A Refined Query-by-Example Approach to Spoken-Term-Detection (STD) on ESL Learners' Speech

Jingyong Hou[1*], *Wenping Hu*[2], *Frank K. Soong*[2], *Lei Xie*[1]

[1]School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an, China
[2]Microsoft Research Asia, Beijing, China

{jyhou,lxie}@nwpu-aslp.org, {wenh,frankkps}@microsoft.com

Abstract

A refined Query-by-Example (QbE) approach is proposed to improve Spoken-Term-Detection (STD) performance on L2 English learners' speech data. A Hidden Markov Model (HMM) is built for each keyword and a computationally efficient, iterative Viterbi decoding is adopted to detect spoken keywords in test. The approach is evaluated on an English as Second Language (ESL) speech database collected over L2 learners with different English proficiency levels. The experimental results show that the new approach achieves a performance better than the traditional DTW-based QbE. Also, it is comparable to that of an LVCSR-based STD but with significant lower complexities and computations. The refined QbE and LVCSR approach to STD are complementary to each other. By fusing the two systems together, we can further improve the MAP and MP@N performance by 6.1%-13.4% and 7.5%-14.4%, respectively, in testing sets of 3 different English proficiency levels over the best performance of either system.

Index Terms: CALL, QbE, STD, HMM, Viterbi

1. Introduction

For L2 language learners, spoken terms in their conversational speech may not well match with the acoustic and language models trained with native speakers' data. In Computer Assisted Language Learning (CALL), it is highly desirable to customize a learning program for each learner, based upon his learning progress and speech data accumulated over the learning period. However, such a program cannot be made easily if keywords spoken by such an L2 learner cannot be spotted with a decent detection rate, particularly in an open conversation scenario. To overcome this difficulty, A high performance Spoken-Term-Detection (STD) system is essential.

Different from traditional STD tasks, detecting keywords from L2 learners' conversational speech is non-trivial because of learners' accented speech, irregular pronunciation, poor grammar, etc. How to detect the spoken terms for L2 learners effectively and efficiently is the main scope of this paper. STD, also known as keyword spotting (KWS), aims at detecting the occurrence of a pre-defined keyword in speech [1]. STD has been a very active topic in speech recognition research and can be roughly divided into two categories: text based and Query-by-Example (QbE) based, depending upon whether the input keyword is given as text or example(s) of spoken speech.

In the text-based STD system, a Large Vocabulary Continuous Speech Recognition (LVCSR) system is indispensable to decode the test speech into phoneme or word sequences, usually compactly represented as a lattice [2]. Then some scoring process is evoked, e.g., word posterior [3], to obtain the detection score of a given keyword. This approach is also called as LVCSR-based STD. In the last decade, a QbE-based approach has been widely used in scenarios where rich text transcription is unavailable or fast detection response is required, e.g, low resource language STD [4, 5, 6, 7] or personalized wake-up word detection [8, 9], respectively.

A QbE-based STD system consists of three modules: feature extraction, keyword model and keyword search. For the feature extraction, Gaussian posterior-gram [10] and unsupervised neural network (NN) based features [11, 12, 13, 14], are widely used in low resource language based keyword detection. Most recently, phoneme posterior-grams [15] and Stacked Bottleneck (SBN) features [16] trained from a rich resource language has shown better performance than features obtained from an unsupervised manner [7, 17]. To model a keyword, template averaging is a typical approach and different template fusion strategies have been investigated for various QbE tasks. Luis *et al.* [5] suggested the selection of the longest instance as the reference and aligning all the remaining instances to the reference for length normalization. This fusion approach has shown more robust performance than other fusion methods, such as aligning to average iteratively [8]. For keyword detection and scoring, variations of Dynamic Time Warping (DTW) are usually adopted. For example, segmental-DTW (S-DTW) [10] has been proposed for STD, also, a more efficient Segmental Local Normalized-DTW (SLN-DTW) [18, 19] has been introduced to replace S-DTW with a better performance [5, 6, 7, 17, 9].

