A LANGUAGE SPACE REPRESENTATION FOR SPEECH RECOGNITION

A. Ragni, M. J. F. Gales and K. M. Knill

Department of Engineering, University of Cambridge
Trumpington Street, Cambridge CB2 1PZ, UK
{ar527,mjfg,kate.knill}@eng.cam.ac.uk

ABSTRACT

The number of languages for which speech recognition systems have become available is growing each year. This paper proposes to view languages as points in some rich space, termed *language space*, where bases are *eigen*-languages and a particular selection of the projection determines points. Such an approach could not only reduce development costs for each new language but also provide automatic means for language analysis. For the initial proof of the concept, this paper adopts cluster adaptive training (CAT) known for inducing similar spaces for speaker adaptation needs. The CAT approach used in this paper builds on the previous work for language adaptation in speech synthesis and extends it to Gaussian mixture modelling more appropriate for speech recognition. Experiments conducted on IARPA Babel program languages show that such language space representations can outperform language independent models and discover closely related languages in an automatic way.

Index Terms— language space, cluster adaptive training, babel

1. INTRODUCTION

Recently there has been interest in developing speech recognition systems simultaneously for multiple languages [1, 2, 3, 4, 5, 6, 7, 8]. One example is the IARPA Babel program [9]. Building systems from scratch for each new language however is time consuming. This has promoted the use of schemes that may help to reduce development costs. One example are language independent approaches [8, 10]. The performance of these approaches may become unsatisfactory when the new language is not well represented by any of the training languages [8]. Though these approaches could alternatively be used for bootstrapping language dependent systems [10], the development cost remains high. This makes schemes capable of rapid adaptation to *any* given language of particular interest.

The concept of rapid adaptation is well known in speaker adaptation where schemes such as maximum likelihood linear regression (MLLR) [11, 12] and cluster adaptive training (CAT) [13] have been developed. In MLLR, a set of linear transforms are used to map an existing model set into a new adapted model set such that the likelihood of adaptation data is maximised [14]. The same concept could be applied in training, in the form of speaker adaptive training (SAT) [15], to factorise speaker variability from the model set [16]. Though a powerful approach, it treats training data as a single block [14] which may be suboptimal if multiple languages occur in the training data. In these situations the use of CAT schemes may be more appropriate. In CAT schemes [13, 17], training speakers are clustered together to form a relatively small number of clusters or *eigen*-speakers which combined would yield the adapted model set [14]. One popular way of combining clusters is to linearly interpolate the underlying cluster parameters [18, 17, 13]. As the parameters themselves can be clustered [19] this offers flexibility to adapt as few parameters as there are clusters [13].

The previous work with CAT has focused on sharing the underlying parameter tying structure, such as phonetic decision trees, among clusters. Though for speaker adaptation this may be reasonable [20], for language adaptation this is expected to be suboptimal as different *eigen*-languages may have different acoustic realisations of context-dependencies. This constraint was relaxed in [21, 22] by letting each cluster maintain its own set of trees. The use of CAT with cluster-dependent trees has been examined for rapid language adaptation in speech synthesis [23]. This paper extends that work to speech recognition. More generally, it tries to establish the possibility of representing languages as *points* in a rich *language space* where bases are *eigen*-languages and a particular projection into the space determines points. The use of CAT clusters and linear projection aims to facilitate the proof of this concept.

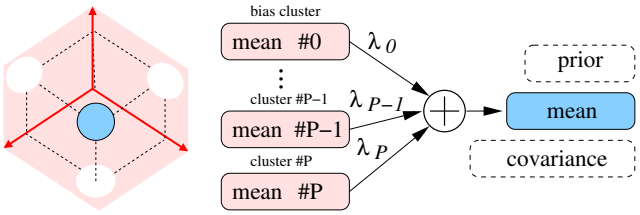
The rest of this paper is organised as follows. Section 2 discusses language spaces. Section 3 provides details on training and adaptation with the particular form of language space examined in this work. Section 4 presents experimental results. Section 5 provides conclusions and outlines future work.

