USE OF ARTICULATORY BOTTLE-NECK FEATURES FOR QUERY-BY-EXAMPLE SPOKEN TERM DETECTION IN LOW RESOURCE SCENARIOS

Gautam Mantena, Kishore Prahallad

International Institute of Information Technology - Hyderabad, India

gautam.mantena@research.iiit.ac.in, kishore@iiit.ac.in

ABSTRACT

For query-by-example spoken term detection (QbE-STD), generation of phone posteriorgrams requires labelled data which would be difficult for languages with low resources. One solution is to build models from rich resource languages and use them in the low resource scenario. However, phone classes are not language universal and alternate representation such as articulatory classes is explored. In this paper, we use articulatory information and their derivatives such as bottle-neck (BN) features (also referred to as articulatory BN features) for QbE-STD. We obtain Gaussian posteriorgrams of articulatory BN features in tandem with the acoustic parameters such as frequency domain linear prediction cepstral coefficients to perform the search. We compare the search performance of articulatory and phone BN features and show that articulatory BN features are a better representation. We also provide experimental results to show that low amounts (30 mins) of training data could be used to derive articulatory BN features.

Index Terms— Query-by-example spoken term detection, multi-layer perceptron, articulatory features, bottle-neck features, low resource.

1. INTRODUCTION

The task of a query-by-example spoken term detection (QbE-STD) is to search a spoken query in a spoken audio data. A traditional QbE-STD approach is to convert spoken audio into a sequence of symbols and then perform text based search. In [1–3], the audio is first converted to a sequence of symbols using large vocabulary continuous speech recognition (LVCSR) and then lattice based search techniques are incorporated. LVCSR based approaches have been shown to be accurate for well resourced languages. However, such approaches are not scalable for languages where there is no availability or the resources to build an LVCSR system. To overcome this limitation dynamic time warping (DTW) based techniques are exploited for QbE-STD [4–8].

Phone [4, 5] and Gaussian posteriorgrams [6–8] are some of the feature representations used for DTW-based QbE-STD. Generation of phone posteriorgrams require labelled data which would be difficult for languages with low resources. One solution is to build models from rich resource languages and use them in the low resource scenario [5, 9]. However, phone classes are not language universal and thus alternate representation such as articulatory classes is explored. Articulatory classes are language independent representation of speech sounds and classifiers could be trained on relatively low amounts of data [10, 11]. Articulatory information has been extensively used in LVCSR for (a) Robust recognition in noisy conditions [11–13], and (b) Multi-lingual and cross-lingual speech recognition [14–16]. In [17], spoken audio is decoded to a sequence of articulatory classes which is used to prune out the spoken audio before performing the DTW-based search.

In this paper, we use articulatory information and their derivatives such as bottle-neck (BN) features (also referred to as articulatory BN features) for QbE-STD. BN features have been used extensively in multi-lingual LVCSR and were shown to improve the word error rate [18–21]. In the context of QbE-STD, BN features of phone classes have been used to build a hierarchical neural network structure (referred to as BN universal context network) [22]. To our knowledge, BN features of articulatory classes have not been explored in the context of DTW-based QbE-STD.

The contributions of our work are as follows: (a) Use of articulatory information and its derivatives such as BN features for QbE-STD, (b) Use of BN features in tandem with the acoustic parameters such as frequency domain linear prediction cepstral coefficients to compute Gaussian posteriorgrams, (c) Comparison of Gaussian posteriorgrams obtained using articulatory and phone BN features, and (d) Experimental results to show that low amounts of training data could be used to obtain articulatory BN features.

The organization of the paper is as follows: Section 2 describes the database used in this work. In Section 3, we describe the DTW-based algorithm used to perform the search. Section 4 describes the acoustic parameters of the speech signal and the computation of Gaussian posteriorgrams. Section 5 describes the use of articulatory BN features for QbE-STD and its comparison with the phone BN features. In Section 6, we provide experimental results to show that 20-30 mins of training data can be used to derive articulatory BN features.

