Investigating Neural Network based Query-by-Example Keyword Spotting Approach for Personalized Wake-up Word Detection in Mandarin Chinese

Jingyong Hou[1], *Lei Xie*[1], *Kaisheng Yao*[2], *Zhonghua Fu*[1]

[1]School of Computer Science, Northwestern Polytechnical University, Xi'an
[2] Microsoft Corporation, Redmond, 98052, WA

{jyhou,lxie}@nwpu-aslp.org, kaisheny@microsoft.com, mailfzh@nwpu.edu.cn

Abstract

We use query-by-example keyword spotting (QbyE-KWS) approach to solve the personalized wake-up word detection problem for small-footprint, low-computational cost on-device applications. QbyE-KWS takes keywords as templates, and matches the templates across an audio stream via DTW to see if the keyword is included. In this paper, we use neural networks as acoustic models to extract DNN/LSTM phoneme posterior features and LSTM embedding features. Specifically, we investigate the LSTM embedding feature extractor for different modeling units in Mandarin, spanning from phonemes to words. We also study the performances of two popular DTW approaches: S-DTW and SLN-DTW. SLN-DTW manages to accurately and effectively search the keyword in a long audio stream without the segmentation procedure that is used in S-DTW approaches. Our study shows that DNN phoneme posterior plus SLN-DTW approach achieves the highest computation efficiency and the state-of-the-art performance with 78% relative miss rate reduction as compared with the S-DTW approach. Word level LSTM embedding feature shows superior performance as compared with other embedding units.

Index Terms: Keyword Spotting, Wake-up Word Detection, DTW, Query-by-Example, DNN, LSTM

1. Introduction

With the rapid development of speech recognition in recent years, a variety of applications with speech interfaces have emerged, such as direct voice input, mobile assistant and intelligent loudspeaker *etc*. As the very first step to activate these speech interfaces, *wake-up word detection* aims to use a specific word to wake-up an automatic speech recognition (ASR) module with a fully hand-free experience. Besides, as a keyword spotting (KWS) technique, it can also be used for command and control functions. For example, some smartphones have voice controlled camera enabled by a speech keyword.

According to whether the keyword is set in advance or can be defined by users, wake-up word detection or keyword spotting can be divided into two categories: fixed KWS and personalized KWS. The former uses predefined keyword to activate devices. For instance, Amazon Echo uses "Alexa" to activate its voice assistant and Google uses "Okay Google" to access voice services. The latter supports users to customize their own keywords or set different wake-up words for different devices. Users can enroll their own wake-up word by speaking it several times, and use it to activate their own devices. With the proliferation of smart devices, personalized KWS has many potential applications due to its flexibility and individuality.

In the speech-enabled applications mentioned above, a wake-up module has to listen consistently on an embedded or mobile device. A small memory footprint and low computational cost solution, suitable for on-device applications, is thus highly desired. To effectively use personalized keywords for on-device applications, a QbyE-KWS system can be favorably considered. As a typical solution, QbyE-KWS takes keywords as templates, and matches the templates across an audio stream via dynamic time warping (DTW) to see if the keyword is included.

During the past several years, neural networks have re-emerged as a powerful tool in acoustic modeling [1, 2] and feature learning [3, 4]. Recently, deep neural networks (DNNs) have been suggested as exceptional feature extractors in DTW-based QbyE-KWS [5, 6, 7, 8]. For example, keyword and segment in testing audio can be represented respectively by a sequence of DNN-generated phoneme posteriors, and a sliding variant of DTW – segmental DTW (S-DTW) [9], is employed to measure the distance between them. Very recently, Chen *et al.* use recurrent neural network (RNN) with long short-term memory (LSTM) cells as a sequence feature extractor to embed keywords and testing audio segments into a fixed-dimension vector representation, respectively. With fixed-dimension vectors, the similarity between keywords and testing audio can be easily measured by a typical distance measure, e.g., cosine, bypassing the time-consuming DTW computation. Specifically, they suggest that using whole word as the embedding target can achieve superior performance in personalized wake-up word detection because of LSTM's long context modeling ability.

