SELECTIVE EM TRAINING OF ACOUSTIC MODELS BASED ON SUFFICIENT STATISTICS OF SINGLE UTTERANCES

Tobias Cincarek, Tomoki Toda, Hiroshi Saruwatari and Kiyohiro Shikano

Graduate School of Information Science
Nara Institute of Science and Technology, Japan
cincar-t@is.naist.jp

ABSTRACT

In this paper, a new algorithm for selective training of acoustic models is proposed. The algorithm is formulated for an HMM-based model with Gaussian mixture densities, but works in principle for any statistical model, which has sufficient statistics. Since there are too many possibilities for selecting a data subset from a larger database, a heuristic has to be employed. The algorithm is based on deleting single utterances from a data pool temporarily or alternating between successive deletion or addition of utterances. The optimization criterion is the likelihood of the new model parameters given some development data, which can be calculated in a short amount of time based on sufficient statistics. The method is applied to automatically obtain task-dependent acoustic models for infant and elderly speech by selecting utterances from a data pool which are acoustically close to the development data. The proposed method is computationally practical and also addresses the issue of reducing the high costs evolving from the development of applications which make use of speech recognition technology.

1. INTRODUCTION

Statistical models like HMMs employed for acoustic modeling in speech recognition have a large number of parameters. To reliably train such models a huge speech database is required. However, the collection of speech data, including recording and transcription, is a very time-consuming and costly process. For example, about half of the relative costs to develop an interactive dialog system are due to database preparation [1]. Moreover, the performance of an acoustic model depends on various factors such as speaker characteristics (e.g. gender, age, accent), speaking style (e.g. read, spontaneous), acoustic conditions (e.g. noise, microphone) and the application. It is impractical to provide enough training data for each possible combination.

Instead of task-dependent acoustic modeling, there are also attempts to build a task-independent acoustic model, which is portable among different applications. In [2] several ways to obtain a generic model are examined: adaptation or retraining with speech data from multiple sources. Unsupervised adaptation with task-specific data to address the issue of cost reduction is also investigated. The overall gap of performance between the generic and task-dependent models was small. While there was an improvement over a task-dependent model w.r.t. some tasks (e.g. spontaneous dictation), for others (e.g. digit recognition) a degradation was observed.

This work is supported by the MEXT COE and e-Society project.

Our goal is the development of an automatic method which enables the construction of a task-dependent model within a short amount of time and at most very few additional costs. The approach of employing speech data from multiple speech databases as described in the last passage is effective for building a generic model. Here, our idea is to employ only a subset of the speech data available from multiple sources to build a task-dependent model. This requires a method to select those items from the data pool which are close to the desired target task and acoustic conditions.

In recent years, proposals for training procedures which make selective use of training data, emerged in literature. A selective training method for HMMs is described in [3]. Each training sample is weighted by a confidence measure in order to control the influence of outliers. The approach was applied to improve the statistical models for accent and language identification. Active learning is employed in [4] in order to reduce the effort necessary for database preparation. Only those utterances with a low recognition confidence score are transcribed and employed for training. Collection of additional data is only carried out as long as the likelihood of the trained model given the selected training data is no longer increasing. Experiments revealed, that the best model is not necessarily obtained when using the whole database for training, but when only using a subset of the whole data.

There are also adaptation methods which make selective use of data. Training speakers which are close to the test speaker are chosen based on the likelihood of speaker GMMs given the adaptation data [5]. The adapted model is constructed from combining precomputed HMM sufficient statistics for the training data of the selected speakers. A similar paradigm is employed in [6], where cohort models close to the test speaker are selected, transformed and combined linearly.

In order to select speech data from a large data pool, the selection procedure from [5, 6] is not applicable, if the speaker label of each utterance is unknown or if there are only few utterances per speaker. This can be the case for data which was collected automatically. e.g. by a dialogue system for public use such as Takemaru-kun [7]. In the following, a new algorithm is proposed, which is able to select single utterances from a data pool based on the ML criterion. Different to [3] is, that no weighting of single patterns is done. The selection unit is larger than in [3] but smaller than in [5]. Other than in [4, 3], the optimization criterion is the model likelihood given a small amount of development data, which can be calculated in a short amount of time based on sufficient statistics. The development set has to be designed to represent the desired target task well. However, the costs for providing this data are far lower than the collection of a larger amount of task-specific training data.