2. DATABASE

The experiments conducted in this work use MediaEval 2012 data which is a subset of Lwazi database [23]. The data consists of audio recorded via telephone in 4 of 11 South African languages. We consider two data sets, development (dev) and evaluation (eval) which contain spoken audio (reference) and spoken query data. The statistics of the audio data is shown in Table 1.

Table 1: Statistics of MediaEval 2012 data.

Data	Utts	Total(mins)	Average(sec)
dev reference	1580	221.863	8.42
dev query	100	2.372	1.42
eval reference	1660	232.541	8.40
eval query	100	2.537	1.52

All the evaluations are performed using 2006 NIST evaluation criteria [24, 25] and the corresponding actual term weighted values (ATWV) and maximum term weighted values (MTWV) are re-

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

ported. To compute the ATWV and MTWV, an average miss probability and false alarm probabilities are computed for all the queries. In this paper, an optimum threshold to retrieve the search results is computed using the dev dataset. This threshold is then applied on the eval dataset to obtain the ATWV.

3. QBE-STD USING NON-SEGMENTAL DTW

QbE-STD is performed using a variant of DTW-based search referred to as non-segmental DTW (NS-DTW) [5, 8, 26]. Let $\mathcal{Q} = \{\mathbf{q_1}, \mathbf{q_2}, \ldots, \mathbf{q_i}, \ldots, \mathbf{q_n}\}$ be a spoken query (or query) containing n feature vectors. Let $\mathcal{R} = \{\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_j}, \ldots, \mathbf{u_m}\}$ be the spoken audio (or reference) containing m feature vectors.

Each of these feature vectors represent a Gaussian, articulatory or phone posteriorgrams as computed in Sections 4 and 5. The distance measure between a query vector $\mathbf{q_i}$ and a reference vector $\mathbf{u_j}$ is given by:

$$d(i,j) = -log\left(\frac{\mathbf{q_i}}{||\mathbf{q_i}||} \cdot \frac{\mathbf{u_j}}{||\mathbf{u_j}||}\right) \quad (1)$$

We define the term *search hit* as the region in the reference $\mathcal{R}$ that is likely to contain the query $\mathcal{Q}$. The query can start from any point in the reference. Initially, $S(1,j) = d(1,j)$, where $d(1,j)$ is the distance measure. The entries in the rest of the similarity matrix for NS-DTW is given by Eq. (2).

$$S(i,j) = min\left\{\begin{array}{c}\frac{d(i,j)+S(i-1,j-2)}{T(i-1,j-2)+1} \\ \frac{d(i,j)+S(i-1,j-1)}{T(i-1,j-1)+2} \\ \frac{d(i,j)+S(i-1,j)}{T(i-1,j)+1}\end{array}\right\}, \quad (2)$$

where T is called the transition matrix. $T(i,j)$ represents the number of transitions required to reach i,j from a start point. In order to detect the start and end time stamps of the *search hit*, we obtain the reference index that contains the best alignment score, i. e., the end point of the *search hit* is given by $j = \min_j\{S(n,j)\}$ for $j = 1, 2, ..., m$. Once the end point j is obtained, the corresponding start point could be obtained by a path trace back. Thus we obtain the location of the query in the reference.

4. FEATURE REPRESENTATION USING GAUSSIAN POSTERIORGRAMS

In general, Gaussian posteriorgrams are obtained by a two step process [7, 8]. In the first step, acoustic parameters such as Mel-frequency cepstral coefficients (MFCC) or frequency domain linear prediction cepstral coefficients (FDLP) are extracted from the speech signal. In the second step, Gaussian posteriorgrams are computed by training a Gaussian mixture model (GMM) on the speech data and the posterior probability obtained from each Gaussian is used to represent the acoustic parameter. In this paper, we train a GMM containing 128 Gaussians to obtain 128 dimensional Gaussian posteriorgrams.

In [8], we show that the Gaussian posteriorgrams of FDLP perform better than that of MFCC. In MFCC, the short-time spectral properties of the speech signal is captured. In order to capture the temporal dynamics of the speech signal, FDLP was developed [27–29].

A 25 ms window length with 10 ms shift is considered to extract 13 dimensional features along with delta and acceleration coefficients for MFCC and FDLP. An all-pole model of order 160 poles/sec and 37 filter banks are considered to extract FDLP. A set of 26 filter banks are used for computing MFCC.

