Chapter 8

FAST ONLINE SPEAKER ADAPTATION FOR SMART ROOM APPLICATIONS

S.Kadambe
HRL Laboratories, LLC, USA
skadambe@hrl.com

Keywords: Smart room, remote monitoring, online speaker adaptation, stochastic matching, maximum likelihood

1. Introduction

In smart room applications such as remote monitoring of (a) meetings, (b) conference presentations and (c) progress of a trainee an automatic speech recognition (ASR) engine is used to extract excerpts from the speech of different people attending the meeting or conference or the training and to convert it to text so that it can be transmitted to the remote location. The performance of an ASR engine has to be robust to noise, to different room acoutics and to different speakers for it to meet the user's satisfaction. In this chapter, we are concentrating on making an ASR engine robust to different speakers which includes non-native speakers i.e., the speakers whose first language is different from the language for which an ASR engine is trained for.

In order to make an ASR engine robust for non-native speakers, the acoustic models it uses have to be adapted for that particular speaker. Generally, a separate adaptation training speech data is collected from different speakers and the acoustic models are adapted for that particular speaker using the training data associated with that person. However, this is not a practical solution since in a meeting or in a conference presentation scenarios, speakers can be dynamically changing as people will be coming in and out of a meeting or a presentation. This implies that anybody's speech excerpts could be extracted at any given point of time. This necessitates a need for on-line adaptation without using any adaptation training data - physically it is not possible to collect any training data in these scenarios. Some of the techniques developed hether to [1]- [7] can be modified for on-line adaptation with some degradation in performance. However, a practical solution is to adapt the models on-line without the requirement of adaptation training data with no or minimal degradation in performance. In this chapter, such a technique that uses modified maximum likelihood stochastic matching is described.

The adaptation conceptually corresponds to projecting the trained models to the testing environment. This kind of projection can be performed at signal, feature and model spaces as shown in Figure 8.1. Most of the adaptation

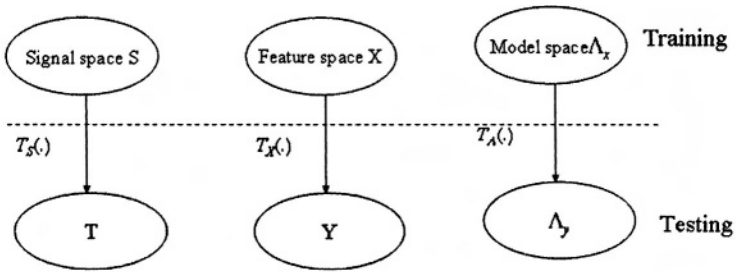

Figure 8.1. The conceptual diagram of adaptation

techniques developed so far perform the projection in the model space and are based on linear or piece-wise linear transformation. These techniques are computationally expensive, need separate adaptation training data and are not applicable for the derivatives of cepstral coefficients. The technique described in this chapter also applies the projection in the model space but is applicable for the derivatives of cepstral coefficients and does not require separate adaptation training data as most of the other adaptation techniques do and hence, is a very practical approach. To achieve near real time speech recognition, the adaptation process should not take much time. Therefore, for quick adaptation, the described technique adapts the acoustic models utterance by utterance and, chooses the models and the mixture components of the chosen model that need to be projected to match the testing environment. The main idea behind adapting the models utterance by utterance is to have the flexibility and ability to adapt to new speakers as they change dynamically. The organization of the chapter is as follows: section 2 describes the proposed adaptation technique; section 3 gives the implementation details with respect to a continuous speech recognizer; section 4 provides the experimental details and results and section 5 concludes and discusses the future research directions.

2. Description of the proposed on-line adaptation technique

If the bias (noise, speaker, etc.) in signal space is convolutive, the bias in feature space using Cepstral features will be additive: $y_t = x_t + b_t$ where b_t corresponds to bias or distortion. Let the distortion model λ_B be a single Gaussian density with a diagonal covariance matrix and is of the form: $p(b_t) = N(b_t : \mu_b, \sigma_b^2)$. Under these assumptions the structure of the parameter set λ_y

of the training model space remains the same as that of the training set λ_x. This implies that the means and variances of λ_y are derived by adding a bias:

$$\mu_y = \mu_x + \mu_b \tag{8.1}$$

$$\sigma_y^2 = \sigma_x^2 + \sigma_b^2 \tag{8.2}$$

where $(\mu_{\mathbf{x}}, \sigma_{\mathbf{x}}{}^2)$ corresponds to parameters of the training model λ_x and $(\mu_{\mathbf{y}}, \sigma_{\mathbf{y}}{}^2)$ to λ_y. These equations define the model transformation $G_\eta(.)$ with the parameters to estimate $\eta = (\mu_{\mathbf{b}}, \sigma_{\mathbf{b}}^2)$. The bias parameters estimation problem can be written as:

$$\begin{aligned} \eta' &= (\mu'_{\mathbf{b}}, \sigma_{\mathbf{b}}^{2'}) \\ &= \underset{\mu_{\mathbf{b}}, \sigma_{\mathbf{b}}^2}{\operatorname{argmax}} \sum_S \sum_C p(\mathbf{Y}, S, C | \eta, \Lambda_X) \end{aligned} \tag{8.3}$$

Here, S is a set of all possible selected models for a given set of trained models Λ_X and C is a set of all selected mixture components of chosen models. The auxiliary function in a maximum likelihood sense (ignoring the prior word probabilities of the language model) that is used in solving the above optimization problem iteratively with the Expectation Maximization (EM) algorithm is:

$$\begin{aligned} Q(\eta'|\eta) &= -\sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{m=1}^{M} \gamma_t(n,m) \sum_{i=1}^{D} \xi(m,n,i) \\ \textit{where} & \\ \xi(m,n,i) &= [\frac{1}{2} \log(\sigma_{n,m,i}^2 + \sigma_{b_i}^{2'} + \frac{(y_{t,i} - \mu'_{b_i} - \mu_{n,m,i})^2}{2(\sigma_{n,m,i}^2 + \sigma_{b_i}^{2'})})] \end{aligned} \tag{8.4}$$

Here T corresponds to total observation time, N corresponds to number of chosen acoustic models for adaptation (in our case this corresponds to a boundary model since the ASR engine that we use is based on segments and which are modeled as boundaries), M is the chosen number of mixture components for each chosen model and D is the total number of bias parameters that is estimated for each model. Taking above equation's derivative with respect to η' does not result in a closed form solution for $\sigma_{b_i}^{2'}$; however, in [1] the authors show an alternate approach. In this paper, we use that approach to derive the estimates of the bias parameters. Equations corresponding to estimated bias parameters are provided below.

$$\mu'_{b_i}(n) = \frac{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(n,m) E(b_{t,i} | y_{t,i}, model = n, c_t = m, \eta, \Lambda_x)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(n,m)} \tag{8.5}$$

$$\sigma_{b_i}^{2'}(n) = \tag{8.6}$$

$$\frac{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(n,m) E(b_{t,i}^2 | y_{t,i}, model=n, c_t=m, \eta, \Lambda_x)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(n,m)} - \mu_{b_i}'^2$$

Because our proposed approach uses a bias for the mean and variance parameters for each boundary model that is modeled by a GMM, the estimated bias parameters equation differs from the one derived in [1]. From the above two equations, it can be seen that We have one set of bias parameters for each boundary model n. Readers familiar with HMM's may think of these GMM's or *boundary models* as *states,* though in our case the boundary models (GMM's) are concatenated to segments, which represent the states in the FST.

The derivation of the conditional expectations E that are needed in the estimation ofbias parameters in the above two equations are derived by modifying the algorithm that is described in [2] for the additive noise model. Using this algorithm together with our Cepstral data, leads to a convolutive noise model, which is appropriate for the purposes ofchannel equilazation and speaker adaptation. The conditional expectations are defined as

$$E(b_{t,i}|y_{t,i}, model = n, c_t = m, \eta, \Lambda_X) = \qquad (8.7)$$
$$\mu_{b_i} + \frac{\sigma^2_{n,m,i}}{\sigma^2_{n,m,i} + \sigma^2_{b_i}}(y_{t,i} - \mu_{n,m,i} - \mu_{b_i})$$

$$E(b^2_{t,i}|y_{t,i}, model = n, c_t = m, \eta, \Lambda_X) = \qquad (8.8)$$
$$\frac{\sigma^2_{b_i}\sigma^2_{n,m,i}}{\sigma^2_{n,m,i}+\sigma^2_{b_i}} + \{E(b_{t,i}|y_{t,i}, model = n, c_t = m, \eta, \Lambda_X)\}^2$$