LVCSR-based STD is in general too complex to be affordable for language learning applications on mobile devices in terms of its footprint and latency. In addition, the N-best or lattice is not provided by most commercial speech recognition services. Therefore, a QbE-based approach is chosen in our on-device English learning application. For each keyword, a large scale of non-native instances with different pronunciation qualities can be obtained through our learning system. But, in traditional template averaging based keyword modeling method, the STD performance saturates quickly as the training instances increased, which is unable to make good use of the collected training instances.

In this paper, we propose a novel QbE-based STD framework to improve both the efficiency and accuracy in ESL speech. Instead of an average template, a multi-state HMM is built for each individual keyword and an efficient **iterative** Viterbi decoding based keyword search approach is used in testing. Experiments show that the proposed HMM-based approach can achieve a better performance than the traditional DTW-based one. Its performance is comparable to LVCSR-based system but with a much lower complexity. The rest of the paper is

[*]This work is done when the first author worked as an intern in speech group at Microsoft Research Asia.

organized as follows. In section 2, we give an introduction of our proposed STD framework. The experimental setup and results are presented in section 3 and 4, respectively. Finally, a conclusion is drawn in section 5.

2. Proposed QbE based STD Framework

Different modules of our QbE-based STD system, including: SBN feature, keyword modeling and template matching approach, are presented in this section. The diagram of our QbE based STD system is shown in Fig. 1.

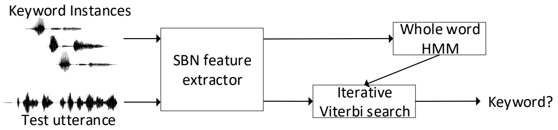

Figure 1: *Framework of QbE based STD system*

2.1. SBN feature extractor

Similar to [16], a cascaded, two-stage feed-forward neural network is used for extracting stacked bottleneck features as shown in Fig. 2. The first stage network use the raw acoustic spectrum as input and a low-dimensional bottleneck feature vector as output. The augmented output bottleneck features are further compressed in the second stage neural network. The final SBN features are used for keyword detection. From the raw acoustic features to the stacked bottleneck features, different external disturbances, e.g., speaker, channel and other recording noises, are suppressed and equalized. Therefore, the SBN features are expected to be more effective for keyword spotting on non-native speech compared with raw acoustic features.

The neural network in each stage is trained to classify "senones" with the cross-entropy objective function independently. The SBN feature extractor is a part of the network of the acoustic model, which is used in the following LVCSR-based STD system.

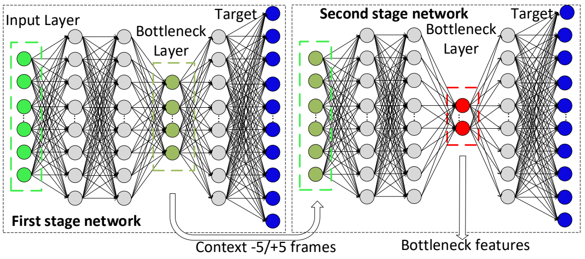

Figure 2: *SBN feature extractor*

2.2. Whole word HMM for keyword modeling

A left to right HMM with several states is built for each keyword. The topology is shown in Fig. 3, where q^i is the i^{th} hidden state, $P(q^i|q^j)$ is the transition probability from state i to state j, $P(x|q^i)$ is the observation probability of feature x for state q^i, which is modeled by a Gaussian Mixture Model (GMM).

Compared with the traditional template averaging approach, our HMM modeling approach is more elegant and yields a more powerful capability in modeling the variability contained in different training samples. In addition, it avoids the length normalization and template fusion efforts needed in template averaging approaches. Actually, a traditional averaged template can be regarded as a special form of HMM by setting the number of states equal to the length of the template, and with each state is modeled as a single Gaussian with unity variance. In addition, unlike previous acoustic KWS systems [20, 21, 22, 23], here we use word as the modeling units of HMM, and each HMM is equivalent to an independent keyword model (That's why we still call it QbE based methods).

