MLLR TRANSFORMS OF SELF-ORGANIZED UNITS AS FEATURES IN SPEAKER RECOGNITION

Man-hung Siu, Omer Lang, Herbert Gish, Steve Lowe, Arthur Chan and Owen Kimball

Raytheon BBN Technologies
Cambridge, MA 02138, USA

ABSTRACT

Using speaker adaptation parameters, such as maximum likelihood linear regression (MLLR) adaptation matrices, as features for speaker recognition (SR) has been shown to perform well and can also provide complementary information for fusion with other acoustic-based SR systems, such as GMM-based systems. In order to estimate the adaptation parameters, a speech recognizer in the SR domain is required which in turn requires transcribed training data for recognizer training. This limits the approach only to domains where training transcriptions are available. To generalize the adaptation parameter approach to domains without transcriptions, we propose the use of self-organized unit recognizers that can be trained without supervision (or transcribed data). We report results on the 2002 NIST speaker recognition evaluation (SRE2002) extended data set and show that using MLLR parameters estimated from SOU recognizers give comparable performance to systems using a matched recognizers. SOU recognizers also outperform those using cross-lingual recognizers. When we fused the SOU- and word recognizers, SR equal error rate (EER) can be reduced by another 15%. This suggests SOU recognizers can be useful whether or not transcribed data for recognition training are available.

Index Terms— speaker recognition, unsupervised learning, self-organized units.

1. INTRODUCTION

Most speaker recognition (SR) algorithms are based on Gaussian mixture models and use short-term spectral properties for making SR decisions. The most successful alternatives that consider longer acoustic spans are large vocabulary speech recognizers that use speech recognition adaptation parameters, such as maximum likelihood linear regression [1] adaptation transforms, as SR features. Intuitively, adaptation parameters can be made to be phoneme-dependent or phoneme class dependent. This should exploit phoneme specific speaker characteristics. Past research [2] has shown that using MLLR matrices as features for a support vector machine (SVM)-based speaker classifier can be competitive with a GMM-based SR system. In addition, because of the differences in the two approaches, they have been shown to fuse well and can result in better overall system performance.

One necessary component of a MLLR-SVM system is a suitable automatic recognition system (ASR) to estimate adaptation parameters on the SR data. For languages such as English, and channels such as broadcast news or telephone, there are many public domain resources for building such recognition systems. However, for less common acoustic channels or languages, the availability of resources for training an ASR system can be limited. Recently, there has been a surge in interest in building ASR systems for resource constrained domains with techniques such as subspace Gaussian mixture models [3], semi-supervised learning [4] or universal phoneme models. In the case of zero resources, one obvious possibility is to use mis-matched recognizers trained on some other domains with sufficient resources. However, with mis-matched recognizers, the degree of SR performance degradation may be difficult to predict and could be quite large. We have proposed an alternative using the HMM-based self-organized units [5] (SOUs). SOU are multi-frame phoneme-like, acoustic units that represent different acoustic patterns, and can be trained without any transcribed data. The SOUs were shown to work well in topic identification [6] and keyword discovery tasks [7].

In this paper, we explore the application of SOUs in MLLR-SVM-based speaker recognition. Instead of estimating the MLLR transforms from traditional speech recognizers trained with transcribed data in the language of the SR data, we use an SOU recognizer trained without supervision. Because the SOUs require no transcription for training, there is no issue with training resources [1] as we can, at the extreme, use the SR training data to train the SOU recognizer. We performed SR experiments on a subset of the NIST 2002 Speaker Recognition evaluation (SRE 2002) extended testset. We compared the SR performance of using a SOU recognizer against a state-of-the-art English recognizer which matched the SR data, as well as using recognizers from other mis-matched languages. We found that the SOU performance is very competitive with the English recognizer, and is significantly better than recognizers from mis-matched languages.

The rest of the paper is organized as follows. In Section 2, we review the SOU technology. In Section 3, we describe our MLLR-SVM SR system. This is followed by a description of our experiments in Section 4 and we conclude the paper in Section 5.

