Acta Cybernetica **18** (2008) 379–390.

An On-line Speaker Adaptation Method for HMM-based Speech Recognizers

András Bánhalmi* and András Kocsor*

Abstract

In the past few years numerous techniques have been proposed to improve the efficiency of basic adaptation methods like MLLR and MAP. These adaptation methods have a common aim, which is to increase the likelihood of the phoneme models for a particular speaker. During their operation, these speaker adaptation methods need precise phonetic segmentation information of the actual utterance, but these data samples are often faulty.

To improve the overall performance, only those frames from the spoken sentence which are well segmented should be retained, while the incorrectly segmented data should not be used during adaptation. Several heuristic algorithms have been proposed in the literature for the selection of the reliably segmented data blocks, and here we would like to suggest some new heuristics that discriminate between faulty and well-segmented data. The effect of these methods on the efficiency of speech recognition using speaker adaptation is examined, and conclusions for each will be drawn.

Besided post-filtering the set of the segmented adaptation examples, another way of improving the efficiency of the adaptation method might be to create a more precise segmentation, which should then reduce the chance of faulty data samples being included. We suggest a method like this here as well which is based on a scoring procedure for the N-best lists, taking into account phoneme duration.

Keywords: speech recognition, speaker adaptation, faulty transcripts, confidence measures, a posteriori phoneme probabilities

1 Introduction

The probabilistic models for speech recognition are normally trained on a large amount of data samples that contain utterances recorded from many speakers. While these speaker-independent models usually operate with a quite similar and acceptable performance for most speakers, speaker-dependent models which are

*Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged H-6720 Szeged, Aradi vértanúk tere 1., Hungary, E-mail: {banhalmi, kocsor}@inf.u-szeged.hu

trained just on the sample utterances of a particular speaker are much more efficient at the recognition task for this specific speaker. The problem with developing speaker-dependent systems for each speaker separately, however, is that large amounts of speech training data for each speaker may be unavailable and even difficult to acquire. In order to fine-tune the speaker independent model to achieve the efficiency of a speaker dependent model, the following important techniques have been already proposed.

The usual approaches for improving the performance of the speaker-independent models are the transformation of the incoming feature vectors (e.g. by VTLN, CMN or CVN) and the fine-tuning of the parameters in statistical acoustic models (speaker adaptation techniques). The aim of feature vector transformation-based methods is to transform (normalize) both the training and the test data in such a way that the classes are easier to separate. In most cases these methods try to normalize the input data with respect to a given parameter. The VTLN (vocal tract length normalization) method normalizes the spectrum of the input speech data by converting it as if all the samples had been pronounced by speakers with the same vocal tract length [10],while the basic CMN (cepstral mean normalization) method converts the cepstral coefficients of the input data so that the samples for each speaker have the same mean value [5].

The other approach for adjusting a speaker independent model in order to better approximate the performance of a speaker dependent model is speaker adaptation. In classical HMM-based systems various speaker-adaptation techniques have been used with success. These techniques fine-tune the parameters of the speaker-independent system to more 'suitable' ones corresponding to the adaptation data examples of a particular speaker.

In most cases, adaptation can be applied using three strategies: batch adaptation, self-adaptation and on-line adaptation [4, 15]. Batch adaptation performs the reestimation of the model parameters only after all the adaptation data samples have been collected, so it is an off-line method. Self-adaptation is performed on the testing data at runtime, without collecting adaptation data, so this is normally an unsupervised method. The on-line (or incremental) adaptation technique alters the parameters of the statistical model only after a block of adaptation data samples have been enrolled, and this block of data is thrown away after the adaptation method has been applied. Recognition errors and faulty transcripts pose an important problem when the above-mentioned algorithms are used. The main advantage of the on-line adaptation technique over the self-adaptation one is that it has access to much more information to separate the non-faulty adaptation data samples, which could be used in the adaptation phase with more success. The other important advantage of on-line adaptation is that it will adapt the previous model to those data samples which have the maximal probability after enrolling a block of data, hence this method should be much more stable than the self-adaptation one.

