*INTERSPEECH 2016*
September 8–12, 2016, San Francisco, USA

# Rescoring Hypothesized Detections of Out-of-Vocabulary Keywords Using Subword Samples

*Van Tung Pham*[1,2], *Haihua Xu*[2], *Xiong Xiao*[2], *Nancy F. Chen*[3], *Eng Siong Chng*[1,2], *Haizhou Li*[1,2,3]

[1]School of Computer Engineering, Nanyang Technological University, Singapore.
[2]Temasek Laboratories, Nanyang Technological University, Singapore.
[3]Institute for Infocomm Research, Singapore

vantung001@e.ntu.edu.sg

## Abstract

Rescoring hypothesized detections, using keyword's audio samples extracted from training data, is an effective way to improve the performance of a Keyword Search (KWS) system. Unfortunately such rescoring framework cannot be applied directly to Out-of-Vocabulary (OOV) keywords since there is no sample in the training data. To address this limitation, we propose two techniques for OOV keywords in this work. The first technique generates samples for an OOV keyword by concatenating samples of its constituent subwords. The second technique splits hypothesized detections into segments, then estimates the acoustic similarities between detections and subword's samples according to the similarities between segments and these samples. The similarity scores from these two techniques are used to rescore and re-rank the list of detections returned by the automatic speech recognition (ASR) systems. The experiments show that incorporating the proposed similarity scores results in a better separation between the correct and false alarm detections than using the ASR scores alone. Furthermore, experimental results on the NIST OpenKWS15 Evaluation show that rescoring with the proposed similarity scores significantly outperforms the raw ASR scores, and other methods that do not use the similarity scores, in both Maximum Term Weighted Value (MTWV) and Mean Average Precision (MAP) metrics.

**Index Terms**: Spoken term detection, OOV keywords, graph based rescoring, acoustic similarity, template concatenation.

## 1. Introduction

With the prevalence of smart phone devices and high Internet bandwidth, there is an increasing amount of spoken data to be archived, managed, and analyzed. Speech retrieval is thus an important research area.

This work focuses on the Spoken Term Detection (STD) [1] or Keyword Search (KWS) [2] task, which detects the present of a textual keyword in an audio corpus. Generally, a two-stage approach is utilized for a KWS system [3–6]. Specifically, audio files of the corpus are first segmented and transcribed into lattices by an automatic speech recognition (ASR). Then indexing/retrieval techniques such as weighted finite state transducer (WFST) [7–9] or ngram inverted index [10–13], are applied on these lattices to generate list of hypothesized detections.

It is observed that the original detection scores, which are normally the posterior probabilities [14] of keywords at detected locations, might not be robustly estimated in adverse conditions. Thus, various approaches has been proposed to rescore the KWS detections [15–23, 33].

In our previous work [24], keyword samples extracted from training data are helpful in rescoring the list of detections. The main idea is that if a hypothesized detection is acoustically more similar to the keyword samples in the training data, it is more likely to be a true detection, and its score should be boosted. The acoustic similarity can be estimated using Dynamic Time Warping (DTW) which is a template-based and non-parametric approach; hence is complementary with the standard model-based parametric ASR system. One of the major drawbacks of the rescoring framework is that it cannot be applicable for Out-of-Vocabulary (OOV) keywords, which contain words that do not appear in the ASR dictionary (and do not have any training samples).

We observed that although an entire OOV keyword cannot be found in the training data, its constituent units, i.e. individual words or subwords, can apprear in the training data. Therefore, in this work, we proposed two techniques to address the lack-of-samples problem for OOV keywords using samples of their subwords. The first technique generates entire samples for an OOV keyword by concatenating samples of its subwords. The second technique splits hypothesized detections into segments then estimates the acoustic similarities between detections and these subword's samples based on the similarities at the subword level. We will show that when interpolating with ASR scores, the acoustic similarities estimated by the proposed techniques result in a better separation between correct detections and false alarm ones. We also show that rescoring KWS detections, using the proposed similarity scores, significantly outperforms the raw ASR score as well as other rescoring methods that do not use the similarity scores.

