Query-by-example spoken term detection based on phonetic posteriorgram Query-by-example spoken term detection based on phonetic posteriorgram

Beili Song[1, a], Wei-Qiang Zhang[1,b], Meng Cai[1], Jia Liu[1] and Michael T. Johnson[2]

[1]Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

[2] Department of Electrical and Computer Engineering, Marquette University, WI 53233, USA

[a]songbl12@mails.tsinghua.edu.cn, [b]wqzhang@tsinghua.edu.cn

Keywords: query-by-example, spoken term detection, softmax output features, dynamic time warping.

Abstract. Spoken term detection in low-resource situations is a challenging problem, because traditional large vocabulary continuous speech recognition (LVCSR) approaches are often unusable. This paper introduces a method to use deep neural network (DNN) softmax outputs as input features in a query-by-example (QBE) spoken term detection (STD) system. Matches between queries and test utterances are located using a modified dynamic time warping (DTW) search approach. Subsystems are built with unsupervised Gaussian mixture model (GMM) and DNN monophone models trained on Chinese and English languages and evaluated on the SWS 2013 multilingual database of low-resource languages. The score-level fusion of these different subsystems are shown to improve performance significantly over the baseline results.

Introduction

Spoken term detection (STD) is the task of finding and locating a given query term in a set of audio documents. With the rapid development of the internet and mobile devices, spoken term detection offers an approach to obtain valuable training resources from the large amount of online speech data. Systems based on large vocabulary continuous speech recognition (LVCSR) have significantly increased the accuracy of STD [1], but LVCSR systems require sufficient language resources, including tens of hours of transcription and a reliable pronunciation dictionary. In many real world scenarios, however, getting such resources is either too difficult or too expensive, which makes traditional LVCSR methods infeasible. Even the few resources that are available are often acoustically mismatched, which leads to severe performance degradations.

Instead of giving queries in the form of text, the QBE approach uses a few audio segments containing keywords of interest as queries. The system then can search for segments that closely match these query examples from test data. The effectiveness of QBE systems relies on two key factors, the choice of features and an effective search algorithm. Choosing suitable features is the most important. A typical template-based approach simply uses acoustical features and a dynamic time warping (DTW) algorithm to match templates to test data [2, 3]. The use of posteriorgram features in template-matching can alleviate mismatches in speakers and environments. These posteriorgrams can be derived from phonetic lattices [4] or directly from a phone tokenizer [5,7]. For unsupervised keyword spotting tasks, a Gaussian posteriorgram based system has been proposed [6]. All these different types of posteriorgrams try to represent a frame of audio data representing the speaker/environment as closely as possible, so the acoustic model will model the probability distributions over vectors of input acoustic features accurately. In [5] and [7], artificial neural network (ANN) based phoneme-state estimators are used, which achieve the best results in the SWS2013 evaluations [8].

Deep neural networks (DNNs) with many hidden layers have shown to outperform Gaussian mixture models (GMMs) at acoustic modeling, and are now widely used in a variety of speech

recognition tasks [9]. In this paper, we apply two DNN models trained on datasets of two different languages, Mandarin Chinese and English, to generate two sets of phone posteriorgrams as system inputs, and then combine these systems at the score level. The Chinese model is trained on 700 hours of Chinese speech. It has 5 hidden layers and an output layer with 98 units, which present 96 Chinese monophones and 2 non-speech monophones. The second DNN model is trained on 700 hours of English, with 5 hidden layers and an output layer with 41 units corresponding to English phonemes. The softmax output of the DNN models are used directly as input features of our QBE system.

For the search algorithm, we adopt the modified DTW search used in [4] and [5]. Topological constraints are tied to the search process to improve search speed.

For each match between the query and the test utterance that is reported, we get two scores. The baseline system uses the average distance as the score of the query. This score is then adjusted by subtracting the average of the highest two scores of all other queries in the same position. We calibrate these two sets of scores separately using z-norm. These two sets are then linearly added to produce the final scores of a single subsystem. The results of the different models are linearly combined at the score level.

Details of the QBE system are described in next section. Then we introduce the database and show experimental results. Finally, conclusions are summarized.