2. LANGUAGE SPACE REPRESENTATION

Speech recognition systems have started to appear for an increasing number of languages. For instance, the IARPA Babel program has so far released resources for 17 languages. Given the wide range of the world's languages, it is clear that treating each language as a new task is becoming too costly to follow. This paper asks and attempts to answer the following question - is there a space where languages could be represented as points in that space? Figure 1 (a) shows an imaginary three dimensional language space employing linear projection with one point representing a language. An affirmative answer to this question may provide a systematic approach to speech recognition for a large number of the world's languages. In addition, it may provide insights into the nature of *eigen*-languages and contribute to the knowledge about language similarity.

For the initial investigation, this paper adopts cluster adaptive

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U. S. Army Research Laboratory (DoD/ARL) contract number W911NF-12-C-0012. The U. S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoD/ARL, or the U. S. Government.



(a) language space (b) cluster adaptive training

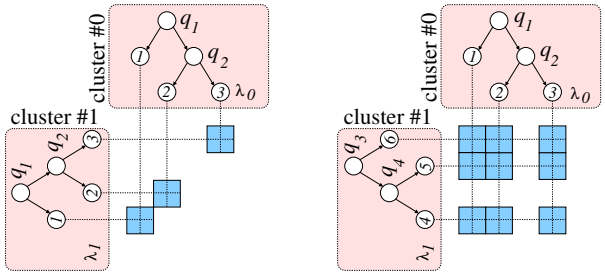
Fig. 1. Language space representation using cluster adaptive training.

training (CAT) [13, 23] to induce the language space and project languages into it. CAT can be viewed as a multiple-cluster scheme where the adapted model set is obtained by combining the clusters together [14]. There are many options to select the form of representation of the clusters and the combination method to employ [18, 17, 13, 24, 25]. One popular option is to use hidden Markov models (HMM) and apply linear interpolation to combine the mean parameters of Gaussian mixture model (GMM) state output distributions [13]. This is illustrated for one Gaussian component in Figure 1 (b) where $P+1$ cluster means are interpolated using $P+1$ weights to give the language adapted mean. For robust estimation it is important to tie these parameters appropriately. The weights could be tied among the components by clustering them using regression class trees [19] similar to the MLLR schemes [12]. The cluster means could be tied using phonetic decision trees [26]. Thus the language adapted mean in this work may be expressed as

$$\boldsymbol{\mu}_{l,m} = \sum_{p=0}^{P} \lambda_{l,r(m),p} \boldsymbol{\mu}_{c_p(m)} \tag{1}$$

where $r(m)$ is a weight regression class for component m, $c_p(m)$ is a cluster p tree leaf node to which m belongs. The priors and covariances usually are not interpolated in CAT and tied along with the means of one of the clusters such as $p=0$.

Phonetic decision trees in CAT are borrowed from a speaker-independent system and shared among clusters [13, 18, 17]. For language adaptation, where languages may have different acoustic realisation of context-dependencies, such an approach may not be advantageous [23]. One option to address this issue would be to borrow language-dependent trees. This however has a drawback of promoting languages to adopt "hard" rather than "soft" assignments to the clusters. Another option would be to re-build the trees during training thus letting each cluster have trees that best represent the current assignment [21, 23]. Figure 2 shows two language spaces with cluster-independent and cluster-dependent trees. The



(a) cluster-independent (b) cluster-dependent

Fig. 2. Intersection of cluster dependent and independent trees.