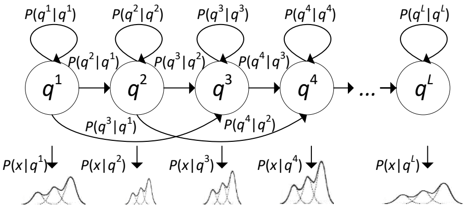

Figure 3: *Topology of adopted HMM for keyword modeling*

2.3. Iterative Viterbi search

An efficient iterative Viterbi decoding based approach is used for keyword search, which was first proposed in paper [23]. Different from traditional DTW based search, it is able to find the globally optimized subsequence corresponding to the keyword after a few iterations. It takes the observation probability of a filler model, used to match the non-keyword segments, as a constant value in the whole iteration and updates it iteratively. Therefore, compared with traditional HMM based STD system, no filler model is required indeed, which simplifies the model training progress largely.

Given the keyword HMM M, for any testing utterance $X_1^N = \{x_1, x_2, ..., x_N\}$, we use the Averaged Observation Likelihood (AOL) to quantify how well the detected subsequence X_b^e matches the keyword model M, where b or e is the start or end frame index of matched subsequence, respectively. We use $Q = \{q^1, ..., q^L\}$ to represent the states of M from left to right. AOL can been defined as:

$$\begin{aligned} AOL(M, X, b, e) &= \frac{1}{(e-b+1)} \max_S \log P(X_b^e, S|M) \\ &= \frac{1}{(e-b+1)} \max_S \{\log P(x_e|s_e) \\ &+ \sum_{n=b}^{e-1} [\log P(x_n|s_n) + \log P(s_{n+1}|s_n)]\}, \end{aligned}$$

where $S = \{s_b, \cdots, s_n, \cdots, s_e\}$, $s_n \in Q$.

The keyword search problem is then formulated as an optimization problem, i.e., finding the optimal frame index b and e to maximize the AOL. This problem can be solved in an iterative way. For simplicity, we use symbol $\bar{M}$ to represent the extended HMM where an extra filler model q^G is added at the beginning and ending of M and symbol $\bar{Q} = \{q^G, q^1, ..., q^L, q^G\}$ to represent the states of $\bar{M}$. The iterative search process is described as Algorithm 1.

This algorithm has been proved to be able to find the global optimal solution within a few limited iterations [23]. Our experimental results further validate that the algorithm can converge in 3-4 iterations.

Algorithm 1 Iterative Viterbi decoding

```
Input: $X_1^N$: test utterance, $\bar{M}$: extend HMM, $\epsilon_0$: initial score
Output: $< b^\star, e^\star >$: optimal frame indexes, $\epsilon^\star$: matching score
Set $k = 0, \epsilon_k = 0$
Calculate $P(x_n|q^i), \forall i \in [1, ..., L], \forall n \in [1, ..., N]$
repeat
    Set $P(x_n|q^G) = \exp(\epsilon_k), \forall n \in [1, ..., N]$
    Find optimal state sequence $S^* = \arg\max_S P(X_1^N, S|\bar{M})$ by ordinary Viterbi search, $S = \{s_1, \cdots, s_n, \cdots, s_N\}, s_n \in \bar{Q}$
    $b^*$=$\arg\min_n\{s_n = q^1\}$, $e^*$=$\arg\max_n\{s_n = q^L\}$, $\epsilon_{k+1} = AOL(M, X, b^*, e^*)$
    $\epsilon^\star = \epsilon_{k+1}, k = k + 1$
until $\epsilon_k$==$\epsilon_{k-1}$
```

Table 1: *Keyword list*

human	twenty	Tuesday	laundry	expenses	american
hotel	travel	welcome	o'clock	computer	position
table	sunday	college	company	recently	insurance
golden	orange	someone	working	thursday	interview
people	united	seattle	receive	actually	resources
period	ticket	service	morning	previous	scheduled
second	salary	baggage	project	thousand	employees
mister	double	meeting	problem	tomorrow	interested
before	credit	program	minutes	training	university
afraid	because	evening	dollars	anything	reservation

Table 2: *Details of 3 search datasets*

	score range	#utt	avg length (s)	prior (%)
G1	15-30	866	4.7	0.64
G2	40-55	953	5.1	0.81
G3	65-80	872	4.8	0.74

3. Experimental setup

3.1. Database

To train the acoustic model or SBN feature extractor, two databases are available. One is a native speakers' speech database, i.e., switchboard [24] containing 282 hrs spontaneous speech and 4,803 speakers in total. The other one is a non-native ESL learners' speech database, denoted as 'Xiaoying'. It is collected from 11,530 English learners, whose mother tongue are mainly Chinese, via Microsoft English learning service available on WeChat [25]. The Xiaoying database is collected in a "read after me" way (read speech) and 160 hrs in total.