System description

In this Section, we give a simple description of our QBE-STD system, including feature extraction, DTW-based query search and score calibration.

Feature extraction. Our QBE system is designed to deal with low resource conditions. There are two basic methods, using unsupervised training to get language independent features, or applying phone decoders trained on a specific language to produce features regardless of the actual language.

For the unsupervised method, a GMM is trained on all the data in the database and is used to produce a Gaussian posteriorgram vector for each speech frame:

$$q_i = (P(C_1 \mid s_i), P(C_2 \mid s_i), ..., P(C_m \mid s_i)) . \quad (1)$$

where s_i denotes the i_{th} speech frame and C_j is the j_{th} Gaussian component of the GMM. In this paper the number of Gaussian components m is set as 50.

Compared to GMMs, DNNs have three basic advantages: (1) they have long window frames and can model feature context; (2) they have many nonlinear layers, with a corresponding strong capacity for feature transformation; (3) they model acoustic states directly, which more directly ties to the recognition task. The application of DNNs can significantly improve the performances of STD, as has been verified in a series of evaluations, most notably at OpenKWS 2013. In this paper, we try to generalize the use of DNNs in model training for QBE-STD systems. A DNN decoder is used to produce phone posteriorgrams for both the queries and audio documents.

For the Chinese model, a DNN with 5 hidden layers was trained on 700 hours of Chinese data collected by the Tsinghua University Speech Technology lab. The Chinese softmax output layer has 98 units, include 96 Chinese monophones and 2 monophones representing non-speech. For the English model, a DNN with 5 hidden layers and a softmax output layer with 41 units was trained on 700 hours of English data from Switchboard dataset and Fisher dataset. More details about DNN model training can be seen in [11].

Observing specific audio signals in the data sets, we can see two characteristics that may affect the feature match accuracy. The first is that there are many non-speech fragments in the audio signals. Especially for queries, there may be non-speech segments near the beginning, middle and end of the signal, which can significantly affect recognition performance. Another problem is that in many query examples, the amount of effective audio is insufficient to represent a complete word. Using such audio as training templates may lead to a higher false alarm rate.

To solve these problems, a speech activity detection (SAD) step is necessary. Similar to the method given in [4], we calculate the non-speech posterior probability (the sum of phone probabilities representing short and long silence segments). If the non-speech posterior exceeds a pre-set threshold, the frame is classified as non-speech and discarded from the posteriorgram feature vector. After removing non-speech frames, if the number of remaining frames is still insufficient, the entire utterance is discarded.

Dynamic Time Warping Search. When comparing a query posteriorgram sequence of length m, $\mathbf{q} = (q_1, q_2, ..., q_m)$, against a test posteriorgram sequence of length n, $\mathbf{x} = (x_1, x_2, ..., x_n)$, we use the dot product log probability $D[i, j]$ to measure the similarity between q_i and x_j:

$$D[i, j] = -\log(q_i \cdot x_j). \tag{2}$$

The average distance between q and x is calculated as:

$$\bar{D} = \frac{1}{L}\sum_{l=1}^{L} D[i_l, j_l]. \tag{3}$$

where i_l and j_l represent the l_{th} alignment of the q and x vectors. Endpoint conditions are given by $i_1 = 1, i_l = m$, $j_1 = k_1$ and $j_l = k_2$, $1 \le k_1 < k_2 \le n$.

To find the best match of a query q against a test utterance x efficiently, we adopt the modified DTW search algorithm used in [4, 5]. The DTW procedure is given as follows:

(1) Initialize the boundaries:

i =1:

$$\begin{cases} a[1, j] = D[1, j] \\ l[1, j] = 1 \end{cases} \quad \text{for } j = 1, \ldots, n \tag{4}$$

j =1:

$$\begin{cases} a[i,1] = \sum_{k=1}^{i} D[\mathrm{k},1] \\ l[i,1] = i \end{cases} \quad \text{for } i = 1, \ldots, n \tag{5}$$

(2) Implement dynamic programming with a topology constraint:

i >1:

$$\Omega = \{(i, j-1), (i-1, j), (i-1, j-2)\}. \tag{6}$$

$$(r, s) = \arg \min_{(p,q)\in\Omega} \frac{a[p, q] + D[i, j]}{l[p, q] + 1}. \tag{7}$$

$$\begin{cases} a[i, j] = a[r, s] + D[i, j] \\ l[i, j] = l[r, s] + 1 \end{cases}. \tag{8}$$

(3) Apply end conditions $i = m$.

