USING PARALLEL TOKENIZERS WITH DTW MATRIX COMBINATION FOR LOW-RESOURCE SPOKEN TERM DETECTION

Haipeng Wang[†]*, Tan Lee*[†]*, Cheung-Chi Leung*[*]*, Bin Ma*[*]*, Haizhou Li*[*]

[†]Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong
[*]Institute for Infocomm Research, A*STAR, Singapore

[†]{hpwang,tanlee}@ee.cuhk.edu.hk [*]{ccleung,mabin,hli}@i2r.a-star.edu.sg

ABSTRACT

Recently the posteriorgram-based template matching framework has been successfully applied to query-by-example spoken term detection tasks for low-resource languages. This framework employs a tokenizer to derive posteriorgrams, and applies dynamic time warping (DTW) to the posteriorgrams to locate the possible occurrences of a query term. Based on this framework, we propose to improve the detection performance by using multiple tokenizers with DTW distance matrix combination. The proposed approach uses multiple tokenizers in parallel as the front-end to generate different posteriorgram representations, and combines the distance matrices of the different posteriorgrams into a single matrix. DTW detection is then applied to the combined distance matrix. Lastly score post-processing techniques including pseudo-relevance feedback and score normalization are used for further improvement. Experiments were conducted on the spoken web search datasets of MediaEval 2011 and MediaEval 2012. Experimental results show that combining multiple tokenizers significantly outperforms the best single tokenizer, and that the DTW matrix combination method consistently outperforms the score combination method when more than three tokenizers are involved. Score post-processing techniques show further gains on top of using multiple tokenizers.

Index Terms— query-by-example spoken term detection, DTW matrix combination, tandem tokenizer, pseudo-relevance feedback

1. INTRODUCTION

Spoken term detection (STD) refers to the task of automatically locating the occurrences of a specified query term in a large audio archive. The query term can be given in the form of orthographic representation or speech utterance example. The latter case is known as query-by-example (QbyE) STD [1]. Aiming at *searching audio using audio*, the QbyE STD techniques allow the absence of the prior linguistic information, such as phoneme inventories and pronunciation dictionaries. Therefore, this kind of techniques are particularly applicable to low-resource languages, for which labeled training data is very limited or even does not exist. QbyE STD for low-resource languages has received increasing attention in recent years [2, 3, 4].

The posteriorgram-based template matching framework [1] has been successfully applied to QbyE STD tasks. This framework utilizes a tokenizer to convert both the query examples and the test utterances into posteriorgrams, and matches the query posteriorgrams with the test posteriorgrams using dynamic time warping (DTW), which has been widely used in template-based speech recognition. Posteriorgram representation is believed to be more robust and more informative than spectral features [5, 6].

In this paper, we focus on low-resource STD tasks, and propose to enhance the posteriorgram-based template matching framework by using multiple tokenizers with DTW distance matrix combination. It is expected to be beneficial to use multiple tokenizers that are obtained from different resources, such as unlabeled data of the test languages, and labeled data from other rich-resource languages. In our approach, three categories (unsupervised, supervised and semi-supervised) of tokenizers are used in parallel as the front-end to generate parallel posteriorgrams, which lead to parallel DTW distance matrices. These DTW matrices are then merged into a single matrix. DTW detection is performed on the combined distance matrix to derive the detection score. Lastly, score post-processing techniques are applied to further improve the performance.

In the previous studies related to the posteriorgram-based template matching framework, many efforts were contributed to introducing novel modeling for the tokenizers, such as deep Boltzmann machine [7], discriminant GMM [8], acoustic segment model (ASM) [9], etc. However few works were dedicated to the combined use of different tokenizers. In [10, 11], a score combination was adopted. Score combination method usually involves two stages. First some heuristic criteria are used to select the candidate hit regions. Then score combination is conducted on the selected candidate regions. Since those unselected parts are not taken into account in the second stage, the score combination approach may suffer from the inaccurate selection of candidate regions. Our approach of combining multiple tokenizers is not applied to the detection scores, but to the DTW distance matrices. Given the query example and the test utterance, the size of the DTW matrix is the same for different tokenizers, so it becomes straightforward to do the combination on the DTW matrices without much information loss. We carried out experiments using the spoken web search (SWS) datasets [12, 13] of MediaEval 2011 and MediaEval 2012. Experimental results demonstrate the effectiveness of the proposed approach.

