FAST SPEAKER ADAPTATION IN AUTOMATIC ONLINE SUBTITLING

Aleš Pražák
SpeechTech s.r.o., Morseova 5, 301 00 Plzeň, Czech Republic
Ales.Prazak@speechtech.cz

Z. Zajíc, L. Machlica, J. V. Psutka
Department of Cybernetics, University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
zzajic@kky.zcu.cz, machlica@kky.zcu.cz, psutka_j@kky.zcu.cz

Keywords: ASR, online subtitling, speaker adaptation, fMLLR, MAP.

Abstract: This paper deals with speaker adaptation techniques well suited for the task of online subtitling. Two methods are briefly discussed, namely MAP adaptation and fMLLR. The main emphasis is laid on the description of improvements involved in the process of adaptation subject to the time requirements. Since the adaptation data are gathered continuously, simple modifications of the accumulated statistics have to be carried out in order to make the adaptation more accurate. Another proposed improvement efficiently employs the combination of fMLLR and MAP. In the case of online adaptation no prior transcriptions of the data are available. They are handled by a recognition system, thus it is suitable to assign a well-applied confidence measure to each of the transcriptions. We have performed experiments focused on the trade-off between the adaptation speed and the amount of adaptation data. We were able to gain a relative reduction of WER 16.2 %.

1 INTRODUCTION

The automatic online subtitling (closed captioning) of live TV programs using automatic speech recognition (ASR) is a very promising way, mainly for its considerable cost reduction. Several years ago, BBC introduced so-called "assisted subtitling" (Evans, 2003). This was intended for the production of well-timed subtitles for TV programs using the program transcript and the recording, based on alignment using Speaker Independent (SI) speech recognition. Another introduced approach, now in common use, employs a Speaker Dependent (SD) recognition of so-called "shadow speaker" who re-speaks the original speech of TV program.

In these days, with support of advanced acoustic modeling techniques and powerful computer technology, we can use online speaker independent large vocabulary continuous speech recognition of original audio stream for direct subtitling of some TV programs. This fully automatic approach comes into question for the speech-only TV programs with a noiseless background, where high recognition accuracy is reached. However, some speaker adaptation techniques with suitable fast online speaker change detection can further increase the accuracy of generated subtitles.

This paper brings an overview of online adaptation techniques with some modifications and improvement suggestions for a discussion. Our implementation of online speaker adaptation in the task of automatic online subtitling is presented and some adaptation strategies are discussed. Some experimental results are presented too.

2 ADAPTATION TECHNIQUES

The difference between the adaptation and ordinary training methods stands in the prior knowledge about the distribution of model parameters, usually derived from the SI model. The adaptation adjusts the model in order to maximize the probability of adaptation data. Hence, the new, adapted parameters can be chosen as

$$\boldsymbol{\lambda}^* = \arg\max_{\boldsymbol{\lambda}} p(\boldsymbol{O}|\boldsymbol{\lambda})p(\boldsymbol{\lambda}), \qquad (1)$$

where $p(\boldsymbol{\lambda})$ stands for the prior information about the distribution of the vector $\boldsymbol{\lambda}$ containing model parameters, $\boldsymbol{O} = \{\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T\}$ is the sequence of T fea-

ture vectors related to one speaker, $\boldsymbol{\lambda}^*$ is the best estimation of parameters of the SD model. We will focus on HMMs with output probabilities of states represented by GMMs. GMM of the j-th state is characterized by a set $\boldsymbol{\lambda}_j = \{\omega_{jm}, \boldsymbol{\mu}_{jm}, \boldsymbol{C}_{jm}\}_{m=1}^{M_j}$, where M_j is the number of mixtures, ω_{jm}, $\boldsymbol{\mu}_{jm}$ and $\boldsymbol{C}_{jm}$ are weight, mean and variance of the m-th mixture, respectively.