In this paper, we provide a systematic investigation on neural network based QbyE-KWS approach for personalized wake-up word detection in Mandarin Chinese. Specifically, we use neural networks to extract features, which include LSTM embedding features [5] and DNN/LSTM phoneme posterior features. We also study the QbyE-KWS performances of two popular DTW approaches: S-DTW and segmental local normalized DTW (SLN-DTW). Our contributions are as follows.

- We investigate the LSTM embedding feature extractor for different modeling units in Mandarin. As we know, Chinese has multiple phonetic and linguistic units: tonal/non-tonal phonemes, tonal/non-tonal syllables, characters and words. We would like to see which unit provide the best performance in LSTM feature extractor based personalized wake-up word detection.
- We introduce the SLN-DTW [10, 11, 12, 6] to QbyE-KWS based wake-up word detection. SLN-DTW manages to accurately and effectively search the keyword in a long audio stream without the segmentation procedure that is used in S-DTW approaches.

Our study shows that DNN phoneme posterior plus SLN-DTW approach has highest computation efficiency and achieves the state-of-the-art performance with lowest miss rate of 0.029 at 0.005 false alarm rate. Word level LSTM embedding feature shows superior performance as compared with other embedding units.

2. Neural Network based QByE KWS

As shown in Figure 1, a typical QbyE-KWS system consists of two modules – feature extraction and keyword searching. A representative feature is critical to the performance of keyword searching. Although popular acoustic features, e.g., MFCC, FBank, PLP, can be directly used, model-based features, e.g., phoneme and Gaussian posteriors [13, 9], have shown more discriminative power over acoustic features and thus lead to superior performance. Especially with the dominance of deep neural networks, DNN-derived features like DNN posteriors [13] and bottleneck features [6, 7, 8] achieve extraordinary performance in QbyE-KWS. Meanwhile, LSTM models are used to embed an audio sequence to a fixed-length representation [5]. In the second module, a distance measure is used for keyword search that compares the keyword template against the testing audio. Based on the feature representation used, usually DTW or some common distances, e.g., cosine, are employed in the matching.

2.1. Segmental DTW based KWS

We cannot directly measure the similarity between the enrollment keyword and testing audio using an ordinary DTW. This is because a KWS system designed for wake-up applications needs to listen continuously to the input sound. Zhang *et al.* [9] proposed to divide the testing audio stream into segments using a sliding window. On each segment, DTW is used to measure its distance with the keyword template. If the matching score goes beyond a pre-set threshold, a keyword is considered to be spotted. In practice, the segment length is set to that of the keyword template. This approached is called *segmental DTW* or S-DTW for short. Please note that, to minimize the missing rate of the keyword, segment overlapping is usually used.

As shown in Figure 1(a), the enrollment keyword and the runtime utterance firstly go through an NN feature extractor, resulting in frame-level NN phoneme posteriorgram features. The runtime utterance, represented in frame-level NN phoneme posterior vectors, is cut to keyword-sized segments and a distance matrix between the keyword and each segment is calculated. A confidence score is then computed from the DTW alignment cost. When the cost exceeds some threshold, a keyword is considered to be detected. Note that different NN models, *e.g.*, feed froward deep neural networks and Recurrent neural network (RNN) can been used.

2.2. LSTM Feature Extractor based KWS

Due to the long sequence modeling ability of LSTM, Chen *et al.* [5] propose an LSTM feature extractor to embed audio segments of varying lengths into a fixed-dimension representation. Hence the similarity of two sequences can be measured directly using a simple distance like cosine. Given an input sequence $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, the output of LSTM hidden layer and last layer are $\mathbf{H} = \{h_1, h_2, ..., h_n\}$ and $\mathbf{Y} = \{y_1, y_2, ..., y_n\}$, respectively. Because of the memory mechanism of LSTM, it is believed that the last k frame of the output $\mathbf{H}^* = \{h_{n-k+1}, h_{n-k+2}, ..., h_n\}$ and $\mathbf{Y}^* = \{y_{n-k+1}, y_{n-k+2}, ..., y_n\}$ contain most information of sequence $\mathbf{X}$. If we select same k frames as feature vectors for varying length of sequences, we could get the LSTM *embedding* representation of these sequences.

