THE LANGUAGE-INDEPENDENT BOTTLENECK FEATURES

Karel Veselý, Martin Karafiát, František Grézl, Miloš Janda and Ekaterina Egorova

Brno University of Technology, Speech@FIT and IT4I Center of Excellence,
Božetěchova 2, 612 66 Brno, Czech Republic

{iveselyk,karafiat,grezl,ijanda,xegoro00}@fit.vutbr.cz

ABSTRACT

In this paper we present novel language-independent bottleneck (BN) feature extraction framework. In our experiments we have used Multilingual Artificial Neural Network (ANN), where each language is modelled by separate output layer, while all the hidden layers jointly model the variability of all the source languages. The key idea is that the entire ANN is trained on all the languages simultaneously, thus the BN-features are not biased towards any of the languages. Exactly for this reason, the final BN-features are considered as language independent.

In the experiments with GlobalPhone database, we show that Multilingual BN-features consistently outperform Monolingual BN-features. Also, cross-lingual generalization is evaluated, where we train on 5 source languages and test on 3 other languages. The results show that the ANN can produce very good BN-features even for unseen languages, in some cases even better than if we trained the ANN on the target language only.

Index Terms— Language-Independent Bottleneck Features, Multilingual Neural Network

1. INTRODUCTION

While the Large Vocabulary Continuous Speech Recognition (LVCSR) for languages with abundant resources (such as US English) has reached certain maturity, fast development of LVCSR systems for new languages with limited resources is still a challenge. Techniques that are able to generalize across languages and to efficiently use data from a set of them to boost performance on a new one are now in the focus of the whole speech recognition community. The aim of this article is to face this challenging problem by creating universal discriminative bottleneck (BN) feature extractor, which can be directly applied to a new language.

This work was partly supported by the Intelligence Advanced Research Projects Activity (IARPA) BABEL program, by Czech Ministry of Trade and Commerce project No. FR-TI1/034, Technology Agency of the Czech Republic grant No. TA01011328, and by European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070). M. Karafiat was supported by Grant Agency of the Czech Republic post-doctoral project No. P202/12/P400.

In the past, many of the multi-lingual efforts were started by the group of Tanja Schulz. In [1], multi-lingual triphone-based acoustic model with cross-lingual phoneme set was created. The group has also invested huge efforts in collecting GlobalPhone database [2].

Another interesting approach to multi-lingual acoustic modelling is based on Subspace Gaussian Mixture Models (SGMM) [3]. Here, the state-dependent GMM models are factored to a subspace, which can be shared across languages. Similar intuition can be found in the Multi-lingual Artificial Neural Network (ANN) by Scanzio [4], here the hidden layers are shared across languages.

Our work is situated in the *Tandem LVCSR framework* [5], where a traditional Hidden Markov Model (HMM) based LVCSR system processes features generated by ANNs. In the recent past, the BN-features [6] have been proved to be beneficial for Tandem systems. Originally the BN-features were seen as language dependent, our objective is to make them universal and language-independent. In [7], we have presented a study of multi-lingual bottleneck features (obtained with unification of phoneme-sets or feature concatenation of several language-dependent ANNs), however, the multi-lingual Tandem systems did not outperform the mono-lingual baselines.

In fact, by observing the results in [1] [4] or [7], it actually seems that the upper bound of the accuracy of multi-lingual systems is given by the performance of mono-lingual systems. This is indeed true if there is sufficient amount of training data. In the case of limited training data, it is advisable to reuse some information from highly represented language(s) by the techniques like cross-language adaptation, bootstrapping [8], or SGMM [3]. For ANNs, adaptation to new language is possible [9][10]. Currently, this is still a very active research area, it is almost sure that new techniques will emerge.

In this work, inspired by [4], we applied the Multi-lingual ANN to produce bottleneck features. The focus of this paper is on the features, thus our GMM-HMM Tandem back-ends are strictly mono-lingual. The core idea of this article is that we create BN-feature space by training Multi-lingual ANN which is trained on all the languages simultaneously. Therefore the resulting BN-features are not biased towards any of

the source languages, which is exactly the reason why the final BN-features are language independent.