Every chosen boundary model and its associated chosen mixture components have its own mean and variance bias which are added according to equations 8.1 & 8.2. To account for the reliability or accuracy of the estimated bias parameters, the update equations 8.1 & 8.2 are modified as:

$$\mu_y = \mu_x + \alpha_\mu \mu_b \qquad (8.9)$$

$$\sigma^2_y = \sigma^2_x + \alpha_\sigma \sigma^2_b, \qquad (8.10)$$

where α_μ and α_σ can be interpreted either as a stepsize or a weight indicating the reliability or accuracy of the bias. A small stepsize means a slower convergence to the optimum solution and/or could indicate a low reliability of the biases μ_b or σ^2_b. The stepsize could be implemented in an adaptive sense. A new idea of reliability measurement has been discussed and implemented as described below.

For each boundary model, the number of mixture components accounted for in equations 8.5 & 8.6 are counted. The more data i.e., the more mixture components has been collected for a boundary, the higher the reliability of its bias. This leads to a model dependent $\alpha(n)$ which is kept same for both mean and variance update in this study; however, we plan to use different functions for mean and variance in our future study. This $\alpha(n)$ is given by:

$$\alpha(n) = \frac{\text{number of mix. comps. used in bias formula for model n}}{\text{sum of all mix. comps. used for all models}} \qquad (8.11)$$

Note that the equations 8.5 & 8.6 are in the form of weighted sum where the weight $\gamma_t(n,m)$ is the joint likelihood (or probability) of producing a symbol $\mathbf{Y}$ at time t, in model n, using the mixture m, given the model (trained) parameters Λ. That is:

$$\gamma_t(n,m) = p(\mathbf{Y}, model = n, c_t = m|\eta, \Lambda_X). \tag{8.12}$$

The calculation of this depends on the framework in which an ASR engine is implemented. The ASR engine that we consider in our study uses the Finite State Transducer (FST) framework. A description of how $\gamma_t(n,m)$ can be implemented in this frame work is provided in the next section.

Note that the technique described here is a modified version of the technique described in [1]. The modifications are: (a) in the bias parameters estimation (eqs. 8.5 & 8.6) – since a chosen set of acoustic models are adapted, these parameters are estimated for each of the chosen models and for a chosen set of mixture components, (b) in the computation of the conditional expectation (eqs. 8.7 & 8.8) and (c) in the update equations (eqs. 8.9 & 8.10) – we use the weighted update equations and the weight determines the accuracy or reliability of the estimated bias parameters.

The overall block diagram of the proposed technique is as shown in Figure 8.2. In short, the proposed on-line, fast speaker/environment adaptation corre-

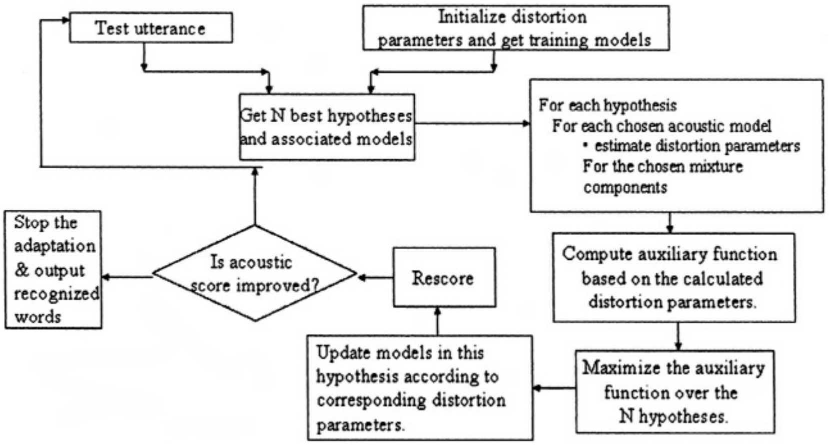
Test utterance
Initialize distortion parameters and get training models
Get N best hypotheses and associated models
For each hypothesis
For each chosen acoustic model
• estimate distortion parameters
For the chosen mixture components
Stop the adaptation & output recognized words
Is acoustic score improved?
Rescore
Compute auxiliary function based on the calculated distortion parameters.
Update models in this hypothesis according to corresponding distortion parameters.
Maximize the auxiliary function over the N hypotheses.