Computationally, two main approaches have been proposed in the literature for the adaptation of HMM parameters. The first is the maximum likelihood (ML) - based framework which contains the maximum likelihood linear regression (MLLR

[11]) approach, the maximum likelihood stochastic matching (SM) approach [18] and the constrained transformation approach [3].Some other adaptation techniques are based on the maximum a posteriori formulation (MAP [6]), where only the parameters of the states corresponding to the Viterbi path (the path with maximal probability values) are reestimated. Because the speech recognition algorithm in our speech recognition system works in a similar way (namely it approximates the forward probability with the maximal value along the Viterbi path), this latter technique might be more feasible in our framework than MLLR-like techniques. The second main approach contains discriminative adaptation techniques like MCE (Minimum Classification Error) [9] and MMI (Maximum Mutual Information) [16, 17], all of which try to maximize the recognition accuracy explicitly for a given vocabulary.

Our goal with the experiments presented in this paper was to improve the performance of a continuous speech recognizer by applying some modifications to the supervised adaptation process. As the first step of HMM phoneme model adaptation, the recognition system has to collect the phonetic data from the utterances of the user. To achieve this an automatic segmentation is carried out by the speech recognizer. In Section 2 we will provide a short description about of the key aspects of our framework. The automatic segmentation phase however can be faulty for a variety of reasons; e.g. the initial model is of poor quality, noise has been introduced by the microphone or by the user, or simply the user stutters, or misreads words. Our aim was to exclude these faulty segmented data items from the adaptation or at least to reduce their number. In the literature several methods have been proposed to tackle this problem. Confidence measures have been investigated to help the adaptation process deal with faulty transcripts [1, 12, 7].In the latter articles confidence measures were used to mark possible recognition errors and to exclude the erroneously segmented words from the adaptation process. In Section 3 we also propose two confidence-measure-like methods for improving the efficiency of speaker adaptation.

Actually, the frequency of the occurrence of faulty adaptation data samples can also be reduced directly at the on-line speech recognition level when the automatic segmentation for the speech signal is being performed. When the speech recognizer fills up the N-best hypothesis stacks, the scores of the hypotheses can be set to eliminate certain (possibly faulty) hypotheses from the stack. Here we give a method like this as well in Section 3.

2 The Speech Recognizer and the Adaptation Module

The speech recognizer employed here is a continuous density HMM-based Viterbi N-best decoder extended with some speed-up and pruning techniques for the purpose of being real-time [2]. The speed-up techniques include, for example, some constraints for the hypothesis extension procedure, thresholds for the stack size and for the maximum number of new hypotheses, and fast Gaussian computation tech-

niques. The adaptation module is based on the same program code as that used in the decoder, with the same adjustable constraints and parameters to guarantee the equivalence between the adaptation process and the continuous speech recognition process. Hence we can say that the program has a recognition mode, and also an adaptation mode. The difference between the recognition mode and the adaptation mode lies in the following points:

- When the adaptation mode is running, a huge amount of auxiliary data has to be stored for each active hypothesis in the N-best list whose data will be used to trace back the Viterbi path and to find the mixture and state indices of the HMMs belonging to the Viterbi path during the adaptation process.
- In order to make the search process efficient, some hypotheses are unified after each hypothesis extension. The unifying technique of the adaptation mode differs from that of the speech recognition mode in the sense that in the adaptation mode not all the hypotheses with the same phonetic history are unified, but just those whose trace-back data can be unified without information loss.
- Because of the large amount of stored data in the adaptation mode, the stack size should be reduced to keep the process real-time.

When the recognizer is in the adaptation mode, it will store the following data that will be used to trace back the Viterbi path and compute the new adapted HMMs. These data items are:

- the probability matrix for each state of each HMM,
- the mixture index matrix for each state of each HMM,
- the state matrix (with the series of phonetic labels, respectively) containing the state from which the given state was attained with maximal probability.