In section 2, we discuss the main characteristics of our model, in 2.1.1, we show in detail how to modify Mono-lingual ANN in order to obtain Multi-lingual ANN. The experimental setup is described in section 3. Finally, section 4 presents the results demonstrating the advantages of the Multi-lingual BN-features with respect to Mono-lingual BN-features. Note that section 4.3 shows clearly the capability of the proposed BN-features to generalize on unseen languages.

2. MULTI-LINGUAL NETWORK

The proposed model is 5-layer Multi-layer perceptron with sigmoid hidden units, linear bottleneck [11] and several output layers, where each language has associated its separate weights and softmax function. The structure of the model is shown in figure 1:

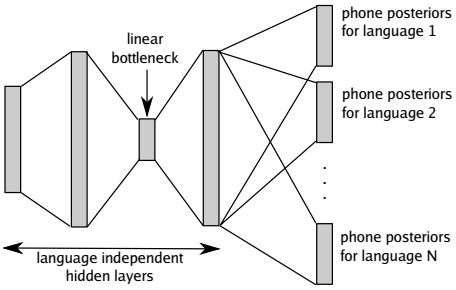

Fig. 1. Multi-lingual Bottleneck Network

From the acoustic modelling perspective, the network is split in two parts: 1) language independent hidden layers 2) language dependent output layers.

Due to the structure of the model, all the language dependent information is concentrated in the neurons which compute the output layer, while the rest of the network produces language independent features.

2.1. Training procedure

The proposed type of ANN can be trained efficiently by the *Stochastic Gradient Descent* algorithm. In our case, we used its optimized parallel implementation from our open-source toolkit *TNet*.

Particularly for this type of heterogeneous data, coming from several languages, it is extremely important to present the training samples at random. Practically speaking, our processing involves both the list-level and frame-level shuffling.

During the training, the Cross-Entropy criterion is optimized. This is done only within the posteriors of a single language, which corresponds to the actual speech frame. The hidden neurons are then trained by standard Backpropagation algorithm. Within the output layer, only the neurons of *"active language"* are trained by given datapoint. The gradient values of all the other output neurons are fixed to zero.

2.1.1. Interval-based Softmax

Although this model might seem complicated to implement, in fact, it is not. The problem can be solved in a very elegant way by interval-based Softmax function, which is able to detect the *"active language"*. During propagation, the posteriors of all the languages get evaluated on per-language basis according to the following Softmax formula:

$$y_i = \frac{\exp(a_i)}{\sum_{j=n_{l,s}}^{n_{l,e}} \exp(a_j)} \tag{1}$$

where a_i corresponds to activation value of i-th neuron; y_i is i-th ANN output; $n_{l,s}$ is *"starting index"* of given language l and $n_{l,e}$ is the *"ending index"*.

Given that one-hot encoding is used for targets t_i and that the derivative of Cross-Entropy wrt. activation a_i value equals to:

$$\frac{\partial E}{\partial a_i} = y_i - t_i \ , \tag{2}$$

we can test for the *"active language"* according to the criterion:

$$\sum_{i=n_{l,s}}^{n_{l,e}} \frac{\partial E}{\partial a_i} = 0 \ , \tag{3}$$

this condition holds only if both the sums of the posterior vector and the target vector are equal to one. Note that the target vector of the *"active language"* must contain single "1" element due to one-hot encoding, the rest of the vector are zeros.

The error derivatives $\partial E / \partial a_i$ of neurons corresponding to *non-active languages* are then forced to be zero, which also ensures zero gradients for these neurons.

Here, we should note that the linear part of all the output layers can be merged into a single linear transform. This greatly simplifies the implementation of the model.

2.1.2. Learning-rate scheduling

Besides the network structure, also the learning-rate scheduling algorithm had to be modified. The original New-Bob algorithm, which is based on frame-level classification accuracies, was modified to use the relative improvement of Cross-Entropy on held-out set as the decisive criterion.

Initially, the learning rate is kept fixed, unless the relative improvement gets smaller than 0.01. Since this point, the learning rate is halved on each epoch, unless the relative improvement is smaller than 0.001, which ends the training.

The ANN weight updates were performed per blocks of 512 frames with initial learning rate 1.0.

3. EXPERIMENTAL SETUP

Database The dataset comes from Multi-lingual database GlobalPhone [2]. The database covers 15 languages with an average of 20 hours of speech from about 100 native speakers per language. The following languages were selected for the experiments: Czech, German, Portuguese, Russian, Spanish, Turkish and Vietnamese. These languages were accompanied with English taken from Wall Street Journal. Table 1 contains phoneme-set sizes and dataset sizes, the data is the same as in [7]. The data were converted to 8kHz, 16-bit, linear PCM, mono format.

