Dialect and Accent Recognition using Phonetic-Segmentation Supervectors

Fadi Biadsy, Julia Hirschberg*, Daniel P. W. Ellis†*

*Department of Computer Science, Columbia University, New York, NY, USA
{fadi,julia}@cs.columbia.edu
† Department of Electrical Engineering, Columbia University, New York, NY, USA
dpwe@ee.columbia.edu

Abstract

We describe a new approach to automatic dialect and accent recognition which exceeds state-of-the-art performance in three recognition tasks. This approach improves the accuracy and substantially lower the time complexity of our earlier phonetic-based kernel approach for dialect recognition. In contrast to state-of-the-art acoustic-based systems, our approach employs phone labels and segmentation to constrain the acoustic models. Given a speaker's utterance, we first obtain phone hypotheses using a phone recognizer and then extract GMM-supervectors for each phone type, effectively summarizing the speaker's phonetic characteristics in a single vector of phone-type supervectors. Using these vectors, we design a kernel function that computes the phonetic similarities between pairs of utterances to train SVM classifiers to identify dialects. Comparing this approach to the state-of-the-art, we obtain a 12.9% relative improvement in EER on Arabic dialects, and a 17.9% relative improvement for American vs. Indian English dialects. We also see a 53.5% relative improvement over a GMM-UBM on American Southern vs. Non-Southern English.

1. Introduction

Recent years have seen increasing interest in automatic identification of regional dialects and accents. Despite much progress on language recognition, dialect recognition is considered to be an even more challenging problem, since dialects of the same language are presumably far more similar than distinct languages. There are many important applications for dialect recognition: Identifying the dialect prior to ASR will enable the system to adapt its pronunciation, acoustic, and language models appropriately. Dialect recognition is also useful for identifying a speaker's regional origin and ethnicity and helpful in speech-to-speech translation and forensic speaker profiling.

Previous approaches in phonotactic-based dialect recognition systems – such as Phone Recognition followed by Language Modeling (PRLM) – have been shown to be effective in identifying languages and dialects (e.g., [1, 2, 3]). Gaussian Mixture Models–Universal Background Model (GMM-UBM) with Shifted Delta Cepstra (SDC) have also achieved considerable success in speaker and language/dialect recognition [4]. Discriminative training has proven important in recent dialect recognition systems (e.g., [5]). The combination of diverse phonotactic-based systems with a GMM approach achieves very good results on Arabic dialect recognition [6]. Prosodic modeling has also shown to be useful [7].

In this paper we describe a new approach to dialect identification which outperforms previous approaches on several dialect and accent recognition tasks. This system also improves in accuracy and time complexity over our own previous systems [8].

We developed this approach for the task of distinguishing Arabic dialects. We have also evaluated the generality of the approach through testing on two English dialect/accent tasks: American English vs. Indian English, and American Southern vs. Non-Southern English. Our new dialect recognition approach and its evaluation on Arabic are presented in Section 2. We describe our experiments on English in Section 3. Finally, in Section 4, we conclude and describe our future work.

2. Dialect Recognition Approach

After front-end pre-processing, the first stage in our approach is to use a phone recognizer to obtain the most likely phone sequence hypothesis for each utterance in the training corpus. We then extract temporally-aligned acoustic feature vectors for each phone instance in the sequence. We train a GMM-UBM for each phone type using all frames from all instances of that phone type from all dialects. We denote this GMM-UBM as *phone GMM-UBM*. In this work, all GMMs are Maximum-Likelihood (ML) -trained using the EM algorithm.

2.1. Phone-Instance Supervectors and their Kernel

In our previous work [8, 9], we model the acoustic-phonetic differences across dialects at the phone level. In particular, we extract a vector that captures these differences for each phone in the hypothesized phone sequence. To do that, we adopt the GMM-supervector representation [10] — but at the level of phone instances. We use the acoustic frames of each phone instance to perform MAP (Maximum A-Posteriori) adaptation of the corresponding phone GMM-UBM. We adapt only the means of the Gaussians using a relevance factor of $r = 0.1$. We denote the resulting GMM as the *adapted phone-GMM*. The intuition is that the modified means of the adapted phone-GMM 'summarize' the variable number of frames in a particular phone instance with a fixed-size representation.