0-7803-9479-8/05/$20.00 © 2005 IEEE

2. PROPOSED APPROACH

The starting point is the following scenario: One or more rather large speech databases are available. The conglomerate of several databases will be called *training data (pool)*. Our goal is to obtain an acoustic model for a certain task (or condition). However, there is only little or no speech data for the target task. If there is no data, just a small development set has to be collected. A variation of this scenario would be, that there are quite a few data for the target task available, but not enough to train a model robustly. Consequently, it is desired to select additional data from the large data pool which are close to the development data.

There are too many possibilities to select a subset of k utterances out of a data pool with n utterances. Even if the number of possibilities is reduced heuristically, it still takes too much time to retrain the model and calculate its likelihood for each considered subset. In the case of a statistical model like HMM, which has sufficient statistics, the time aspect is no longer problematic, however. Sufficient statistics (e.g. see [8]) have the nice property to contain all information necessary to reconstruct the parameters of a model. Furthermore, they can be decomposed w.r.t. the training patterns.

In the following section it is shown, that a likelihood criterion can be defined, which is based only on the sufficient statistics of the selected training data and the development data. This makes it possible to successively delete utterances from the subset of selected utterances or successively add utterances from the data pool as long as the likelihood keeps increasing.

2.1. Optimization Criterion

Since training of HMMs involves missing data (the unknown state sequence), parameters cannot be estimated directly. Instead, an iterative method, the Expectation-Maximization (EM) framework [9] has to be employed. The optimization criterion is the auxiliary Q-function, which can be defined as follows:

$$Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}) = \sum_{\vec{s}} P(\vec{s}|\mathcal{D}, \boldsymbol{\Theta}) \log P(\vec{s}, \mathcal{D}|\hat{\boldsymbol{\Theta}})$$

The meanings of the symbols used in this and the following equations are:

- initial model parameters $\boldsymbol{\Theta}$
- development data $\mathcal{D} = \{\vec{x}\}$
- training data $\mathcal{T} = \{\vec{y}\}$
- state and mixture index sequence $\vec{s}$
- state index q
- mixture component index m
- parameters $\hat{\boldsymbol{\Theta}} = \{\hat{\mu}_{qm}, \hat{\sigma}_{qm}, \hat{w}_{qm}, \hat{a}_{qq'}\}$ which maximize or let increase Q
- mean $\hat{\mu}_{qm}$ of state q, mixture m
- variance $\hat{\sigma}_{qm}$ of state q, mixture m
- weight $\hat{w}_{qm}$ of state q, mixture m
- state occupation and transition probabilities $\hat{a}_{qq'}$
- speech frames x_t, y_t with time index t

For the derivation of the Forward-Backward (FB) algorithm, i.e. the conventional EM algorithm for training HMMs, $\mathcal{D}$ is the training data. Here, $\mathcal{D}$ denotes the development data, since the optimization criterion is modified to the model likelihood given the development data. However, the calculation of new model parameters $\hat{\boldsymbol{\Theta}}$ will be carried out as usual with the FB algorithm based on (a subset of) the training data $\mathcal{T}$. The Q-function has the property that the likelihood increases, i.e. $P(\mathcal{D}|\hat{\boldsymbol{\Theta}}) > P(\mathcal{D}|\boldsymbol{\Theta})$, if Q increases, i.e. $Q(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}) > Q(\boldsymbol{\Theta}, \boldsymbol{\Theta})$. For simplicity of notation, x_t and y_t are assumed to be one-dimensional and the HMM transition probabilities are neglected. Nevertheless, it is easy to define the equations for multivariate data and to include the transition probabilities.