Table 2: MTWV obtained using 128 dimensional Gaussian posteriorgrams (GPost.) of 39 dimensional MFCC and FDLP. The values indicated in the brackets show the ATWV computed for the eval dataset.

Feats.	dim.	GPost. dim.	MTWV (ATWV)	
			dev	eval
MFCC	39	128	0.377	0.325 (0.323)
FDLP	39	128	0.399	0.387 (0.358)

Table 2 shows the MTWV using 128 dimensional Gaussian posteriorgrams of 39 dimensional MFCC and FDLP. The search is performed using NS-DTW as described in Section 3. From Table 2, it can be seen that Gaussian posteriorgrams of FDLP performs better than that of MFCC. Hence, we are motivated to use FDLP as the acoustic features for QbE-STD. A more detailed analysis of the performance of NS-DTW using FDLP is described in [8].

To obtain Gaussian posteriorgrams of the acoustic parameters such as FDLP, no class information such as phone or articulatory classes is used. In this paper, we derive bottle-neck (BN) features from an articulatory model (also referred to as articulatory BN features). We show that the Gaussian posteriorgrams of articulatory BN features in tandem with FDLP perform better than that of FDLP. Section 5 describes the use of articulatory BN features in detail.

5. ARTICULATORY BOTTLE-NECK FEATURES

Availability of labelled data is an issue for building supervised models such as multi-layer perceptron (MLP). To overcome such an issue we train models on a high resource language and use it in a low resource scenario.

Table 3: Articulatory classes of speech sounds

Articulatory Property	Classes	# bits
Voicing	±voicing	1
Vowel length	short, long, diphthong	3
Vowel height	high, mid, low	3
Vowel frontness	front, central, back	3
Lip rounding	±rounding	1
Manner of articulation	stop, fricative, affricative nasal, approximant	5
Place of articulation	velar, alveolar, palatal, labial, dental	5
Aspiration	±aspiration	1
Silence	±silence	1

We train an articulatory MLP using 24 hours of labelled Telugu database consisting of 49 phones [30]. These 49 phones are represented by 23 articulatory classes which characterize the speech production process such as vowel properties, place of articulation, manner of articulation, etc. We modify the articulatory classes described in [31] to suit the training data available. We use nine different articulatory properties (as shown in Table 3). Each articulatory property is further divided into sub classes resulting in a 23 dimensional articulatory posteriorgram.

The architecture used for training an articulatory MLP is 39L 120N 13L 120N 23S. For comparison we also train a phone MLP with an architecture 39L 120N 13L 120N 49S. The integer values in the MLP architecture indicate the number of nodes, and L (linear), N (non-linear) and S (sigmoid) represent the activation functions in

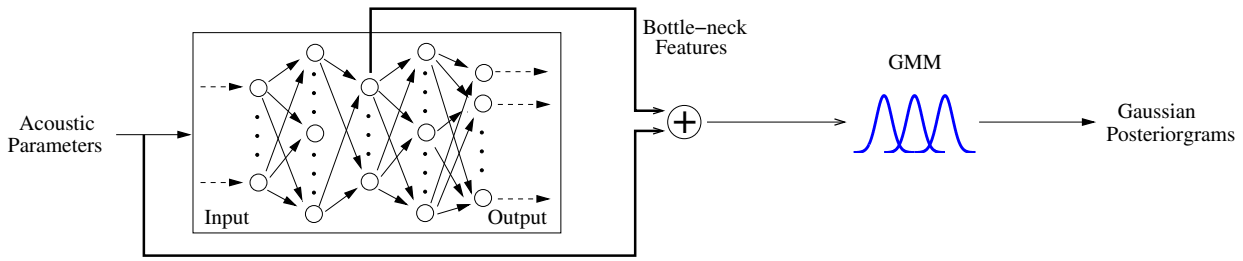

Fig. 1: A general block diagram for computing Gaussian posteriorgrams of bottle-neck features in tandem with the acoustic parameters such as FDLP.

each of the layers. We use 39 dimensional acoustic parameters as the input for the articulatory and phone MLPs.