As shown in Figure 1(b), at the enrollment stage, we stack last k frames of LSTM RNN's last hidden layer activations as keyword's embedding representation. In practice, if multiple templates are used for each keyword, e.g., user says the keyword several times to enroll to the system, we use DTW-based template average [10, 5] to get a new embedding representation of the enrollment keyword. Note that the enrollment is an offline procedure.

At runtime stage, similarly, we extract LSTM embedding features for the runtime utterance, and then a sliding window is employed to segment the test audio. The sliding window size to usually set to k frames, so we can measure the similarity between the keyword and the runtime utterance using cosine distance of stacked vectors between keyword and runtime utterance segments.

2.3. Segmental Local Normalized DTW based KWS

The above KWS systems both need a sliding window to segment the runtime utterance. Besides, the window size and shifting size are usually determined empirically and affect the KWS performance if set improperly. In contrast, segmental local normalized DTW (SLN-DTW) [10, 14, 15] can mange to effectively search the keyword in a long audio stream without the segmentation procedure.

Given two sequences of feature vectors extracted from the DNN feature extractor, $Q = \{q_1, q_2, ..., q_n\}$ and $S = \{s_1, s_2, ..., s_m\}$, corresponding to an enrollment keyword and a test utterance, cosine or other distance matrix $dist$ is calculated, where $dist(i, j)$ represents distance between frame i of the keyword and frame j of the test utterance. SLN-DTW aims to find a path in the distance matrix $dist$, starting from $(1, s)$ to (n, e) (where $1 \leq b \leq e \leq m$), which minimizes the average accumulated distance $cost(i, j) = a(i, j)/l(i, j)$, where $a(i, j)$ is the accumulated distance from $(1, s)$ to (i, j) and $l(i, j)$ is the path length of (1, s) to (i, j). Dynamic programming algorithm is used to find best matching score, as described below.

1. Initialize a and l:

$$\begin{cases} a(i,1) = \sum_{k=1}^{i} dist(k,1) \\ l(i,1) = i \end{cases} \tag{1}$$

$$\begin{cases} a(1,j) = dist(1,j) \\ l(1,j) = 1 \end{cases} \tag{2}$$

   where $i = 1, 2, ..., n$ and $j = 1, 2, ..., m$.

2. For $i > 1$ and $j > 1$, select (u, v) from $\Phi = \{(i - 1, j), (i, j - 1), (i - 1, j - 1)\}$:

$$(u, v) = \underset{(u,v)\in\Phi}{\operatorname{argmin}} \frac{a(u, v) + dist(i, j)}{l(u, v) + 1} \tag{3}$$

   and then,

$$\begin{cases} a(i,j) = a(u,v) + dist(i,j) \\ l(i,j) = l(u,v) + 1 \end{cases} \tag{4}$$

3. Finally, $\min_{j=1,...,m}(cost(n, j))$ is the best matching score between the keyword and the testing utterance.

3. Experimental Setup

3.1. Training of Feature Extractors

The neural network we trained are shown in Table 1. We first trained feed-forward DNN phoneme recognizers and LSTM