2. SELF ORGANIZING UNITS

2.1. Unsupervised HMM

For supervised training of an HMM for speech recognition, its parameters are typically specified by the acoustic model parameters, θ_{am}, and the language model parameters, θ_{lm}. For notational conveniences in this paper, we group them as a single parameter set $\theta = [\theta_{am}, \theta_{lm}]$. Then, the ML parameter estimation finds the parameter set, $\hat{\theta}_{sup}$, that maximizes the joint likelihood, $p(X, W|\theta)$, of observation sequence X and the label sequence W. That is,

$$\hat{\theta}_{sup} = \arg\max_{\theta} p(X, W|\theta). \quad (1)$$

In the case of unsupervised training in which the label sequence W is not known, we maximize the joint likelihood by searching not

[1] We are dealing with data availability instead of computational resources.

only over the model parameters but also all possible label sequences. That is, W becomes a variable to be optimized. The unsupervised ML parameter estimation becomes,

$$\begin{aligned}\hat{\theta}_{unsup} &= \arg\max_{\theta}\max_{W} p(X, W|\theta), &(2)\\ &= \arg\max_{\theta}\max_{W} p(X|W,\theta)p(W|\theta) &(3)\end{aligned}$$

The maximization over both the label sequence and the acoustic model likelihood in Eqn 3 balances the acoustic likelihood and label sequence structure.

Eqn. 2 maximizes over two sets of variables, θ and W, which can be performed using iterative maximization. At each iteration, one set of variables is fixed while the other set is maximized. Then we alternate between them. So, at the i-th iteration, the two maximization steps are:

1. find the best parameter set θ_i on the previously found label sequence W_{i-1}.

$$\theta_i = \arg\max_{\theta} p(X, W_{i-1}|\theta). \quad (4)$$

2. find the best word sequence W_i by using the previously estimated parameter set θ_i.

$$W_i = \arg\max_{W} p(X, W|\theta_i), \quad (5)$$

Comparing Eqns 1 and 4, it is obvious that Step 1 (Eqn 4) is simply the regular supervised HMM training (both acoustic and language models) using the newly obtained transcription W_{i-1} as reference. Finding the best word sequence in the second step would suggest a Viterbi recognition pass. Although recognition is usually viewed as finding the most likely label sequence over the posterior probability, $p(W|X, \theta)$, it is easy to show that the same sequence also maximizes the joint likelihood $p(X, W|\theta)$ as in Eqn. 5. So, Eqn. 5 expresses the recognition of a new transcription using the updated parameters θ_i.[2]

2.2. Initialization

Audio is first segmented based on its spectral discontinuities which are learned without supervision from the audio signal [8]. It is followed by fitting each audio segment with a polynomial (quadratic) trajectory in the cepstral space. The audio segments are then grouped into clusters of similar acoustics based on the distance between their polynomial trajectory parameters [9]. The distance measure currently used on a pair of segments is the area between their polynomial trajectories. These segment clusters represent collections of sound units. Any individual cluster is a collection of variants of a particular sound and forms the basis for generating a segmental Gaussian mixture model (SGMM) with each mixture component representing a segment cluster. The SGMM is trained with the EM algorithm.

The SGMM becomes a speech tokenizer when, for an audio segment, it returns the mixture index by which the segment likelihood is maximized. After building the SGMM, it is used as a tokenizer for the training segments. These segment labels form the initial transcription for HMM training.

2.3. SOU HMM Training

We use the state-of-the-art BBN Byblos recognizer [10] for HMM training. Byblos includes advanced signal processing techniques, such as Vocal Tract Length Normalization (VTLN), Heteroscedastic Linear Discriminant Analysis (HLDA) feature transformation, context-dependent triphone and quinphone models, multi-pass recognition, speaker adaptive training etc. Byblos uses “flat start” HMM training that does not require token time marks. Instead, iterative alignment and model estimation are carried out. While discriminative training is part of Byblos, our current experiments used only the maximum likelihood training. Details about the Byblos training can be found in [10]. In addition to acoustic models, initial bigram and trigram language models are constructed using the label sequences generated by the segmental tokenizer. Following Eqns 4 and 5, we iteratively maximize the model likelihood and find the best label sequence.