The adaptation techniques do not access the data directly, but only through some statistics, which are accumulated beforehand. Let us define these statistics:

$$\gamma_{jm}(t) = \frac{\omega_{jm} p(\boldsymbol{o}(t)|jm)}{\sum_{m=1}^{M} \omega_{jm} p(\boldsymbol{o}(t)|jm)} \tag{2}$$

stands for the m-th mixtures' posterior of the j-th state of the HMM,

$$\varepsilon_{jm}(\boldsymbol{o}) = \frac{\sum_{t=1}^{T} \gamma_{jm}(t)\boldsymbol{o}(t)}{\sum_{t=1}^{T} \gamma_{jm}(t)}, \quad c_{jm} = \sum_{t=1}^{T} \gamma_{jm}(t), \tag{3}$$

where c_{jm} is the soft count of mixture m and $\varepsilon_{jm}(\boldsymbol{o})$ represents the average of features that are assigned to mixture m in the j-th state of the HMM. It is necessary to accumulate also the statistic $\varepsilon_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})$, which can be computed in analogy with (3). Note that $\boldsymbol{\sigma}_{jm}^2 = \mathrm{diag}(\boldsymbol{C}_{jm})$ is the diagonal of the covariance matrix $\boldsymbol{C}_{jm}$.

2.1 Maximum A-posteriori Probability (MAP) Adaptation

MAP is based on the Bayes method for estimation of the acoustic model parameters, with the unit loss function (Gauvain and Lee, 1994). MAP adapts each of the parameters separately; therefore it is necessary to have for all the parameters enough adaptation data. Otherwise, the result of adaptation would be negligible. The balance between old and new parameters is determined using a user-defined parameter.

2.2 Linear Transformations based on Maximum Likelihood

The advantage over the MAP technique is that the number of available model parameters is reduced via clustering of similar model components (Gales, 1996). A well suited clustering method used for this purpose is based on Regression Trees (RTs), where each of the leaves in the tree contains a set of mixture components (e.g. mixture means). The leaves are merged (exploiting a criterion) until the final root node so that a RT is formed. The transformation matrices are estimated only for nodes with sufficient amount of data. Hence, their occupations have to be

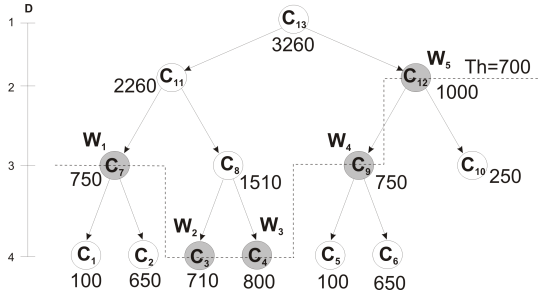

Figure 1: Example of a binary regression tree. The numbers assigned to nodes/clusters are their occupation counts. Nodes C_1 and C_2 have occupations lesser then the threshold $Th = 700$, therefore the mixture components located in C_1 and C_2 will be transformed utilizing transformations computed for node C_7. D denotes the depth in the tree.

greater than an empirically determined threshold Th. Note that the occupation of the n-th node can be computed according to $occ(n) = \sum_{m \in K_n} c_m$, where c_m was specified in (3) and K_n represents the content of the n-th cluster. As the same transformation is used for all parameters from the same cluster $K_n, n = 1, \ldots, N$, less amount of adaptation data is needed. An example of a RT is depicted in Figure 1.

For the task of online recognition the feature transformation is preferable because of implementation reasons (Machlica et al., 2009).

2.2.1 Feature Maximum Likelihood Linear Regression (fMLLR)

The method is based upon feature transformation:

$$\bar{\boldsymbol{o}}_t = \boldsymbol{A}_{(n)}\boldsymbol{o}_t + \boldsymbol{b}_{(n)} = \boldsymbol{W}_{(n)}\boldsymbol{\xi}(t), \tag{4}$$

where $\boldsymbol{W}_{(n)} = [\boldsymbol{A}_{(n)}, \boldsymbol{b}_{(n)}]$ stands for the transformation matrix corresponding to the n-th cluster K_n and $\boldsymbol{\xi}(t) = [\boldsymbol{o}_t^{\mathrm{T}}, 1]^{\mathrm{T}}$ represents the extended feature vector. The objective function that has to be maximized (Povey and Saon, 2006) has the form