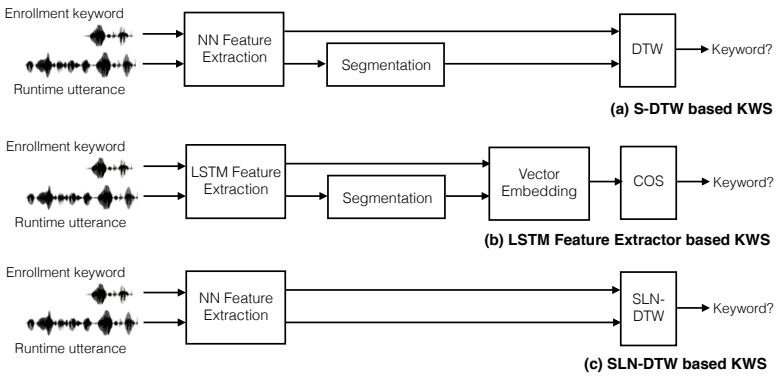

Figure 1: *Three different QbyE-KWS approaches for wake-up word detection.*

Table 1: *Configurations of neural network models*

model	input	layers	output/target	parameters
DNN phoneme recognizers	(10+1+5) Fbank	5-layers, 128 units	62 non-tonal phones	156k
			214 tonal phones	175k
LSTM phoneme recognizers	Fbank at current frame	2-layers, 128 cells	62 non-tonal phones	227k
			214 tonal phones	246k
LSTM embedding feature extractors	Fbank at current frame	2-layers, 128 cells	62 non-tonal phones	219k
			214 tonal phones	219k
			435 non-tonal syllables	219k
			1734 tonal syllables	219k
			4399 Characters	219k
			15000 Words	219k

phoneme recognizers with tonal and non-tonal phoneme targets. They are used for S-DTW and SLN-DTW based KWS experiments. LSTM embedding feature extractors keep the same input and network structure with the LSTM phoneme recognizers. Different from [5], in Mandarin, we tried different embedding units, i.e., tonal/no-tonal phonemes, tonal/no-tonal syllables, characters and words.

We use a window of 25ms with 10ms shifting to extract frame-level 40-dimension log filterbank (FBank) energy as spectral features, used as neural network input. All above NN models were trained using Kaldi [16] on a Mandarin corpus with 1115 hours of speech (sampling rate: 16KHz) recorded from over 4000 speakers. Stochastic gradient descent criterion with momentum parameter is used to optimize the models.

3.2. Wake-up Word Detection Test

Our wake-up word detection evaluation dataset was recorded by 225 speakers. We select 2 keywords as personalized wake-up words with length ranging from two to four words. Each keyword, embedded in a 1-5sec sentence, was recorded 10 times by each speaker. Two of them were used to enroll as templates and the rest 8 were used for evaluation. Evaluation positive samples consisted of utterances which contain recorded keywords and negative samples did not. Same to [5], there was no cross speaker test in our experimental setup. For one enrollment keyword, all the positive and negative samples came from the same speaker. In total, for each wake-up word, we had 1753 positive samples and 7200 negative samples from all the speakers. Modified ROC curve [5] was used to exhibit the KWS performance. We also chose miss rate at 0.005 false alarm rate to quantitatively represent the performance.

4. Results

4.1. LSTM Feature Extractors

Table 2 shows the performance of LSTM feature extractors trained for different embedding units in Mandarin. We can clearly see that the miss rate degrades significantly with the increase of embedding units. The miss rate remains at a every high level for tonal/non-tonal phoneme and non-tonal syllable units. The lowest miss rate (0.089) is achieved by the word LSTM feature extractor. We believe that the good performance of word level feature extractor is due to LSTM's ability to model long context.

Table 2: *Performance of LSTM embedding feature based KWS for different modeling units in Mandarin.*

model	miss rate at 0.005 false alarm
Non-tonal Phoneme LSTM Embedding	0.354
Tonal Phoneme LSTM Embedding	0.264
Non-tonal Syllable LSTM Embedding	0.169
Tonal Syllable LSTM Embedding	0.109
Character LSTM Embedding	0.096
Word LSTM Embedding	**0.089**

Table 3: *Performance of S-DTW based KWS*

model	miss rate at 0.005 false alarm
DNN tonal phoneme posteriors	0.137
DNN non-tonal phoneme posteriors	**0.129**
LSTM tonal phoneme posteriors	0.148
LSTM non-tonal phoneme posteriors	0.139

4.2. S-DTW KWS

As shown in Table 3, compared with LSTM embedding feature extractor based KWS systems (Table 2), S-DTW based KWS systems show inferior results, no matter either DNN phoneme posterior or LSTM phoneme posterior feature is used. This conclusion is consistent with that in [5], in which LSTM feature extractor also prevails in English.