The output density part of Q can be rewritten for Gaussian parameters $\hat{\mu}_{qm}, \hat{\sigma}_{qm}$ to be proportional to the expression

$$\propto \sum_q \sum_m \sum_t \gamma_{qm}(t) \left[\log \frac{\hat{w}_{qm}}{\sqrt{2\pi\hat{\sigma}_{qm}}} - \frac{1}{2}(x_t - \hat{\mu}_{qm})^2 \frac{1}{\hat{\sigma}_{qm}} \right]$$

where $\gamma_{qm}(t)$ is the state occupation probability of speech frame x_t for state q and mixture component m. This expression can be transformed to

$$\propto \sum_q \sum_m \left\{ y_{qm} \log \frac{\hat{w}_{qm}}{\sqrt{2\pi\hat{\sigma}_{qm}}} - \frac{z_{qm} - 2\hat{\mu}_{qm} o_{qm} + \hat{\mu}_{qm}^2 y_{qm}}{2\hat{\sigma}_{qm}} \right\}$$

$$\propto \sum_q \sum_m \left\{ y_{qm} \log \left[\frac{c_{qm}}{\sum_n c_{qn}} \right] - \frac{y_{qm}}{2} \log \left[\frac{\sigma_{qm} c_{qm} - \mu_{qm}^2}{c_{qm}^2} \right] \right\}$$
$$- \sum_q \sum_m \left\{ \frac{z_{qm} c_{qm}^2 - 2\mu_{qm} o_{qm} c_{qm} + \mu_{qm}^2 y_{qm}}{2\sigma_{qm} c_{qm} - 2\mu_{qm}^2} \right\}$$

The variables y_{qm}, o_{qm} and z_{qm} denote the sufficient statistics (SS) of the development data $\mathcal{D}$

$$\begin{aligned} y_{qm} &= \sum_t \gamma_{qm}(t) \\ o_{qm} &= \sum_t \gamma_{qm}(t) x_t \\ z_{qm} &= \sum_t \gamma_{qm}(t) x_t^2 \end{aligned}$$

and c_{qm}, μ_{qm} and σ_{qm} are the SS of the whole training data $\mathcal{T}$ which are decomposable w.r.t. the training utterances $\mathbf{u}_i$.

$$\begin{aligned} c_{qm} &= \sum_i c_{qm}^i &&= \sum_i \sum_t \gamma_{qm}^i(t) \\ \mu_{qm} &= \sum_i \mu_{qm}^i &&= \sum_i \sum_t \gamma_{qm}^i(t) y_t^i \\ \sigma_{qm} &= \sum_i \sigma_{qm}^i &&= \sum_i \sum_t \gamma_{qm}^i(t) (y_t^i)^2 \end{aligned}$$

New model parameters $\hat{\boldsymbol{\Theta}}$ can be calculated easily and fast from the SS $\mathbf{S}_i = (c_{qm}^i, \mu_{qm}^i, \sigma_{qm}^i)^T$ of any subset of training utterances $\{\mathbf{u}_i\}$. The formulas f for the reconstruction of the new model parameters $\hat{\boldsymbol{\Theta}} = f(\sum_i \mathbf{S}_i)$ are

$$\hat{w}_{qm} = \frac{\sum_i c^i_{qm}}{\sum_n \sum_i c^i_{qn}} = \frac{c_{qm}}{\sum_n c_{qn}}$$

$$\hat{\mu}_{qm} = \frac{\sum_i \mu^i_{qm}}{\sum_i c^i_{qm}} = \frac{\mu_{qm}}{c_{qm}}$$

$$\hat{\sigma}_{qm} = \frac{\sum_i \sigma^i_{qm}}{\sum_i c^i_{qm}} - \hat{\mu}^2_{qm} = \frac{\sigma_{qm}}{c_{qm}} - \frac{\mu^2_{qm}}{c^2_{qm}}$$

From this derivation of Q, which is expressible only with the SS of the training and the development data w.r.t. the initial model parameters $\boldsymbol{\Theta}$, the feasibility of selective training becomes clear. Selective training is accomplished by successively adding or substracting the SS of single training utterances, which means modifying c_{qm}, μ_{qm} and σ_{qm}, so that the Q-function increases. The detailed algorithm is explained in the next subsection.