Figure 8.2. The block diagram of the proposed adaptation technique

sponds to estimating η' such that the acoustic likelihood score increases. The iterative procedure is stopped when the acoustic score of the current iteration is same as the previous iteration. The update equations (8.9 & 8.10) or the transformation $G_\eta(.)$ is applied only to the chosen models and their associated chosen mixture components. The models are chosen based on low confidence by using confidence measures where as the mixture components are chosen based on a threshold value.

The reason for choosing acoustic models with low confidence is as follows. The acoustic models with low confidence imply that the trained acoustic models did not match well with the test data. This further indicates that they are the outliers which need to be adapted so that they match better with the test data. Therefore, it makes sense to choose those models with low confidence to adapt.

The threshold value is kept constant for all different speakers. The threshold value is selected empirically which is described in section 4. In that section, the chosen value is also provided.

Note that this type of selection of acoustic models and the mixture components is not done in [1].

For the evaluation and verification of the proposed technique, it is implemented as an additional module in MIT's Summit speech recognizer and was tested using a set of non-native speakers' speech data. Note that even though this has been implemented as part of MIT's speech recognition engine, it can be implemented as part of any other speech recognition engine. Some of the details of (a) MIT's speech recognition engine and (b) the implementation of our proposed approach are provided in the following section.

3. Implementation details of proposed approach

In brief, SUMMIT is part of a spoken dialogue based system that is developed by MIT's Spoken Language Processing (SLS) group [8]. This system from lowest to highest level of processing consists of the SUMMIT, SAPPHIRE and GALAXY systems. The proposed speaker adaptation module has been implemented by making changes to SUMMIT and SAPPHIRE code.

The MIT SUMMIT System is a segment-based probabilistic speech recognition system. Using the Mel-frequency representation of a given signal, acoustic landmarks of varying robustness are located and embedded in an acoustic network. The sequence of landmarks/boundaries represents time. The segments in the network are then mapped to phoneme hypotheses. The result is a phonetic network, in which each arc is characterized by a vector of probabilities for all the possible candidates. Each segment can have one or more "landmarks" or 'boundaries' each of which is represented by a boundary model (GMM). During recognition, the phonetic network is matched to a pre-compiled pronunciation network to determine the best word hypothesis. An N-gram language model is used with the Viterbi search algorithm. SUMMIT uses an FST framework where each speech segment is represented by a state. The Viterbi training is used, resulting in a hypothesis based N-best scoring for recognition.

As mentioned in the previous section the implementation of $\gamma_t(n, m)$ (equation 8.12) depends on the ASR engine. For SUMMIT it can be implemented as follows.

3.1 Calculation of γ in an FST framework

The FST framework allows representation of information sources and data structures used in recognition which includes context-dependent units, pronunciation dictionaries, language models and lattices within a uniform structure.

In particular, a single composition algorithm is used to combine both information sources such as language models and dictionaries in advance and, acoustic observations and information sources dynamically during recognition. During speech preprocessing each utterance is split in to a number of landmarks or boundaries which delineate main changes in speech features. A boundary vector is composed of several adjacent frame feature vectors and is modeled by a boundary model – GMM. In SUMMIT currently about 1500 different boundary models are used; 92 of which are non-transitional. During recognition, SUMMIT groups the boundaries to segments/phones by making use of language model and pronunciation dictionary restrictions which have been set in advance. Up to 200 hypotheses are calculated using an FST model where each state corresponds to a segment/phone. The output consists of N best hypotheses. Each hypothesis can have a different number of segments, different segments or phones, segments differing in lengths, etc. In the current version of SUMMIT, transition probabilities are not used. All phone to phone and boundary to boundary transitions are assumed equally likely and have been assigned a log-based score of zero.