From these data items the Viterbi path, the phonetic segments and all the other data necessary for the computation of the adapted parameters can be easily obtained. In order to efficiently store all these data items for all the hypotheses, these data items are kept in a tree structure. If all hypotheses share a common root, then, up to the end of this common root, only one matrix from each type is stored, so the algorithm has linear storage requirements.

The adaptation module can be used both for supervised adaptation and for unsupervised adaptation. Supervised adaptation means that the uttered word series are known. The possible phonetic transcript variants of the word series are defined by a grammar containing rules which only permit the type of phoneme series that could be the phonetic transcription of the given word series. This simple grammatic model can also take into account assimilation rules and the possibility that there are silent gaps between the words. However, when the adaptation process is unsupervised, not a simple grammar, but a rather complex language model is used by the continuous speech recognizer to build up N-best lists of possible hypotheses.

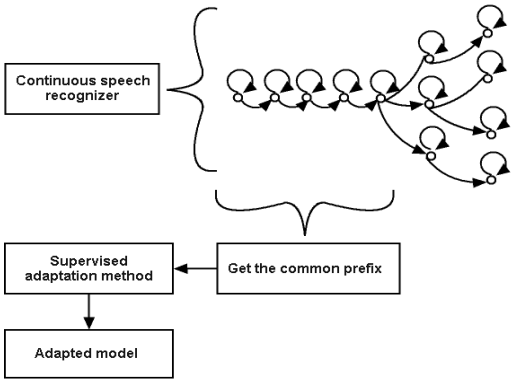

Figure 1: The process of unsupervised adaptation

After processing a few frames, the hypotheses in the N-best stack will have a common phonetic transcription prefix (they will have this, because N-best pruning is used, and the number of hypotheses grows exponentially); afterwards this common segment of the spoken data can be processed by applying the supervised method (see Figure 1).

3 Reducing the Amount of Faulty Adaptation Data

3.1 Extracting the Adaptation Data

As we mentioned earlier, the adaptation process could be more efficient if faulty adaptation data samples were removed from the well-segmented ones before the adaptation process was carried out. We will now discuss a new and simple Viterbi N-best cutting constraint that can be used to efficiently remove faulty adaptation data samples. Methods like this can be computed only with a highly limited stack size because of their high computational cost. So it cannot be normally used in the speech recognition process, but it can be used when adaptation data extraction is being performed, because the data samples necessary for this method are stored and are accessible.

It is a widely accepted property of HMMs that their transitional probability values do not model the duration of phoneme utterances very well. Moreover, the Δ and $\Delta\Delta$ feature components are strongly influenced by the phoneme duration (because they measure how fast the features change). When samples are taken from many different speakers, these features can be very different, so the resulting accuracy of the phonetic segmentation could be quite low. When the duration of the phoneme is modeled incorrectly, long phoneme durations can occur many times. In reality this is very unlikely, except in the case when the phoneme model has to model silent phases between two words. Our aim here is to reduce the number

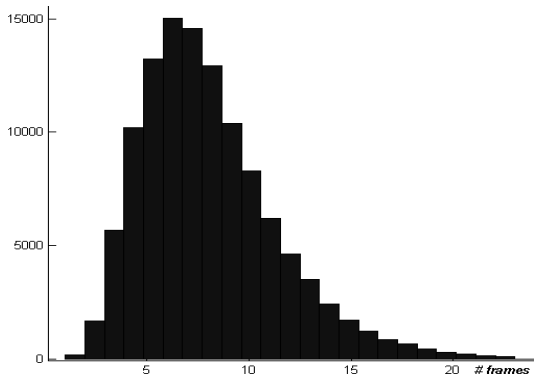

Figure 2: The histogram of the duration of all the phonemes except silence

of hypotheses containing an abnormally long phoneme utterance. Because the Viterbi path is stored for each hypothesis and for each state of the corresponding HMM models, the duration of the last phoneme can be computed using the Viterbi path corresponding to the HMM state having the maximal probability (for each particular state of a HMM there are different Viterbi paths). If the hypotheses with an unlikely duration are punished, then hypotheses with likely phoneme durations will be kept in the N-best list with a higher probability. With this in mind we define an a posteriori phoneme-weighting function by which the probability score of all hypotheses get multiplied:

$$\varphi_L = \begin{cases} 1 & ,\text{if } \mathrm{L} < \theta \\ e^{-\alpha(L-\theta)} & ,\text{if } \mathrm{L} \geq \theta \end{cases}$$