The paper is organized as follows. Section 2 describes the proposed rescoring framework. Section 3 presents the experimental setup for the KWS task. In section 4, we show the experimental results and discussions. Finally, section 5 concludes our work.

## 2. Rescoring framework for OOV keywords

The proposed rescoring framework for an OOV keyword is shown in Figure 1. When presented with an OOV keyword $q$, the subword-based [1] KWS system first searches over the lattices (generated by an ASR system) to generate a list of detections with scores and timing information. The rescoring system then converts the OOV keyword into a subword representation and extracts a list of samples for each subword, which is described

[1]In this work, we use morpheme as the subword unit, which is described in section 3.3



http://dx.doi.org/10.21437/Interspeech.2016-646

in section 2.1. Next, the acoustic similarities between hypothesized detections and subword samples are estimated through two proposed techniques as described in section 2.2. Finally, these similarities scores are used to rescore the list of detections using methods described in section 2.3 .

### 2.1. Subword samples extraction for an OOV keyword

Suppose keyword $q$ consists of $n(q)$ words, i.e. $q = W_1W_2...W_{n(q)}$. For each word $W_i$, if it does not appear in the training data, split it into subword representations, otherwise keep $W_i$ unchanged. Keeping words that appear in the training data helps to reduce the number of concatenation/splitting, as will be mentioned in section 2.2. The conversion process results in a new representation of keyword $q$, i.e. $q = U_1U_2...U_n$ where $U_i$ are either words or subwords. For simplicity, in the rest of the paper we treat all $U_i$ as subwords. Then, given the time boundaries of subwords in the training transcription, we extract all samples of each subword $U_i$ from the training data.

### 2.2. Acoustic similarity estimation

Given samples for each subword $U_i$ of the keyword $q$, we then estimate the acoustic similarities between hypothesized detections and these samples using two techniques described in following subsections.

#### *2.2.1. Concatenating subword samples*

This technique generates samples of $q$ by concatenating samples of each subword $U_i$, then estimates the acoustic similarities between the detections and keyword samples at the keyword level. For each consecutive concatenation between samples $h_{i,j}$ and $h_{i+1,k}$ , where $h_{i,j}$ is the $j^{th}$ sample of $U_i$ and $h_{i+1,k}$ is the $k^{th}$ sample of $U_{i+1}$, we assign a concatenation penalty $P_{i,i+1}$ that reflects the acoustic mismatch between the two samples. Specifically, a mismatch occurrs at one of the following situations:

- They come from different audio files.
- The speaker's genders of two samples are different
- Two samples are spoken by different speakers.
- The right context of $h_{i+1,k}$ is not the same as the first phoneme of $h_{i,j}$ or the left context of $h_{i,j}$ is not the same as the last phoneme of $h_{i+1,k}$. Here, the the left context (or right context) of a subword refers to the phoneme right before (or right after) of the subword.

Then the concatenation penalty between $h_{i,j}$ and $h_{i+1,k}$ is computed as $P_{i,i+1} = 1.1^{g}$ [2],where $g$ is the number of mismatches between the two samples (e.g. different speaker, audio file or contexts as mentioned above).The total penalty for the whole concatenation is $P = \prod_{i=1}^{n-1} P_{i,i+1}$.

Suppose each subword $U_i$ has $N_{U_i}$ samples, the number of concatenated instances will be $N_{U_1} \times N_{U_2} \times ...N_{U_n}$, which is often large. In this work we only select top 20 concatenated instances with lowest penalties as the samples for $q$. We now can easily estimate the acoustic similarity between each detection $d$ and a concatenated sample $x_i$, denoted as $S(d, x_i)$, using Dynamic Time Warping (DTW) [25]. The average similarity between $d$ and all concatenated samples is then estimated as

$$S(d) = \frac{1}{20} \sum_{i=1}^{20} S(d, x_i) \quad (1)$$

#### *2.2.2. Splitting hypothesized detections*

This technique estimates the acoustic similarities between detections and subword's samples according to the similarities at the subword level. The procedure to estimate such similarity is described as follows:

**Step 1**: Split each detection $d$ into segments, i.e. $d = d_1d_2...d_n$ so that each segment $d_i$ corresponds to a subword $U_i$.