In some of the experiments, IPA mapping of phoneme-sets is used. The IPA mapping to 118 phonemes (incl. silence) was designed by a trained phonetician to represent only those qualities of speech that are distinctive across languages based on their perceptive characteristics.

Initial acoustic models The speech recognition system is based on HMM cross-word tied-states triphones. The initial acoustic models were trained from scratch using mixture-up training on mono-lingual training sets. The resulting models contained ≈2500 tied states and 18 Gaussian mixtures per state. The PLP features of 13 coefficients were expanded with derivatives Δ and Δ^2 which leads to 39 dimensional features. These were mean- and variance-normalized on speaker basis.

These baseline PLP systems were used to generate forced alignments for ANN training. Triphone labels were converted to 3-state monophone labels.

For the PLP-HLDA baseline, the 13 dimensional PLPs were expanded by Δ, Δ^2, Δ^3 to 52 dimensions, and then reduced to 39 dimensions by HLDA, which considers HMM states as classes. Also in this case, speaker-based mean- and variance-normalization was applied.

ANN Parameterization TRAPs-DCT features [12] were used as ANN input: The parameters are 15 log Mel-filterbank outputs derived with 25ms window, 10ms shift, and with per-utterance mean-normalization applied. In each band, a temporal context of 31 frames is taken, rescaled by Hamming window and compressed by Discrete Cosine Transform (DCT) with 16 basis (including C0). By concatenating all 15 per-band DCT-outputs, we obtain final feature space with 240 dimensions. The features were finally rescaled to have zero mean and unit variance.

ANN Topologies For all experiments, we used 5-layer Multi-Layer Perceptrons. The feature-producing bottleneck size is always 30, the input dimension is fixed to 240, the output dimensions depends on training targets. For mono-lingual networks, the dimension of 1st and 3rd hidden layer was chosen to have ANN with 1 million parameters. For multi-lingual networks, the hidden layer dimensions were fixed to 1141, to fix the parameter count of feature-extraction front-end.

ANN Initialization Weight matrices were initialized by $\mathcal{N}o\,(0;\,0.01)$, the biases of sigmoid units are samples from $\mathcal{U}\,(-4.1;\,-3.9)$. The biases of the linear and softmax units were set to zero.

Final system The BN-features produced by different ANNs were transformed by Maximum Likelihood Linear Transform (MLLT), which considers HMM states as classes. The transformed bottleneck features were mean- and variance-normalized.

New models were trained by single pass retraining from the PLP based *initial acoustic models*. Next, 12 maximum likelihood iterations followed to better settle down the Mono-lingual HMM-GMMs in the new feature space.

The test sets were decoded with bigram language models based on public newspaper data. More details about the language models and dictionaries are given in table 2, the setup is the same as in [7].

Table 1. *Phoneme-set sizes (incl. silence), dataset sizes in hours.*

Language	#phn	TRAIN [h]	DEV [h]	TEST [h]
German	42	13.2	1.8	1.3
Czech	41	26.8	1.2	1.9
English	40	14.2	1.0	1.0
Spanish	35	13.4	1.2	1.2
Portuguese	34	14.7	1.0	1.0
Russian	54	16.9	1.3	1.4
Turkish	30	12.0	1.6	1.4
Vietnamese	35	14.7	1.2	1.3
All	311	125.9	10.3	10.5

Table 2. *Detailed information about language models and test dictionaries for individual tasks.*

Language	OOV rate	Dict. size	LM corpus size	WWW server
German	1.92	375k	19M	www.faz.net
Czech	3.08	323k	7M	www.novinky.cz
English	2.30	20k	39M	WSJ - LDC2000T43
Spanish	3.10	135k	18M	www.aldia.cr
Portuguese	0.92	205k	23M	www.linguateca.pt/ cetenfolha
Russian	1.44	485k	19M	www.pravda.ru
Turkish	2.60	579k	15M	www.zaman.com.tr
Vietnamese	0.02	16k	6M	www.tintuconline.vn

4. RESULTS

Three sets of experiments have been performed, first, we trained Mono-lingual Bottleneck ANN for each target language separately, this is our baseline. Then, we trained three Multi-lingual networks with different ways to merge the language-specific information. Finally, we evaluated the cross-lingual generalization by training the ANN on 5 source languages and testing on 3 other target languages.