Table 4 shows MTWV obtained using 23 dimensional articulatory, 49 dimensional phone and 128 dimensional Gaussian posteriorgrams. From Table 4, it can be seen that the Gaussian posteriorgrams perform better than the articulatory and phone posteriorgrams. Thus, phone and articulatory posteriorgrams under-perform when the language they were trained on differs from the target language [7, 32].

Table 4: MTWV obtained using 23 dimensional articulatory, 49 dimensional phone and 128 dimensional Gaussian posteriorgrams of FDLP. The values indicated in the brackets show the ATWV computed for the eval dataset.

Posteriorgrams	**Post. dim.**	**MTWV (ATWV)**	
		dev	eval
Art. Post.	23	0.212	0.172 (0.156)
Phone Post.	49	0.265	0.217 (0.209)
Gaussian Post.	128	**0.399**	**0.387 (0.358)**

5.1. Bottle-neck (BN) features

In order to exploit the class information captured by an MLP, we derive features from the bottle-neck layer (as shown in Fig. 1). These are referred to as bottle-neck (BN) features and are of 13 dimensions. The advantages of BN features are as follows [33]: (a) They are compressed features and are of lower dimension, and (b) Classification properties of the target class is reflected in the BN features.

5.2. Compressed (CP) features

An alternative representation to BN features can be obtained by post processing the articulatory posteriorgrams as follows: (a) A negative logarithm is applied on the articulatory posteriorgrams to scale the dynamic range and then followed by dimensionality reduction [14, 16]. These post processed posteriorgram features are referred to as compressed posteriorgram (CP) features, and (b) We then obtain Gaussian posteriorgrams of CP features in tandem with FDLP.

In the literature, CP features are referred to as tandem connectionist features [34] or probabilistic features [20, 33]. In [9], Gaussian posteriorgrams of CP features derived from phone MLPs were used for QbE-STD. However, it was shown that the Gaussian posteriorgrams of CP features were performing similar to that of the acoustic parameters. In this paper, we show that the search performance can be improved by using BN (or CP) features in tandem with the acoustic parameters such as FDLP.

To compress the log posteriorgram features, we perform a non-linear PCA using an auto associative neural network (AANN) with an architecture 23L 100N 13L 100N 23L. Thus we obtain 13 dimensional CP features from 23 dimensional articulatory posteriorgrams. These features are similar to that of the BN features as described in Section 5.1. However, an advantage of BN over CP features is that they do not require an explicit dimensionality reduction.

5.3. Comparison of BN and CP features

Table 5 shows MTWV obtained using Gaussian posteriorgrams of articulatory CP (AR-CP), articulatory BN (AR-BN), FDLP, FDLP + AR-CP and FDLP + AR-BN. From Table 5, it can be seen that: (a) Gaussian posteriorgrams of FDLP + AR-BN (or AR-CP) perform better than that of FDLP, and (b) Gaussian posteriorgrams of FDLP + AR-BN perform better than of FDLP + AR-CP. Thus we choose articulatory BN features to obtain Gaussian posteriorgrams for QbE-STD.

Table 5: MTWV obtained using Gaussian posteriorgrams of AR-CP, AR-BN, FDLP, FDLP + AR-CP and FDLP + AR-BN features. The values indicated in the brackets show the ATWV computed for the eval dataset.

Feats.	**dim.**	**GPost. dim.**	**MTWV (ATWV)**	
			dev	eval
AR-CP	13	128	0.336	0.331 (0.323)
AR-BN	13	128	0.419	0.390 (0.389)
FDLP	39	128	0.399	0.387 (0.358)
FDLP + AR-CP	52	128	0.465	0.467 (0.463)
FDLP + AR-BN	52	128	**0.494**	**0.492 (0.467)**

5.4. Selecting an Optimum Dimension for Articulatory BN Features

In this Section, we perform experiments to select an optimum dimension for AR-BN features. We derive AR-BN features of dimensions 5, 9, 13, 17 and 21 to obtain Gaussian posteriorgrams.