This analysis yields a set of supervectors v_i for the i^{th} phone in an utterance, where ϕ_i is the identity of that phone. Thus, an utterance U is represented as a sequence of tuples $S_U = \{(v_i, \phi_i)\}_{i=1}^{n}$, where n is the number of phones in U. Note that our representation retains the dependency between the phone identity and its supervector.

2.1.1. Designing a Phone-Instance-Based Kernel

From the sequences of tuples S_U produced for the utterances U of the training corpora, we train an SVM classifier for each pair of dialects to distinguish one from the other. We design a kernel function to compute the similarity between pairs of utterances U_a and U_b. Let $S_{U_a} = \{(v_i, \phi_i)\}_{i=1}^{n}$ and $S_{U_b} = \{(u_j, \psi_j)\}_{j=1}^{m}$ be the tuple sequences of U_a and U_b, respectively. Our kernel function is defined in (1), where Φ is the phone inventory:

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi \in \Phi} \sum_{i:\phi_i=\phi} \sum_{j:\psi_j=\phi} e^{-\|v_i - u_j\|^2 / 2\sigma^2} \quad (1)$$

This function computes the sum of RBF kernels between every pair of supervectors of phone instances with the same type across the two utterances. It is straightforward to show that this kernel is positive definite, satisfying the Mercer condition. We term this approach Kernel-GMM-Instance.

Employing the kernel functions above, we first compute a kernel matrix for each pair of dialects using the tuple sequences extracted for all our training utterances. Next we train a standard binary SVM classifier for each pair of dialects using the pair's kernel matrix. The regularization parameter C and σ (in the kernel function (1)) are tuned by 10-fold cross-validation on the training data. Thus, for our four-way Arabic dialect experiment, we train a total of $\binom{4}{2} = 6$ binary classifiers.

2.1.2. *Evaluation and Time Complexity*

We evaluate this kernel on four Arabic dialects. Our Arabic dialect data is taken from spontaneous telephone conversations from the following Appen corpora: Iraqi Arabic (478 speakers), Gulf (976), and Levantine (985). We use 80% of the speakers from each corpus for training and hold out the remaining 20% for testing. We equalize the percentage of test speakers in each of the categories female/male and landline/mobile to avoid a bias towards the prior distributions of these categories during testing. We also test our system on Egyptian Arabic. We use the 280 speakers in CallHome Egyptian and its supplement for training. To test our system under different acoustic conditions, we employ 120 speakers from CallFriend Egyptian for testing. In this paper, we present results from testing our system on 30 s cuts. Each cut consists of consecutive speech segments totaling 30 s in length (after removing silence). Multiple cuts are extracted from each speaker. For Iraqi, we have 477 such test cuts, and 801, 818, and 1912 test cuts for Gulf, Levantine, and Egyptian, respectively. We adopt the NIST language/dialect recognition evaluation framework to report detection results instead of identification. We report our results using Detection Error Tradeoff (DET) figure and Equal Error Rate (EER). To plot an overall DET, our results are pooled across each pair of dialects with dialect priors equalized to discount the impact of different number of per-dialect test trials.

This system uses a trigram context-dependent phone recognizer trained on modern standard Arabic, trained with IBM's Attila system [11]. For each phone type, a phone GMM-UBM is trained with 100 Gaussian components using aligned 40D PLP frames (resulting from Linear Discriminant Analysis of 9 stacked frames) with cepstral mean and variance normalization and fMLLR adaptation (see [9]). We report the pooled DET curve in Figure 1; the EER of our Kernel-GMM-Instance is 4.94%. We can see in this Figure that this kernel approach significantly outperforms several earlier approaches: a standard trigram-based PRLM (EER 17.7%); a standard GMM-UBM (15.3%), our previous GMM-UBM-fMLLR (11.0%), and our recent Discriminative Phonotactics approach (6.0%). The details of all of these approaches are in [9]. The standard GMM-SVM [12] using our fMLLR-transformed features, with the same settings of GMM-UBM-fMLLR, yields an EER of 6.9%.

We now calculate the time complexity of Kernel-GMM-Instance to compute the kernel function between each pair of training utterances. Let m_k^j be the number of instances of phone type k (where $1 \leq k \leq |\Phi|$) in utterance U_j. For simplicity, we assume that the supervectors of every phone type have the same size, D; in our experiments, D = 2,340 to 4,000. Denoting $M = max_{j,k}\{m_k^j\}$, and assuming that computing the Euclidian distance between two D-dimensional vectors takes $\mathcal{O}(D)$, the time complexity of comparing a pair of utterances using the kernel function (1) is upper-bounded by $\mathcal{O}(M^2|\Phi|D)$. For example, the average of the frequency of the most frequent phone \a\ in our Arabic data across all utterances is about 53. Therefore, the expected number of comparisons for this phone is 53^2 for each pair of utterances. Computing the kernel matrix for a total of N utterances for each pair of dialects is thus bounded by $\mathcal{O}(N^2M^2|\Phi|D)$.