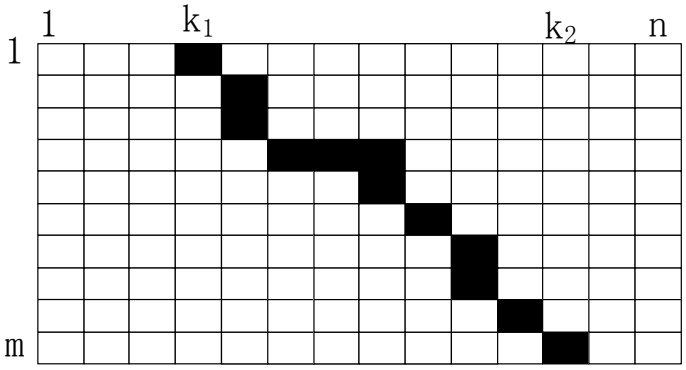

Fig. 1. Example of similarity matrices between a query and a test utterance. The black matrices show a possible crossing path.

Using this algorithm, the average distance is taken from the score of the target keyword during the time interval $[t_s, t_e]$, where the beginning time t_s and ending time t_e can be obtained from the SAD results corresponding to k_1 and k_2. An example can be seen in Figure 1.

The constraints imply that the path can be extended by only one frame, both in the query and in the test utterance. Thus, when calculating the minimum accumulated distance of a node, only the minimum accumulated distances of three former nodes are needed. This is guaranteed by the traversal order of the similarity matrices. In each test utterance, only the best 5 non-overlapping matches are recorded.

Score calibration. The evaluation dataset contains queries with durations across a large range, collected from multiple language families and continents, as discussed in Section 3. This makes it difficult to determine a common decision threshold for all queries. To solve this problem and improve overall performance, score calibration is necessary. To accomplish this, the scores of all possible matches for each query are ranked and the *N* highest scores are retained (we choose *N* = 500 for these experiments).

Observing the DTW search results empirically, we find that there may be a few putative hits with the same ranking. To reduce the influence of test data, we calculate the scores of all other queries in the positions of putative hits and subtract the average score of the highest two queries:

$$s_{adjusted,t} = s_{q,t} - (s_{q_1,t} + s_{q_2,t})/2. \quad (9)$$

where $s_{q,t}$ is the raw score of the t_{th} putative keyword for the q_{th} query, and $s_{q_1,t}$ and $s_{q_2,t}$ are the highest two scores of all other simultaneous queries other than q.

A z-norm score normalization [9] for both the raw score and the adjusted score of each query is then implemented using:

$$\hat{s}_{q,t} = (s_{q,t} - \mu_q)/\delta_q. \quad (10)$$

where $s_{q,t}$ is the score of the t_{th} putative keyword for the q_{th} query., and μ_q and δ_q^2 are the mean and variance of the N-best scores for the q_{th} query.

Finally, the results of raw scores and adjusted scores after z-normalization are linearly combined using MTWV weights. This result will be used for subsystem combination.

Data and experimental results

Our DTW system based on unsupervised Gaussian posteriorgrams and DNN are tested on the SWS 2013 multilingual database [8]. The experiment uses a single example for each development query. The database contains 9 languages from both European and African language families. It has a variety of speaking styles and acoustic conditions, to evaluate the system performance across multiple low/zero resource conditions.

The maximum term weighted value (MTWV) is used as performance metric:

$$TWV(\theta) = 1 - [P_{Miss}(\theta) + \beta \cdot P_{FA}(\theta)]. \quad (11)$$

where θ is the threshold giving highest TWV, $P_{Miss}(\theta)$ is the term-weighted miss probability, $P_{FA}(\theta)$ is the term-weighted false alarm probability, and β is the false alarm cost.