$$\log|\boldsymbol{A}_{(n)}| - \sum_{i=1}^{I} \boldsymbol{w}_{(n)i}^{\mathrm{T}} \boldsymbol{k}_i - 0.5 \boldsymbol{w}_{(n)i}^{\mathrm{T}} \boldsymbol{G}_{(n)i} \boldsymbol{w}_{(n)i}, \tag{5}$$

where the column vector $\boldsymbol{w}_{(n)i}$ equals the transpose of the i-th row of $\boldsymbol{W}_{(n)}$,

$$\boldsymbol{k}_{(n)i} = \sum_{m \in K_n} \frac{c_m \mu_{mi} \varepsilon_m(\boldsymbol{\xi})}{\sigma_{mi}^2}, \; \boldsymbol{G}_{(n)i} = \sum_{m \in K_n} \frac{c_m \varepsilon_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}})}{\sigma_{mi}^2} \tag{6}$$

and

$$\varepsilon_m(\boldsymbol{\xi}) = \left[\varepsilon_m^{\mathrm{T}}(\boldsymbol{o}), 1\right]^{\mathrm{T}}, \; \varepsilon_m(\boldsymbol{\xi}\boldsymbol{\xi}^{\mathrm{T}}) = \begin{bmatrix} \varepsilon_m(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) & \varepsilon_m(\boldsymbol{o}) \\ \varepsilon_m^{\mathrm{T}}(\boldsymbol{o}) & 1 \end{bmatrix}. \tag{7}$$

The solution can be expressed as:

$$\boldsymbol{w}_{(n)i} = \boldsymbol{G}_{(n)i}^{-1} \left(\frac{\boldsymbol{v}_{(n)i}}{\alpha_{(n)}} + \boldsymbol{k}_{(n)i} \right), \tag{8}$$

where $\alpha_{(n)} = \boldsymbol{w}_{(n)i}^{\mathrm{T}} \boldsymbol{v}_{(n)i}$ and $\boldsymbol{v}_{(n)i}$ stands for the transpose of the i-th row of cofactors of the matrix $\boldsymbol{A}_{(n)}$ extended with a zero in the last dimension. Note that $\boldsymbol{w}_{(n)i}$ has to be computed iteratively, thus matrices $\boldsymbol{A}_{(n)}$ and $\boldsymbol{b}_{(n)}$ have to be correctly initialized first. For further details see (Gales, 1997).

2.2.2 Incremental Approach to fMLLR

In the online recognition, the adaptation has to be performed iteratively whenever the amount of adaption data reaches the pre-specified level. Hence, the subsequent recognition becomes more accurate. Recall that fMLLR operates with *model dependent statistics* (2) and (3), which are in the case of incremental adaptation accumulated continuously along with incoming data. Thus, statistics happen to be inconsistent between distinct iterations.

There are two possibilities, either accumulate all the data in the original space, or after each adaptation (iteration) transform the previous statistics into the new space formed by the new transformation matrices $\boldsymbol{A}_{(n)}^{k+1}$, $\boldsymbol{b}_{(n)}^{k+1}$. In the latter case, there is no need to transform directly the statistics $\varepsilon_{jm}(\boldsymbol{o})$, $\varepsilon_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})$. Instead, the matrices in (6) can be transformed, what significantly reduces the number of multiplications (number of mixtures in the HMM vs. number of clusters). The transformation formulas are (Li et al., 2002):

$$\boldsymbol{k}_{(n)i}^{k+1} = \hat{\boldsymbol{W}} \boldsymbol{k}_{(n)i}^{k}, \;\; \boldsymbol{G}_{(n)i}^{k+1} = \hat{\boldsymbol{W}} \boldsymbol{G}_{(n)i}^{k} \hat{\boldsymbol{W}}^{\mathrm{T}}, \qquad (9)$$

where

$$\hat{\boldsymbol{W}} = \begin{bmatrix} \hat{\boldsymbol{A}}_{(n)} & \hat{\boldsymbol{b}}_{(n)} \\ 0 & 1 \end{bmatrix}, \qquad (10)$$