2. BACKGROUND

2.1. Posteriorgram-based Template Matching

Fig. 1 depicts the posteriorgram-based template matching framework for QbyE STD. A tokenizer is first obtained from the training resources. The tokenizer can be a recognizer or classifier for any kind of sound units, e.g., phonemes, Gaussian components [14],

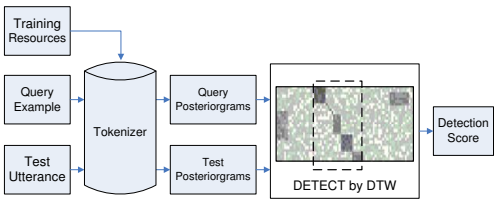

Fig. 1. Posteriorgram-based template matching framework

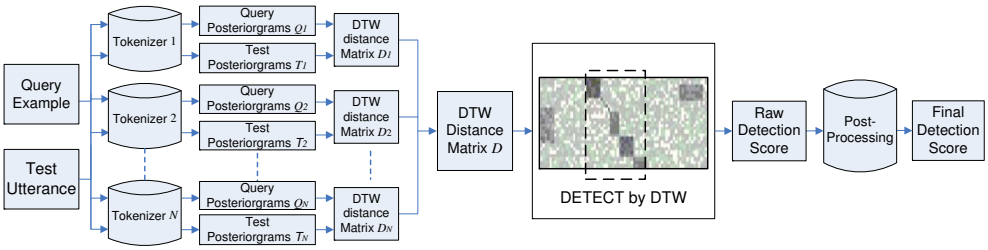

Fig. 2. The proposed system architecture

self-organized units [15, 16], etc. With the tokenizer, both the query examples and the test utterances are converted into posteriorgrams. Then DTW is utilized to scan through the test posteriorgrams, determine the matching regions and provide the detection scores with respect to the query posteriorgrams.

2.2. Posteriorgram Representation

Given a sequence of M observation feature vectors $[\boldsymbol{o}_1, \boldsymbol{o}_2, ..., \boldsymbol{o}_M]$, and a set of K predefined sound units $\{C_1, C_2, ..., C_K\}$. The corresponding posteriorgram $\boldsymbol{PG}$ is a $K \times M$ matrix consisting of M posterior probability vectors,

$$\boldsymbol{PG} = [\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_M], \tag{1}$$

where $\boldsymbol{q}_m$ is the posterior probability vector of the m_{th} frame:

$$\boldsymbol{q}_m = [p(C_1|\boldsymbol{o}_m), p(C_2|\boldsymbol{o}_m), ..., p(C_K|\boldsymbol{o}_m)]^T. \tag{2}$$

Compared to the spectral features, the posteriorgram representation is believed to be more robust against speaker variations. Different tokenizers may use different approaches to derive different types of posteriorgrams. For example, a phoneme recognizer based on multilayer perceptrons (MLP) can directly produce phoneme posteriorgrams, and a GMM tokenizer may generate Gaussian posteriorgrams from the acoustic likelihoods according to the Bayes' theorem.

3. PROPOSED SYSTEM

Fig. 2 shows the proposed system. It is modified from the posteriorgram-based template matching framework, and was used as our main system in the MediaEval2012 SWS evaluation [17]. The system involves N different tokenizers, which are expected to be complementary to each other. Details of the tokenizer implementation are presented in Section 4.1. The N tokenizers convert the query example and the test utterance into N types of posteriorgrams, which are used to compute N distance matrices. These distance matrices are then merged into one distance matrix, on which DTW is performed to derive the raw detection score. Lastly score post-processing techniques are applied.

3.1. DTW Matrix Combination

Let $\boldsymbol{Q}_n$ denote the query posteriorgram generated by the n_{th} tokenizer, and let $\boldsymbol{T}_n$ denote the corresponding test posteriorgram. The distance matrix $\boldsymbol{D}_n$ can be computed with the inner-product distance [1] as follows:

$$\boldsymbol{D}_n = -\log(\boldsymbol{Q}_n^T \times \boldsymbol{T}_n). \tag{3}$$

Given a query example and a test utterance, the sizes of all the distance matrices are the same and independent of the tokenizers . Thus a linear combination of these matrices can be obtained,

$$\boldsymbol{D} = \sum_{n=1}^{N} w_n \boldsymbol{D}_n, \tag{4}$$

where w_n is the weighting coefficient for the DTW distance matrix $\boldsymbol{D}_n$. In this study, w_n was set to $\frac{1}{N}$. We leave the design of more sophisticated weighting strategies for future work.