Besides, an extra Xiaoying database is used for keyword modeling and testing in STD related experiments. These two Xiaoying databases are collected in the same way and there is no overlapping of sentence transcription and speaker between them. The keyword set consists of 60 isolated words, each word contains at least 2 syllables or 5 phonemes. Table 1 shows these 60 selected keywords. For each keyword, we collect $\sim$40 instances to build the keyword model. The keyword search database consists of $\sim$2,700 utterances by English learners with different proficiencies. There is no speaker overlap between keyword enrollment and testing datasets.

To test the sensitivity of our STD systems to users' English proficiency, we divide the testing database into three groups, i.e., G1, G2 and G3, according to its utterance level pronunciation score evaluated by the pronunciation scoring algorithm described in [26]. The detail information of each group are listed in Table 2. To better distinguish different search datasets, there are discontinuities in the score ranges for the three groups, i.e.,the utterances with scores between 31 - 39 and 56 - 64 are discarded. The prior is calculated as the averaged occurrences of each keyword in the whole testing set. It shows that the averaged prior is less than 1%, which indicates the difficulty of our STD tasks.

3.2. Training of SBN feature extractor

Two feed-forward neural networks with a bottleneck layer at the next to last hidden layer are cascaded as the feature extractor in our experiment. Each neural network consists of 1 input layer, 4 hidden layers (each layer with 1500 nodes) and 1 output layer. The number of nodes for the first or second bottleneck layer is 80 or 30, respectively. The input of each feed-forward neural network is an augmented feature vector, which contains 5 preceding frames, the current frame and 5 succeeding frames. For the first stage network, the raw spectrum feature is 36-dim log scaled fbanks and the augmented super feature vector is further converted to a 216-dim vector by Discrete Cosine Transform (DCT) [16]. For the second stage network, the input comes from the augmented outputs of the first bottleneck layer.

Firstly, we train this stacked neural networks with switchboard database, the recipe is similar as that described in paper [16]. This model is served as baseline acoustic models in our following experiments. Then, we refine this model with Xiaoying data to alleviate the mismatch between trained acoustic model and ESL testing set.

Two metrics are used to evaluate the performance of different STD systems, i.e., the Mean Average Precision (MAP) and Mean Precision at N (MP@N), where the AP and P@N are defined as:

- **AP**: Averaged precision at the true hit utterance position over all ranked utterances
- **P@N**: Top N precision of the returned ranked list; N: the number of utterances that contain the keyword

4. Experimental results

4.1. Recognition results on STD testing dataset

Before conducting STD experiments, we check the recognition performance on the collected ESL testing set by utilizing Microsoft Bing speech API and our self-built LVCSR systems. The results are shown in Table 3, where a 3-gram language model, built from the transcription of SWBD training database, is used in our LVCSR systems.

Table 3: *WERs (%) of different LVCSR systems*

	Bing speech API	baseline LVCSR	adapted LVCSR
G1	80.59	99.46	**61.98**
G2	59.28	86.93	**43.07**
G3	36.72	63.88	**32.74**

As show in Table 3, the WER is improved as the proficiency level (rated by corresponding pronunciation score) increased from G1 to G3 consistently in all LVCSR systems. The performance of the baseline LVCSR system is the worst since its acoustic model is only trained on native speakers' speech only, while ESL learners in STD can be heavily accented. After adapting the acoustic model with Xiaoying's field data, adapted

LVCSR system has 37.7%, 50.5%, 48.7% relative WER degradation over the baseline system for G1, G2, G3, respectively. In the following experiments, the adapted LVCSR and SBN feature extraction are used.

4.2. LVCSR based STD

For comparing with the QbE based STD system, an LVCSR based system is built. Since the size of word lattice in LVCSR decoding is critical to the final accuracy of keyword spotting, we vary the lattice beam from 5 to 17 to study its effect on the STD performance. Both MAP and MP@N of all three test sets are shown in Fig. 4.

