SEGREGATION OF SPEAKERS FOR SPEAKER ADAPTATION IN TV NEWS AUDIO

Ulpu Remes, Janne Pylkkönen, Mikko Kurimo

Adaptive Informatics Research Centre
Helsinki University of Technology, Finland

ABSTRACT

Speaker adaptation is commonly used to compensate speaker variation in large vocabulary continuous speech recognition. In a multi-speaker environment where speakers change frequently speaker segregation is needed to divide the input audio stream to speaker turns. Speaker turns define the current speaker at each time and speaker adaptation can thus be done based on speaker turns. The novelty of this paper is that the speaker-specific transformations are estimated incrementally and in tandem with speaker segregation. Therefore we need a transformation that can be reliably estimated based on one speaker turn alone. We propose the constrained maximum likelihood linear regression (CMLLR) for this. In testing with Finnish TV news audio, speaker adaptation reduced the average letter error rate 25 % relative to baseline.

Index Terms— speaker recognition, speech recognition

1. INTRODUCTION

Variation introduced by different speakers and changing environments is still a challenge in large vocabulary continuous speech recognition. A number of speaker adaptation techniques have been developed to handle the consequent mismatch between training and testing conditions. However, in order to use speaker adaptation, speakers must be known. We do not need to know speaker identities, to be exact, but it is important that we know when speakers change and if some have spoken several times. In an application where speakers change frequently and new speakers appear, such information is not often available. Segregation of speakers is then needed before applying speaker adaptation to the data.

Speaker segregation aims at dividing the input audio to speaker turns. Each turn can be associated with one speaker whereas speakers can have many turns. Given no prior information on speakers, the task is to find speaker change boundaries that divide the audio to speaker turns and then label the speaker turns correctly. Speaker segregation is also an essential part in many speech technology applications. Important applications include retrieval and browsing of large automatically transcribed audio files and the analysis of spoken dialogs and multi-party meetings.

Speaker changes are found using a distance measure that illustrates the dissimilarity between two speech segments. Distance measures include divergence shape distance often used with line spectral pair (LSP) features [1] and distance measures based on generalized likelihood ratio [2, 3]. Distance measures may also be used as similarity measures in speaker clustering [1, 2]. This is one option for finding the speaker labels.

In [4] speaker labels are found by decoding each input utterance against a speaker-independent model and speaker-dependent models created with speaker adaptation. Maximum likelihood model is then selected and if this is the speaker-independent model, a new adapted model is created. Speaker-dependent model would be updated. Last, the utterance is re-decoded with the new adapted model.

The original contribution of this paper is to use a likelihood ratio based distance measure to find possible speaker change boundaries and to propose a new speaker tracking method for setting the speaker labels. The speaker tracking method presented in this paper differs from that proposed in [4] in that we take advantage of speaker-specific feature transformations and instead of decoding the input audio several times we calculate the likelihoods using a state sequence hypothesis and features generated with speaker-independent model. The transformations are estimated with constrained maximum likelihood linear regression (CMLLR) [5].

2. SPEAKER ADAPTATION

Linear transformations are a common choice for speaker adaptation, because they need only a modest amount of adaptation data to perform well even with large model sets. A limiting factor is that most transformation methods are model-based. In a multi-speaker environment it would be better if we did not need to create a new adapted model for each speaker.

CMLLR is a model-space transformation, but it assumes the model means and the covariances are adapted with the same transformation matrix [5]. With this assumption, adaptation can be done in the feature space rather than model space. Features are transformed as

$$\hat{\mathbf{o}}(\tau) = \mathbf{A}\mathbf{o}(\tau) + \mathbf{b} = \mathbf{W}\boldsymbol{\zeta}(\tau), \tag{1}$$

where $\mathbf{A}$ is the transformation matrix and $\mathbf{b}$ the constant bias, $\mathbf{W}$ is the extended transformation matrix $\mathbf{W} = [\mathbf{b}'\ \mathbf{A}']'$ and $\boldsymbol{\zeta}(\tau) = [1\ \mathbf{o}(\tau)']'$ is the extended observation vector at time τ.