3.2. DTW Detection

Based on the combined distance matrix $\boldsymbol{D}$, DTW detection is performed with a sliding window and an adaptive adjustment window constraint. The sliding window moves along the test utterance with one frame forward at each step. Let I denote the length of the query example, and let J denote the width of the sliding window. The DTW detection for each window operates on an $I \times J$ matrix $\tilde{\boldsymbol{D}}$, which is extracted from the distance matrix $\boldsymbol{D}$. The detection score S is negative to the normalized alignment distance,

$$S = -\min_{L, i(l), j(l)} \frac{1}{L} \sum_{l=1}^{L} \tilde{\boldsymbol{D}}(i(l), j(l))), \tag{5}$$

where $i(l)$ and $j(l)$ denote the coordinates of the l_{th} step of the alignment path, and L is the length of the alignment path.

To avoid excessive temporal distortion, the adjustment window constraint [18] is imposed on the alignment path: $|i(l) - j(l)| \leq R$. Because of the significant length variation of query examples, R is not set to a fixed number, but made adaptive to the query length I: $R = \alpha I$, where the proportional coefficient α controls the allowed path range. With this restriction, we have $J \geq (1+\alpha)I$ and $(1-\alpha)I \leq j(L) \leq (1+\alpha)I$. α was tuned on the development data and was set to $\frac{1}{3}$ in this work.

3.3. Score Post-processing

3.3.1. Pseudo-Relevance Feedback

Pseudo-relevance feedback (PRF) [19] is commonly used for score re-ranking in information retrieval. PRF has recently been introduced to STD tasks [20, 21]. For QbyE STD, the motivation of using PRF is to expand the query example set by treating the top retrieved speech segments as new query examples. Given a query term, the PRF used in our work is implemented in the following steps:

1) *Relevance example selection:* From all the development and test utterances, the top H ($H \leq 3$ in this study) hit regions with the raw scores greater than a pre-set threshold are selected as the *relevance* examples.

2) *Candidate region selection:* In each test utterance, the hit region with the highest raw score is selected as the candidate region for this query. Only the selected candidate regions are considered in the following steps.

3) *Scoring by relevance examples:* The query relevance examples selected in step 1 are used to score the candidate regions selected in step 2. Here the scoring is an exact DTW operation.

4) *Score fusion:* The scores obtained by the relevance examples are linearly fused with the scores given by the original query examples. For simplicity, equal weighting coefficients are used for the linear fusion.

3.3.2. Score Normalization

One potential problem with the DTW detection scores is the diversity of score distributions of different query terms. This may arise from the phonetic content variation, speaker variation or other factors. The score diversity of different query terms makes it troublesome to find a global threshold for the acceptance or rejection decision. One way to handle this problem is normalizing the score distributions of different query terms. In this work, we used a simple score range normalization scheme for each query term as follows:

$$\hat{S}_t = (S_t - S_{min})/(S_{max} - S_{min}), \quad (6)$$

where S_t is the score from the t_{th} utterance. S_{min} and S_{max} are the minimum and maximum scores for the query term from all the retrieved utterances.

4. EXPERIMENTAL SETUP

4.1. Tokenizer Implementation

If a tokenizer is trained from labeled data, it is called a *supervised* tokenizer, e.g., a phoneme recognizer. For low-resource STD, the supervised tokenizers are usually borrowed from rich-resource languages, and hence suffer from the language mismatch problem. If trained from unlabeled data, the tokenizer is called an *unsupervised* tokenizer, e.g., a GMM tokenizer [14]. Unsupervised tokenizers are trained from unlabeled data from the test languages, thus do not have the language-mismatch problem. However the unsupervised training makes them more sensitive to speaker and channel variations. This motivates the development of a *semi-supervised* tandem tokenizer, in which a supervised tokenizer is used as the front-end to generate more robust features from the unlabeled data, and the generated features are then transformed and modeled in an unsupervised way.

Unsupervised tokenizers: Two unsupervised tokenizers were implemented from the development audio data of the test languages. The first was a GMM tokenizer with 1024 Gaussian components. The other was an ASM tokenizer containing 256 units. Each ASM unit had 3 states with 16 gaussian components at each state. Detailed training procedure of the ASM tokenizer can be found in [9]. Both the GMM tokenizer and the ASM tokenizer took in 39-dimensional MFCC features, which were processed with utterance-based mean and variance normalization (MVN) and vocal tract length normalization (VTLN). Both the GMM posteriorgrams and the ASM posteriorgrams were computed from the acoustic likelihoods.