**Step 2**: Estimate the similarity between each $d_i$ and all samples of the subword $U_i$ as

$$S(d_i, U_i) = \frac{1}{N_{U_i}} \sum_{j=1}^{N_{U_i}} S(d_i, h_{i,j}) \quad (2)$$

where $h_{i,j}$ denotes $j^{th}$ sample of $U_i$, and $S(d_i, h_{i,j})$ denotes the acoustic similarity between two speech segments $d_i$ and $h_{i,j}$.

**Step 3**: The acoustic similarity between $d$ and all subword samples is estimated by averaging $S(d_i, samples(U_i))$ through $d_i$, i.e.

$$S(d) = \frac{1}{n} \sum_{i=1}^{n} S(d_i, U_i) \quad (3)$$

This technique does not need to deal with the mismatch cost as in the samples concatenation method in section 2.2.1. It, on the other hand, requires the time boundary information of all subwords for each detection, estimated by the KWS system, which might not be always precise.

### 2.3. Rescoring OOV detections using keyword samples

#### *2.3.1. Rescoring OOV detections by simple interpolation*

For each detection $d$, given the acoustic similarity $S(d)$ as in equation (1) or (3), we estimate a new confidence score for $d$ by interpolating $S(d)$ with the original ASR score (i.e. its lattice posterior probability [14]):

$$C'(d) = C(d)^{\delta} S(d)^{1-\delta} \quad (4)$$

where $C(d)$ is the ASR score of $d$, $\delta$ is a interpolation factor tuned on the development data.

This method is similar to the Pseudo Relevance Feedback (PRF) [26]. Note that, the PRF assumes that top detections (e.g. top 5) are relevant to the keyword $q$, which might not be true; then uses acoustic similarities between $d$ and the top detections to perform rescoring. This method, however, uses the acoustic similarities between $d$ and actual true keyword samples to perform rescoring.

#### *2.3.2. Rescoring OOV detections through graph random walk*

This section explores the use of a graph-based algorithm, which was previously used for the re-ranking task [20–23]. Our work is different from the previous works in that the keyword samples are incorporated into the graph.

For a keyword $q$, a directed graph is constructed from all detections and all samples of the keyword. Each node of the graph represents a detection or a keyword sample and the weight between a pair of nodes is the acoustic similarity between the two

[2]The factor 1.1 reflects the degree of concatenation penalty. Other values greater than 1 are also appropriate

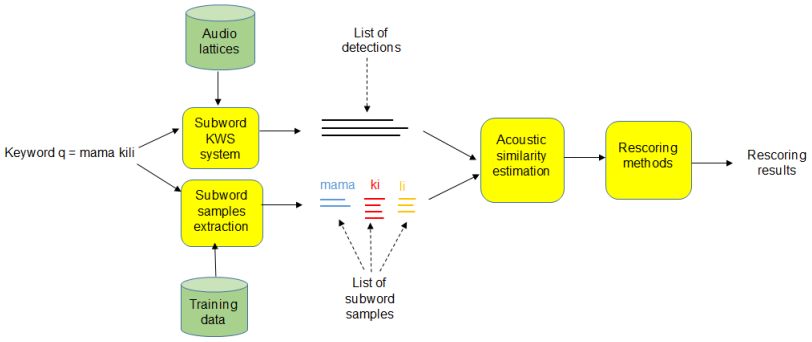


Figure 1: *Proposed rescoring framework for an OOV keyword*

nodes. For the technique in section 2.2.2, since we don't have any sample for the entire keyword $q$, a "representative sample" is created to represent all subword's samples of the keyword $q$. The weight between the representative sample node and a detection node is estimated as equation (3).