To compute $\gamma_t(n, m)$, the probability of the mth mixture component of model n producing the observation $\mathbf{o}$ at time t, given the past and the future path through the models, a probability $P(\mathbf{o}_t | model = n, mixt = m, \lambda)$ can be defined as

$$P_t(n,m) = \frac{w_{n,m}\mathcal{N}(\mathbf{o}_t; \mu_{n,m}, C_{n,m})}{\sum_{j=1}^{M} w_{n,j}\mathcal{N}(\mathbf{o}_t; \mu_{n,j}, C_{n,j})}, \tag{8.13}$$

where $w_{n,m}$ is the weight of the mixture component m of model n. With this probability and the fact that all boundary transitions are equally likely, we can write:

$$\gamma_t(n,m) = \underbrace{\prod_{ti=1}^{t-1} \left(\sum_{k=1}^{M_{s(ti)}} P_{ti}(s(ti),k) \right)}_{\alpha} \cdot P_t(n,m) \cdot \underbrace{\prod_{ti=t+1}^{T} \left(\sum_{k=1}^{M_{s(ti)}} P_{ti}(s(ti),k) \right)}_{\beta}, \tag{8.14}$$

where $M_{s(ti)}$ is the number of mixture components of the hypothesized model at time ti and T is the total number of boundaries.

The first product term in the above equation corresponds to forward probability α and the second product term corresponds to backward probability β. These two probabilities are computed as follows: At each boundary there is a model for each of the N-best hypotheses. The same model may be selected by more than one hypothesis. For the N-best hypotheses and series of landmarks there is a set of unique models for the utterance. In the adaptation only this set of models are considered. At each time boundary or landmark, forward and backward probabilities are obtained by calculating the probability to be in a model m at time landmark t by summing the mixture log probability score along each hypothesis to the boundary t and the associated partial score with the model m for that hypothesis. The log probabilities were then re-scaled (to prevent machine overflow) and then converted to a relative probability by exponentiation. These numbers were then scaled to add to 1.0 when summed over the N-best hypotheses (i.e. converted to probabilities.). Finally if a model occurred for more than one hypothesis, the results were combined so that it occurred only once at t.

4. Experimental details and results

After implementing the proposed fast on-line adaptation technique as part of SUMMIT as an additional module called "Adaptation", it was tested using non-native speakers' data.

Basically, the adaptation is performed with an inner loop adjusting the models and an outer loop to re-process the same utterance several times as shown in Figure 8.2 to make sure there is convergence. After the EM calculation, new scores are calculated by running the recognizer SUMMIT as shown in Figure 8.2 with the updated models. If the update increases the total probability score, the models are updated again. If not, the previous set of models (the best) are restored.

First, a threshold value was set empirically for the selection of number of components of a Gaussian mixture associated with a chosen acoustic model to be adapted. This threshold value corresponds to the coefficient or the weight or the probability associated with each mixture component. This weight in SUMMIT is in the form of 10^{-d}. For different values of d, an experiment of adaptation was conducted by considering N-best hypotheses of ASR output where N was set to 2, 3, 5 and 10. For each d value, word recoginition accuracy (WA) was computed for twenty seven utterances. In Table 8.1, the threshold value, the number of hypotheses that were used in the adaptation and WA are tabulated. From this table, it can be seen that for all threshold values, there is an improvement in WA as compared to no adaptation (the first row in the Table 8.1); however, the maximum improvement in WA is for the threshold value $d = 6$ and for $N = 2$. Hence, these value were selected and was kept the same for all the speakers for whose speech utterances the proposed algorithm was tested. Next, the adaptation experiment was conducted by considering ten non-native

Table 8.1. Example of threshold selection

threh	nbest	WA
0	-	87.5
10^{-3}	2	88.4
10^{-4}	2	89.2
10^{-5}	2	90.1
10^{-6}	2	90.9
10^{-7}	2	89.2
10^{-4}	3	88.4
10^{-5}	3	88.8
10^{-6}	3	88.8
10^{-7}	3	88.8

speakers and twenty four utterances from each speaker. The acoustic models and the number of mixture components associated with each model to adapt were selected using the confidence scores and the threshold value, respectively, as mentioned before. The adaptation procedure as described in Figure 8.2 was applied to each utterance by discarding all the models that were adapted for the previous utterance. Such an adaptation procedure was applied since the next utterance could be from a different speaker especially, in the scenarios considered in this study. If the scenario is a dialogue between one speaker and an ASR engine, it makes sense not to restart the adaptation process for each utterance. However, we are interested in making an ASR engine robust to anybody at any time and any where. Hence, we are adapting utterance by utterance.

