CUHK System for the Spoken Web Search task at Mediaeval 2012

Haipeng Wang, Tan Lee
DSP-STL, Dept. of EE,
The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{hpwang,tanlee}@ee.cuhk.edu.hk

ABSTRACT

This paper describes our systems submitted to the spoken web search (SWS) task at MediaEval 2012. All the systems were based on a new framework which is modified from the posteriorgram-based template matching approach. This framework employs parallel tokenizers to convert audio data into posteriorgrams, and then combine the distance matrices from the posteriorgrams of different tokenizers to derive a combined distance matrix. Lastly dynamic time warping (DTW) is applied to the combined distance matrix to detect the possible occurrences of the query terms. For this SWS task, we used three types of tokenizers, namely Gaussian mixture model (GMM) tokenizer, acoustic segment model (ASM) tokenizer, and phoneme recognizers of rich-resource languages. Pseudo-relevance feedback (PRF) and score normalization were also used in some of the systems.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation, Languages

Keywords

Query-by-example, spoken term detection, low-resource language, parallel tokenizers followed by DTW detection

1. INTRODUCTION

The spoken web search (SWS) task of MediaEval 2012 is to locate the occurrences of the query terms in the audio archive using audio content queries [3]. All our systems were based on the same framework: parallel tokenizers followed by DTW detection (PTDTW). This framework was modified from the posteriorgram-based template matching approach [2]. The motivation of developing this framework was to exploit the complementary nature of different tokenizers when building the DTW distance matrix.

2. RESOURCES

We submitted the results of five systems for this task. System 1 and system 2 were built using only the provided development audio data without transcriptions. System 3, 4 and 5 used more resources, including the provided development audio data without transcriptions, the three Brno phoneme recognizers (Czech, Hungarian and Russian), which were trained on the SpeechDat-E corpus with transcriptions [1, 4], one English phoneme recognizer trained on about 15-hour audio data with transcriptions from the Fisher corpus and Swichboard Cellular corpus, and one Mandarin phoneme recognizer trained on about 15-hour audio data with transcriptions from the CallHome corpus and the CallFriend corpus.

MediaEval 2012 Workshop, October 4-5, 2012, Pisa, Italy

3. SYSTEM DESCRIPTION

3.1 PTDTW framework

Fig. 1 illustrates our proposed PTDTW framework. It involves N parallel tokenizers, which are either trained from the development data given in the SWS task, or developed from some rich-resource languages. Using these tokenizers, the query examples and test utterances are converted into frame-level posteriorgrams. Each of the N tokenizers may use a different algorithm to compute its posteriorgrams. For each type of posteriorgrams, the DTW distance matrix $D_i (1 \leq i \leq N)$ is computed. To take advantage of the complementary information from different tokenizers, the N distance matrices are combined linearly to give a new distance matrix, i.e., $D = \sum_{i=1}^{N} w_i D_i$. For simplicity, the weighting coefficient w_i was set to $\frac{1}{N}$ in our systems. Subsequently DTW detection is applied to D to compute the raw detection score.

3.2 Tokenizers

The first type of tokenizer we used is the GMM tokenizer. In our implementation, the GMM tokenizer contained 1024 Gaussian components trained from the provided development data. The input of the GMM tokenizer was 39-dimensional MFCC feature vector. The MFCC features were further processed with voice activity detection (VAD), sentence-based mean and variance normalization (MVN) and vocal tract length normalization (VTLN). This tokenizer is referred to as MFCC-GMM.

The second type of tokenizer is the ASM tokenizer, which was also trained using only the development data. The ASM tokenizer was trained in an unsupervised manner. Details of ASM training are shown in [5]. The ASM tokenizer contained 256 units. Each unit owned 3 states with 16 gaussian components for each state. The input features for the ASM tokenizer were the same as those for the GMM tokenizer. The tokenizer is named as MFCC-ASM.

The third type of tokenizers are phoneme recognizers. These recognizers were trained on rich-resource languages, for which labeled training data are readily available. As mentioned in Section 2, we used five phoneme recognizers, namely Czech (CZ), Hungarian (HU), Russian (RU), English (EN) and Mandarin (MA) phoneme recognizers. All these phoneme recognizers used the split temporal context network structure [4]. They performed as the front-end to produce posterior features. The posterior features were further processed by taking logarithm and PCA transformation, and then modeled by 256 Gaussian mixtures which were trained on the development data. The 256-dimensional Gaussian posteriorgrams were the final output of this type of tokenizers. These tokenizers are respec-

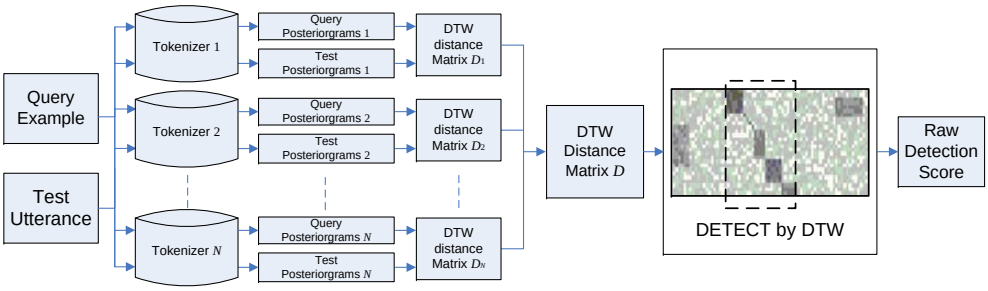

Figure 1: PTDTW Framework

tively referred to as CZ-GMM, HU-GMM, RU-GMM, EM-GMM, and MA-GMM.