and $\hat{\boldsymbol{A}}_{(n)}$, $\hat{\boldsymbol{b}}_{(n)}$ denote the newly computed transformation matrices. The final transformation matrices $\boldsymbol{A}_{(n)}^{k+1}$, $\boldsymbol{b}_{(n)}^{k+1}$ are given as (assuming no change in clusters K_n, $n = 1, \ldots, N$):

$$\boldsymbol{A}_{(n)}^{k+1} = \hat{\boldsymbol{A}}_{(n)} \boldsymbol{A}_{(n)}^{k}, \;\; \boldsymbol{b}_{(n)}^{k+1} = \hat{\boldsymbol{A}}_{(n)} \boldsymbol{b}_{(n)}^{k} + \hat{\boldsymbol{b}}_{(n)}. \qquad (11)$$

It is appropriate to wait until the increase of information is sufficient so that the newly formed transformation matrices are well-conditioned and the new iteration reasonably improves the recognition (Machlica et al., 2009).

2.3 Combination of MAP and fMLLR

As MAP and fMLLR work in different ways, it would be suitable to combine them. A simple method would be to run MAP and fMLLR subsequently in two passes, where fMLLR should be followed by MAP. The fMLLR method transforms all the mixtures from the same cluster at once, thus also mixtures with insufficient amount of adaptation data. The second pass (MAP adaptation) can be thought of as a refinement stage of mixtures with sufficient amount of data (MAP affects each of the mixtures separately – more precise). The suitability of MAP after fMLLR combination was also proved by experiments (Zajíc et al., 2009).

The main disadvantage of such an approach is the need to compute the statistics defined in Section 2 twice. Hence, it would be suitable to adjust the statistics accumulated in the first pass (fMLLR) without the need to see the feature vectors once again. Because fMLLR is in use, the adaptation stands for the transformation of feature vectors. Thus, instead of accumulating new statistics of transformed features, it is possible to transform the already computed statistics $\varepsilon_{jm}(\boldsymbol{o})$, $\varepsilon_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})$ in the following way (consider expressions (3) and (4)):

$$\begin{aligned} \bar{\varepsilon}_{jm}(\boldsymbol{o}) &= \boldsymbol{A}_{(n)} \varepsilon_{jm}(\boldsymbol{o}) + \boldsymbol{b}_{(n)}, \\ \bar{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) &= \boldsymbol{A}_{(n)} \varepsilon_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}}) \boldsymbol{A}_{(n)}^{\mathrm{T}} + \\ &\quad + 2\boldsymbol{A}_{(n)} \varepsilon_{jm}(\boldsymbol{o}) \boldsymbol{b}_{(n)}^{\mathrm{T}} + \boldsymbol{b}_{(n)} \boldsymbol{b}_{(n)}^{\mathrm{T}}, \end{aligned} \qquad (12)$$

and perform the MAP adaptation utilizing $\bar{\varepsilon}_{jm}(\boldsymbol{o})$, $\bar{\varepsilon}_{jm}(\boldsymbol{o}\boldsymbol{o}^{\mathrm{T}})$. Note that the only approximation consists in the use of untransformed mixture posteriors defined in (2).

2.4 Adaptation of Silence

When using regression trees (RTs) in fMLLR, the speech and silence segments usually share the same cluster. In cases where only adaptation data containing small amount of silence frames are available, a situation may occur that the states of silence, presented in the HMM, are bended toward the speech data. Hence, the silence segments can be more often recognized as speech, mainly when channel of adaptation data significantly varies. Generally, the speech and silence are so much different that the idea to separate them is straightforward. Therefore, it is suitable to establish a special node in RT intended only for states of silence. It should be mentioned that the adaptation has to be performed only when sufficient amount of data is available for both silence and speech parameters.

2.5 Model Normalizations

Because the data used to train the SI model come from a large set of speakers, the SI model is strongly influenced by speaker variations. Therefore, it would be wise to suppress/remove such variations before the

adaptation process itself. Method used for this purpose is called Speaker Adaptive Training (SAT) and is based on Linear Transformations (e.g. fMLLR). The details can be found in (Gales, 1997).