Fig. 2 shows MTWV obtained for dev data using Gaussian posteriorgrams of FDLP + AR-BN. We derive 5, 9, 13, 17 and 21 dimensional AR-BN features and use them in tandem with 39 dimensional FDLP parameters. MLP architecture used to derive AR-BN features is as follows: 23L 100N ΦL 100N 23L, where Φ = 5, 9, 13, 17, 21. From Fig. 2, it can be seen that the best performance is with 13 dimensional AR-BN features in tandem with FDLP. Thus, we choose 13 as the optimum AR-BN feature dimension.

5.5. Comparison with Phone BN Features

In this Section, we derive 13 dimensional phone BN features and compare it with articulatory BN features. The MLP architecture used to derive phone BN features is 39L 120N 13L 120N 49S. Table

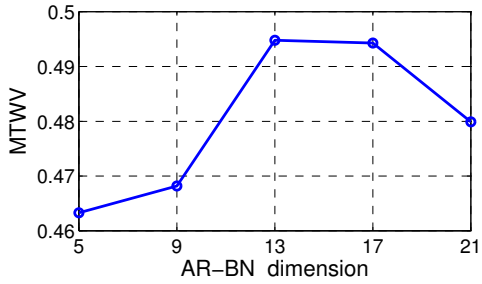

Fig. 2: MTWV obtained for dev data using Gaussian posteriorgrams of FDLP + AR-BN. AR-BN features of 5, 7, 13, 17 and 21 dimensions are used in tandem with 39 dimensional FDLP.

6 shows MTWV obtained using Gaussian posteriorgrams of phone and articulatory BN features in tandem with FDLP. The phone and articulatory BN features are denoted as PH-BN and AR-BN respectively.

Table 6: MTWV obtained using Gaussian posteriorgrams of FDLP + PH-BN and FDLP + AR-BN features. The values indicated in the brackets show the ATWV computed for the eval dataset.

Feats.	dim.	GPost. dim.	MTWV (ATWV)	
			dev	eval
FDLP + PH-BN	52	128	0.469	0.452 (0.425)
FDLP + AR-BN	52	128	**0.494**	**0.492 (0.467)**

From Table 6, it can be seen that the Gaussian posteriorgrams of FDLP + AR-BN perform better than that of FDLP + PH-BN. Articulatory classes are more language universal than phones. Thus AR-BN features are a better representation than PH-BN features to obtain Gaussian posteriorgrams.

6. USE OF LOW AMOUNTS OF TRAINING DATA FOR ARTICULATORY AND PHONE MLPS

In Section 5, we use an articulatory and phone MLPs trained on 24 hours of spoken audio data. However, access to such large amounts of labelled data is expensive and not always feasible. In this Section, we derive BN features from articulatory and phone MLPs trained on low amounts of spoken audio data.

Fig. 3 shows MTWV obtained for dev data using Gaussian posteriorgrams of FDLP, FDLP + PH-BN and FDLP + AR-BN. The articulatory and phone MLPs are trained using 10, 20, 30, 50 and 75 mins of audio data. MTWV obtained using Gaussian posteriorgrams of FDLP is the baseline performance and is denoted as an horizontal line (as shown in Fig. 3). From Fig. 3, we observe that 20-30 mins of training data can be used to derive AR-BN features. This is because each phone is represented by more than one articulatory class. This leads to a large amount of training material for each articulatory class, which often exceeds the amount of phone training data [11, 35].

Table 7 shows MTWV obtained using Gaussian posteriorgrams of FDLP, FDLP + PH-BN and FDLP + AR-BN. PH-BN and AR-BN features are derived from MLPs trained on 30 mins of labelled data. From Table 7, it can be seen that 30 mins can be used to derive AR-BN features to obtain Gaussian posteriorgrams. However, there is a trade-off between the performance of the BN features and the amount of data used for training (as shown in Fig. 3)

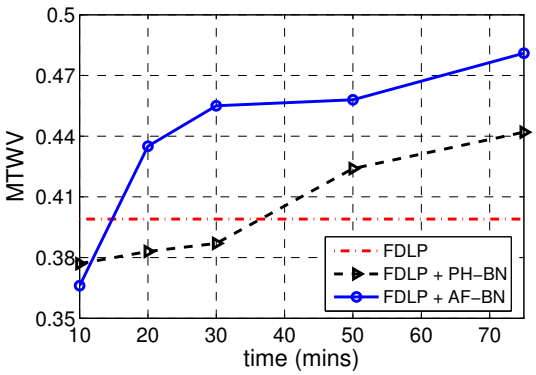

Fig. 3: MTWV obtained for dev data using Gaussian posteriorgrams of FDLP, FDLP + PH-BN and FDLP + AR-BN. The x-axis represent the amount of labelled data used to train the MLPs.