2.2. Selective Training Algorithm

Figure 1 depicts the overall setup for selective training (ST). There are several possibilities to define a concrete ST algorithm based on the auxiliary Q-function. Here, we will consider two variants: The delete scan algorithm *ST_DelScan*, which considers every training utterance only once, and the *ST_DelAdd* algorithm which successively deletes or adds utterances for several iterations. The *ST_DelScan* variant works as follows:

1. Let R be the set of all (selected) training utterances.
2. Obtain $\{\mathbf{S}_i\}$, the SS of each training utterance $\mathbf{u}_i$.
3. Obtain $\mathbf{S}_\mathcal{D}$, the SS of the whole development data.
4. Obtain $\mathbf{S}_\mathcal{T}$, the SS of the whole training data.
5. Evaluate $q := Q(\boldsymbol{\Theta}, f(\mathbf{S}_\mathcal{T}))$.
6. For each utterance $\mathbf{u}_i \in R$ do:
   a. Evaluate $q' := Q(\boldsymbol{\Theta}, f(\mathbf{S}_\mathcal{T} - \mathbf{S}_i))$.
   b. If $q' > q$, then discard utterance $\mathbf{u}_i$: $R := R - \{\mathbf{u}_i\}$
7. Use $\hat{\boldsymbol{\Theta}} = f(\sum_{\mathbf{u}_i \in R} \mathbf{S}_i)$ as new model parameters.
8. Retrain with utterance set R for several iterations.

The idea is, that if the independent deletion of single training utterances leads to an increase of model likelihood, it should not be used for training. Consequently, the decision to discard one utterance is independent from the deletion of a previous or following utterance.

Instead of considering every utterance only once for deletion, step (6.) could also be carried out iteratively, while alternating between deleting (already) selected utterances or adding unselected utterances. This is realized in the *ST_DelAdd* variant of the algorithm. Step (6.) has to be modified as follows:

6. Repeat for a predefined number of iterations:
   I. For each $\mathbf{u}_i \in R$ do:
      a. Evaluate $q' := Q(\boldsymbol{\Theta}, f(\mathbf{S}_\mathcal{T} - \mathbf{S}_i))$.
      b. If $q' > q$, then discard utterance $\mathbf{u}_i$: $R := R - \{\mathbf{u}_i\}$, $\mathbf{S}_\mathcal{T} := \mathbf{S}_\mathcal{T} - \mathbf{S}_i$ and $q := q'$
   II. For each $\mathbf{u}_i \notin R$ do:
      a. Evaluate $q' := Q(\boldsymbol{\Theta}, f(\mathbf{S}_\mathcal{T} + \mathbf{S}_i))$.
      b. If $q' > q$, then remember utterance $\mathbf{u}_i$: $R := R \cup \{\mathbf{u}_i\}$, $\mathbf{S}_\mathcal{T} := \mathbf{S}_\mathcal{T} + \mathbf{S}_i$ and $q := q'$

A drawback of this approach is, that the decision to delete or add an utterances depends on the order of presenting training utterances to the algorithm. On the other hand, the value of the auxiliary Q-function can increase more than in case of the *ST_DelScan* variant. Figure 2 illustrates the processing of both variants of the selective training algorithm in detail.

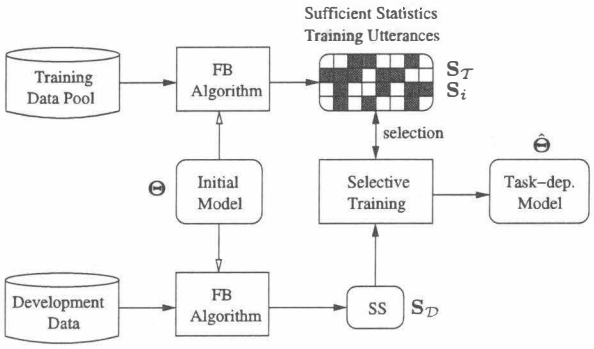

Fig. 1. Overall setup for selective training. After calculating the sufficient statistics for each training utterance in the data pool, and the sufficient statistics for the whole development data, the selective training procedure *ST_DelScan* or *ST_DelAdd* is carried out.

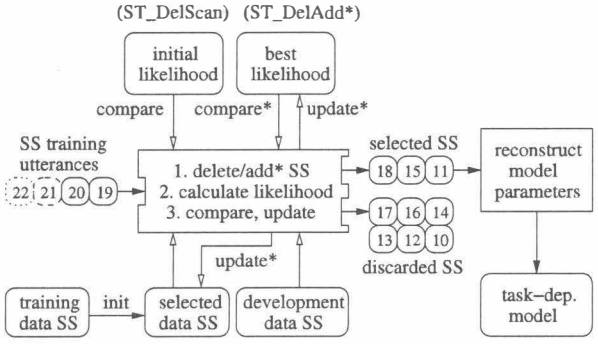

Fig. 2. Detailed illustration of the selective training procedure. Actions marked with (*) are only carried out by the *ST_DelAdd* variant. The sufficient statistics (SS) of each training utterance are processed one by one in series. An utterance is only selected for parameter estimation if the model likelihood increases. While *ST_DelScan* considers each utterance only once for deletion, *ST_DelAdd* examines each discarded utterance again for addition.