The maximum likelihood solution for i-th row in $\mathbf{W}$ is [5]

$$\mathbf{w}_i = (\alpha\mathbf{p}_i + \mathbf{k}(i))\mathbf{G}(i)^{-1}, \tag{2}$$

where $\mathbf{p}_i$ is the extended cofactor vector $\mathbf{p}_i = [\,0\ c_{i1} \ldots c_{in}]$ ($c_{ij} = \mathrm{cof}(\mathbf{A}_{ij})$) and n is the feature dimension. Factor α is solved from a quadratic equation presented in [5]. $\mathbf{G}(i)$ and $\mathbf{k}(i)$ are calculated as

$$\mathbf{G}(i) = \sum_{\tau=1}^{T} \boldsymbol{\zeta}(\tau)\boldsymbol{\zeta}(\tau)' \sum_{k=1}^{K} \frac{1}{\sigma_k(i)^2}\gamma_k(\tau) \tag{3}$$

$$\mathbf{k}(i) = \sum_{\tau=1}^{T} \boldsymbol{\zeta}(\tau)' \sum_{k=1}^{K} \frac{1}{\sigma_k(i)^2}\mu_k(i)\gamma_k(\tau), \tag{4}$$

where $\mu_k(i)$ and $\sigma_k(i)^2$ denote the i-th component of mean and variance of Gaussian k and $\gamma_k(\tau)$ is the posterior probability of being in Gaussian k at time τ. The beauty is that $\mathbf{G}(i)$ and $\mathbf{k}(i)$ can be calculated incrementally as more data becomes available. With speaker-specific $\mathbf{G}(i)$ and $\mathbf{k}(i)$ information extracted from a new utterance can be merged with information from all the previous utterances that share the same speaker.

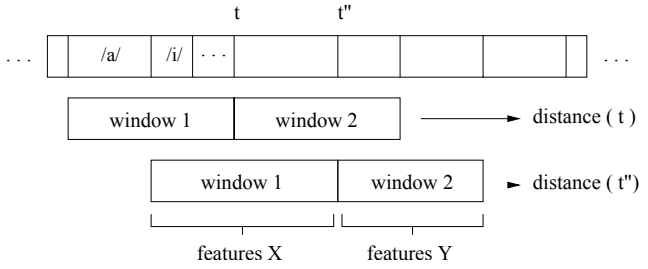

Fig. 1. Moving windows positioned to test for speaker change at hypothesized phone boundaries at times t and t".

3. SPEAKER SEGREGATION

Speaker segregation as described here assumes we have no prior information on speaker turns. Speaker change detection (SCD) is used to find the speaker change boundaries and speaker tracking to label the speaker segmented audio. In our approach, also the transformation matrices for speaker adaptation are estimated during speaker tracking.

3.1. Speaker change detection

Features extracted from speech signal characterize both the speech and the speaker. However, features collected from more than a few seconds of speech are expected to fill the feature space in a way that depends primarily on the speaker [6]. Speaker changes are thus detected using a pair of moving windows both which hold a short segment of speech (Figure 1). Distance between the speech segments is calculated to determine whether there exists a speaker change in between the windows. Minimum window size is set to 5.6 seconds. Window sizes are not constant, for windows move according to hypothesized phone boundaries. The phone level time resolution was proposed in [3].

Assuming we have two speech segments with the set of features $X = \{\mathbf{x}_n\}$ and $Y = \{\mathbf{y}_m\}$ and we need to know if they were uttered by the same speaker, we wish to test the hypothesis:

H_0: X and Y are generated by the same speaker

H_1: X and Y are generated by different speakers

Taking X and Y as coming from independent Gaussian processes, we may test our hypothesis using the generalized likelihood ratio test [2]

$$\lambda = \frac{\mathcal{L}(X, Y \mid \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})}{\mathcal{L}(X \mid \hat{\boldsymbol{\mu}}_X, \hat{\boldsymbol{\Sigma}}_X)\, \mathcal{L}(Y \mid \hat{\boldsymbol{\mu}}_Y, \hat{\boldsymbol{\Sigma}}_Y)} \ , \tag{5}$$

where $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$ are the maximum likelihood estimates calculated for mean and covariance from features in X and Y. The generalized likelihood ratio is always greater than zero and less than unity, so the distance between speech segments is then defined as

$$d = -\log \lambda. \tag{6}$$

The distance is calculated as [2]

$$d = \frac{1}{2}[N \log |\mathbf{S}_X| + M \log |\mathbf{S}_Y| - (N+M) \log |\mathbf{S}_{X,Y}|] \, , \tag{7}$$

where $\mathbf{S}$ are the sample covariance matrices calculated from features and N, M are the number of features in X and Y, respectively. Note, that the distance depends on both the mean and covariance estimates. To test the similarity between the covariances only, as suggested in [1, 2], the sample covariance matrix $\mathbf{S}_{X,Y}$ should be replaced with $(N\,\mathbf{S}_X + M\,\mathbf{S}_Y)/(N+M)$.