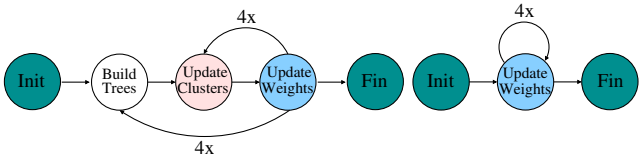
difference between these approaches lies in the number of language adapted means at the intersection of tree leaves (shaded squares). For the cluster-dependent tree language space in Figure 2 (b), any pair of leaves could be selected as the topology and questions need not be the same. Such an inter-dependency between language adapted means however makes cluster parameter estimation and tree building more complicated than in the original CAT approaches [13, 17] in Figure 2 (a). Another issue is that standard clustering procedures assume a Gaussian distribution in the leaves [26] whereas GMMs are commonly used in speech recognition. Although mixing-up procedures [27] could be applied they are costly to perform. This paper proposes to maintain GMMs in the first $p=0$ cluster, termed the *bias cluster* [13], and Gaussians in the rest. This requires no modification to the form of the language adapted mean in equation (1) if GMM bias cluster tree leaves are split into the constituent Gaussians. The bias cluster trees can be borrowed from language independent systems and only non-bias cluster trees need to be built.

3. TRAINING AND ADAPTATION

The language space parameters $\mathcal{M}$ comprise HMM transition probabilities, covariances, priors, cluster means and weights. For estimation it is possible to use the maximum likelihood (ML) criterion. The auxiliary function is given by [23]

$$\mathcal{Q}(\mathcal{M};\hat{\mathcal{M}}) = C + \mathcal{Q}(\mathcal{M};\hat{\mathbf{a}},\hat{\mathbf{c}}) - \frac{1}{2}\sum_{l,t,m}\gamma_{l,t,m}\left\{(\mathbf{o}_{l,t}-\hat{\boldsymbol{\mu}}_{l,m})^{\mathsf{T}}\hat{\boldsymbol{\Sigma}}_m^{-1}(\mathbf{o}_{l,t}-\hat{\boldsymbol{\mu}}_{l,m}) + \log(|\hat{\boldsymbol{\Sigma}}_m|)\right\} \tag{2}$$

Estimation is performed by interleaving optimisation of clusters and weights. Prior to each iteration the trees associated with non-bias clusters can be re-built as shown in Figure 3 (a). A language in-



(a) training (b) adaptation

Fig. 3. Language space training and adaptation flow diagrams.

dependent system may be used to initialise transition probabilities, priors, covariances and bias cluster means [23]. Each language $l = 1,\ldots,L$ defines a new cluster. Trees for non-bias clusters are initialised to have zero mean root nodes. Points for all languages are initialised to zero for every cluster apart from the bias and the corresponding language, i.e., $\lambda_{l,r,p}=1$ if $p=0$ or $p=l$ and zero otherwise. Such an initialisation ensures that the language space system initially yields the same log-likelihood as the language independent system. For rapid adaptation it is possible to estimate only the point associated with the new $L+1$ language whilst keeping clusters and trees fixed to those obtained during training as shown in Figure 3 (b). The point is initialised to have zero elements for all but the bias cluster, i.e., $\lambda_{L+1,r,p}=1$ for $p=0$ and zero otherwise.

The rest of this section will discuss estimation of the clusters in Section 3.1, weights in Section 3.2 and tree building in Section 3.3.

3.1. Clusters

The cluster parameters include means $\boldsymbol{\mu}=\{\boldsymbol{\mu}_i\}$, covariances $\boldsymbol{\Sigma}=\{\boldsymbol{\Sigma}_k\}$, priors $\mathbf{c}$ and transition probabilities $\mathbf{a}$.

For optimising means, the auxiliary function in equation (2) may be simplified as [23]

$$\mathcal{Q}(\boldsymbol{\mu}, \hat{\boldsymbol{\mu}}) = C - \frac{1}{2} \sum_{m,p} \Big(\hat{\boldsymbol{\mu}}_{c_p(m)}^{\mathsf{T}} \mathbf{G}_{p,p,m} \hat{\boldsymbol{\mu}}_{c_p(m)} + 2 \sum_{b \neq p} \hat{\boldsymbol{\mu}}_{c_p(m)}^{\mathsf{T}} \mathbf{G}_{p,b,m} \hat{\boldsymbol{\mu}}_{c_b(m)} - 2 \hat{\boldsymbol{\mu}}_{c_p(m)}^{\mathsf{T}} \mathbf{k}_{p,m} \Big) \quad (3)$$