3. DATA

The two variants of the ST algorithm are evaluated with Japanese speech from the Takemaru database. Takemaru-kun [7] is a

Table 1. Speech data employed for evaluation: The purpose of experiment (A) is to obtain an infant-dependent AM only from speech data of elementary school children. In experiment (B), utterances from adult speakers are selected to better model the speech of elderly persons.

	Training Data Pool		Development Data		Test
Exp.	Group	# Utter.	Group	# Utter.	# Utter.
(A)	Element.	29,776	Infant	500	1,554
(B)	Adult	17,874	Elderly	53	400

speech-oriented dialogue system intended to provide the user information on the weather, news, the surrounding environment, public transportation system, Internet pages, a.s.o. The system is very popular among children, because it is based on an animated character. It is a working system installed in a public place in Nara, Japan. Speech data is collected automatically since November 2002 from users who speak to the system. Each recorded utterance is transcribed, labeled with tags (e.g. noise) and classified subjectively into one of five speaker groups: infants (preschool children), elementary school children, junior-high school children, adults and elderly persons. The selective training procedure is applied to obtain an acoustic model for the two groups for which only few data are available: infants and elderly persons. Table 1 gives details about the speech data employed for experiments.

4. EXPERIMENTAL CONDITIONS

The initial acoustic model is obtained from scratch by training with all utterances in the data pool. It consists of one 3-state HMM each for 35 phonemes and three silence models. Each HMM state has at most 16 Gaussian mixture densities with diagonal covariance matrices. The acoustic feature vector is 25-dimensional, including ΔE, 12 MFCC and 12 Δ MFCC. The sufficient statistics necessary for selective training are calculated with this initial model. The implementation of the ST algorithm is based on HTK source code. In order to prevent flooring of variances, a threshold of 200 is set for the minimum number of examples required per phoneme HMM. Step (6.) is repeated five times for the *ST_DelAdd* variant. Only those training utterances which were discarded during the previous delete step are examined for insertion again. The open-source LVCSR engine Julius [10] is used for decoding test utterances. None of the test utterances is part of the training or development data employed for selective training of the acoustic model. The perplexity of the test set (5,742 words) in experiment A is 8.3 for a closed task-dependent language model trained on transcriptions of infant utterances. For experiment B an open language model is trained on transcriptions of adult utterances. The perplexity of the corresponding test set (1,609 words) is 16.3.

5. RESULTS AND DISCUSSION

Table 2 shows the result for building an infant-dependent (A) and an elderly-dependent (B) acoustic model. Both variants of the ST algorithm are compared to training without selection. For experiment A, there is only an improvement of 1.3% absolute (2.8% relative) over the initial model when retraining the initial model with all utterances in the data pool. With selection, the performance increases up to 5.1% absolute (11.0% relative). Although most is gained by the first iteration, retraining with the selected set of utterances several times leads to further improvements in word accuracy.

Table 2. Comparison of performance (word accuracy in %) for the acoustic model obtained by conventional EM training without selection to the proposed selective training algorithm. The development and test data employed in experiment A consists of infants' utterances, in experiment B of utterances from elderly persons.

Experiment A	Training Iteration					
Algorithm	init	1	2	3	4	5
No Selection	46.4	46.7	47.3	**47.4**	47.3	47.3
ST_DelScan	46.4	50.4	50.6	50.8	**51.4**	51.3
ST_DelAdd	46.4	50.5	51.1	51.2	51.2	**51.5**
Experiment B	Training Iteration					
Algorithm	init	1	2	3	4	5
No Selection	72.3	72.2	72.0	72.4	**72.5**	72.1
ST_DelScan	72.3	74.4	**75.1**	74.7	74.6	74.5
ST_DelAdd	72.3	73.5	73.8	73.8	**74.1**	73.7

Table 3. Number and percentage of utterances chosen from the data pool by the proposed selective training algorithm.