4.3. Effects of Window Shifting Size

S-DTW based KWS and LSTM embedding feature extractor based KWS both need a window shifting along the testing audio stream. Thus we investigate the impacts from different window shifting size. As shown in Figure 2, with the increase of the shifting size, the miss rate elevates dramatically. Therefore, in real applications, we need to choose a small shifting size to main a reasonable missing rate, at the cost of increasing computation heavily.

4.4. SLN-DTW KWS

Table 4 shows the results of SLN-DTW based KWS systems. We can see that the SLN-DTW lowers the miss rate to a new level, as compared with LSTM embedding feature extractor (Table 2) and S-DTW based KWS (Table 3). Non-tonal phoneme DNN posterior feature with SLN-DTW achieves the lowest miss rate of 0.029. Compared with LSTM embedding feature extractor based KWS, this SLN-DTW system achieves

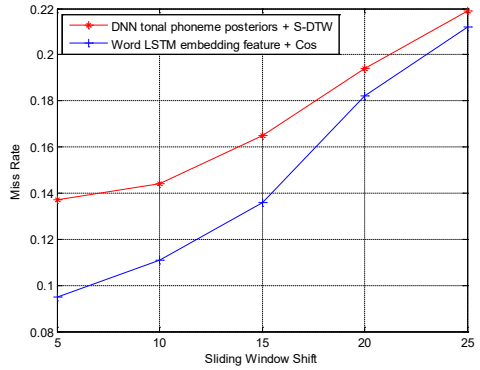

Figure 2: *Impact of window shifting size for S-DTW and LSTM embedding feature extractor based KWS.*

Table 4: *Performance of SLN-DTW based KWS.*

model	miss rate at 0.005 false alarm
DNN tonal phoneme posteriors	0.035
DNN non-tonal phoneme posteriors	**0.029**
DNN non-tonal phoneme posteriors (2 enrollments)	**0.007**
LSTM tonal phoneme posteriors	0.045
LSTM non-tonal phoneme posteriors	0.047

67% relative miss rate reduction. Compared with S-DTW based KWS, this SLN-DTW system achieves 78% relative miss rate reduction. We also notice that LSTM-based SLN-DTW cannot outperform DNN-based SLN-DTW, but LSTM achieves an error rate at a similar level with DNN. As demonstrated in [10, 5], using an average keyword template of multiple enrollment entries can improve the performance. Similarly, we follow [10, 5] to perform an extra experiment on KWS with multiple enrollments. By using 2 enrollment entries for each keyword, results (in Table 4) show that the miss rate reaches 0.007. The ROC curves of several typical QbyE-KWS systems are shown in Figure 3.

4.5. Efficiency Test

We empirically compare the runtime efficiency of different models on a server (CPU:Intel Xeon E5-2643, 96G RAM), and the real-time ratio (time to process the speech/duration of the speech) is summarized in Table 5. The ratio is averaged on the evaluation samples of all keywords. Results show that DNN phoneme posterior plus SLN-DTW approach has the highest running efficiency, running faster than the S-DTW approach. But the speed-up is not salient than expected. This is because the neural network-based feature extraction takes a substantial amount of computation time. When we compare LSTM and DNN, we find that the DNN approaches are much faster than the LSTM approaches. Comparing the two LSTM approaches, the word LSTM embedding feature approach runs faster. This is because the similarity measure is cosine in this approach. Please note that we do not consider any optimization strategies in the efficiency test. In real on-device use, some practical optimization strategies can be used to further speech-up the computation.