Since the EM iteration seems to converge in at most two trials the adaptation can be achieved in close to real-time. In Table 8.2, the recognition time for different utterances is provided with and without adaptation. From this, it can be

Table 8.2. **Recognition time of an utterance with (w) and without (w/o) adaptation**

Utterance length in secs	Recog. time w/o adaptation in secs	Recog. time w adaptation in secs
1.42	0.422	0.65
3.0	0.838	0.975
1.39	0.414	0.62
3.64	0.977	1.046
1.8	0.637	0.844
7.1	2.344	2.808
3.5	0.955	1.028
3.7	1.038	1.207
1.52	0.441	0. 667
2.76	0.933	1. 092

seen that the recognition time with adaptation is not significantly much higher than the recognition time without adaptation. For the selection of mixture components, a thresholding technique was used as mentioned before. This threshold value was selected empirically and then kept constant for all speakers. From the above discussion it can be seen that the selected threshold value is $d = 6$ and $N = 2$. The experimental results of Table 8.3 indicate that this empirically chosen threshold did not get affected by the speaker variability. Note that no separate adaptation training data was used. The recognition accuracies that we obtained for twenty four utterances from each of 10 non-native speakers are tabulated in Table 8.3. The total number of words in twenty four utterances varied from speaker to speaker. Hence, this is also tabulated in this table. From this table it can be seen that an average of 7.4 % improvement in word accuracy can be obtained across ten non-native speakers. Note that in systems that are used for remote monitoring, the N-best hypotheses output of an ASR engine is further processed by applying natural language processing techniques to reduce the error in extracting the excerpts from speech in the form of text. The recognition accuracy improvement of an ASR engine by 4 to 7 % will

Table 8.3. Word recognition accuracy (WRA) for 24 utterances

Speaker	WRA w/o adaptation in %	WRA w. adaptation in %	Relative Improvement in %	# of words
1	87.5	90.9	3.9	232
2	61.5	65.6	6.7	195
3	87	90	3.3	192
4	77.4	84	7.9	186
5	72.4	77.2	6.0	225
6	74.9	80.4	6.8	200
7	65.4	69.2	5.5	285
8	66.7	74.6	10.6	246
9	75.2	87.7	14.25	488
10	63.6	69. 9	9.01	855

improve the overall performance of the system by 10 to 15 % [8] which would definitely helps in improving the user's satisfaction.

5. Conclusions

The results indicate that the proposed on-line fast adaptation technique adapts the required models fast and improves the recognition accuracy significantly. The main advantages of this technique are (a) it does not need separate adaptation training data which is impractical to obtain since the ASR systems can be used by anybody, anytime and anywhere especially, in the smart room applications and (b) it performs adaptation in near real-time since it adapts only a small set of models that need to be adapted for any given utterance and the EM algorithm converges within two iterations. Future work warrants the usage of mixture Gaussian components for the distortion model to further improve the recognition accuracy.

References

[1] A. Sankar and C-H. Lee, "A maximum likelihood approach to stochastic matching for robust speech recognition," IEEE Trans. On Speech and Audio Processing, 4:190-202, May 1996.

[2] R. C. Rose, E. M. Hoftsetter and D. A. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," IEEE transactions on Speech and Audio processing, 2(2):245-257, April 1994.

[3] Hui Jiang and Li Deng, "A robust compensation strategy for extraneous acoustic variations in spontaneous speech recognition," IEEE Transactions on Speech and Audio Processing, 7(1):9 -17, Jan. 2002.

[4] J. McDonough, T. Schaaf, and A. Waibel, "On maximum mutual information speaker-adapted training", ICAASP 2002, 1:601-604, 2002.

[5] Bowen Zhou and J. Hansen, "Rapid speaker adaptation using multi-stream structural maximum likelihood eigenspace mapping," ICASSP

2002, 4:4166-4169, 2002.

[6] J.-T. Chien, “Online unsupervised learning of hidden Markov models for adaptive speech recognition,” IEE Proceedings on Vision, Image and Signal Processing, 148(5): 315 -324, October 2001.

[7] Shaojun Wang and Yunxin Zhao, “Online Bayesian treestructured transformation of HMMs with optimal model selection for speaker adaptation,” IEEE Transactions on Speech and Audio Processing, 9(6): September 2001.

[8] V. Zue, et al, “JUPITER: A telephone-based conversational interface for weather information,” IEEE trans. On speech and audio processing, 8(1): 85-96, January 2000.