4.1. Baseline system

We defined three baselines, all the three are mono-lingual systems, the features are different: (I.) PLPs, which are by design language-independent, (II.) PLP-HLDA, which contain language-dependent linear transform, and finally (III.) language-dependent bottleneck features.

In table 3, we see that Mono-lingual Bottleneck Features (III.) mostly outperform the PLP-HLDA systems (II.), with exception of Spanish and Portuguese.

The cross-lingual generalization of Mono-lingual BN-features was also evaluated. As can be seen in table 4, the language mismatch between the BN-features and the GMM-HMM back-end results in WER degradation within range 0.4%-8.5% absolute. Very interesting is to compare the mismatched language pairs with the PLP-HLDA baseline (II.). Often, the PLP-HLDA systems perform better than Mono-lingual BN-feature systems with mismatched languages. This results show that the Mono-lingual BN-features do not generalize well on unseen languages.

In the next section, we will experiment with Language-Independent Bottleneck-Features. We might also be tempted to test Language-Independent PLP-HLDA Features, however these were already studied in our lab [13], showing that the 8-language HLDA performs about the same as the mono-lingual one, the small improvements were obtained in 5 cases out of 8. The improvement was never better than 0.4% absolute, ie. smaller than we get in the next section.

Table 3. *Baseline results [WER%] for PLP, PLP-HLDA and Mono-lingual Bottleneck-feature systems*

Language	WER with features		
	PLP (I.)	PLP-HLDA (II.)	Mono-lingual Bottleneck, (III.)
Czech	24.5	22.6	**19.7**
English	17.8	16.8	**15.9**
German	28.5	26.6	**25.5**
Portuguese	28.7	**27.0**	27.2
Spanish	25.1	**23.0**	23.2
Russian	35.4	33.5	**32.5**
Turkish	34.4	32.0	**30.4**
Vietnamese	30.2	27.3	**23.4**

Table 5. *Results [WER%] for different approaches to merge language-specific information.*

Language	WER with Multi-lingual Bottleneck features		
	ANN output layer		
	1-Softmax (lang-dep.) (a)	1-Softmax (IPA map.) (b)	8-Softmax (lang-dep.) (c)
# targets	933	354	933
Czech	20.3	19.4	**19.3**
English	16.1	15.5	**14.7**
German	25.9	24.8	**24.0**
Portuguese	27.2	25.6	**25.2**
Spanish	24.2	23.2	**22.6**
Russian	33.4	32.5	**31.5**
Turkish	31.3	30.3	**29.4**
Vietnamese	26.9	25.9	**24.3**

4.2. Multi-lingual Bottleneck Features

The multi-lingual information can be merged by Bottleneck-ANN by different approaches: (a) by simple concatenation of language-specific phoneme-sets, (b) by mapping to global phoneme-set based on IPA notation or (c) by Multi-lingual ANN [4] where each language has its output layer.

The first column (a) in table 5 corresponds to ANN with single output layer, where individual phoneme-sets with tagged language[1] were simply concatenated. This leads to performance degradation wrt. Mono-lingual BN-feature baseline (III.) for all the languages. The problem is that very similar phones from different languages are considered as different classes and part of the bottleneck "encoding capacity" is spoilt to discriminate them.

The second column (b) corresponds to ANN with single output layer, where the per-language phoneme sets are mapped to a global phoneme-set based on IPA notation, according to prior expert-knowledge of a phonetician. Here, the results are better, however the prior knowledge may not be always accurate, so the resulting phonemes may be "too disparate", while the bottleneck must encode them as single classes, which is again inefficient.

Finally the third (c) column corresponds to Multi-lingual ANN with eight output layers. Here the between-class competition of the phoneme-states is only within a single language. In this way we have effectively bypassed the issue of phoneme-set unification, and the bottleneck "encoding capacity" is finally used efficiently. From table 5 it is obvious that this model gives consistently better results than all the three baselines. Only in the case of Vietnamese the baseline in table 3 was 0.9% better.

If we compare the last column (III.) of the baseline table 3 with column (c) of the multi-lingual table 5, we see that

[1]Tagged for example like : English_A German_A Turkish_A ...