2.4. Tokenization

With the trained acoustic models and language models, the tokenization of audio into SOU sequences is no different from regular phoneme recognition. To create context dependent models in Byblos, we need to create “phoneme” classes to drive the decision-tree based phoneme-state clustering. To produce “linguistic questions” to drive the decision tree-based state clustering, we cluster the 64 SGMM’s into 16 classes to act as “phoneme classes”. Because we used the phonemes (or SOUs) as words, true context modeling occurs only in cross-word models. To use context-based model, recognition lattices are re-scored using cross-word models.

3. MLLR-SVM

3.1. Recognition System

Our MLLR-SVM SR systems use the BBN Byblos speech recognition system [10]. Byblos is a multi-pass, HMM-based recognizer. The decoding can be broadly divided into two stages: the Speaker Independent (SI) stage and the Speaker Adapted (SA) stage. The SI stage uses models shared across all speakers. The SA stage adapts the models and features to better fit with the observed data. It uses speaker adapted training (SAT) models together with various constrained MLLR adaptation and model-based MLLR adaptation.

In terms of signal processing, 12 cepstral features, normalized after applying vocal tract length normalization, are computed at a rate of 100 frames per second. A moving window of 15 consecutive frames of cepstral features (plus normalized energy) are concatenated into a 265-dimensional vector which is then projected down to a 46-dimensional feature vector using heteroscedastic linear discriminant analysis (HLDA). Byblos has multiple-decoding passes. The fast first pass decoding in the forward direction uses an low-complexity, non-crossword, state-tied-mixture (STM) models [10] that is followed by a backward search using more detailed, non-crossword, phoneme-tied-mixture (PTM) model. The fast forward search reduces the search space and estimates path scores of feasible paths to be used for pruning in the backward search. The backward search generates a lattice that can be re-scored using the most detailed, crossword, state-clustered-tied-mixture (SCTM) model. At the end of the SI stage, a one-best transcription is produced which is used as the reference for speaker adaptation.

Multiple adaptation stages are embedded in the Byblos system. In the feature space, the cepstral features are adapted before concatenation, called “pre-transform” and adapted again after HLDA pro-

[2] We ignored all the approximations typically associated with practical recognizers, such as using language modeling weights, or pruning.

jection is applied (“post-transform”). Both are constrained MLLR transforms. Then, separate sets of model-based MLLR matrices are estimated for the STM, PTM and SCTM models based on a Gaussian clustering tree.

Byblos can be configured to run at a faster speed by reducing some of the passes. For example, our two-times-faster-than-real-time (2xFtRT) system performs only the forward backward passes on both the SI and SA stages.

Byblos can also be configured as a sub-word recognizer by changing the vocabulary in the dictionary from words to frequent subwords and phoneme n-grams. Because single phonemes are added as part of the lexical unit set, the subword recognizer can backoff to pure phoneme recognition if the test vocabulary does not match that of training. The subword recognizer can be viewed as a sophisticated phoneme recognizer. Before acoustic model training, the transcription is broken into subword units through a greedy left-to-right, longest match segmentation process. For language model (LM) training, all possible subword segmentation is considered by constructing a subword-graph for each sentence. This is similar to the approach used for LM training for using compound words.

3.2. Adaptation Parameters and SVM Classifier

There are multiple sets of adaptation parameters estimated by Byblos, including the pre-transforms, the post-transforms and the 3 sets of MLLR model transforms. The number of the MLLR model transforms is controlled by the depth of the Gaussian clustering tree which is a binary tree with a MLLR transform on each node. For example, a tree with a depth of 2 contains 2 leaf-nodes and 1 intermediate nodes, resulting in a total of 3 transforms. All or some of these transforms can be used as features for the SVM SR classifier.

We use the SVMlight [11] package for speaker classification. Features are rank-normalized on a per component-basis with a normalizer estimated on the background data set. Instead of using the SVM decision, the distance between the SVM score for a test, which is test sample and the decision boundary, is extracted as the classification score.

3.3. Score Fusion

SR scores from different classifiers can be fused together by treating the system-specific scores as features to another classifier. In our case, we use the generalized linear model (GLM) [12] which has been shown to perform well as a backend fusion module [13]. In our work, we used the publicly available GLM package in R [12].