Chinese/English DNN subsystem. We use the results of the 96-output Chinese DNN and 41 output English DNN as two DNN subsystem. The MTWV results of both raw and adjusted scores, with and without z-norm normalization are shown in Table 1.

Table 1. Results using Chinese/English DNN.

Type of scores	DNN	
	Chinese	English
raw	0.1175	0.1207
adjusted	0.1704	0.1871
raw/z-norm	0.2195	0.2232
adjusted/z-norm	0.2178	0.2323
raw-adjusted score fusion	0.2344	0.2527

Fusion with adjusted scores. Table 1 also shows that the adjusted scores obtain better MTWV than raw scores. After z-normalization, raw and adjusted scores from Chinese DNN have similar MTWV. These are then combined to give the final subsystem, which has better MTWV than either types of score individually.

Fusion of different languages. Combining the previously shown Chinese DNN-based baseline system with an English DNN system, Table 2 shows the final MTWV scores of the final composite system. The search results of both DNN models are calibrated and combined with linear fusion using MTWV weights, which improves the MTWV score to 0.3060.

Table2. Fusion of different languages.

Model		MTWV
DNN	Chinese	0.2344
	English	0.2527
	Chinese-English fusion	0.3060

Fusion of DNNs and unsupervised GMM. As shown in Table 3, the result of a comparable unsupervised GMM subsystem is much worse than the DNN system. To combine these, the z-normalized GMM and DNN scores are MTWV weighted and summed. This combination yields a substantial additional MTWV improvement, from 0.3060 to 0.3369.

Table3. Fusion of DNNs and unsupervised GMM.

Model		MTWV
DNN	Chinese	0.2344
	English	0.2527
GMM		0.1790
DNN-GMM fusion		0.3369

Summary

We adopt unsupervised Gaussian posteriorgram and DNN phonetic posteriorgram features as inputs to a DTW-based QBE system. DNN results are significantly better than comparable GMM-based systems, consistent with recent findings on other speech applications. We calibrate the scores with an N-best query approach to normalize the score range, and combine multiple normalized adjusted scores for best results. The combination of multiple language complementary DNN models and unsupervised GMM significantly improves overall system performance.

References

[1] D. Miller, et al, “Rapid and accurate spoken term detection,” in *Proc. Interspeech*, 2007, pp. 314-317.

[2] H. Ney, "The use of a one-stage dynamic programming algori-thm for connected word recognition," *IEEE Transactions on Audio, Speech and Language Processing*, 1984, vol. 32, no. 2, pp. 263-271.

[3] W.-Q. Zhang and J. Liu, “Two-stage method for specific audio retrieval,” in *Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, Hawaii, USA, Apr. 15-20, 2007, pp. IV-85-88.

[4] Timothy Hazen, Wade Shen, and Christopher White, “Query-by-example spoken term detection using phonetic posteri-orgram templates,” in *Proc. ASRU*, Merano, Italy, December 13-17, 2009, pp. 421–426.

[5] Rodriguez-Fuentes, Luis J., et al, "High-performance query-by-example spoken term detection on the SWS 2013 evaluation," in *Proc. ICASSP*, 2014.

[6] Yaodong Zhang and James R. Glass, “Unsupervised Spoken Keyword Spotting via Segmental DTW on Gaussian Posterior-grams,” in *Proc. ASRU*, 2009, pp. 398–403.

[7] Szoke, Igor, et al, "Calibration and fusion of query-by-example systems—But SWS 2013," in *Proc. ICASSP*. 2014, pp. 7849-7853.

[8] Anguera, Xavier, et al, "Query-by-example spoken term detection evaluation on low-resource languages," http://speech.fit.vutbr.cz/files/sltu_2014_MediaEval_SWStask_results.pdf.

[9] G.E. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, November 2012, vol. 29, pp. 82–97.

[10] Haipeng Wang and Tan Lee, “CUHK system for the Spoken Web Search task at MediaEval 2012,” in *Proc. MediaEval 2012 Workshop*, Pisa, Italy, October 2012, vol. 2012, no. 927.

[11] Meng Cai, Yongzhe Shi, and Jia Liu, "Deep maxout neural networks for speech recognition," in *Proc. ASRU*, 2013, pp. 291-296.