Experiment A		Experiment B	
ST_DelScan	*ST_DelAdd*	*ST_DelScan*	*ST_DelAdd*
10,697 (36%)	4,299 (14%)	7,704 (43%)	3,165 (18%)

The same can be observed for building an elderly-dependent model in experiment B. An increase of up to 2.8% absolute (3.9% relative) in recognition accuracy by selective training versus almost no improvement without selection. Since the difference in performance between the two variants of the ST algorithm in both experiments is rather small, it is hard to say which variant should be preferred. Statistics about the number of utterances selected are given in Table 3. While the *ST_DelScan* variant employs more than one third of the utterances in the data pool, only about 10-20% are considered as important by *ST_DelAdd*.

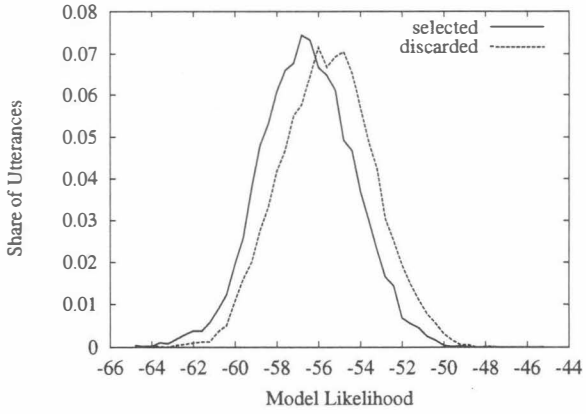

Fig. 3. Likelihood distribution of the initial model given the selected and the discarded training utterances (Experiment B, *ST_DelScan*).

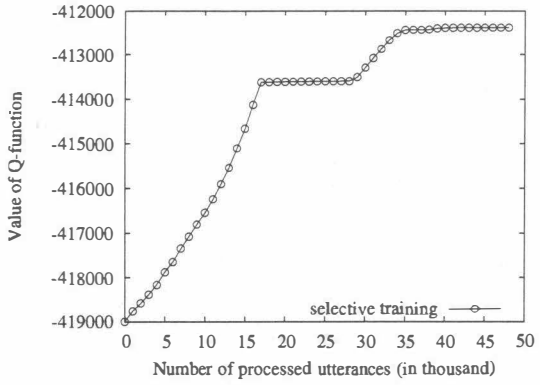

Fig. 4. Behavior of the Q-function (Experiment B, *ST_DelAdd*).

Table 4. Relationship between the number of utterances in the development set and the performance (word accuracy in %) of selective training. (Experiment A, *ST_DelScan*).

Devel. Set Size	5	10	20	50	100
ST (1st iter.)	49.1	49.3	50.4	50.4	50.5
ST (5th iter.)	49.6	50.3	50.2	50.8	51.2
# selected utter.	9,633	8,715	9,393	9,617	10,287

From Figure 3 it is clear, that rather utterances with a lower model likelihood are selected. Nevertheless, there is much overlap between the likelihood distributions of selected and discarded utterances, so that a simple selection rule such as "select all utterances with a likelihood below a threshold" would be far less effective than the proposed ST algorithm. The same tendency could be observed in both experiments.

How the value of the Q-function changes is depicted in Figure 4. The largest increase can be observed during deleting utterances in the first and second iteration. The number of discarded (-) and added (+) utterances during the five iterations was: -11,715, +203; -2,870, +44; -341, +21; -42, +1; -10, +0. Almost nothing is gained when adding previously deleted utterances again. Consequently, step (6.II.) of *ST_DelAdd* could also be omitted.

Table 4 shows the performance of selective training depending on the development set size. There is already an improvement with only five development utterances. Maximum possible performance seems to be reached with about 100 utterances. The number of selected utterances does not differ remarkably.

The results show that a better model can be obtained by selective training based on a small set of development data. Consequently, it is important to know, how many utterances of the target task must be collected in order to reach the same performance by conventional EM training. From Figure 5 it becomes clear, that 500 or 1,000 utterances are not enough to train a model of the given complexity (monophone, about 94,000 parameters) robustly. With 2,500 infant utterances, the same performance as with selective training can be reached. Here, it has to be mentioned, that recognition of infant speech is a very difficult task. The maximum recognition performance is almost reached: The word accuracy with a monophone model of the same complexity build from scratch with 7,500 infant utterances was 52%.