5. Related Work

KWS has been studied for fixed or personalized keyword applications. The keyword-filler HMM approach [17, 18, 19, 20, 21] has been a dominating approach for many years. But with the fast development of deep neural networks, they have been suggested to solve the KWS problem [22, 23, 24]. In [24], a popular small footprint KWS approach is proposed, where a DNN is trained to predict keyword targets and a garbage target. The above HMM and DNN based solutions are only suitable for

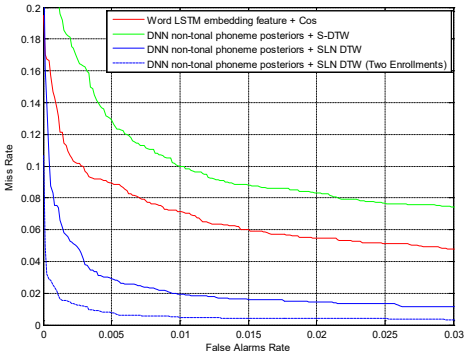

Figure 3: *ROC curves for several typical QbyE-KWS systems.*

Table 5: *Real time rate of different KWS systems (the smaller the more efficient)*

model	KWS	real time rate
LSTM non-tonal phoneme posteriors	SLN-DTW	0.018
DNN non-tonal phoneme posteriors	SLN-DTW	0.006
DNN non-tonal phoneme posteriors	S-DTW	0.008
Word LSTM embedding	Cosine	0.016

fixed keywords.

Another solution that can be considered for personalized KWS relies on a large vocabulary continuous speech recognizer (LVCSR). It decodes audio to symbolic representation like a phoneme/word sequence or a lattice [25, 26, 27, 28, 29, 30] and text retrieval techniques or symbolic search [31, 11] are used to detect keywords. If a keyword enrollment stage is involved, then both keyword and testing audio need to be decoded by the LVCSR. This KWS solution may be a good one if no limit to the resources. But LVCSR needs quite a lot of computing resources and does not fit our on-device applications.

As discussed in Section 1, DTW-based QbyE-KWS is an appropriate solution for on-device personalized keyword detection [9, 10, 11, 12, 6]. Because the keyword template is usually much shorter than runtime utterance, strategies like S-DTW [9] and SLN-DTW [10, 11, 12, 6] are used to solve this problem. Chen *et al.* [5] propose an LSTM feature extractor approach to embed audio segments of varying lengths into a fixed-dimension representation. Hence the similarity of two sequences can be measured directly using a simple distance (e.g., cosine), bypassing the time-consuming DTW computation.

6. Conclusion

We have investigated neural network based query-by-example keyword spotting (QbyE-KWS) approach for personalized wake-up word detection in Mandarin Chinese. Specially, we have studied both neural network based features and two different DTW algorithms for KWS. Wake-up word detection experiments show that DNN phoneme posterior plus SLN-DTW approach has achieved the state-of-the-art performance with highest runtime efficiency. Compared with other embedding units like phonemes, syllables and characters, word level LSTM embedding feature shows superior performance. But the LSTM networks, either used as embedding feature extractor or posterior feature extractor, cannot outperform the feed-forward networks. This is probably because LSTMs may need more parameter tunings and training tricks. As our future work, we will perform further studies on LSTM-based feature extractors.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 61571363).

8. References

[1] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[3] P. Schwarz, P. Matejka, and J. Cernocky, "Hierarchical structures of neural networks for phoneme recognition," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[4] F. Grézl, M. Karafiát, S. Kontár, and J. Cernocky, "Probabilistic and bottle-neck features for lvcsr of meetings," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, vol. 4. IEEE, 2007, pp. IV–757.

[5] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5236–5240.

[6] J. Hou, C.-C. L. Van Tung Pham, L. Wang, H. Xu, H. Lv, L. Xie, Z. Fu, C. Ni, X. Xiao, H. Chen, S. Zhang, S. Sun, Y. Yuan, P. Li, T. L. Nwe, S. Sivadas, B. Ma, E. S. Chng, and H. Li, "The NNI query-by-example system for MediaEval 2015," *Working Notes Proceedings of the MediaEval 2015 Workshop, Wurzen, Germany, Sept*, 2015.

