IIIT-H SWS 2013: Gaussian Posteriorgrams of Bottle-Neck Features for Query-by-Example Spoken Term Detection

Gautam Mantena, Kishore Prahallad
International Institute of Information Technology-Hyderabad, India
gautam.mantena@research.iiit.ac.in, kishore@iiit.ac.in

ABSTRACT

This paper describes the experiments conducted for spoken web search (SWS) at MediaEval 2013 evaluations. A conventional approach is to train a multi-layer perceptron using high resource languages and then use it in the low resource scenario. However, phone posteriorgrams have been found to under-perform when the language they were trained on differs from the target language.

In this paper, we use bottle-neck features derived from MLP to generate Gaussian posteriorgrams. We also use a variant of dynamic time warping (DTW) based technique which exploits the redundancy in speech signal and thus averages the successive Gaussian posteriorgrams to reduce the length of the spoken query and spoken reference.

1. INTRODUCTION

Gaussian and phone posteriorgrams are a popular feature representation for query-by-example spoken term detection (QbE-STD). Gaussian posteriorgrams are typically trained in an unsupervised manner often referred to as zero-resource scenario, whereas, phone posteriorgrams are obtained by training a multi-layer perceptron (MLP) in a supervised manner. For low/zero resource languages, an MLP is trained on high resource languages and then it is used in the low resource scenario. However, phone posteriorgrams have been found to under-perform when the language they were trained on differs from the target language. These MLP classifier outputs, though capture acoustic phonetic properties of a speech signal, are not sufficient as a feature representation. This is because the language used for training MLP is not enough to capture the complete acoustic characteristics of the multi-lingual data. To utilize this complimentary information captured, we derive features from an MLP for obtaining Gaussian posteriorgrams. A similar kind of feature representation has been explored in paper [1] for a better search performance.

An alternative representation for phone posteriorgrams are the articulatory features (AFs). AFs are a better representation as they are more language universal than phones.

This paper describes the experiments conducted for spoken web search (SWS) at MediaEval 2013 [2]. The primary focus of this work is to explore the use of bottle-neck (BN) features for QbE-STD derived from phone and AF MLPs.

MediaEval 2013 Workshop, October 18-19, 2013, Barcelona, Spain

2. FEATURE EXTRACTION

We use a three step process to generate the features for QbE-STD: (a) Extracting speech parameters such as frequency domain linear prediction (FDLP) [3](b) Train a phone or AF MLP and extract the bottle-neck features for each of the speech parameters, and (c) Compute Gaussian posteriorgrams using speech parameters in combination with the derived BN features.

In [4], we show that Gaussian posteriorgrams computed from FDLP perform better than those obtained from short-time spectral analysis such as Mel-frequency cepstral coefficients. In this paper, we use FDLP as the acoustic parameters of the speech signal.

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

2.1 Phone and AF Bottle-Neck Features

In this paper, we train phone and AF MLPs using labelled Telugu database ($\approx$ 24 hours) consisting of 49 phones [5]. MLP is trained to obtain 49 dimensional phone posteriorgrams and 23 dimensional articulatory features (AFs) using 39 dimensional FDLP features.

Table 1: Articulatory Features

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

The articulatory features (AFs) used in this work represent the characteristics of speech production process, which include vowel properties, place of articulation, manner of articulation, etc. We modified the AFs described in [6] to suit the training data available. We use nine different articulatory properties as shown in Table 1. Each articulatory property is further divided into sub classes resulting in a 23

dimensional AF vector.

Table 2: Architecture of the MLPs trained to derive bottle-neck features

	Architecture
PH MLP	39L 120N 13L 120N 49S
AF MLP	39L 120N 13L 120N 23S

Table 2, shows the architectures used to build phone and AF MLPs. 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 each of the layers.

3. EXPERIMENTS AND RESULTS

Gaussian posteriorgrams are computed by training a Gaussian mixture model (GMM) on the spoken data and the posterior probability obtained from each Gaussian is used to represent the speech parameters. The number of Gaussians represent the approximate number of acoustic units present in the spoken data. We computed Gaussian posteriorgrams as described in [7]. We trained the Gaussian mixture models (GMM) using 128 Gaussians. Before performing the DTW search we removed the Gaussian posteriorgrams corresponding to silence regions as described in [8]. All the experiments were conducted on a HPC cluster with HP SL230s compute nodes. Each HP SL230s node is equipped with two Intel E5-2640 processors with 12 cores each

We used a variant of DTW-based approach, referred to as non-segmental DTW (NS-DTW), for obtaining the search results [4]. NS-DTW is similar to that of the DTW-based search given in [7] but differs in the local constraints. Table 3 show the maximum term weighted values (MTWV) obtained by using each of the features. From Table 3, it can be seen that the use of bottle-neck features has improved the performance of the system. To perform the search our algorithm requires approximately 10 GB of memory.

Table 3: MTWV using Gaussian posteriorgrams computed from various features

Feats.	dev	eval
FDLP	0.1652	0.1557
PH-BN	0.2491	0.2133
AF-BN	0.2627	0.2122
FDLP + PH-BN	0.2741	0.2492
FDLP + AF-BN	0.2765	0.2413

To improve the computational performance, we reduce the query and reference Gaussian posteriorgrams vectors before performing search. Given a reduction factor $\alpha \in \mathbb{N}$, a window of size α is considered over the posteriorgram features and a mean is computed. The window is then shifted by α and another mean vector is computed. The posteriorgram vectors are replaced with the reduced number of posteriorgram features during this process. The averaging of Gaussian posteriorgrams also reduce the amount of memory required to compute the similarity matrix. In a conventional approach the space complexity required to compute the similarity matrix between a query and reference is of order $O(mnd^2)$ where m,n are the length of reference and query and d is the dimension of the feature vector. The averaging of Gaussian posteriorgrams will reduce the space complexity to an order of $O(\frac{mnd^2}{\alpha^2})$.

Table 4: Evaluation using FNS-DTW for various values of α

α	dev		eval	
	MTWV	RT (10^{-4})	MTWV	RT (10^{-4})
1	0.2765	16.55	0.2413	15.67
2	0.2530	4.21	0.2236	4.16
3	0.2252	1.92	0.1995	1.85
4	0.2043	1.11	0.1773	1.11

Table 4 show the MTWV and the runtime factor (RT) for various values of α using FDLP + AF-BN features. The results show an improvement in speed at the cost of the search accuracy. We have considered $\alpha = 2$ as an optimum value based on MTWV and the speed improvements.

4. CONCLUSIONS

In this work we have used the bottle-neck features obtained from phone and articulatory MLPs. We have shown that these BN features perform better than the conventional Gaussian posteriorgrams computed from FDLP. This motivates us to build models using high resource languages and use it in the low resource scenario.

5. REFERENCES

[1] 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 *in Proc. of ICASSP*, 2013.

[2] X. Anguera, F. Metze, A. Buso, I. Szoke, and L. J. Rodriguez-Fuentes, "The spoken web search task," in *MediaEval 2013 Workshop*, Barcelona, Spain, October 18-19 2013.

[3] 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.

[4] G. Mantena, S. Achanta, and K. Prahallad, "Query-by-example spoken term detection using frequency domain linear prediction and non-segmental dynamic time warping," *submitted to IEEE Trans. Audio, Speech and Lang. Processing*, 2013.

[5] 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.

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

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

[8] X. Anguera, "Telefonica Research system for the spoken web search task at MediaEval 2012," in *MediaEval 2012 Workshop*, Pisa, Italy, October 2012.