Finally, selective training is compared to adaptation with the

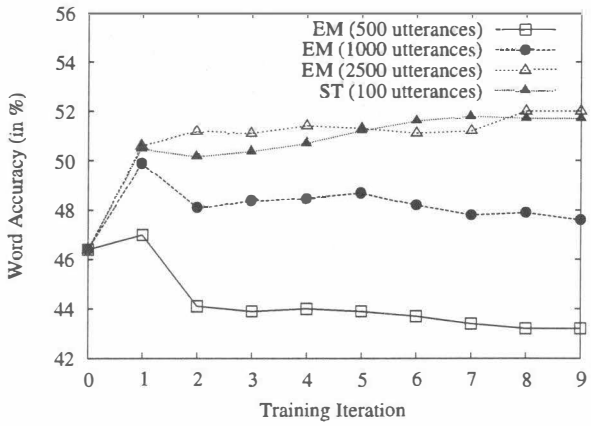

Fig. 5. Comparison of conventional EM training to selective training (Experiment A, *ST_DelScan*).

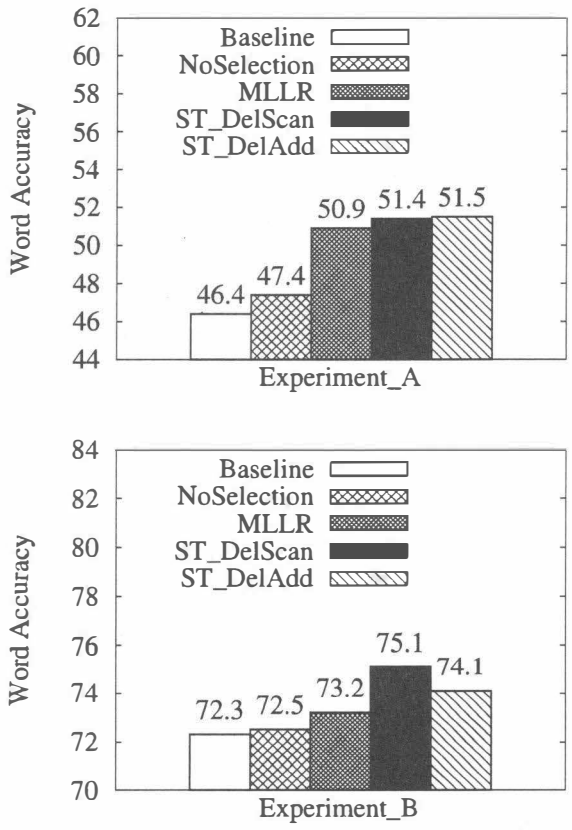

Fig. 6. Comparison of performance between the baseline model, training without selection, MLLR adaptation with the development data and the proposed approach for selective training.

development data. A standard adaptation method for relatively few adaptation data is Maximum Likelihood Linear Regression (MLLR) [11]. The performance of MLLR is obtained by considering the best result among evaluation for 2, 4, 8, 16 and 32 regression classes and two adaptation iterations. All utterances in the development set (500 for experiment A, 53 for experiment B) are used for adaptation. The result in Figure 6 shows, that selective training is superior to both MLLR adaptation and retraining without selection. The advantage of selective training over MLLR also is, that no parameter such as the number of regression classes has to be set, which influences the performance.

6. COMPUTATIONAL REQUIREMENTS

This section gives information about the computational requirements in (disk) space and (CPU) time for selective training. The time to extract the sufficient statistics (SS) for each training utterance is the same as for conventional EM training. Additional time is only needed for storing the SS and running the ST algorithm. Reconstruction of model parameters is possible within milliseconds. Rather than CPU time, physical disk space and data transfer rate are important issues. The size of the SS is proportional to the number of model parameters. Consequently, much more disk space is needed to store the SS in comparison to the feature vector sequence or the discrete time speech signal. Fortunately, an utterance usually contains only a small subset of all target language phonemes. This means that most SS are zero. Hence, a high compression ratio (e.g. 1:5 for experiment A) can be achieved for most utterances.