Distance is calculated at phone boundaries and a threshold is used to detect speaker changes — or rather speaker change intervals. Such interval is seen to begin when the distance surpasses threshold value, and end when the distance drops down again. Speaker change boundaries are then placed where the distance met its maximum value within an interval. This should prevent us from detecting multiple change boundaries where one speaker change occurs.

3.2. Speaker tracking

The speaker tracking method presented here is based on the assumption that each speaker has an optimal adaptive transformation that is not optimal for any other speaker. Thus, provided that we can get estimates close to the optimal transformations, different speakers can be recognized. Transformations may have to be estimated from relatively small amounts of data, for the average length of speaker turn in our TV news data is around 30 seconds. Linear transformations do not generally need a lot of data, so we should do fine with CMLLR.

The method presented in [4] has the same basis, but the speaker adaptation methods used are all model-based and the input audio is either decoded against all models that are being considered or a supplementary speaker model is used for model selection. In our approach, audio is decoded once with the speaker-independent model to get a state sequence hypothesis, which is then used to evaluate all feature transformations. Transformation matrices do not need a lot of memory space [7] so we can keep a good many of them available during speaker tracking.

Speaker tracking is carried out as shown in Figure 2. Features extracted from audio signal are adapted with the transformations estimated for previous speakers, if such exist. State information is read from hypothesis and log-likelihoods are calculated for the features as [5]

$$\mathcal{L}(\mathbf{o}(\tau)|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \mathbf{A}, \mathbf{b}) = \ln \mathcal{N}(\hat{\mathbf{o}}(\tau)|\boldsymbol{\mu}, \boldsymbol{\Sigma}) + \frac{1}{2} \ln |\mathbf{A}|^2, \tag{8}$$

where $\hat{\mathbf{o}}(\tau)$ are the transformed features and $\mathbf{A}$, $\mathbf{b}$ are the transformation matrix and constant bias. Addition of the term $\ln |\mathbf{A}|^2$ is due to effects of adaptation.

Log-likelihoods are summed over time and thus they become likelihood values for the different transformations. At speaker change boundary we then find the highest log-likelihood value. Should this belong to the speaker-independent model represented by unadapted features, a new speaker label is created and the feature information collected from the speaker turn is used to estimate CMLLR transformation for the new speaker. If instead the maximum likelihood features were produced with a speaker-dependent transformation, the feature information is merged with previously collected information (see Section 2) and new transformation is estimated for this speaker. Speaker turns are also labeled accordingly.

Sometimes a transformation estimated for one speaker can also help another. This would result to both speakers being labeled the same. To handle this problem we added a threshold to accepting the decision made in comparing the likelihoods: if the ratio between the likelihood value calculated for unadapted features and the highest speaker adapted likelihood value does not surpass a given threshold, we take it that the selected transformation would not significantly benefit our current speaker and we decide we have a new speaker.

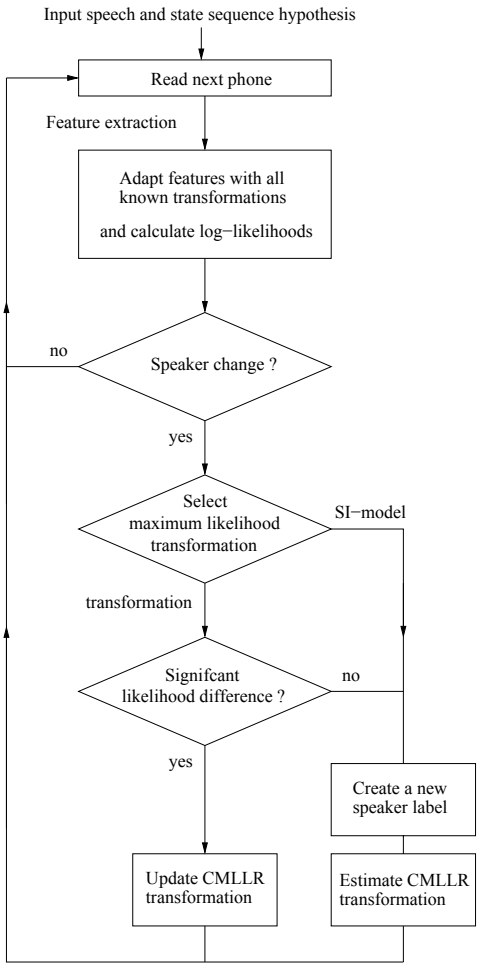

Fig. 2. Speaker tracking labels the given speaker turns and incrementally estimates CMLLR transformations for speaker adaptation.