3.3 DTW Detection

DTW detection is performed with a sliding window, which moves along the test utterance with one frame forward for 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 a $I \times J$ distance matrix, which is extracted from the whole distance matrix D. The DTW alignment path maximize the detection score, which is negative to the following normalized distance:

$$\hat{d} = \min_{K,i(k),j(k)} \frac{\sum_1^K d(i(k),j(k))w_k}{Z(w)} \quad (1)$$

where $i(k)$ and $j(k)$ denote the coordinates of the $k_{th}(1 \leq k \leq K)$ step of the alignment path. For this SWS task, $d(i(k),j(k))$ was computed as the inner-product distance [2]. The weighting coefficient w_k was set to 1, and the normalization factor $Z(w)$ was set to K.

An additional restriction was imposed on the alignment path: $|i(k) - j(k)| \leq R$, where R defines the allowed range of the path. With this restriction, we have $J \geq I + R$ and $I - R \leq j(K) \leq I + R$. Because of the significant variation of the query length, R was not set to a fixed number but was made proportional to the query length I, i.e.,$R = \alpha I$. α was set to $\frac{1}{3}$ in our experiment.

3.4 PRF and Score Normalization

After obtaining the DTW detection raw scores, the Pseudo-relevance feedback (PRF) we used could be divided into 3 steps for each query:

1) The top H hits from all the test utterances were selected as the *relevance* examples. Selection criterion included: a) $H \leq 3$; b) raw detection score should be larger than a pre-set threshold.

2) The *relevance* examples were used to score the top $\hat{H}$ ($\hat{H} = 2$ for this task) hits from each utterance.

3) The scores obtained by the *relevance* examples were linearly fused with the scores of the original query examples.

Score normalization was applied to normalize the score as $\hat{s}_{q,t} = (s_{q,t} - \mu_q)/\delta_q$, where $s_{q,t}$ is the score of the q_{th} query on the t_{th} hit region. μ_q and δ_q^2 are the mean and variance of the scores for the q_{th} query estimated from the development data.

4. CONFIGURATION AND PERFORMANCE

We submitted the results of five different systems. Their configurations and performances are given in Table 1. The performance metric is ATWV [3]. System 1 and 2 belong to the *required run* condition as defined in [3]. The other three utilized the phoneme recognizers of other languages, so they belong to *general run* condition. System 1 and system 4 were the *primary* systems for the two conditions respectively. As can be seen, using parallel phoneme recognizers from rich-resource languages provide quite promising performances. The combination of unsupervised tokenizers (GMM and ASM) and these phoneme recognizers improves the system performance. Moreover, PRF leads to consistent improvements.

Table 1: System Configurations (rows 2-10) and ATWV performances (rows 11-14).

System No.	1	2	3	4	5
MFCC-GMM	√	√		√	√
MFCC-ASM	√	√		√	√
CZ-GMM			√	√	√
HU-GMM			√	√	√
RU-GMM			√	√	√
MA-GMM			√	√	√
EN-GMM			√	√	√
PRF	√			√	
Score Normalization	√	√	√	√	√
devQ - devC	0.68	0.63	0.73	0.78	0.74
devQ - evlC	0.60	0.55	0.70	0.75	0.70
evlQ - devC	0.68	0.65	0.73	0.77	0.75
evlQ - evlC	0.64	0.59	0.72	0.74	0.74

5. CONCLUSION

We have presented the proposed PTDTW framework, which can effectively combine different tokenizers for the query-by-example spoken term detection task. The modification of DTW detection, the PRF technique and the score normalization have also been described. Promising results are obtained from the SWS evaluation.

6. ACKNOWLEDGMENTS

The authors would like to thank Cheung-Chi Leung for helpful discussions in developing these systems. This research is partially supported by the General Research Funds (Ref: 414010 and 413811) from the Hong Kong Research Grants Council.

7. REFERENCES

[1] http://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context.

[2] T. Hazen, W. Shen, and C. White. Query-by-example spoken term detection using phonetic posteriorgram templates. In *ASRU*, pages 421–426, 2009.

[3] F. Metze, E. Barnard, M. Davel, C. van Heerden, X. Anguera, G. Gravier, and N. Rajput. The spoken web search task. In *MediaEval 2012 Workshop*, 2012.

[4] P. Schwarz. Phoneme recognition based on long temporal context, PhD thesis. 2009.

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