3 ONLINE ADAPTATION IMPLEMENTATION

Described adaptation techniques require an assignment of feature vectors to the HMM states and mixtures. This assignment is usually acquired using the force alignment of adaptation utterances to the HMM states and mixtures based on manual word transcriptions (supervised approach). In case of online adaptation, where no manual transcriptions are available, recognized word sequence is used instead (unsupervised approach). Since the recognition process is not error-free, some technique for confidence tagging of recognized words should be used to choose only well-recognized segments of speech for speaker adaption.

3.1 Confidence Measure

To apply the online speaker adaption as soon as possible, word confidences have to be evaluated very fast for partial word sequences generated periodically along with incoming acoustic signal. We use posterior word probabilities computed on the word graph as a confidence measure (Wessel et al., 2001). For fast evaluation of word confidences, the size of partial word graphs is reduced in the time axis to limit the time of the confidence measure evaluation. In addition, a special modification of the word graph topology is applied in the beginning and at the end of the graph for correct estimation of word confidences near word graph ends.

3.2 Force Alignment

The force alignment of adaptation utterances to the HMM states and mixtures is performed only for well-recognized segments of speech. To use only the trustworthy segments of speech, we use a quite strict criterion for word selection - only words, which have confidence greater than 0.99 and their neighboring words have confidence greater than 0.99 too, are selected. This ensures that the word boundaries of selected words are correctly assigned. The force alignment is then performed in three steps. In the first step, a state network is constructed based on phonetic transcriptions of recognized words. A lexical tree structure is used in the case of more phonetic transcriptions for one word to reduce the network size. In the second step, the Viterbi search with the beam pruning is applied on the state network to produce a state sequence corresponding to the selected words. Finally, feature vectors are assigned to the HMM state mixtures based on their posterior probability densities.

4 EXPERIMENTS

We have performed some experiments of the automatic online subtitling related to a real task running in the Czech public service television. The task concerns subtitling of live transmissions of the Czech Parliament meetings without the use of a shadow speaker. Hence, the original speech signal was being recognized.

4.1 Experimental Setup

An acoustic model was trained on 100 hours of parliament speech records with manual transcriptions. We use three-state HMMs and 8 mixtures of multivariate Gaussians for each state. The total number of 43 080 Gaussians is used for the SI model. In addition, discriminative training techniques were used (Povey, 2003). The analogue input speech signal is digitized at 44.1 kHz sampling rate and 16-bit resolution format. We use PLP parameterization with 19 filters and 12 PLP cepstral coefficients with both delta and delta-delta sub-features. Feature vectors are computed at the rate of 100 frames per second.

A language model was trained on about 24M tokens of normalized Czech Parliament meeting transcriptions (Chamber of Deputies only) from different electoral periods. To allow subtitling of arbitrary (including future) electoral period, five classes for representative names in all grammatical cases were created. See (Pražák et al., 2007) for details. The vocabulary size is 177 125 words. For the fast online recognition, we use a class-based bigram language model with Good-Turing discounting trained by SRI Language Modeling Toolkit (Stolcke, 2002). For a more accurate confidence measure of recognized words, the class-based trigram language model is used.

The experiments were performed on 12 test records from different parliament speakers, 5 minutes each, 6 612 words in total. To simulate conditions during a real subtitling, the data for the adaptation were accumulated from the beginning of each test record and individual adaptation steps were performed iteratively whenever the amount of adaption data reaches the pre-specified level. Evaluation of the recognition accuracy was done on the whole test

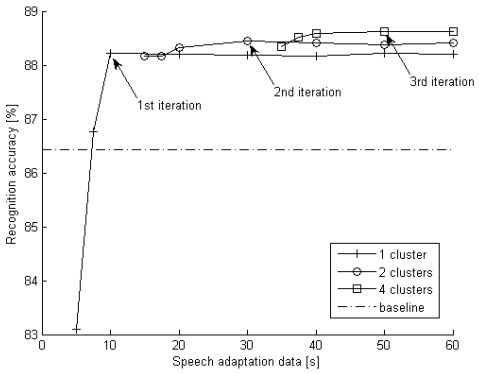

Figure 2: Recognition results using online incremental fMLLR adaptation.

records, thus the influence of each adaptation step approved itself only on parts of records after its application.