4. SYSTEM AND DATA

Our large vocabulary continuous speech recognition system uses a morph-based growing n-gram language model [8, 9] trained on book and newspaper data. Text data contains 145 million words. Since all words and word forms can be represented with morphs, we have an unlimited decoding vocabulary. Our decoder is an efficient time-synchronous beam-pruned Viterbi token-pass system [10].

For acoustic modeling we have state-clustered hidden Markov triphone models constructed with a decision-tree method [11]. Each state is modeled with 8 Gaussians and states are also associated with gamma probability functions to model the state durations [12]. Speech signal is represented with 12 MFCC and the log-energy along with their first and second differentials. Features are treated with cepstral mean subtraction and maximum likelihood linear transformation that is estimated in training.

Models were trained with data taken from the Finnish SPEECON database [13]. The selected training data set had 26 hours of clean speech data recorded with close-talk microphone from 208 speakers both male and female. Among utterances were words, sentences and free speech.

System is tested with a set of speech clips taken from the Finnish Broadcasting Company (YLE) evening news. We chose 48 speech clips from 7 TV news broadcasts to the test set. The complete test set had around one hour of speech data from 49 speakers. This test set could be divided to 153 speaker turns. For parameter optimization we had a small set of speech clips taken from a separate TV news broadcast. This set had 10 minutes of speech data from 9 speakers one of which is also present in the test set. All speech clips were selected to contain only planned speech from newscasters and reporters. Background music and other noise is present in some parts.

In Finnish, speech recognition performance is best measured with letter error rate (LER). Word error rate (WER) is more common in speech recognition measurements, but it is not well applicable for Finnish where words tend to be rather long. Finnish words often correspond to more than one English words and constitute of several concatenated morphemes like “kahvin+juoja+lle+kin” which means “also for a coffee drinker”.

5. RESULTS

Speaker adaptation is tested under three different conditions. First, the speaker-specific CMLLR transformations are estimated based on true speaker turns (a). Then, in order to test speaker tracking alone, system is given the true speaker change boundaries but no information about the speakers (b). In speaker tracking, the given turns are labeled and speaker-specific transformations are estimated based on this labeling. Last, system is not given any information on speaker turns (c), but speaker change boundaries are searched with speaker change detection. This divides the audio to supposed speaker turns that are then labeled and used in estimating the speaker-specific transformations. In all three cases, audio is re-decoded after we have the transformations. Results are presented in Table 1.

Table 1. Speech recognition performance.

	WER	LER
Baseline	23.0	7.9
Speaker adaptation		
(a) true speaker turns	19.8	6.0
(b) speaker tracking	19.5	5.9
(c) SCD + tracking	19.4	5.9

Speaker adaptation significantly improves the recognition results. With true speaker turns (a) the relative error reduction is 24 %. Results from speaker segregation and adaptation experiments are even better. Compared to baseline, the relative error reduction is 25 % in experiments (b) and (c).

It is evident from the results in Table 1 that our speaker segregation method can provide a good basis for speaker adaptation. The small difference in results (b) and (c) compared to (a) indicates that the automatic methods lead to at least as low error rate as the manually marked speaker turns. They do make some mistakes in speaker segregation, but from speaker adaptation point of view these are, at least on average, better decisions. A similar difference was marked in [4] where speaker tracking was tested with data manually segmented to sentences. Note here, that the results in Table 1 indicate our speaker tracking method works fine with both true and detected

speaker change boundaries, even if speaker change detection creates quite a many false boundaries that make the detected speaker turns shorter than they really are, leaving less data for speaker tracking.

The results from speaker segregation were also compared to the true speaker turns, so that the results could be evaluated in isolation from speaker adaptation. Speaker change detection performance is evaluated on the SCD evaluation metrics suggested in [3]. With 0.5 s tolerance factor the results are good. False acceptance is 51 % and false rejection is 16 %. Thus, there are many false speaker change boundaries, but also most true changes were found. False boundaries are not so critical in this task because the speaker tracking phase often clears them. After speaker tracking, we have false acceptance 26 % and false rejection 17 %.

Success in speaker tracking can be measured also with average cluster purity (*acp*) and average speaker purity (*asp*) measures [14]. Our proposed speaker tracking method succeeded remarkably well when given the true change boundaries: average cluster purity is 97 % and average speaker purity 95 %. Thus, the speaker turns that get the same label are most often from the same speaker, and the turns from one speaker most often have a common label.