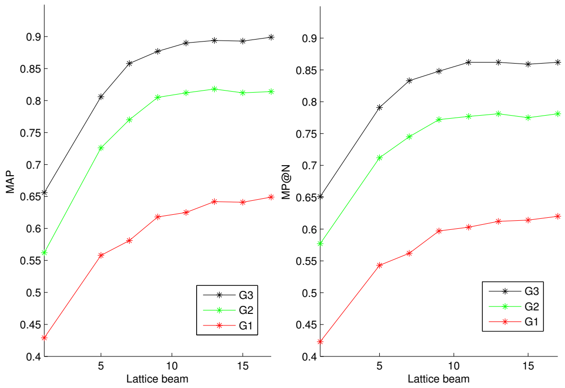

Figure 4: *Performance of LVCSR based STD with different lattice beam*

As shown in the figure, the detection performance is quite poor when there is no provided lattice. By increasing the lattice beam, both MAP and P@N are improved and saturated around the beam size of 15. The optimal STD performance on each test subset is shown in the third column of Table 4.

4.3. QbE based STD

In our proposed QbE approach, SBN features are used. The number of HMM states for each keyword is 9 times of the number of its corresponding phonemes and each state is trained with a single Gaussian and all states share a global, diagonal variance for all keywords. Here we have tried different setup of the HMM based keyword model, for example, the number of states, the number of components for each GMM, etc., the above configuration has achieved the best experimental results. The iterative Viterbi search as described in section 2.3 is used.

In Fig. 5, we compare the proposed QbE system with the traditional DTW based one, where template fusion and DTW search are implemented as [5]. The STD results show that the proposed approach yields higher MAP and MP@N than DTW based one on all testing sets and all number of instances. In addition, the performance of DTW based approach saturates at around 15 keyword instances, while the STD performance of our proposed system can be further improved with increased instances.

4.4. Boosted performance by score fusion

In Table 4, a breakdown of the STD performance of LVCSR-based, DTW-based and the proposed HMM-based QbE systems. The results show that our proposed system can achieve roughly the same performance as LVCSR-based one, i.e., with a slightly better MAP and a slightly worse MP@N. The performance of LVCSR based system depends upon the quality of language model used in decoding. When we turned off the language model, the STD performance is significantly degraded, particularly for the G1 set, which is shown in the second column.

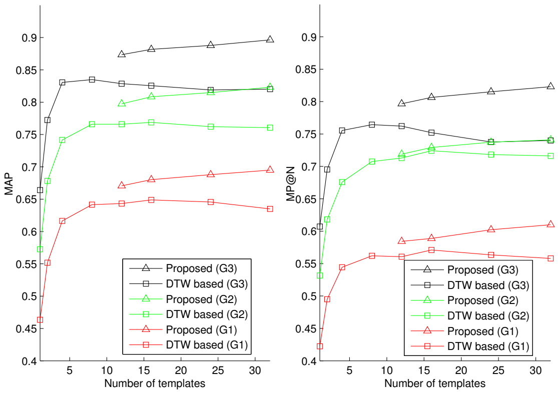

Figure 5: *Performance comparison of the proposed HMM-based and DTW-based QbE systems*

Finally, we fuse the LVCSR based and our proposed systems by weighting its corresponding detection score with 0.5. The detection results are shown in the last column in Table 4. It shows that 6.1%-13.4% in MAP and 7.5%-14.4% in MP@N can be achieved, when compared with the best single system on 3 testing set, which indicates that the information contained in those two STD approaches are complementary to each other.