Table 7: MTWV obtained using Gaussian posteriorgrams of FDLP, FDLP + PH-BN and FDLP + AR-BN. The BN features are obtained from 30 mins of training data. The values indicated in the brackets show the ATWV computed for the eval dataset.

Feats.	dim.	GPost. dim.	MTWV (ATWV)	
			dev	eval
FDLP	39	128	0.399	0.387 (0.358)
FDLP + PH-BN	52	128	0.387	0.391 (0.338)
FDLP + AR-BN	52	128	**0.455**	**0.442 (0.425)**

7. CONCLUSIONS

In this paper, we have used articulatory information and its derivatives such as bottle-neck (BN) features (also referred to as articulatory BN features) for query-by-example spoken term detection (QbE-STD). We compared the search performance using Gaussian posteriorgrams of articulatory BN (AR-BN) and phone BN (PH-BN) features and have shown that AR-BN features are a better representation. We have also provided experimental results to show that 30 mins of training data could be used to derive AR-BN features.

Acknowledgements

We would like to thank Florian Metze, CMU for clarification on the literature of articulatory and bottle-neck features. We would also like to thank Tata Consultancy Services (TCS) for partially supporting Gautam's PhD fellowship at IIIT-H, India.

8. REFERENCES

[1] I. Szöke, M. Fapso, L. Burget, and J. Cernocky, "Hybrid word-subword decoding for spoken term detection," in *Workshop on Searching Spontaneous Conversational Speech*, 2008, pp. 4–11.

[2] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. of HLT-NAACL*, 2004, pp. 129–136.

[3] D. R. H. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. of INTERSPEECH*, 2007, pp. 314–317.

[4] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. of ASRU*, 2009, pp. 421–426.

[5] V. Gupta, J. Ajmera, A., and A. Verma, "A language independent approach to audio search," in *Proc. of INTERSPEECH*, 2011, pp. 1125–1128.

[6] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Proc. of ASRU*, 2009, pp. 398–403.

[7] X. Anguera, "Speaker independent discriminant feature extraction for acoustic pattern-matching," in *Proc. of ICASSP*, 2012, pp. 485–488.

[8] G. Mantena, S. Achanta, and K. Prahallad, "Query-by-example spoken term detection using frequency domain linear prediction and non-segmental dynamic time warping," *accepted for publication in IEEE Trans. Audio, Speech and Lang. Processing*, 2014.

[9] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," in *Proc. of ICASSP*, 2013.

[10] A. W. Black, T. Bunnell, Y. Dou, P. Muthukumar, F. Metze, D. Perry, T. Polzehl, K. Prahallad, S. Steidl, and C. Vaughn, "Articulatory features for expressive speech synthesis," in *Proc. of ICASSP*, Kyoto, Japan, 2012.

[11] K. Kirchhoff, G. A. Fink, and G. Sagerer, "Combining acoustic and articulatory feature information for robust speech recognition," *Speech Communication*, vol. 37, no. 3-4, pp. 303–319, 2002.

[12] K. Livescu, Ö. Cetin, M. Hasegawa-johnson, S. King, C. Bartels, N. Borges, A. Kantor, P. Lal, A. Bezman, S. Dawson-Haggerty, B. Woods, J. Frankel, M. Magimai-Doss, and K. Saenko, "Audiovisual speech recognition with articulator positions as hidden variables," in *in Proc. of ICASSP*, 2007.

[13] V. Mitra, W. Wang, A. Stolcke, H. Nam, C. Richey, J. Yuan, and Mark Liberman, "Articulatory features for large vocabulary speech recognition," in *in Proc. of ICASSP*, 2013.