4.2 Online Adaptation Strategy

There are several online adaptation strategies that come into question. Firstly, incremental fMLLR approach should be used since it requires only moderate number of adaptation data. Moreover, the number of transformation matrices should be continuously increased as the amount of adaptation data grows. The optimum adaptation strategy should generate new transformation matrices whenever new adaptation data are available, this means after each newly recognized word. Unfortunately, such an approach is very time consuming since it takes a while to recompute new fMLLR matrices. In practice, it is appropriate to wait until the increase of information is sufficient so that the benefit of a new adaptation iteration (with increased number of transformation matrices/used clusters – see Section 2.2) is appreciable. The results of this adaptation strategy on our test records are presented in Figure 2.

Individual iterations of the fMLLR adaptation should be performed when sufficient amount of adaptation data for each cluster is available (marked by arrows). The Word Error Rate (WER) reduction after 3rd iteration of adaptation (using 4 clusters/RT nodes) over the baseline (without any adaption) is 16.2 % relatively. It is important to note that the real speech length is about twice longer than the speech adaptation data declared in Figure 2.

Another adaptation strategy combines the benefits from both fMLLR and MAP techniques described in Section 2. An incremental fMLLR approach is used as described above, but as soon as sufficient adaptation data for MAP are available; the whole model is recomputed and applied. From that moment new incremental fMLLR adaptation is started and so on. Anyhow, the best adaptation strategy for each speaker should be optimized based on the estimated length of his/her speech.

5 CONCLUSION

We have discussed some of the techniques of online speaker adaptation extended with several modifications and improvements. Based on the proposed online adaptation strategy we have performed experiments, where we were able to gain relative reduction of WER 16.2 % over the baseline (without adaptation). In the future work, we are going to investigate the combination of fMLLR and MAP adaptation enhanced with the SAT approach.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education of the Czech Republic under projects MŠMT 2C06020 and MŠMT LC536.

REFERENCES

Evans, M. J. (2003). Speech recognition in assisted and live subtitling for television. Technical report, BBC.

Gales, M. (1996). The generation and use of regression class trees for MLLR adaptation. Technical report, Cambridge University, Engineering Department.

Gales, M. (1997). Maximum likelihood linear transformations for HMM-based speech recognition. Technical report, Cambridge University, Engineering Department.

Gauvain, J.-L. and Lee, C.-H. (1994). Maximum a-posteriori estimation for multivariate gaussian mixture observations of Markov chains. *IEEE Transactions On Speech and Audio Processing*, 2(2):291 – 298.

Li, Y., Erdogan, H., Gao, Y., and Marcheret, E. (2002). Incremental on-line feature space MLLR adaptation for telephony speech recognition. In *ICSLP 2002, International Conference on Spoken Language Processing*.

Machlica, L., Zajíc, Z., and Pražák, A. (2009). Methods of unsupervised adaptation in online speech recognition. In *SPECOM 2009, International Conference on Speech and Computer*.

Povey, D. (2003). *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, Engineering Department.

Povey, D. and Saon, G. (2006). Feature and model space speaker adaptation with full covariance gaussians. In *INTERSPEECH 2006*.

Pražák, A., Müller, L., Psutka, J. V., and Psutka, J. (2007). LIVE TV SUBTITLING - fast 2-pass LVCSR system for online subtitling. In *SIGMAP 2007, International Conference on Signal Processing and Multimedia Applications*.

Stolcke, A. (2002). SRILM - an extensible language modeling toolkit. In *ICSLP 2002, International Conference on Spoken Language Processing*.

Wessel, F., Schlüter, R., Macherey, K., and Ney, H. (2001). Confidence measures for large vocabulary continuous speech recognition. *IEEE Transactions on Speech and Audio Processing*, 9(3).

Zajíc, Z., Machlica, L., and Müller, L. (2009). Refinement approach for adaptation based on combination of MAP and fMLLR. In *TSD 2009, International Conference on Text, Speech and Dialogue*.