where $\mathbf{G}_{p,b,m}$ and $\mathbf{k}_{p,m}$ are statistics given by

$$\mathbf{G}_{p,b,m} = \sum_{l,t} \gamma_{l,t,m} \lambda_{l,r(m),p} \boldsymbol{\Sigma}_m^{-1} \lambda_{l,r(m),b} \quad (4)$$

$$\mathbf{k}_{p,m} = \sum_{l,t} \gamma_{l,t,m} \lambda_{l,r(m),p} \boldsymbol{\Sigma}_m^{-1} \mathbf{o}_{l,t} \quad (5)$$

Due to the inter-dependency imposed by cluster-dependent trees on language adapted means (see Figure 2 (b)), all cluster mean parameters should be estimated simultaneously [23]

$$\mathbf{G} \hat{\boldsymbol{\mu}} = \mathbf{k} \quad (6)$$

where $\mathbf{G}$ and $\mathbf{k}$ are block matrix and vector with elements given by

$$\mathbf{G}_{i,j} = \sum_{\substack{p,b,m: \\ c_p(m)=i \\ c_b(m)=j}} \mathbf{G}_{p,b,m}, \qquad \mathbf{k}_i = \sum_{\substack{p,m: \\ c_p(m)=i}} \mathbf{k}_{p,m} \quad (7)$$

The across-cluster statistics $\mathbf{G}_{i,j}$ serves to measure the usefulness of node j in cluster b to node i in cluster p. The more useful nodes are to each other the less across-cluster statistics are equal to $\mathbf{0}$. Though dimensionality of equation (6) is expected to be high, the statistics matrix $\mathbf{G}$ is sparse and hence it can be solved using sparse linear routines [28]. The degree of sparsity thus can be used to quantify how much of the intersect space, such as in Figure 2 (b), is covered.

The covariance parameters are updated using [23]

$$\boldsymbol{\Sigma}_k = \frac{\sum_{\substack{l,t,m: \\ c_0(m)=k}} \gamma_{l,t,m} (\mathbf{o}_{l,t} - \boldsymbol{\mu}_{l,m})(\mathbf{o}_{l,t} - \boldsymbol{\mu}_{l,m})^{\mathsf{T}}}{\sum_{\substack{l,t,m: \\ c_0(m)=k}} \gamma_{l,t,m}} \quad (8)$$

The CAT estimates for priors and transition probabilities are similar to the standard ML estimates [13].

3.2. Weights

For optimising cluster weights, the auxiliary function in equation (2) can be expressed as [13]

$$\mathcal{Q}(\boldsymbol{\lambda}, \hat{\boldsymbol{\lambda}}) = C + \sum_{l,r} \hat{\boldsymbol{\lambda}}_{l,r} \mathbf{s}_{l,r} - \frac{1}{2} \hat{\boldsymbol{\lambda}}_{l,r}^{\mathsf{T}} \mathbf{Z}_{l,r} \hat{\boldsymbol{\lambda}}_{l,r} \quad (9)$$

where $\mathbf{Z}_{l,r}$ and $\mathbf{s}_{l,r}$ are statistics given by

$$\mathbf{Z}_{l,r} = \sum_{m \in r} \mathbf{M}_m^{\mathsf{T}} \boldsymbol{\Sigma}_m^{-1} \mathbf{M}_m \sum_t \gamma_{l,t,m} \quad (10)$$

$$\mathbf{s}_{l,r} = \sum_{m \in r} \gamma_{l,t,m} \mathbf{M}_m^{\mathsf{T}} \boldsymbol{\Sigma}_m^{-1} \sum_t \gamma_{l,t,m} (\mathbf{o}_{l,t} - \mathbf{b}_m) \quad (11)$$