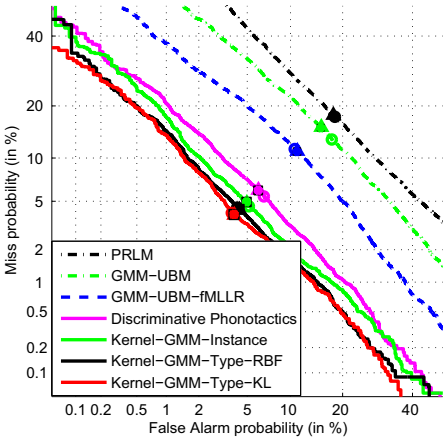

Figure 1: DET curves for several approaches on four Arabic dialects

We can see that computing the kernel function in (1) is quite expensive, partly due to the cross-comparison between every phone *instance* of the same type across each pair of utterances. This becomes increasingly significant when the training/testing utterances are long (leading to large M). Note that a smaller phone inventory would result in a larger number of instances of each type within each utterance; because the cost is linear in inventory size, but quadratic in instance count, this would increase cost. Another significant disadvantage of the Kernel-GMM-Instance approach is that since each phone instance typically consists of just a few frames, performing the MAP adaptation at this level leads to robustness issues with the parameter estimates for the adapted phone-GMM.

2.2. Phone-Type Supervectors and their Kernels

Instead of comparing supervectors of phone *instances*, we next experiment with comparing supervectors of phone *types*. This gives us a constant number of comparisons between a pair of utterances, equal to the number of phone types. Similar to the Kernel-GMM-Instance, we first run the phone recognizer to obtain the most likely phone sequence hypothesis for U, along with the frame alignment for each phone instance. However, unlike Kernel-GMM-Instance, we perform MAP adaption not on individual phone instances, but instead on all the frames of every instance of a given phone type in U to MAP-adapt the corresponding phone GMM-UBM. Thus, we obtain $|\Phi|$ adapted phone-GMMs from each utterance. Again, we adapt only the means of the Gaussians using a relevance factor of $r = 0.1$. The adapted GMM means are then stacked to construct a supervector for each phone type. This representation captures the 'general' realization of each phone type as opposed to the individual realization of each phone instance, as in Kernel-GMM-Instance. We term this approach Kernel-GMM-Type.

2.2.1. *Designing a Phone-Type-Based SVM Kernel*

An utterance U is represented by a set S_U of supervectors, each supervector corresponding to one phone type. Therefore, the size of S_U is at most the size of the phone inventory ($|\Phi|$). We denote the supervector u of phone type ϕ, as u_ϕ. Let $S_{U_a} = \{u_\phi\}_{\phi \in \Phi}$ and $S_{U_b} = \{v_\phi\}_{\phi \in \Phi}$ be the phone-type supervector sets of utterances U_a and U_b, respectively. Our new

kernel function is:

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi \in \Phi} e^{-\|u_\phi - v_\phi\|^2 / 2\sigma^2} \quad (2)$$

It compares the general phonetic realization of the same phone types across a pair of utterances, as opposed to the realization of every pair of individual-phone instances of the same type across the pair of utterances, as in (1). We call this Kernel-GMM-Type-RBF.

2.2.2. A Phone-Type-Based Kernel with KL-Divergence

The kernel functions we have designed thus far are sums of RBF kernels between phone supervectors. Recall that these supervectors are created from means of MAP-adapted phone-GMMs. Instead of comparing GMM mean vectors, we can therefore compare the Kullback–Leibler (KL) divergence between the two adapted phone-GMMs, following [13, 12]. Unfortunately, the KL-divergence does not satisfy the Mercer condition, and thus does not meet the requirements to be used as the kernel function for an SVM. However, Campbell et al. [12] have proposed a kernel function between GMMs based on an upper bound for their KL-divergence proposed by Do [14].