Table 4: *Performance comparison of different systems*

	LVCSR based (No LM)	LVCSR based	DTW based	Proposed	**Fused**
			MAPs		
G1	0.566	0.668	0.642	**0.695**	**0.788**
G2	0.775	0.814	0.766	**0.823**	**0.899**
G3	0.851	**0.899**	0.835	0.896	**0.954**
			MP@Ns		
G1	0.524	**0.620**	0.562	0.610	**0.709**
G2	0.733	**0.781**	0.707	0.741	**0.854**
G3	0.810	**0.862**	0.764	0.823	**0.927**

5. Conclusions

A new QbE based approach is proposed for improving spoken term detection performance on ESL learners' speech data. When compared with the traditional DTW-based approach, the proposed one yields significantly better performance on testing sets of ESL learners with 3 different proficiency levels. In comparison with the LVCSR-based system, similar performance is achieved but with significantly smaller system size and lower computational complexities, hence more preferable for on-line, mobile oriented language learning services. By fusing the LVCSR-based system with the proposed one, we can further improve the performance by another 6.1%-13.4% in MAP and 7.5%-14.4% in MP@N on 3 test sets at different proficiency levels, compared with the best individual system. Bridging the performance gap between learners of different proficiencies is very helpful for providing timely and focused feedbacks to ESL learners, particularly to the beginners.

6. References

[1] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. SIGIR*,

2007, pp. 51–57.

[2] M. Saraclar, "Lattice-based search for spoken utterance retrieval," in *Proc. HLT-NAACL*, 2004, pp. 129–136.

[3] F. Wessel, R. Schluter, K. Macherey, and H. Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on speech and audio processing*, vol. 9, no. 3, pp. 288–298, 2001.

[4] F. Metze, N. Rajput, X. Anguera *et al.*, "The spoken web search task at MediaEval 2011," in *Proc. ICASSP*, 2012, pp. 8121–8125.

[5] 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 *Proc. ICASSP*, 2014, pp. 7819–7823.

[6] I. Szöke, M. Skácel, and L. Burget, "BUT QUESST 2014 system description," in *Proc. Multimedia Benchmark Workshop*, 2014.

[7] I. Szoke, L. J. Rodriguez-Fuentes, A. Buzo, X. Anguera, F. Metze, J. Proenca, M. Lojka, and X. Xiao, "Query by example search on speech at MediaEval 2015," in *Proc. Multimedia Benchmark Workshop*, 2015.

[8] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. ICASSP*, 2015, pp. 5236–5240.

[9] J. Hou, L. Xie, and Z. Fu, "Investigating neural network based query-by-example keyword spotting approach for personalized wake-up word detection in Mandarin Chinese," in *Proc. ISCSLP*, 2016, pp. 1–5.

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

[11] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. ICASSP*, 2015, pp. 5818–5822.

[12] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. ICASSP*, 2016, pp. 4950–4954.

[13] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. ICASSP*, 2014, pp. 7634–7638.

[14] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio Word2Vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *Proc. INTERSPEECH*, 2016, pp. 765–769.

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

[16] M. Karafiát and F. Grézl, "Hierarchical neural net architectures for feature extraction in ASR," in *Proc. INTERSPEECH*, 2010, pp. 1201–1204.

[17] H. Xu, J. Hou, X. Xiao, C.-C. Leung *et al.*, "Approximate search of audio queries by using DTW with phone time boundary and data augmentation," in *Proc. ICASSP*, 2016, pp. 6030–6034.

[18] A. Muscariello, G. Gravier, and F. Bimbot, "Audio keyword extraction by unsupervised word discovery," in *Proc. INTERSPEECH*, 2009.

[19] ——, "Variability tolerant audio motif discovery," in *Proc. MMM*. Springer, 2009, pp. 275–286.

[20] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. ICASSP*, 1990, pp. 129–132.

[21] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden Markov modeling for speaker-independent word spotting," in *Proc. ICASSP*, 1989, pp. 627–630.

[22] J. Wilpon, L. Miller, and P. Modi, "Improvements and applications for key word recognition using Hidden Markov Modeling techniques," in *Proc. ICASSP*, 1991, pp. 309–312.

[23] M. C. Silaghi and H. Bourlard, "Posterior-based keyword spotting approaches without filler models," *Swiss Federal Institute of Technology Lausanne (EPFL)*, 1999.

[24] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP*, 1992, pp. 517–520.

[25] "Microsoft Xiaoying," http://www.engkoo.com/.

[26] W. Hu, Y. Qian, and F. K. Soong, "A new DNN-based high quality pronunciation evaluation for Computer-Aided Language Learning (CALL)." in *Proc. INTERSPEECH*, 2013, pp. 1886–1890.