If bias cluster is used as in this work then $\mathbf{M}_m = \begin{bmatrix} \boldsymbol{\mu}_{1,m} & \dots & \boldsymbol{\mu}_{P,m} \end{bmatrix}$ and $\mathbf{b}_m = \boldsymbol{\mu}_{0,m}$ otherwise $\mathbf{M}_m = \begin{bmatrix} \boldsymbol{\mu}_{0,m} & \dots & \boldsymbol{\mu}_{P,m} \end{bmatrix}$ and $\mathbf{b}_m = \mathbf{0}$. Differentiating equation (9) with respect to cluster weight vector associated with language l and regression class r, equating to zero yields [13]

$$\mathbf{Z}_{l,r} \hat{\boldsymbol{\lambda}}_{l,r} = \mathbf{s}_{l,r} \quad (12)$$

This can be solved using standard matrix inversion since $\mathbf{Z}_{l,r}$ is expected to be dense and have a low dimensionality.

3.3. Decision trees

Due to the inter-dependency between language adapted means, the tree building should be performed for all clusters simultaneously. As this is computationally expensive [21], an interleaving scheme could be adopted where trees are built only for one of the clusters at a time. Under the assumptions used for the conventional tree building [26], the total log-likelihood associated with cluster p node k excluding constant terms can be calculated as [23]

$$\ell(k) = -\frac{1}{2} \sum_{\substack{m: \\ c_p(m)=k}} \Big(\boldsymbol{\mu}_{c_p(m)}^{\mathsf{T}} \mathbf{G}_{p,p,m} \boldsymbol{\mu}_{c_p(m)} + 2 \sum_{b \neq p} \boldsymbol{\mu}_{c_p(m)} \mathbf{G}_{p,b,m} \boldsymbol{\mu}_{c_b(m)} - 2 \boldsymbol{\mu}_{c_p(m)} \mathbf{k}_{p,m} \Big) \quad (13)$$

Since all mean vectors $\boldsymbol{\mu}_{c_p(m)}$ associated with node k will be tied to give $\boldsymbol{\mu}_k$ the above expression can be simplified as [23]

$$\ell(k) = \frac{1}{2} \hat{\boldsymbol{\mu}}_k^{\mathsf{T}} \left(\sum_{\substack{m: \\ c_p(m)=k}} \mathbf{G}_{p,p,m} \right) \hat{\boldsymbol{\mu}}_k \quad (14)$$

where

$$\hat{\boldsymbol{\mu}}_k = \left(\sum_{\substack{m: \\ c_p(m)=k}} \mathbf{G}_{p,p,m} \right)^{-1} \sum_{\substack{m: \\ c_p(m)=k}} \left(\mathbf{k}_{p,m} - \sum_{b \neq p} \mathbf{G}_{p,b,m} \boldsymbol{\mu}_{c_b(m)} \right) \quad (15)$$

is the ML estimate of $\boldsymbol{\mu}_k$ under the assumption that all parameters associated with other trees remain unchanged [23]. The question q splitting node k components into a yes $k_+(q,k)$ and no $k_-(q,k)$ subset that results in the largest gain in the total log-likelihood

$$\Delta \ell = \ell(k_+(q,k)) + \ell(k_-(q,k)) - \ell(k) \quad (16)$$

is selected to split the node k. The procedure can be terminated using cross-validation [23], information theoretic criteria [29], or a heuristic threshold on the gain in the total log-likelihood [26].

4. EXPERIMENTS

This section describes the setup and presents experimental results with the language space approach examined in this work.

4.1. Setup

Limited language packs (LLP) released within the IARPA Babel program are used for evaluation. Table 1 provides a summary of the 11 languages used. To be consistent with previous work in [8], the same set of 7 held-in languages is adopted. The remaining 4 languages serve as the held-out languages. Each LLP contains roughly 10 hours of training data, an equivalent amount of development data, X-SAMPA phone set and lexicon. Full language pack language models are used to minimise the impact of non-acoustic phenomena.