Here L means the duration of the phoneme computed from the Viterbi path. For the phoneme duration threshold (θ) we used a value of 15, and for the punishment constant (α) we used a value of -5. The duration threshold value was obtained from the histogram above (see Figure 2), this histogram was computed on the training database described later in the Section 4.

3.2 Confidence Measures for Adaptation

Many heuristic methods have been proposed in the literature for separating faulty adaptation data from correctly segmented ones. Another family of methods performs a weighting of the learning data samples when the adapted model is computed, here the weights are based on some particular confidence values. In this section we propose some new confidence-measure-like heuristics to reduce the number of faulty adaptation data samples.

Our first confidence measure is based on the observation that many of the incorrect segment boundaries are wrongly positioned by just a small amount. This

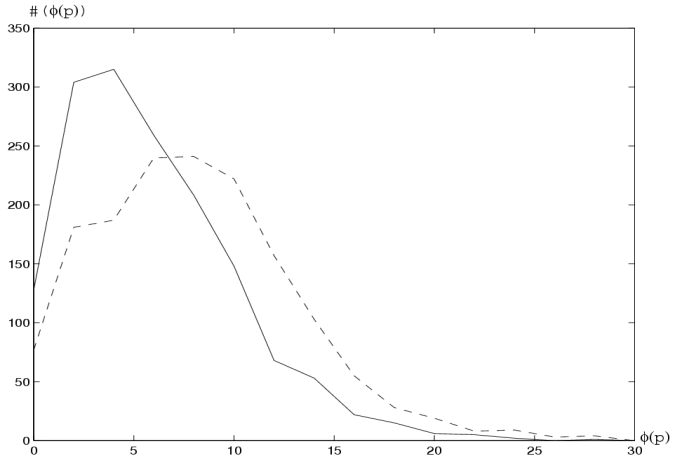

Figure 3: The histogram of the first confidence-measure scores on well-segmented samples (solid line), and on incorrectly segmented samples (dashed line)

means that the faulty segment parts at the boundaries will generally give a lower probability for the first and last states of the HMM. The difference between the probability values at the segments boundaries and the probability of the intermediate part will be higher for the faulty data than for the correctly segmented data. Based on this idea, we devised a simple formulation to measure this difference:

$$\varphi(p) = \left| \frac{\log P(f_1|\Theta = p) + \log P(f_N|\Theta = p)}{2} - \frac{\sum\limits_{i=2\ldots N-1} \log P(f_i|\Theta = p)}{N-2} \right|,$$

where p is the recognized phoneme, Θ represents the HMM model, and f_i is a feature vector for the recognizer. Here φ is the function ranking the test phonetic data, and a classification between the correct and faulty samples can be done using an acceptance threshold.

In order to determine the proper threshold value, we had run an algorithm on the training set which computed the distribution of the above-mentioned scores on correctly segmented data and on incorrectly segmented data. The two histograms are shown in Figure 3 above.

The second scoring method proposed by us is based on the Fisher score [13], [8]. The Fisher score of a probabilistic model (which fits a probabilistic distribution to the data, and uses the Bayes rule for classification) is the gradient of the log-likelihood of a feature series with respect to the model parameters. Put formally,