Supervised tokenizers: The supervised tokenizers used in the experiments were the three MLP-based phoneme recognizers developed by Brno University of Technology: Czech (CZ), Hungarian (HU) and Russian (RU) phoneme recognizers [22]. All these phoneme recognizers used the split temporal context network structure [23]. Phoneme posteriorgrams were directly obtained from the merger MLP outputs.

Semi-supervised tandem tokenizer: The tandem tokenizer was built and used in a hybrid way. First a MLP-based phoneme recognizer was used to produce posterior features on the unlabeled data of the test languages. Then the posterior features were transformed by taking logarithm, PCA transformation, and MVN. Quantitative analysis in [6] has shown that the Log-MLP features are more robust than spectral features, and are suitable for Gaussian modeling. Lastly a GMM with 256 components was trained from the transformed features of the development data. With the resultant GMM, all the transformed features could be converted into posteriorgrams, which were the final outputs of the tandem tokenizer for this task. Using the three phoneme recognizers mentioned above, we developed corresponding tandem tokenizers which are referred to as CZ-GMM, HU-GMM, and RU-GMM.

4.2. Datasets and Evaluation Metrics

We carried out experiments on the MediaEval SWS2011 dataset [12] and MediaEval SWS2012 dataset [13]. Both datasets contain their own development set and evaluation set which are all telephone speech data. The development set and the evaluation set involve separate query terms and test utterances. The system parameters were tuned on the development set. The experimental results on the evaluation set are reported in the next section. The audio data from the development set was used to train the unsupervised tokenizers and the tandem tokenizers. The SWS2011 development set contains about 2 hours of 400 utterances and 64 query terms, and the SWS2011 evaluation set contains about 0.8 hour of 200 utterances and 36 query terms. The SWS2012 development set contains about 3.7 hours of 1580 utterances and 100 query terms, and the SWS2012 evaluation set contains about 3.9 hours of 1660 utterances and 100 query terms. For both datasets, each query term has only one audio example. The SWS2011 data involves four different Indian languages, namely English, Hindi, Gujarati and Telugu. The SWS2012 data involves four different African languages, namely isiNdebele, Siswati, Tshivenda and Xitsonga. No language labels were used in the experiments.

The performances were evaluated in terms of the non-interpolated mean average precision (MAP) and equal error rate (EER) calculated on the evaluation sets of both datasets. Both the metrics were measured on a per utterance basis [1], i.e., a trial involves a query term and a test utterance.

5. EXPERIMENTAL RESULTS AND ANALYSIS

5.1. Performances of Individual Tokenizers

Table 1 shows the performance of individual tokenizers. Comparing the unsupervised tokenizers with the supervised tokenizers, it can be observed that the two unsupervised tokenizers generally performed better. We believe that this is due to the language mismatch between the supervised tokenizers and the evaluation data.

Table 1. Performances of individual tokenizers

Tokenizer		SWS2011		SWS2012	
		MAP	EER	MAP	EER
Unsupervised	GMM	**0.451**	0.350	**0.487**	0.170
	ASM	0.414	0.345	0.444	0.174
Supervised	CZ	0.344	0.381	0.361	0.208
	HU	0.363	0.371	0.335	0.234
	RU	0.362	0.350	0.372	0.217
Semi-supervised	CZ-GMM	0.369	0.381	0.439	0.162
	HU-GMM	0.407	**0.335**	0.480	**0.153**
	RU-GMM	0.368	0.371	0.477	0.164

From Table 1, it can also be observed that the tandem tokenizers showed consistent improvements over the supervised tokenizers. And the performances of the tandem tokenizers were comparable to those of the unsupervised tokenizers. Specifically, compared to