Speaker tracking performance decreases slightly when we replace the true change boundaries with detected speaker changes. Now, we have average cluster purity 96 % and average speaker purity 84 %. Again, speakers do not share labels, but this time not all the speech from one speaker has been labeled the same. 56 speakers were found in speaker tracking, whereas 49 were found when system had the true change boundaries. This is also the true number of speakers. However, most extra labels have been assigned to short speaker turns that speaker change detection tends to create around noise. Having them labeled as new speakers is probably good from speaker adaptation perspective.

Finally, average cluster and speaker purity give also means to examine how thresholding the likelihood values affected speaker tracking. We had the threshold so that there should be at least 10 % difference in likelihood values for the speaker-dependent transformation to be selected. Most often the difference is over 15 % if the transformation is correct and should be selected, and less than 5 % if not. With the threshold removed, we get *acp* 69 % and *asp* 99 % when system is given the true change boundaries. Similarly, we get *acp* 66 % and *asp* 97 % with detected speaker changes. There are now less speaker labels than true speakers and several speakers have been given the same label.

6. CONCLUSIONS

Methods for segregating speech from different speakers in TV news audio were described and tested along with speaker adaptation. We used the generalized likelihood ratio test [2] for speaker change detection and for speaker tracking we suggested a method that labels the given speaker turns and equips each speaker with a CMLLR transformation. Transformations are estimated incrementally and are based on all the data belonging to their respective speaker label.

The suggested method is related to that proposed in [4] although some key features are different. The differences are that we have feature transformations and we calculate transformation likelihoods based on state sequence hypothesis generated with the baseline speaker-independent model. We save all estimated transformations and do speaker adaptation offline. As we also test several transformations for each speaker turn, we utilize thresholding in transformation selection.

Test results denote that speaker segregation and speaker adaptation significantly improve system performance in speech recognition task. The proposed speaker segregation method is most worthwhile when followed by speaker adaptation because it directly provides the speaker-specific transformations. Speech recognition results also suggest that speaker adaptation may actually benefit from automatic segregation. However, our speaker segregation method suits also other purposes, for it did well in correctly partitioning and labeling the speech data.

7. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland and the Finnish National Technology Agency (Tekes) in the projects: *New adaptive and learning methods in speech recognition* and *New methods and applications for speech technology*.

8. REFERENCES

[1] L. Lu and H. Zhang, "Speaker change detection and tracking in real-time news broadcasting analysis," in *Proc. ACM Multimedia*, 2002, pp. 602–610.

[2] H. Gish, M. Siu, and R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification," in *Proc. ICASSP*, 1991, vol. 2, pp. 873–876.

[3] D. Liu and F. Kubala, "Fast speaker change detection for broadcast news transcription and indexing," in *Proc. EUROSPEECH*, 1999, vol. 3, pp. 1031–1034.

[4] Z. Zhang, S. Furui, and K. Ohtsuki, "On-line incremental speaker adaptation with automatic speaker change detection," in *Proc. ICASSP*, 2000, vol. 2, pp. 961–964.

[5] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[6] H. Gish and N. Schmidt, "Text-independent speaker identification," *IEEE Signal Processing Magazine*, pp. 18–31, 1994.

[7] D. Liu, D. Kiecza, A. Srivastava, and F. Kubala, "Online speaker adaptation and tracking for real-time speech recognition," in *Proc. INTERSPEECH*, 2005, pp. 281–284.

[8] V. Siivola and B. Pellom, "Growing an n-gram language model," in *Proc. INTERSPEECH*, 2005, pp. 1309–1312.

[9] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to Finnish," *Computer Speech and Language*, vol. 20, no. 4, pp. 515–541, 2006.

[10] J. Pylkkönen, "An efficient one-pass decoder for Finnish large vocabulary continuous speech recognition," in *Proc. 2nd Baltic Conference on Human Language Technologies*, 2005, pp. 167–172.

[11] J. J. Odell, *The use of context in large vocabulary speech recognition*, Ph.D. thesis, Univ. of Cambridge, 1995.

[12] J. Pylkkönen and M. Kurimo, "Duration modeling techniques for continuous speech recognition," in *Proc. INTERSPEECH*, 2004, pp. 385–388.

[13] D. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kiessling, "SPEECON - speech databases for consumer devices: Database specification and validation," in *Proc. LREC*, 2002, pp. 329–333.

[14] J. Ajmera, H. Bourland, I. Lapidot, and I. McCowan, "Unknown-multiple speaker clustering using HMM," in *Proc. INTERSPEECH*, 2002, pp. 573–576.