$$U_f = \nabla_\Theta \log(P(f|\Theta))$$

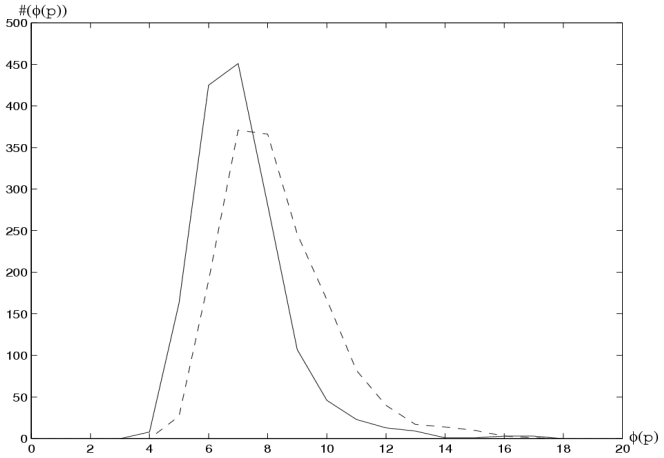

Figure 4: The histogram of the second confidence-measure scores on well-segmented samples (solid line), and on incorrectly segmented samples (dashed line)

Here the parameter Θ represents the model, while the parameter f stands for the feature vector series to be modeled. The gradient vector here measures how much the log-likelihood of the feature data series changes when the model parameters are varied. When using a Gaussian distribution, the Fisher score components with respect to the means of the Gaussian mixtures can be computed via the following formula:

$$\nabla_{\mu_\mathrm{k}} \log(P(f|\mu_\mathrm{k})) = \sum_{i=1}^{m} P(k|f_i)\Sigma_k^{-1}(f_i - \mu_\mathrm{k})$$

Here m stands for the number of feature vectors modeled by the k-th Gaussian distribution. The feature vector series are denoted by f, and their elements are denoted by f_i.

We used the above-defined Fisher score to construct a confidence measure which has the following form:

$$\varphi(p) = \frac{1}{N} \sum_{i=1}^{N} ||\nabla_{\mu_\mathrm{k}} \log(P(f_i|\mu_\mathrm{k}))||,$$

where N is the number of feature vectors, and k is the index of the mixture that corresponds to the Viterbi path. Figure 4 above shows the distribution of the scores for the correctly segmented data samples and for the incorrectly segmented data samples.

4 Experiments and Results

The adaptation technique used here was the MAP (maximum a posteriori) method, which can be used for incremental adaptation by applying the following recursive formula [19]:

$$\mu_{d,N+1} = \frac{x_{N+1} + (N+\alpha) \cdot \mu_{d,N}}{N+1+\alpha}.$$

Here the parameter N represents the number of adapting examples for the given mixture component, while the parameter α controls how fast the mean (μ_d) is altered by this linear regression procedure.

For testing purposes we used the following settings (only the main parameters are given here):

- The continuous speech recognizer had a stack size value of 1000. The maximum number of new hypotheses for the phonetic hypothesis extension was set to 250, and the log-likelihood cut-off parameter was set to 260.
- The phoneme HMM models were monophone HMMs with 3 states, each state having a mixtures of 3 Gaussian distributions.
- The stack size of the adaptation method was 30, and the maximum number of new hypotheses in adaptation mode was restricted to 20. The log-likelihood cut-off parameter was set to 260.
- The recognition system used the conventional mel-frequency cepstral coefficient (MFCC) features. More precisely, 13 coefficients were extracted from each 25 msec frame, along with their Δ and $\Delta\Delta$ values, at a frame rate of 100 frames/sec.
- The α value of the MAP adaptation formula was set to 0.3. Other values were also tested in the interval $(0, 1)$, but no significant difference was observed in the results.
- The threshold for the first confidence-measure method had a value of 5, and the threshold value used by the second method was 8. These settings were selected to reduce a relatively high amount of faulty adaptation data (see Figures 3 and 4).

In the experiments our own training, test and speaker adaptation databases were employed. These databases and the continuous speech recognition system were created by the Research Group on Artificial Intelligence, the University of Szeged and the Laboratory of Speech Acoustics of the Budapest University of Technology and Economics within the framework of the Hungarian Medical Dictation Project financially supported by the national fund IKTA-056/2003. The adaptation database contained spoken sentences of 2 male(denoted by L and T) and 2 female (denoted by A and B) speakers, each speaker uttered 17 paragraphs from the same text.