[7] H. Xu, J. Hou, X. Xiao, C.-C. Leung, L. Wang, H. Lv, L. Xie, B. Ma, E. S. Chng, H. Li *et al.*, "Approximate search of audio queries by using dtw with phone time boundary and data augmentation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6030–6034.

[8] I. Szöke, M. Skácel, L. Burget, and J. Černockỳ, "Copingwith channel mismatch in query-by-example-but quesst 2014," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5838–5842.

[9] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 398–403.

[10] L. J. Rodriguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "High-performance query-by-example spoken term detection on the SWS 2013 evaluation," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7819–7823.

[11] P. Yang, H. Xu, X. Xiao, L. Xie, C.-C. Leung, H. Chen, J. Yu, H. Lv, L. Wang, S. J. Leow *et al.*, "The NNI query-by-example system for MediaEval 2014," *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Spain, Oct*, 2014.

[12] I. Szöke, M. Skácel, and L. Burget, "But QUESST 2014 system description." *Working Notes Proceedings of the MediaEval 2014 Workshop, Barcelona, Spain, Oct*, 2014.

[13] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 421–426.

[14] A. Muscariello, G. Gravier, and F. Bimbot, "Audio keyword extraction by unsupervised word discovery," in *INTERSPEECH 2009: 10th Annual Conference of the International Speech Communication Association*, 2009.

[15] P. Yang, C.-C. Leung, L. Xie, B. Ma, and H. Li, "Intrinsic spectral analysis based on temporal context features for query-by-example spoken term detection." in *INTERSPEECH*, 2014, pp. 1722–1726.

[16] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[17] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989, pp. 627–630.

[18] R. C. Rose and D. B. Paul, "A hidden markov model based keyword recognition system," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 129–132.

[19] J. Wilpon, L. Miller, and P. Modi, "Improvements and applications for key word recognition using Hidden Markov Modeling techniques," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*. IEEE, 1991, pp. 309–312.

[20] M.-C. Silaghi and H. Bourlard, "Iterative posterior-based keyword spotting without filler models," in *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*. Citeseer, 1999, pp. 213–216.

[21] M.-C. Silaghi, "Spotting subsequences matching an HMM using the average observation probability criteria with application to keyword spotting," in *Proceedings of the National Conference on Artificial Intelligence*, vol. 20, no. 3. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005, p. 1118.

[22] K. Li, J. Naylor, and M. Rossen, "A whole word recurrent neural network for keyword spotting," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 2. IEEE, 1992, pp. 81–84.

[23] S. Fernández, A. Graves, and J. Schmidhuber, "An application of recurrent neural networks to discriminative keyword spotting," in *International Conference on Artificial Neural Networks*. Springer, 2007, pp. 220–229.

[24] G. Chen, C. Parada, and G. Heigold, "Small-footprint keyword spotting using deep neural networks," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4087–4091.

[25] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," *Urbana*, vol. 51, p. 61801, 2004.

[26] C. Chelba and A. Acero, "Position specific posterior lattices for indexing speech," in *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 443–450.

[27] D. Vergyri, I. Shafran, A. Stolcke, V. R. R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 spoken term detection system." in *INTERSPEECH*. Citeseer, 2007, pp. 2393–2396.

[28] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 615–622.

[29] D. R. 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 *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[30] K. Ng and V. W. Zue, "Subword-based approaches for spoken document retrieval," *Speech Communication*, vol. 32, no. 3, pp. 157–186, 2000.

[31] H. Xu, P. Yang, X. Xiao, L. Xie, C.-C. Leung, H. Chen, J. Yu, H. Lv, L. Wang, S. J. Leow *et al.*, "Language independent query-by-example spoken term detection using N-best phone sequences and partial matching," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5191–5195.