4. EXPERIMENTS

We performed a number of experiments on the first testset of the SRE2002 extended testset. To speed up our experiments, we used the 2xFtRT recognition setup that performs only two passes in the SI-stage and SA stages. Because the SOUs are phoneme-like units, we also experimented with a subword recognizer. So, we tested the MLLR-SVM SR using word, subword and SOU recognizers across different languages. The specifics of their training are shown in Table 1 in which “wd”, “sub” denote “word” and “subword”, “Eng”, “Spa” and “Man” denote “English”, “Spanish” and “Mandarin” respectively. For all word and subword models, discriminative training was applied. For SOUs, models were trained with 5 iterations of re-training using only maximum likelihood training [6]. For all recognitions, speech activity detection was performed using a four-state HMM that considered cepstral features from both sides of a conversation.

Table 1. Different recognition systems and their acoustic training conditions with CH and CF denote the Callhome and CallFriend corpora, Eng, Spa and Man denote English, Spanish and Mandarin respectively.

	Language	Mode	Corpora	Am. Trn
Eng-wd	English	word	Fisher	370hr
Man-wd	Mandarin	word	HKUST, CH, CF	250hr
Spa-wd	Spanish	word	Fisher, CH, CF	250hr
Eng-sub	English	subword	Fisher	370hr
Spa-sub	Spanish	subword	Fisher, CH, CF	250hr
Eng-SOU	English	sou	Fisher	60hr
Man-SOU	Mandarin	sou	HKUST, CH, CF	60hr
Spa-SOU	Spanish	sou	Fisher, CH, CF	15hr

Table 2. SR EER on testset 1 of SRE2002 extended set trained and tested on 1 cut using different recognizers for the MLLR-SVM approach.

Language Mode	English	Spanish	Mandarin
Word	4.82%	7.08%	7.98%
Subword	5.20%	6.25%	–
SOU	5.05%	5.74%	6.37%

Our adaptation and classification configuration roughly follows [2]. Instead of using all the possible speaker specific transformations as SVM features, we used only the model-based MLLR matrices based on a Gaussian clustering regression tree of depth 2. This results 3 matrices, two from leaf-nodes and 1 from the root node. Two sets of matrices were generated for the two different models sets in Byblos’ forward and backward passes. This resulted in an SVM feature vector of 3x2x46x47 = 12972 dimensional vector.

Following SRE2002 setup, we trained the SVM using a background set of 6758 conversation sides taken from the Switchboard corpus. We used a linear kernel with all the default parameter settings in SVMlight, including setting the margin error weight to 1. The training and test conditions followed the NIST evaluation plan for extended test side with the exception that we only evaluated performance using single training cut. The results are shown in Table 2.

Our first set of experiments (first row) used word-based ASR models. As expected the English model, matching the SRE’s language, produces the best results and for the mis-matched models, Spanish performs slightly better than Mandarin. This may be because Spanish is phonetically more similar to English than Mandarin. Our second set of experiments (second row) used subword-based ASR models. Our expectation was that these ”looser” models will perform worse when the ASR model matches the SRE language and better when it mismatches. As expected, the English subword-based model performs worse than the English word-based model, while the Spanish subword-based model performs better than the Spanish word-based model. Because of the significant difference between Mandarin-word and English, we did not run the Mandarin subword experiment. Our third set of experiments (last row) used SOU-based ASR models. In the case of English and Spanish, the SOU-based models produce an improvement over the subword-based models but the improvements are small. The Mandarin SOU-based model performs better than the Mandarin word-based model. Once again, English, the matching model, performs best. The better performance of the Spanish and Mandarin over the corresponding word or subword system suggests that SOUs are flatter models and are not as language specific. *However, by the unsupervised nature of the SOU training, one could train an SOU system using the SR training data and remove the need of using mis-match models.* The advantage of using English SOU over word or subword of a mis-matched language (Spanish and Mandarin) is substantial. This confirms our intuition that when there is no transcriptions available, SOUs, which are trained without supervision on the target domain, can be superior than using a recognizer from mis-matched domains.

Because the SOU and word based systems are modeling the speech data with different resolutions, one may gain improved performance by combining their scores. Using a GLM with default settings, we combined the scores from English word-based MLLR-SVM with the English SOU-MLLR-SVM. The combination has 3 parameters, one for the linear offset, two for the weights of the scores of the two systems and are learned from the test data. By combining the two different MLLR-SVM systems, the EER was reduced to 4.13%. In Figure 1, we plot the Detection Error Tradeoff (DET) curves of the Spanish subword system (Spa-sub), which is the best cross-lingual system, together with the English word system (Eng-wd), which is the best system overall, English SOU system (Eng-sou) and the fused system of English word and SOU (Eng-wd+sou). Across different operating points and consistent with the EER results, the English SOU system is only slightly worse than the English word system and significantly better than the Spanish subword system. The fusion of the English word and SOU systems provides a substantial improvement.