[14] Ö. Çetin, M. Magimai-Doss, K. Livescu, A. Kantor, S. King, C. D. Bartels, and J. Frankel, "Monolingual and crosslingual comparison of tandem features derived from articulatory and phone MLPs," in *Proc. of ASRU*, 2007, pp. 36–41.

[15] S. Stüker, F. Metze, T. Schultz, and A. Waibel, "Integrating multilingual articulatory features into speech recognition.," in *in Proc. INTERSPEECH*, 2003.

[16] L. Tóth, J. Frankel, G. Gosztolya, and S. King, "Cross-lingual portability of MLP-based tandem features - a case study for English and Hungarian," in *Proc. of INTERSPEECH*, 2008, pp. 2695–2698.

[17] F. Metze, N. Rajput, X. Anguera, M. H. Davel, G. Gravier, C. J. V. Heerden, G. V. Mantena, A. Muscariello, K. Prahallad, I. Szöke, and J. Tejedor, "The spoken web search task at MediaEval 2011," in *Proc. of ICASSP*, 2012, pp. 5165–5168.

[18] F. Grézl and P. Fousek, "Optimizing bottle-neck features for LVCSR," in *Proc. of ICASSP*, 2008, pp. 4729–4732.

[19] N. T. Vu, F. Metze, and T. Schultz, "Multilingual bottle-neck features and its application for under-resourced languages," in *Proc. of SLTU*, 2012.

[20] F. Grézl, M. Karafiát, and M. Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proc. of ASRU*, 2011, pp. 359–364.

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

[22] J. Tejedor, I. Szöke, and M. Fapso, "Novel methods for query selection and query combination in query-by-example spoken term detection," in *in Proc. of SSCS*, 2010, pp. 15–20.

[23] E. Barnard, M. H. Davel, and C. J. V. Heerden, "ASR corpus design for resource-scarce languages," in *Proc. of INTERSPEECH*, 2009, pp. 2847–2850.

[24] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddington, "Results of the 2006 spoken term detection evaluation," in *Proc. of Workshop on Searching Spontaneous Conversational Speech*, 2007, pp. 45–50.

[25] F. Metze, E. Barnard, M. H. Davel, C. J. V. Heerden, X. Anguera, G. Gravier, and N. Rajput, "The spoken web search task," in *MediaEval*, 2012.

[26] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for query-by-example spoken term detection," in *Proc. of ICME*, 2013.

[27] S. Ganapathy, *Signal analysis using autoregressive models of amplitude modulation*, Ph.D. thesis, Johns Hopkins University, Baltimore, Maryland, USA, Jan. 2012.

[28] S. Thomas, S. Ganapathy, and H. Hermansky, "Recognition of reverberant speech using frequency domain linear prediction," *IEEE Signal Processing Letters*, vol. 15, pp. 681 –684, 2008.

[29] S. Ganapathy, S. Thomas, and H. Hermansky, "Temporal envelope compensation for robust phoneme recognition using modulation spectrum," *Journal of Acoustical Society of America*, vol. 128, pp. 3769–3780, 2010.

[30] G. K. Anumanchipalli, R. Chitturi, S. Joshi, S. Singh R. Kumar, R.N.V Sitaram, and S.P. Kishore, "Development of Indian language speech databases for LVCSR," in *Proc. of SPECOM*, Patras, Greece, 2005.

[31] B. Bollepalli, A. W. Black, and K. Prahallad, "Modelling a noisy-channel for voice conversion using articulatory features," in *Proc. of INTERSPEECH*, 2012.

[32] A. Muscariello, G. Gravier, and F. Bimbot, "Audio keyword extraction by unsupervised word discovery," in *Proc. of INTERSPEECH*, 2009, pp. 2843–2846.

[33] F. Grézl, M. Karafiat, S. Kontar, and J. Cernocky, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. of ICASSP*, 2007, vol. 4, pp. 757–760.

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

[35] K. Kirchhoff, *Robust speech recognition using articulatory information*, Ph.D. thesis, Bielefeld University, 1999.