Once the graph is constructed, graph-based scores can be estimated by algorithms such as random walk [20, 21]. Specifically, let $C(d)$ be the initial score of a node for detection $d$, a graph-based score $G(d)$ is computed as

$$G(d) = (1-\alpha-\gamma)C(d) + \alpha \sum_{d' \in D(d)} G(d')S'(d,d') + \gamma \sum_{x \in E(d)} G(x)S'(d,x) \quad (5)$$

where $D(d)$ is the set of detections that connect to $d$, $E(d)$ is the set of training samples that connect to $d$, $0 \leq \alpha, \gamma \leq 1$ are scaling factors. $S'(d,d')$ and $S'(d,x)$ are the normalized similarities computed as in [24]. The set of graph scores $G(d)$ can be easily estimated using an iterative method as in [20]. The final confidence score for each detection $d$ is then estimated as

$$C'(d) = C(d)^{\delta} G(d)^{1-\delta} \quad (6)$$

# 3. Experiment setup

## 3.1. Evaluation metric

To evaluate the KWS performance, we use the term-weighted value (TWV) which is the primary metric defined by NIST [1] as following:

$$TWV(\theta) = 1 - \frac{1}{M} \sum_{k=1}^{M} ((P_{miss}(q_k,\theta) + \beta P_{fa}(q_k,\theta)) \quad (7)$$

where $\theta$ is a threshold, $M$ is the number of keywords, $q_k$ is a keyword, $P_{miss}$ and $P_{fa}$ are probabilities of miss and false alarm (FA) respectively. The weight $\beta$ is related with the prior probability of a keyword, and the cost ratio between the false alarm and the miss errors.

The TWV value at a specific threshold $\theta$ is called Actual term-weighted value (ATWV) while the best TWV over all possible $\theta$ is named as maximum term-weighted value (MTWV). In this work, we report the KWS performance on MTWV since it is less sensitive to threshold selection than the ATWV metric.

It is expected that after rescoring, correct detections should rank higher than the false alarm ones. Thus to measure the efficiency of rescoring methods with respect to ranking performance, Mean Average Precision at top 100 (MAP@100) is also used as the evaluation metric in this work. It is worth noting that MTWV and MAP are not always correlated to each other. Moreover, MAP only characterizes the relative order, i.e. the rank, of the detections but not the absolute detection scores as in MTWV metric.

## 3.2. NIST OpenKWS15 Data

The KWS experiments are conducted on Swahili, which is the surprise language for the NIST OpenKWS15 Evaluation. The released acoustic data includes 40 hours of training data, 10 hours development data and 15 hours of part 1 of evaluation data (denoted as $evalpart1$). The training data is used to develop the ASR systems. The $dev10h$ data is used for parameter tuning of KWS, and we evaluate the KWS performance on $evalpart1$. The DTW-based acoustic similarities are estimated using the multilingual bottleneck features trained on various corpora [28], e.g. Cantonese, Turkish. Our initial experiments have shown that these features are better than other well-known features such as MFCC or PLP.

The training transcription is converted to subword representation, as will be mentioned in section 3.3, to build the subword KWS system. As the pronunciation dictionary was not provided during the OpenKWS15 Evaluation, in this work we use the grapheme-based dictionary, i.e. letters are considered as phonemes.

NIST released the evaluation keyword list which contains 4454 keywords. However only 73 OOV keywords appear in the evaluation data. To verify our proposed approach more thoroughly, we manually add 942 OOV keywords to the evaluation keyword list. This results in a final keyword list with 1015 OOV keywords. When estimating MTWV, we only consider detected keywords, i.e. 798 keywords, since the MTWV of undetected keywords are always 0. Similary, when estimating MAP scores, we only consider keywords (535 keywords) that have at least one correct detection.