Database	A	B	L	T	Average
Normal Adaptation	93.32%	92.28%	97.52%	93.29%	94.10%
Using PAPL	94.48%	93.48%	98.10%	94.10%	95.04%
WER reduction	17.36%	15.54%	23.38%	12.07%	17.08%

Table 1: Relative word error rate reductions achieved when using the phoneme a posteriori likelihood (PAPL) method on the adaptation data sets of four speakers (A, B, L, T).

Database	A	B	L	T	Average
Base Adaptation	94.48%	93.48%	98.10%	94.10%	94.97%
1st method	94.92%	93.16%	98.10%	94.46%	95.16%
2nd method	95.77%	93.16%	98.33%	94.32%	95.41%
WER reduction 1.	7.9%	-1.32%	0%	6.10%	3.9%
WER reduction 2.	23.36%	-1.32%	13.68%	3.7%	8.8%

Table 2: Relative word error rate reductions achieved when using the proposed two confidence measures on the adaptation data sets of four speakers (A, B, L, T).

The average duration of this speech data was 6 minutes per speaker, and the total number of phoneme examples for the 44 monophone classes was 3500 on average. The test database contained the utterances of the same four speakers, the recordings of 20 medical reports being taken from each speaker. The HMMs were trained using the MRBA database that was created by the Research Group on Artificial Intelligence, the University of Szeged and the Laboratory of Speech Acoustics of the Budapest University of Technology and Economics within the framework of the Hungarian Medical Dictation Project. This database contains 85365 phoneme examples, 26 female speakers, and 74 male speakers. The grammar model built for testing purposes was a simple word 3-gram, with a dictionary containing some 500 words. This grammar was trained on a text corpus built from 2500 thyroid gland medical reports. The grammar model for the supervised adaptation contained 162 assimilation rules.

Table 1 shows the results of our experiments when using the a posteriori probability multiplier for punishing the hypotheses with extremely long phoneme durations in the N-best stack. The results show that there is a definite improvement in the efficiency of the adaptation when this kind of scoring technique is applied. The experimental results using our confidence-measure-based methods to select the adaptation data are listed in Table 2. As the reader can see, using theses methods, a relative word error rate (WER) reduction of 4-8% was achieved on average, but the WER reduction was not always positive. The reason for the instability of these methods might be due to a significant reduction in the amount of adaptation data.

5 Conclusions and Further Work

Our results show that the efficiency of an adaptation method can be improved in two ways suggested here: by increasing the robustness of the method which extracts the adaptation data so that the automatically segmented data of the uttered sentence should contain fewer incorrect segments, and by selecting and dropping probably faulty adaptation data samples after the segmentation process. Building upon these promising results, more advanced adaptation data filtering methods will be tried in the near future which apply Data Description (One-Class Classification) methods that have a better scoring mechanism for phonetic segments, and are also better able to separate faulty data samples from the good data samples.

References

[1] T. Anastasakos, S.V. Balakrishnan, *The Use of Confidence Measures in Unsupervised Adaptation of Speech Recognizers,* Proc. Int. Conf. on Spoken Language Processing, Vol. 6, pp. 2303–2306., Sydney, NSW, Australia, Dec. 1998.

[2] András Bánhalmi, Dnes Paczolay, Lszl Tth *An Empirical Study on the Performanc of a CSR System Respect to Various Hypothesis-Space Pruning Techniques,* V. MSZNY, pp. 56–68., Szeged, Hungary, 2007.

[3] Digalakis, V. Rtischev, D. and Neumeyer, L., *Speaker Adaptation Using Constrained Reestimation of Gaussian Mixtures,* IEEE Trans. on Speech Audio Processing, 3, pp. 357–366., 1995.

[4] V. Digalakis, *On-line Adaptation of Hidden Markov Models Using Incremental Estimation Algorithms,* Proc. Eurospeech '97, pp. 1859–1862., Rhodes, Greece, 1997.