Table 4. *Cross-lingual mismatch of Mono-lingual Bottleneck Features; comparison with PLP-HLDA baseline*

ANN Language	Test-set language [WER%]							
	Czech	English	German	Portuguese	Spanish	Russian	Turkish	Vietnamese
Czech	**19.7**	16.3	26.6	27.6	25.1	33.7	32.0	29.2
English	21.9	**15.9**	27.4	29.2	26.1	35.9	33.8	30.2
German	21.9	17.6	**25.5**	29.7	27.3	36.3	35.1	31.9
Portuguese	21.4	17.4	27.9	**27.2**	24.7	34.8	32.7	28.4
Spanish	21.3	16.7	27.4	28.1	**23.2**	35.3	32.5	28.1
Russian	20.7	16.8	26.9	27.9	25.0	**32.5**	32.4	30.1
Turkish	22.0	17.4	28.0	29.4	25.1	35.8	**30.4**	28.8
Vietnamese	23.9	18.3	30.9	31.9	26.3	38.3	34.7	**23.4**
PLP-HLDA (II.)	22.6	16.8	26.6	**27.0**	**23.0**	33.5	32.0	27.3

by using more languages, we can observe a synergy effect, which leads to lower error rates. This might be caused by the fact that we use more training data for the ANN training. Also, from the same observation we can deduce, that there *definitely must exist some commonalities in the structure of speech patterns across the languages*, otherwise we would observe degradations rather than improvements, while adding more languages to the training set.

The ANNs corresponding to the first column (a) and the third column (c) have both 933 outputs, the difference is in grouping into languages via the Softmax function. The ANN from the second column (b) has 354 outputs due to mapping to common phoneme set. In all the cases the targets are three-state monophones.

Table 6. *Results [WER%] for cross-lingual generalization experiment with Language-Independent Bottleneck Features. The 5-Softmax ANN is trained on the first five languages, the unseen languages are Russian, Turkish and Vietnamese.*

Language	baselines		ANN output : 5-Softmax (lang-pooled) (d)
	PLP-HLDA (II.)	Mono-BN (III.)	
Czech	22.6	19.7	19.2
English	16.8	15.9	14.7
German	26.6	25.5	24.5
Portuguese	27.0	27.2	26.0
Spanish	23.0	23.2	23.0
Russian	33.5	32.5	**32.3**
Turkish	32.0	**30.4**	30.7
Vietnamese	27.3	**23.4**	26.8

4.3. Cross-lingual generalization

The previous promising results lead us to investigate into the cross-lingual generalization. In this experiment we trained the ANN on 5 source languages (Czech, English, German, Portuguese, Spanish) and tested on 3 other languages (Russian, Turkish, Vietnamese).

In table 6, we see that the cross-language generalization is very good. In the case of Russian, the 5-Softmax ANN system (d) outperformed the Mono-lingual BN-feature baseline (III.) by 0.2% absolute. Here we should clearly recall, that Russian plays role of unseen language. This unexpected result can be interpreted in the way, that for Russian, a better BN-feature extractor can be obtained by unification of feature-spaces from 5 other languages, rather than training solely on Russian data, which has *no precedent in the case of so far published BN-feature experiments*.

In the case of Turkish, there is a slight hit of 0.3%, which is still a very good result, if we consider that Turkish is unseen language. The improvement over PLP-HLDA baseline (II.) is still solid 1.3% absolute.

In the case of Vietnamese, the cross-language generalization is poorer, this may be caused by the fact that the tonal Vietnamese is very different from all the 5 source languages, which all come from the Indo-European family. Anyway, the performance is still 0.5% better than the Mono-lingual PLP-HLDA baseline (II.).

Very interesting is to compare the performance of 5 source languages with the 8-language system from column (c) in table 5. The slight degradation for German 0.5% Portuguese 0.8% and Spanish 0.4% shows us that the synergy effect is stronger when training on more languages and of course on more training data.

At this point it is also good to look back at table VIII in [7]. By comparing the results of the 3 unseen languages, we see an absolute improvement between 0.3% for Russian and 1.7% for Vietnamese.