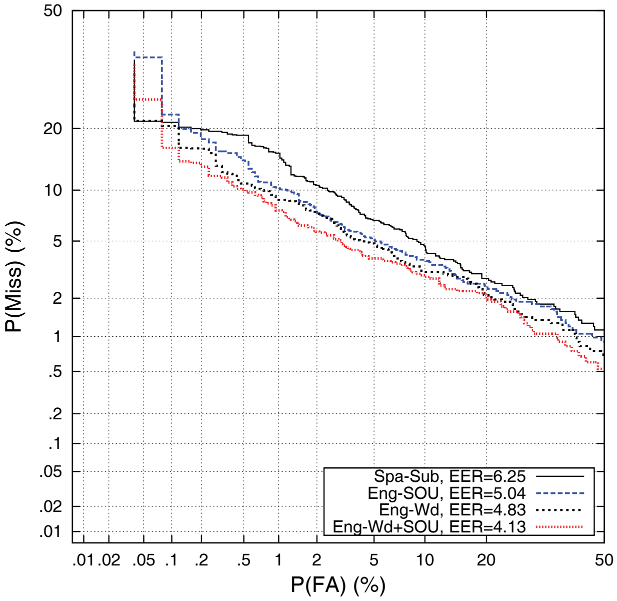

Fig. 1. DET curves comparing the English word, SOU, their fusion against the Spanish subword system

5. CONCLUSIONS

In this paper, we demonstrate the use of SOU recognition, which can be trained without supervision, for MLLR-SVM in speaker recognition. We showed that SOU-based MLLR-SVM SR performed only slightly worse than English word-based MLLR-SVM which matched the test domain, and outperformed other MLLR-SVM systems trained on mis-matched languages. Furthermore, a combination of the SOU-based system and English word recognition system gave a 15% relative EER reduction compared to using the English word recognition alone.

6. REFERENCES

[1] M. Gales and S. Young, “Maximum likelihood linear transformations for HMM-based speech recognition” in *CSL* volume 12, 1998.

[2] A. Stolcke, L. Ferrer, S. Kajareka, E. Shriberg and A. Venkataraman, “ MLLR transforms as features in speaker recognition” “Eurospeech 2005”

[3] D. Povey *et. al* “Subspace Gaussian mixture models for speech recognition,” in ICASSP 2010.

[4] J. Ma and R. Schwartz, “Unsupervised versus supervised training of acoustic models,” in *Interspeech* 2008.

[5] H. Gish, M. Siu and W. Belfield, “Unsupervised training of an HMM-based speech recognition system for topic classification,” in *Interspeech* 2009.

[6] M. Siu and H. Gish and A. Chan and W. Belfield, “Improved topic classification and keyword discovery using an HMM-based speech recognizer trained without supervision”, in *Interspeech* 2010.

[7] M. Siu, H. Gish, S. Lowe and A. Chan, “Unsupervised Audio Patterns Discovery using HMM-based Self-Organized Units”, in *Interspeech 2011*

[8] J. Cohen, “Segmenting Speech Using Dynamic Programming”, in *JASA*, May 1981.

[9] H. Gish and K. Ng, “A segmental speech model with applications to word spotting”, in *ICASSP* 1993.

[10] S. Matsoukas *et. al*. “Advances in Transcript of broadcast news and conversational telephone speech within the combined EARS BBN/LIMSI system”, in *IEEE TASLP*, vol. 14, 2006.

[11] ”T. Joachims”, “Making large-Scale SVM Learning Practical”, in *Advances in Kernel Methods - Support Vector Learning*, MIT-Press, 1999.

[12] W. Venables and B. Ripley, “Modern Applied Statistics with S”, New York, Springer, 2002.

[13] N. Brummer *et. al* “Fusion of Heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006”, in *IEEE TASLP* vol 15, 2007.