Using this KL-divergence-based kernel between two adapted GMMs modeling phone ϕ (with mean supervectors μ^a and μ^b), we obtain the kernel function:

$$K_\phi(\mu^a, \mu^b) = \sum_i (\sqrt{\omega_{\phi,i}} \Sigma_{\phi,i}^{-\frac{1}{2}} \mu_i^a)^T (\sqrt{\omega_{\phi,i}} \Sigma_{\phi,i}^{-\frac{1}{2}} \mu_i^b) \quad (3)$$

where $\omega_{\phi,i}$ and $\Sigma_{\phi,i}$ respectively are the weight and diagonal covariance matrix of Gaussian i of the phone GMM-UBM of phone-type ϕ. We define a new kernel function between a pair of utterances:

$$K(S_{U_a}, S_{U_b}) = \sum_{\phi \in \Phi} K_\phi(u_\phi - \mu_\phi, v_\phi - \mu_\phi) \quad (4)$$

where μ_ϕ is the stacked mean vectors of the phone-GMM-UBM of phone-type ϕ. The subtraction of μ_ϕ in (4) from the supervectors allows zero contributions from Gaussians that are not affected by the MAP adaptation, which will result in sparse supervectors.[1] We term this approach Kernel-GMM-Type-KL.

It is interesting to note that for a linear kernel K_ϕ such as (3), we can represent each utterance S_{U_x} in (4) with a single vector. This vector, say W_x, is formed by stacking the phone-type supervectors (after scaling by $\sqrt{\omega_\phi} \Sigma_\phi^{-\frac{1}{2}}$ and subtracting the corresponding μ_ϕ) in some (arbitrary) fixed order, with zero supervectors for phone types not in U_x. This representation allows the kernel in (4) to be written as:

$$K(S_{U_a}, S_{U_b}) = W_a^T W_b \quad (5)$$

The vector-of-supervectors W_x can be viewed as the 'phonetic fingerprint' of the utterance. We hypothesize that such a representation can be useful for multiple speech applications, including speaker verification and identification.

2.2.3. Evaluation and Time Complexity

We evaluate Kernel-GMM-Type-RBF (employing the kernel in (2)) using the same data sets and settings as for Kernel-GMM-Instance. As shown in Figure 1, this approach yields a slight (EER of 4.35%) but not significant improvement over Kernel-GMM-Instance. However, when we use the KL-based kernel function in (4), we achieve our best results (EER of 3.96%) with significant improvement in EER over Kernel-GMM-Instance (19.8% relative improvement). We hypothesize that this reduction in EER is due to the utilization of all frames from all instances in the utterance to MAP-adapt the corresponding phone GMM-UBM, leading to more robust estimates of the adapted phone-GMM.

Torres-Carrasquillo et al. [5] has shown that a GMM-UBM-based model, discriminatively trained with SDC features, eigenchannel compensation, and VTLN, can achieve an EER of 7.0% on three Arabic dialects (Gulf, Iraqi, and Levantine) using the same Appen corpora employed here. For a direct comparison, we trained our system using both the training *and* the development data used by [5]; we evaluate on the exact test cuts. Our Kernel-GMM-Type-KL approach achieves an EER of 6.1%, a 12.9% relative improvement over the state-of-the-art results reported in [5]. Our Kernel-GMM-Instance approach achieves an EER of 6.4% in these conditions.

Not only do we obtain the best results with Kernel-GMM-Type-KL, but also the time complexity of this approach is also substantially lower than that of Kernel-GMM-Instance. Assuming that all utterances have at least one instance of each phone type, the complexity of computing the kernel function (4) on a pair of utterances is $\mathcal{O}(|\Phi|D)$. Thus, constructing the kernel matrix for N utterances takes $\mathcal{O}(N^2|\Phi|D)$. Note that unlike Kernel-GMM-Instance, the complexity here is independent of the duration of utterances, since we compare phone types as opposed to phone instances. We observed run-time speed improvements of about 12-15×.

3. Experiments on English Accents/Dialects

To test whether our approach generalizes for dialects and accents of other languages, we evaluate it on two NIST tasks: American English vs. Indian English accents, and Southern vs. Non-Southern American English.