5. CONCLUSIONS

The results that we have observed in all the previous experiments can be summarized as follows:

1. Multi-lingual ANN is an effective framework for obtaining Language-Independent Bottleneck Features.
2. The resulting Language-Independent Bottleneck Features consistently outperform both the PLP-HLDA and Mono-lingual Bottleneck-feature ones.
3. In order to merge internal structure of 8 languages to 1 feature space, it is more efficient to use Multilingual ANN with 8 output layers, rather than use simple phoneme-set concatenation or mapping to common phoneme-set based on IPA notation.
4. The key point is, that the network should not be biased to any of the source languages, which is assured by simultaneous training on all the languages.
5. The Language-Independent Bottleneck Features generalize well on unseen languages, if the languages are not very different.

In case of Russian as unseen language, these BN-features outperformed the mono-lingual network that was trained on Russian data only. Even if the unseen language was very different, as in the case of Vietnamese, the result was still better than PLP-HLDA baseline.

The results have also shown, that there definitely must exist some commonalities in the structure of speech patterns across the languages, which is in agreement with "common sense intuition".

With the Multi-lingual ANN, it is straightforward to use even more languages, however care should be taken to data balancing. In our case, the training sets were almost balanced: 12h-26h. If this was not true, a compensation by per-language learning rate could be used to prevent bias towards any of the source languages.

Further WER reductions might be possible by using hierarchical ANNs such as Universal Context Network, Convolutive Bottleneck Network [11] or Deep architectures.

The application of Multi-lingual ANN is not limited only to feature extraction. Similarly to [14], it can also be used to generate data-driven universal phoneme set. This can be done efficiently by accumulation of posterior-based multilingual confusion matrix. Universal phoneme recognizers have successful application for example in Language Identification [15].

Yet another very interesting application would be to use these Language-Independent Bottleneck Features in low-resourced language LVCSR experiments in Tandem with Multi-lingual SGMM-based [3] acoustic modelling. This will be subject of further experiments.

6. REFERENCES

[1] Tanja Schultz and Alex Waibel, "Development of Multilingual Acoustic Models in the GlobalPhone Project," in *Proc. TSD'1998*.

[2] Tanja Schultz, "GlobalPhone: A Multilingual Speech and Text Database Developed at Karlsruhe University," in *Proc. ICSLP'2002*.

[3] Liang Lu, Arnab Ghoshal, and Steve Renals, "Regularized Subspace Gaussian Mixture Models for Cross-lingual Speech Recognition," in *Proc. ASRU'2011*.

[4] Stefano Scanzio, Pietro Laface, Luciano Fissore, and al., "On the use of a Multilingual Neural Network Front-end," in *Proc. INTERSPEECH'2008*.

[5] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem Connectionist Feature Extraction for Conventional HMM Systems," in *Proc. ICASSP'00*.

[6] František Grézl, Martin Karafiát, Stanislav Kontár, and Jan Černocký, "Probabilistic and Bottle-Neck Features for LVCSR of Meetings," in *Proc. ICASSP'07*.

[7] František Grézl, Martin Karafiát, and Miloš Janda, "Study of Probabilistic and Bottle-Neck Features in Multilingual Environment," in *Proc. ASRU'2011*.

[8] Wheatley et al., "An Evaluation of Cross-language Adaptation for Rapid HMM Development in a new Language," in *Proc. ICASSP'1994*.

[9] N. Vu et al., "Multilingual Bottle-Neck Features and its Application for Under-resourced Languages," in *Proc. SLTU'12*.

[10] S. Thomas et al., "Multilingual MLP Features for Low-resource LVCSR Systems," in *Proc. ICASSP'12*.

[11] Karel Veselý, Martin Karafiát, and František Grézl, "Convolutive Bottleneck Network Features for LVCSR," in *Proc. ASRU'2011*.

[12] Petr Schwarz, Pavel Matějka, and Jan Černocký, "Towards Lower Error Rates In Phoneme Recognition," in *Proc. TSD'2004*.

[13] Martin Karafiát, Miloš Janda, Jan Černocký, and Lukáš Burget, "Region Dependent Linear Transforms in Multilingual Speech Recognition," in *Proc. ICASSP'2012*.

[14] Paul Dalsgaard, Ove Andersen, and William J. Barry, "Cross-language Merged Speech Units and Their Descriptive Phonetic Correlates," in *Proc. ICSLP'98*.

[15] A. Stolcke, M. Akbacak, and al., "Improving Language Recognition with Multilingual Phone Recognition and Speaker Adaptation Transforms," in *Proc. Odyssey'10*.