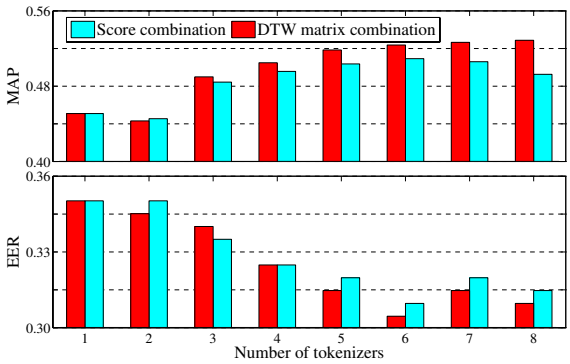

Fig. 3. Performances on SWS2011 evaluation set

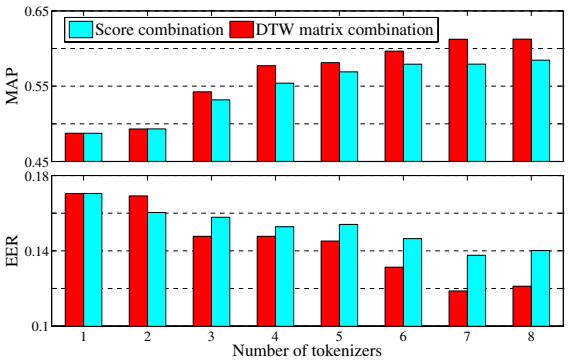

Fig. 4. Performances on SWS2012 evaluation set

the performances of the Hungarian phoneme recognizer (HU), HU-GMM tandem tokenizer brought 12.1% and 43.3% relative MAP improvements on SWS2011 set and SWS2012 set respectively, and the HU-GMM tokenizer achieved the best EER performances among all the 8 tokenizers.

5.2. Effectiveness of DTW Matrix Combination

We then evaluated the performances of the DTW matrix combination approach. As mentioned in Section 3.1, equal weights were used for combining the DTW matrices. Corresponding performances versus the number of tokenizers used in the system are given in Fig. 3 and Fig. 4. Tokenizers were gradually added into the system in the top-down order listed in the second column of Table 1, i.e., in Fig. 3 and Fig. 4, 1 tokenizer corresponds to the GMM tokenizer, 2 tokenizers correspond to the combination of the GMM and the ASM tokenizers, etc. Roughly speaking, with the DTW matrix combination approach, more tokenizers led to better performances. When using all the 8 tokenizers, the DTW matrix combination approach significantly outperformed the best single tokenizer, with a 17.3% relative MAP improvement on SWS2011 evaluation set and a 25.7% relative MAP improvement on the SWS2012 evaluation set.

Fig. 3 and Fig. 4 also show the performances of the score combination method. We implemented the score combination method in a two-pass way. For the first pass, we used the GMM tokenizer to determine the best matching region in each utterance as the *candidate regions*. For the second pass, all the tokenizers were used for scoring between the query examples and the *candidate regions* obtained in the first pass. Then the second-pass scores from all the tokenizers were linearly fused with equal weights. The reason for choosing GMM tokenizer in the first pass is because it gave the best MAP performances among all the single tokenizers. As can be seen, the advantage of the DTW matrix combination approach over the score combination method becomes more significant as the number of tokenizers increases. On the SWS2012 evaluation set, using all the 8 tokenizers with DTW matrix combination approach could provide a 4.78% relative MAP improvement and a 13.5% relative EER reduction over the score combination approach. These demonstrate the effectiveness of the DTW matrix combination approach.

5.3. Effectiveness of Score Post-processing

In the final part of the experiments, we examined the performances of the two post-processing techniques: pseudo relevance feedback (PRF) and score normalization (Norm.). Corresponding results are listed in Table 2. The raw detection scores were obtained using all the 8 tokenizers with the DTW matrix combination approach. PRF brought consistent performance improvements on both the evaluation sets in both evaluation metrics. This verifies the effectiveness of using top retrieved speech segments as relevance examples to enhance the acoustic representation ability. Since the score normalization approach used in this paper did not change the ranking positions of the retrieved utterances, it did not affect the MAP performances. When applying the score normalization to the raw detection scores, it led to a 6.77% relative EER reduction on the SWS2011 set, and a 12.4% relative EER reduction on the SWS2012 set. However, when applying score normalization to the results of PRF, the performances did not change significantly, i.e., a relative EER reduction of 5.45% was obtained on SWS2012 set, but a relative 1.73% EER degradation was observed on SWS2011 set. This is probably because the score distribution differences of different query terms have already been reduced by using multiple speech examples.