Two baseline configurations are selected: language-independent (LI) and language-dependent (LD) tandem [30, 31] systems [8, 9]. The topology of the multi-layer perceptron (MLP) for the LI system is adjusted to accommodate roughly 7 times more output layer units than the 1000 units used for LD systems. The MLP is trained using the cross-entropy criterion to produce 26 dimensional bottleneck features [30] which are appended to perceptual linear prediction coefficients (PLP) [32] and pitch [33] to yield observation vectors. A

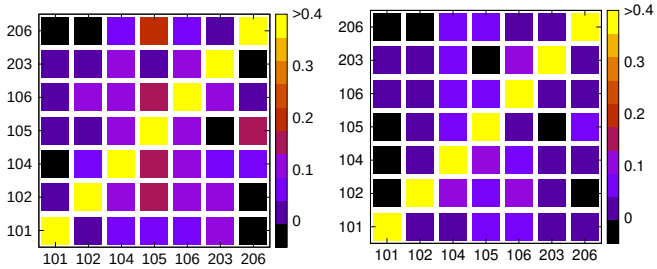
Held	Language	Code	Release
in	Cantonese	101	IARPA-babel101b-v0.4c
	Assamese	102	IARPA-babel102b-v0.5a
	Pashto	104	IARPA-babel104b-v0.4bY
	Turkish	105	IARPA-babel105b-v0.4
	Tagalog	106	IARPA-babel106b-v0.2g
	Lao	203	IARPA-babel203b-v3.1a
	Zulu	206	IARPA-babel206b-v0.1d
out	Bengali	103	IARPA-babel103b-v0.4b
	Vietnamese	107	IARPA-babel107b-v0.7
	Creole	201	IARPA-babel201b-v0.2b
	Tamil	204	IARPA-babel204b-v1.1b

Table 1. A summary of held-in and held-out languages.

heteroscedastic linear discriminant analysis (HLDA) [34] and global semi-tied covariance (STC) transforms [35] are used to de-correlate PLP and BN and pitch coefficients, respectively. The language space (LS) system is initialised from the LI system as discussed in Section 3. A single weight vector is used for each language. A total of 4 training cycles is performed, each consisting of tree building followed by 5 estimations of clusters and weights. For the held-out languages, adaptation consists of 4 estimations of the corresponding weights only using alignments from LI system.

4.2. Results

Prior to reporting results it is interesting to examine language spaces that can be learned with this approach. Figure 4 illustrates two language spaces obtained with tree building thresholds of 1000 and 500 respectively. Each column represents one of the clusters and



(a) threshold 1000 (b) threshold 500

Fig. 4. Cluster weights for held-in languages obtained with tree building threshold (a) 1000 and (b) 500.

is marked by the code of the language used for initialisation. It can be seen that the clusters are centred around the languages used for initialisation and make little use of other languages' data. This is expected since held-in languages are quite distinct from each other. It can be also seen that the use of a tighter threshold leads to even less usage of other languages' data. The number of non-bias cluster leaf nodes in the language spaces is 1031 and 3079 respectively which represents 15% and 44% of the number of the bias cluster leaves. The degrees of sparsity in the final systems are close to 100% which is consistent with the language space representations in Figure 4.

It is also interesting to examine the points that can be estimated for the held-out languages. Figure 5 illustrates these points for the language space in Figure 4 (b). These points suggest that two Indian languages in the held-out set, Bengali (103) and Tamil (204), will mostly benefit from the only Indian language in the held-in set, Assamese (102). Whilst both Bengali (103) and Tamil (204) benefit from Assamese (102), it is interesting to note that the language

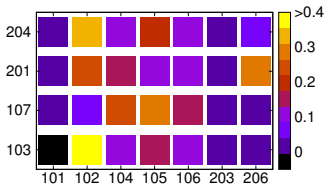


Fig. 5. Cluster weights for held-out languages obtained with the language space in Figure 4 (b).

which is from the same family, Bengali, derives the most contribution. Apart from Cantonese (101) and Lao (203) all held-in languages seem to be useful for these held-out languages.