For the American vs. Indian English experiments, we use the American English speaker training files (30 s cuts) from the 2005 NIST LRE; The CallHome American English Speech corpus and 27 hours of randomly selected speech of native American English speakers from Fisher English Training Part 1 for the American data. For the Indian English data, we use the Indian English speaker training files (30 s cuts) from the 2005 NIST LRE and the Indian and Tamil English speakers from Fisher English Training Part 1 and 2, augmented with the CallFriend Hindi Speech corpus, following [5]. We segment both corpora to 30 s training segments and use multiple segments from the same speaker, resulting in 2589 Indian training segments, and 4877 American English training cuts. We test our kernel-based systems on the official 2007 NIST LRE Test Set (the 30 s task). This set contains 79 American English speakers and 160 trials of Indian English speakers. This official set allows us to directly compare the performance of our approach to published work. Torres-Carrasquillo et al. [5] tested their system on this accent task as well, employing a superset of our training data. They obtain an EER of 10.6% on this official set. The EER using the GMM-SVM approach [12] on this task, according to [5] is 11.3%. Chen et al. [3] evaluated their system on this test set for this task as well. Their approach when fused with PRLM also yields an EER of 10.6%. We compare the performance of our approaches to these systems.

For our English data, we obtain phone hypotheses using the Brno University's English phone recognizer [15]. For each phone type, we train a phone GMM-UBM with 60 Gaussian components using a random sample of frames that were aligned to phone instances of this phone type from the training data. The acoustic features are 13 RASTA-PLP features (including energy) plus delta and delta-delta, resulting in a 39D feature vector for each frame. The rest of the steps/settings for the English sys-

[1] We have observed that this subtraction slightly improves EER.

tem are exactly the same as those of the Arabic experiments. We train another system based on Kernel-GMM-Instance, but, instead of the RBF kernel in (1), we use the KL-divergence-based kernel in (3). We term this system Kernel-GMM-Instance-KL.[2]

Evaluating these two systems on this official test set, we obtain an EER of 8.7% for Kernel-GMM-Type-KL (17.9% relative improvement in EER over [5] and [3]) and a slightly, but not significantly, better EER of 7.8% using Kernel-GMM-Instance-KL. Combining the output of these two systems by simply summing the SVM posteriors, we achieve our best EER: 6.3%.[3] All these systems outperform both [5] and [3]'s systems. Although the combined system achieves 40.6% relative improvement over both baselines, this is not statistically significant due to the small number of test trials.

For the American Southern vs. Non-Southern dialects (a 1996 NIST LRE task), we compare the performance of our Kernel-GMM-Type-KL system to the standard GMM-UBM approach (with 2048 Gaussians). Our corpus includes speakers from the CallFriend American English – Southern Dialect and American English – Non-Southern Dialect corpora. Each is divided into 40 speakers for training, 40 for development, and 40 for testing. In this work, we use both the training and development portions to train our models and the 40 test speakers for evaluation. Similar to our other experiments, we segment each file in both corpora to 30 s segments. We use multiple cuts from each speaker, resulting in 839 southern cuts and 871 non-southern cuts for testing.

We find that the EER using the GMM-UBM approach with the same front-end as the American vs. Indian-English experiments is 31.4%. This is significantly above chance, but still appears low in comparison to the other evaluations. Recall, however, that for this task we have only 80 training speakers per dialect, a small number relative to our Arabic and American vs. Indian English experiments. Evaluating Kernel-GMM-Type-KL, we obtain an EER of 15.7%, a substantial improvement over GMM-UBM (50% relative reduction in EER).

In Section 2.2.2 we saw that for the Kernel-GMM-Type-KL approach, an utterance can be represented as a single vector W_x of phone supervectors. Such a representation allows us to investigate classifiers other than SVMs. We experiment with logistic regression using this vector representation for the Southern vs. Non-Southern American English task. We train a logistic regression with L_2 regularizer on the same vectors used for the Kernel-GMM-Type-KL experiment, and test on the same test cuts. Unsurprisingly, due to the close relationship between SVM and logistic regression, the logistic classifier on this task performs slightly but not significantly better (EER of 14.6%) than the SVM using the kernel in (4). We also find that the DET curve corresponding to the logistic regression has a slope much closer to -1. This may be useful for speaker verification.

4. Conclusions and Future Work

We have introduced a new approach to dialect recognition based on modeling a speaker's 'general' realization of each phone type in their utterance. In our approach, given an utterance, we first obtain a phone segmentation hypothesis using a phone recognizer, and then extract GMM-supervectors for each phone type in the utterance. We design a novel kernel function that computes similarities between phone types across pairs of utterances. Using this kernel, we train an SVM classifier for each pair of dialects. We have conducted a series of experiments to test our approach on four Arabic dialects, American English vs. Indian Accent, and American Southern vs. Non-Southern English from 30 s speech segments. For all of these tasks, our approach outperforms state-of-the-art approaches. Moreover, our new kernel function can be computed substantially faster than that of our previous work, with speed improvements of about 12-15$\times$.

