knill and S. J. Young, "Low-cost implementation of open set keyword spotting," *Computer Speech and Language*, pp. 243-266, Jul. 1999.
- [4] K. Thambiratnam and S. Sridharan, "Dynamic match phone lattice searches for very fast and accurate keyword spotting," *Proc. ICASSP*, pp. 465-468, Mar. 2005.
- [5] L. ten Bosch, A. Hamalainen, O. Scharenborg, and L. Boves, "Acoustic scores and symbolic mismatch penalties in phone lattices," *Proc. ICASSP*, pp. 437-440, May 2006.
- [6] K. Audhkhasi and A. Verma, "Keyword search using modified minimum edit distance measure," *Proc. ICASSP*, pp. 929-932, Apr. 2007.
- [7] C. W. Han, S. J. Kang, and N. S. Kim, "Estimation of phone mismatch penalty matrices for two-stage keyword spotting," *IEICE Trans. Inf. Syst.* (submitted in Jan. 2010).
- [8] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. Information Theory*, vol. IT-21, no. 44, pp. 404-411, Jul. 1975.
- [9] S. Nakagawa and M. M. Milan, "Syllable-based connected spoken word recognition by two pass O(n) DP matching and hidden Markov models," *Proc. ICASSP*, pp. 1117-1120, Apr. 1986.
- [10] Q. Fu, X. He, and L. Deng, "Phone-discriminating minimum classification error (P-MCE) training for phonetic recognition," *IEEE Trans. Speech, Audio Process.*, vol. 10, no. 8, pp. 542-550, Nov. 2002.
- [11] F. Sha and L. K. Saul, "Comparison of large margin training to other discriminative methods for phonetic recognition by hidden Markov models," *Proc. ICASSP*, pp. 313-316, May 2007.
- [12] SiTEC website. [Online]. Available: <http://www.sitec.or.kr/>