Table 2. System performances with score post-processing

	SWS2011		SWS2012	
	MAP	EER	MAP	EER
raw score	0.529	0.310	0.612	0.121
+PRF	0.546	0.289	0.623	0.110
+Norm.	0.529	0.289	0.612	0.106
+PRF +Norm.	0.546	0.294	0.623	0.104

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a DTW matrix combination approach for using multiple tokenizers as the front-end in the low-resource QbyE STD task. Our approach is based on the combination of the DTW matrices rather than the detection scores, and thereby does not need to make hypotheses on the possible detection regions. The advantage of our approach was verified on SWS2011 and SWS2012 evaluation datasets. To reduce the language mismatch between the supervised tokenizers and the test languages, the use of semi-supervised tandem tokenizers was proposed and validated in the experiments. Moreover, two score post-processing approaches further show promising gains on top of the use of multiple tokenizers. For future work, we may investigate more sophisticated strategies to adjust the weighting coefficients of the DTW matrix combination, as well as more advanced score post-processing techniques.

7. ACKNOWLEDGMENT

This research is partially supported by the General Research Funds (Ref: 414010 and 413811) from the Hong Kong Research Grants Council.

8. REFERENCES

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

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

[3] A. Jansen, K. Church, and H. Hermansky, "Towards spoken term discovery at scale with zero resources," in *Proc. INTERSPEECH*, 2010, pp. 1676–1679.

[4] J. Tejedor, M. Fapšo, I. Szöke, J. Černockỳ, F. Grézl, et al., "Comparison of methods for language-dependent and language-independent query-by-example spoken term detection," *ACM Trans. on Information Systems*, vol. 30, no. 3, 2012.

[5] G. Aradilla, J. Vepa, and H. Bourlard, "Using posterior-based features in template matching for speech recognition," in *Proc. INTERSPEECH*, 2006, pp. 1186–1189.

[6] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "Shifted-delta MLP features for spoken language recognition," *Signal Processing Letters, IEEE*, vol. 20, no. 1, pp. 15–18, 2013.

[7] Y. Zhang, R. Salakhutdinov, H.A. Chang, and J. Glass, "Resource configurable spoken query detection using deep boltzmann machines," in *Proc. ICASSP*, 2012, pp. 5161–5164.

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

[9] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. ICASSP*, 2012, pp. 5157–5160.

[10] A. Muscariello, G. Gravier, and F. Bimbot, "Zero-resource audio-only spoken term detection based on a combination of template matching techniques," in *Proc. INTERSPEECH*, 2011, pp. 921–924.

[11] A. Abad and R. Astudillo, "The L2F spoken web search system for MediaEval 2012," in *Working Notes Proceedings of the MediaEval 2012 Workshop*, 2012.

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

[13] F. Metze, E. Barnard, M. Davel, C. van Heerden, X. Anguera, G. Gravier, and N. Rajput, "The spoken web search task," in *Working Notes Proceedings of the MediaEval 2012 Workshop*, 2012.

[14] P. Torres-Carrasquillo, D. Reynolds, and J. Deller, "Language identification using Gaussian mixture model tokenization," in *Proc. ICASSP*, 2002, pp. I.757–I.760.

[15] C.-H. Lee, F. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Proc. ICASSP*, 1988, pp. 501–541.

[16] M.-H. Siu, H. Gish, A. Chan, and W. Belfield, "Improved topic classification and keyword discovery using an HMM-based speech recognizer trained without supervision," in *Proc. INTERSPEECH*, 2010, pp. 2838–2841.

[17] H. Wang and T. Lee, "CUHK system for the spoken web search task at MediaEval 2012," in *Working Notes Proceedings of the MediaEval 2012 Workshop*, 2012.

[18] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 26, no. 1, pp. 43–49, 1978.

[19] C. Buckley, G. Salton, and J. Allan, "Automatic retrieval with locality information using smart," in *Proc. TREC-1*, 1993, pp. 59–72.

[20] C. Chen, H. Lee, C. Yeh, and L. Lee, "Improved spoken term detection by feature space pseudo-relevance feedback," in *Proc. INTERSPEECH*, 2010, pp. 1672–1675.

[21] C. Chan and L. Lee, "Integrating frame-based and segment-based dynamic time warping for unsupervised spoken term detection with spoken queries," in *Proc. ICASSP*, 2011, pp. 5652–5655.

[22] P. Schwarz, *Phoneme recognition based on long temporal context*, Ph.D. thesis, Brno University of Technology, 2009.

[23] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *Proc. ICASSP*, 2006, pp. 325–328.