Table 5 shows the run time and disk space required for conducting experiments A and B. A state-of-the-art personal computer with a 3.2 GHz CPU was employed. The selective training procedure took only about 20 minutes for experiment A, and 27 minutes for experiment B. Most of the CPU time is used to evaluate the optimization criterion (Q-function). The disk space required to store the SS is 2.5 GB. Since the selection works utterance-based it is possible to reduce the additional space necessary to store the SS to zero at the cost of doubling computation time, if the *ST_DelScan* variant is used. However, the reduction of disk space is not a recommendable option for the *ST_DelAdd* variant.

Table 5. Time and space required for selective training.

Experiment	A, *ST_DelScan*	B, *ST_DelAdd*
Total Run Time	$\approx$ 20 minutes	$\approx$ 27 minutes
Total CPU Time	$\approx$ 10 minutes	$\approx$ 18 minutes
CPU Time Q-function	216 seconds	366 seconds
Model Size (ASCII)	1300 KB	1300 KB
Development data SS	368 KB	313 KB
Training data SS	400 KB	379 KB
Single utterance SS	78 KB	84 KB
Total disk space SS	2.5 GB	1.4 GB

It is clear, that building a task-dependent model with the proposed algorithm is feasible within a short period of time. Even if the model complexity and the size of the data pool increase, enough disk space can be provided easily and the additional computation time needed for utterance selection is only a fraction of the time necessary for one conventional EM training iteration.

7. SUMMARY AND FUTURE WORK

In this paper a new approach for selective training was introduced. It was shown, that it is possible to select relevant training utterances from a large data pool given only a small amount of task-dependent development data. The selection is based on a likelihood criterion, the auxiliary Q-function. The method was applied to obtain an infant-dependent and elderly-dependent acoustic model. There was a relative improvement over the baseline of up to 11.0%. The approach leads also to better results than MLLR adaptation with the development data. The additional time necessary for selective training is only a fraction of a conventional EM training iteration. As future work, selective training for acoustic models with a higher complexity (e.g. triphone) has to be examined. Moreover, the new algorithm will be evaluated for different speech databases and tasks, and it has to be investigated how the algorithm behaves for data selection across multiple databases.

8. REFERENCES

[1] Y. Gao, L. Gu, and H.-K. J. Kuo, "Portability Challenges in Developing Interactive Dialogue Systems," in *International Conference on Acoustics, Speech, and Signal Processing*, 2005, pp. 1017–1020.

[2] F. Lefevre, J.-L. Gauvain, and L. Lamel, "Genericity and Portability for Task-dependent Speech Recognition," *Computer Speech and Language*, vol. 19, pp. 345–363, 2005.

[3] L. M. Arslan and J. H. L. Hansen, "Selective Training in Hidden Markov Model Recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 1, pp. 46–54, 1999.

[4] T. M. Kamm and G. G. L. Meyer, "Robustness Aspects of Active Learning for Acoustic Modeling," in *Proceedings of the International Conference on Spoken Language Processing*, 2004, pp. 1095–1098.

[5] S. Yoshizawa, A. Baba, K. Matsunami, Y. Mera, M. Yamada, A. Lee, and K. Shikano, "Evaluation on Unsupervised Speaker Adaptation based on Sufficient HMM Statistics of Selected Speakers," in *European Conference on Speech Communication and Technology*, 2001, pp. 1219–1222.

[6] C. Huang, T. Chen, and E. Chang, "Transformation and Combination of Hidden Markov Models for Speaker Selection Training," in *Proceedings of the International Conference on Spoken Language Processing*, 2004, pp. 1001–1004.

[7] R. Nishimura, Y. Nishihara, R. Tsurumi, A. Lee, H. Saruwatari, and K. Shikano, "Takemaru-kun: Speech-oriented Information System for Real World Research Platform," in *International Workshop on Language Understanding and Agents for Real World Interaction*, 2003, pp. 70–78.

[8] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, Inc., 2001.

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J.R. Statistical Society*, vol. 1, no. 39, pp. 1–38, 1977.

[10] "Julius, an Open-Source Large Vocabulary CSR Engine - `http://julius.sourceforge.jp/`,".

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