[5] S. Furui, *Cepstral Analysis Technique for Automatic Speaker Verification,* J. Acoust. Soc. Amer., Vol. 55, pp. 1204–1312., June, 1974.

[6] J.-L. Gauvain and C.-H. Lee, *Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains,* IEEE Transactions on Speech and Audio Processing, 2(2):pp. 291–298., April 1994.

[7] S. Homma, K. Aikawa and S. Sagayama, *Improved Estimation of Supervision in Unsupervised Speaker Adaptation,* Proc. of ICASSP-97, Vol. II, pp. 1023–1026., 1997.

[8] Jaakkola, T., Diekhans, M. and Haussler, D., *Using the Fisher Kernel Method to Detect Remote Protein Homologies,* Proceedings of the Internation Conference on Intelligent Systems for Molecular Biology, pp. 149–158., Aug. 1999.

[9] B.-H. Juang, S.Katagiri, *Discriminative Learning for Minimum Error Classification,* IEEE Trans. on Signal Processing, Vol. 40, No. 12, pp. 3043–3054., 1992.

[10] L. Lee and R.C.Rose, *Speaker Normalisation Using Efficient Frequency Warping Procedures,* Proc. ICASSP96, pp. 353–356., Atlanta, GA, 1996.

[11] C. Leggetter, P. Woodland, *Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density HMMs,* Computer Speech and Language, Vol. 9, pp. 171–185., 1995.

[12] T. Matsui and S. Furui, *N-Best-Based Instantaneous Speaker Adaptation Method for Speech Recognition,* Proc. of ICSLP-96, Vol. III, pp. 973–976., 1996.

[13] P.J.Moreno P.Ho and N.Vasconceles, *A Kullback-Leibler Divergence Based Kernel for SVM Classification in Multimedia Applications,* Advances in Neural Information Processing Systems 16. MIT Press, Cambridge, 2004.

[14] P. Nguyen, P. Gelin, J.C. Junqua, J.T. Chien, *N-Best Based Supervised and Unsupervised Adaptation for Native and Non-Native Speakers in Cars,* Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Vol. 1, pp. 173–176., Phoenix, AZ, USA, March 1999.

[15] P. Nguyen, L. Rigazio, R. Kuhn, JC. Junqua, C. Wellekens, *Self-adaptation using eigenvoices for large-vocabulary continuous speech recognition,* In Adaptation-2001, pp. 37–40, 2001.

[16] Y. Normandin, R. Cardin, R. De Mori, *High-Performance Connected Digit Recognition Using Maximum Mutual Information Estimation,* IEEE Trans. on Speech and Audio Processing, Vol. 2, pp. 299–311., April 1994.

[17] W. Reichl, G. Ruske, *Discriminative Training for Continuous Speech Recognition,* Proc. EUROSPEECH, Madrid, Spain, pp. 537–540., 1995.

[18] Sankar, A. and Lee, C.H., *A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition,* IEEE Trans. on Speech and Audio Processing, 4 (3), pp. 190–202., 1996.

[19] E. Thelen, *Long Term On-Line Speaker Adaptation for Large Vocabulary Dictation,* IEEE ICSPL, 4: pp. 2139–2142., October 1996.

[20] F. Wallhoff, D. Willet, G. Rigoll, *Frame-Discriminative and Confidence-Driven Adaption for LVCSR,* Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Vol. 3, pp. 1835–1838., Istanbul, Turkey, June 2000.

[21] Vicsi, K., Kocsor, A., Teleki, Cs., Tth, L., *Hungarian Speech Database for Computer-using Environment in Offices* II. MSZNY, pp. 315–318., Szeged, Hungary, 2004.

[22] T. Zeppenfeld, M. Finke, K. Ries, M. Westphal, A. Waibel, *Recognition of Conversational Telephone Speech Using the Janus Speech Engine,* Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Vol. 3, pp. 1815–1818., Munich, April 1997.