The LD, LI and LS systems in Figures 4 (b) and 5 are evaluated on two held-in, Assamese (102) and Lao (203), and one held-out, Creole (201), languages. The performance of the LI system on the remaining three held-out languages does not go below 90% error rate making any results unreliable. Table 2 shows speech recognition results including decoding with bigram language model (BG), lattice rescoring with trigram language model (+TG) and confusion network rescoring (+CN) in terms of word error rate (WER). For the

Held	Language	Code	System	WER (%)		
				BG	+TG	+CN
in	Assamese	102	LI	73.0	72.4	70.2
			LS	72.7	72.2	69.9
			LD	73.3	73.0	70.8
	Lao	203	LI	66.4	65.7	64.5
			LS	65.7	64.9	64.0
			LD	69.1	68.5	67.4
out	Creole	201	LI	86.0	85.4	83.6
			LS	85.6	84.8	82.8
			LD	68.7	67.7	65.6

Table 2. Recognition results with held-in and held-out languages.

held-in languages, the LS system shows consistent yet small gains over LI and LD systems. This suggests the need for building more powerful language space systems. For the held-out language, the LS system shows small gains over the LI system, however, both LI and LS lag far behind the LD system. This illustrates the need for a more powerful adaptation to accommodate larger mismatches between seen and unseen languages, such as the one discussed in [23] where a new basis can be introduced to considerably enhance language adaptation limited in this work to only 7 free parameters.

5. CONCLUSIONS

This paper has discussed the possibility of representing the world's languages as points in a language space to enable efficient bootstrapping and rapid adaptation of speech recognition systems to any language. As a proof of the concept it has adopted cluster adaptive training (CAT) to induce language spaces and the use of linear interpolation between cluster mean vectors to project languages into the space. Experiments conducted on IARPA Babel program languages showed that such an approach is capable of automatically discovering related languages and exceeding language independent performance levels. It is expected that more powerful representations, CAT and non-CAT, will permit more expressive language spaces. In addition, more complex adaptation approaches are hoped to reduce the current gap between language space and language dependent system performance for unseen languages.

6. REFERENCES

[1] L. Burget, P. Schwarz, M. Agarwal, P. Akyazi, K. Feng, A. Ghoshal, O. Glembek, N. K. Goel, M. Karafiát, D. Povey, A. Rastrow, R. C. Rose, and S. Thomas, "Multilingual acoustic modeling for speech recognition based on subspace Gaussian mixture models," in *ICASSP*, 2010, pp. 4334–4337.

[2] S. Thomas, S. Ganapathy, and H. Hermansky, "Cross-lingual and multi-stream posterior features for low-resource LVCSR systems," in *Interspeech*, 2010, pp. 877–880.

[3] H. Bourlard, J. Dines, M. Magimai-Doss, P. N. Garner, D. Imseng, P. Motlicek, H. Liang, L. Saheer, and F. Valente, "Current trends in multilingual speech processing," *Sadhana*, vol. 36, no. 5, pp. 885–915, 2011.

[4] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *ICASSP*, 2013, pp. 7319–7323.

[5] Z. Tüske, J. Pinto, D. Wilett, and R. Schlüter, "Investigation on cross- and multilingual MLP features under matched and mismatched acoustical conditions," in *ICASSP*, 2013, pp. 7349–7353.

[6] K. Vesely, M. Karafiat, F. Grezl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *SLT*, 2012, pp. 336–341.

[7] N. T. Vu and T. Schultz, "Multilingual multilayer perceptrons for rapid language adaptation between and across language families," in *Interspeech*, 2013, pp. 515–519.

[8] K.M. Knill, M.J.F. Gales, A. Ragni, and S.P. Rath, "Language independent and unsupervised acoustic models for speech recognition and keyword spotting," in *Interspeech*, 2014.

[9] M. J. F. Gales, K. M. Knill, A. Ragni, and S. P. Rath, "Speech recognition and keyword spotting for low-resource languages: BABEL project research at CUED," in *SLTU*, 2014.