We have seen that a speaker's utterance can be represented in a single vector which summarizes the general realization of the speaker's individual phones. It is important to note that, in our vector representation, the phone labels constrain which Gaussians can be affected by the MAP adaptation, i.e., the comparison incorporates the linguistic constraints realized by the phone recognizer. This is in contrast to the GMM-supervector representation [10] for which, in theory, any Gaussian in the GMM-UBM can be affected by any frame of any phone – ignoring the linguistic context of each frame.

It has been shown that VTLN and channel compensation techniques improve language and dialect recognition systems. In future, we will test the impact of such techniques on our approach (by compensating at the level of our vector of phonetic-supervectors). We hypothesize that our approach can potentially be combined with other state-of-the-art approaches. Finally, using our vector of phonetic-supervectors, we plan to evaluate our approach on speaker recognition.

Acknowledgements: We would like to thank Jason W. Pelecanos, Lidia Mangu, Hagen Soltau and the rest of the IBM T.J. Watson speech team for their useful discussions.

[2]Here we also shift the mean vectors by the UBMs' means.

[3]Kernel-GMM-Instance-RBF (1) achieves an EER of 10.3%.

5. References

[1] M.A. Zissman, T. Gleason, D. Rekart, and B. Losiewicz, "Automatic dialect identification of extemporaneous conversational, Latin American Spanish speech," in *Proceedings of the IEEE ICASSP*, Atlanta, USA, 1996.

[2] O. Koller, A. Abad, and I. Trancoso, "Exploiting variety-dependent Phones in Portuguese Variety Identification," in *Odyssey*, Brno, Czech Republic, 2010.

[3] N.F. Chen, W. Shen, and J.P. Campbell, "A linguistically-informative approach to dialect recognition using dialect-discriminating context dependent phonetic models," in *ICASSP'10*, 2010.

[4] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 19 – 41, 2000.

[5] P.A. Torres-Carrasquillo, D. Sturim, D. Reynolds, and A. McCree, "Eigenchannel Compensation and Discriminatively Trained Gaussian Mixture Models for Dialect and Accent Recognition," in *INTERSPEECH*, Brisbane, Australia, 2008.

[6] M. Akbacak, D. Vergyri, A. Stolcke, N. Scheffer, and A. Mandal, "Effective Arabic dialect classification using diverse phonotactic models," in *Proceedings of Interspeech'12*, 2012.

[7] F. Biadsy and J. Hirschberg, "Using Prosody and Phonotactics in Arabic Dialect Identification," in *Proceedings of INTERSPEECH'09*, Brighton, UK, 2009.

[8] F. Biadsy, J. Hirschberg, and M. Collins, "Dialect Recognition Using a Phone-GMM-Supervector-Based SVM Kernel," in *Proceedings of Interspeech'10*, Japan, 2010.

[9] F. Biadsy, H. Soltau, L. Mangu, J. Navratil, and J. Hirschberg, "Discriminative phonotactics for dialect recognition using context-dependent phone classifiers," in *Proceedings of Odyssey*, Brno, Czech Republic, 2010.

[10] W.M. Campbell, D.E. Sturim, and D.A. Reynolds, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 308–311, May 2006.

[11] H. Soltau, G. Saon, B. Kingsbury, H.K.Kuo, L. Mangu, D. Povey, and A. Emami, "Advances in Arabic speech transcription at IBM under DARPA GALE program," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 5, pp. 884–895, 2009.

[12] W.M. Campbell, D.E. Sturim, D.A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and nap variability compensation," in *Proceedings of ICASSP'06*, France, May 2006.

[13] P.J. Moreno, P.P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," in *Advances in Neural Information Processing Systems 16, MIT Press, Cambridge*, 2004.

[14] M.N. Do, "Fast approximation of Kullback-Leibler distance for dependence trees and hidden Markov models," *IEEE Signal Processing Letters*, vol. 10, no. 4, pp. 115–118, 2003.

[15] P. Matejka, P. Schwarz, J. Cernocky, and P. Chytil, "Phonotactic Language Identification using High Quality Phoneme Recognition," in *Proceedings of Eurospeech'05*, 2005.