[10] T. Schultz and A. Waibel, "Language-independent and language-adaptive modeling for speech recognition," *Speech Communication*, vol. 35, pp. 31–51, 2001.

[11] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[12] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

[13] M. J. F. Gales, "Cluster adaptive training of hidden Markov models," *IEEE Tran SAP*, vol. 8, no. 4, pp. 417–428, 2000.

[14] M. J. F. Gales and S. J. Young, "The application of hidden Markov models in speech recognition," *Foundations and trends in signal processing*, vol. 1, no. 3, pp. 195–304, 2007.

[15] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *ICSLP*, 1996, pp. 1137–1140.

[16] M. J. F. Gales, "Acoustic factorisation," in *ASRU*, 2001, pp. 77–80.

[17] R. Kuhn, P. Nguyen, J.-C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation," in *ICSLP*, 1998, vol. 5, pp. 1771–1774.

[18] T. J. Hazen and J. R. Glass, "A comparison of novel techniques for instantaneous speaker adaptation," in *Eurospeech*, Rhodes, Greece, 1997, pp. 2047–2050.

[19] M. J. F. Gales, "The generation and use of regression class trees for MLLR adaptation," Tech. Rep. CUED/F-INFENG/TR263, Cambridge University, 1996.

[20] K. Yu and H. Xu, "Cluster adaptive training with factorized decision trees for speech recognition," in *Interspeech*, 2013, pp. 1243–1247.

[21] H. Zen and N. Braunschweiler, "Context-dependent additive $\log f_0$ model for HMM-based speech synthesis," in *ICSLP*, 2009, pp. 2091–2094.

[22] K. Yu, H. Zen, F. Mairesse, and S. Young, "Context adaptive training with factorized decision trees for HMM-based statistical parametric speech synthesis," *Speech communication*, vol. 53, pp. 914–923, 2011.

[23] H. Zen, N. Braunschweiler, S. Buchholz, M. J. F. Gales, K. Knill, S. Krstulovic, and J. Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *IEEE Tran ASLP*, vol. 20, no. 6, pp. 1713–1724, 2012.

[24] V. Diakoloukas and V. Digalakis, "Maximum likelihood stochastic transformation adaptation of hidden Markov models," *IEEE Tran SAP*, vol. 7, no. 2, pp. 177–187, 1999.

[25] B. Mak, J. T. Kwok, and S. Ho, "Kernel eigenvoice speaker adaptation," *IEEE Tran SAP*, vol. 13, pp. 984–992, 2005.

[26] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *ARPA HLT*, 1994, pp. 307–312.

[27] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book (for HTK Version 3.4.1)*, University of Cambridge, `http://htk.eng.cam.ac.uk`, 2009.

[28] T. A. Davis, *Fundamentals of algorithms. Direct methods for sparse linear systems*, SIAM, 2006.

[29] K. Shinoda and T. Watanabe, "Acoustic modelling based on the MDL criterion for speech recognition," in *Eurospeech*, 1997, pp. 99–102.

[30] F. Grézl, M. Karafiát, S. Kontár, and Černocky. J., "Probabilistic and bottle-neck features for LVCSR of meetings," in *ICASSP*, 2007, vol. IV, pp. 757–760.

[31] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ICASSP*, 2000, vol. 3, pp. 1635–1638.

[32] H. Hermansky, "Perceptual Linear Predictive (PLP) analysis of speech," *ASA*, vol. 87, no. 4, pp. 1738–1752, 1990.

[33] D. Talkin, "A robust algorithm for pitch tracking (RAPT)," in *Speech Coding and Synthesis*, chapter 14, pp. 495–518. Elsevier Science B. V., 1995.

[34] N. Kumar, *Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition*, Ph.D. thesis, Johns Hopkins University, 1997.

[35] M. J. F. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Tran SAP*, vol. 29, pp. 82–97, 2012.