## 3.3. Keyword Search system

We built a morpheme-based KWS system since morpheme was shown to be effective for detecting OOV keywords [5, 27, 28]. We adopted the Morfessor toolkit [29] to segment both word-based dictionary and word transcriptions into morpheme units. As we used the grapheme-based dictionary, it is straightforward to infer the subword pronunciation dictionary. The open-source Kaldi toolkit [30] is used to build the morpheme-based ASR system. We used filter-bank features to train a deep neural network (DNN) acoustic model. The ASR system used tri-

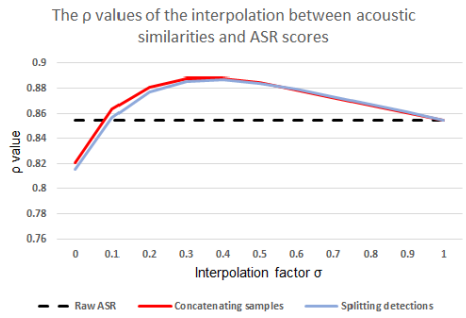


Figure 2: The $\rho$ values of the interpolation (equation (4)) between the acoustic similarites in section 2.2 and raw ASR scores

gram language model. For indexing and search, we utilized the WFST algorithm [8] which is a part of the Kaldi recipe [30].

## 4. Experiment results and discussion

In this section, we first demonstrate that the proposed acoustic similarities estimated in section 2.2 are effective for rescoring KWS detections. Specifically, we show that when interpolating with raw ASR scores, these acoustic similarities help to effectively separate correct detections out of false alarm ones. Then, in the next subsection, the KWS performance (MTWV and MAP@100 scores) are reported on the $evalpart1$ data set.

### 4.1. Effect Size of MannWhitney U test

In this subsection, by adopting the idea of measuring the separation between two populations using Mann-Whitney U test [32], we show that the proposed acoustic similarities in section 2.2 are effective for rescoring KWS detections. Specifically, we show that when interpolating with raw ASR scores, the acoustic similarities result in a better separation between two populations, i.e. correct detections and false alarm detections.

Let $D_{corr}$ and $D_{fa}$ be two set of scores of correct and false alarm detections respectively; and $D = D_{corr} \cup D_{fa}$. We define a non-parametric score $\rho$ [31] as measument of the overlapping between two populations, i.e. $D_{corr}$ and $D_{fa}$. $\rho$ is estimated as follows :

$$R_{corr} = \sum_{d \in D_{corr}} rank(d) \tag{8}$$

$$U_{corr} = R_{corr} - \frac{\| D_{corr} \| (\| D_{corr} \| + 1)}{2} \tag{9}$$

$$\rho = \left| \frac{U_{corr}}{\| D_{corr} \| \times \| D_{fa} \|} - 0.5 \right| \times 2 \tag{10}$$

where $rank(d)$ is the rank of $d$ in $D$, and $\|.\|$ is the cardinal of the set. $\rho$ is between 0 and 1; and higher $\rho$ indicates a better separation of $D_{corr}$ and $D_{fa}$.

Figure 2 shows the $\rho$ values when interpolating the acoustic similarities with the raw ASR scores as in equation (4) at different interpolation factor $\delta$. It can be seen that the interpolation can improve $\rho$ values significantly, e.g. from 0.8548 to 0.8879 (at $\delta = 0.4$). In other words, the acoustic similarities are potential for rescoring KWS detections. Another observation is that the similarity scores generated by the two techniques in section 2.2 provide similar performance.

Table 1: The MTWV and MAP@100 values of two rescoring methods, i.e. simple interpolation and GBRWS, as compared to baselines on $evalpart1$ data set

| Methods | | MTWV | MAP@100 |
|---|---|---|---|
| Raw ASR (baseline) | | 0.4482 | 0.5670 |
| PRF (baseline) | | 0.4581 | 0.5774 |
| Simple interpolation | Concatenating samples | 0.4789 | **0.6304** |
| | Splitting detections | **0.4799** | 0.6243 |
| GBR (baseline) | | 0.4752 | 0.5834 |
| GBRWS | Concatenating samples | **0.4827** | **0.6274** |
| | Splitting detections | 0.4792 | 0.6208 |

### 4.2. KWS performance

This section reports the KWS performance of rescoring methods described in section 2.3. When simple interpolation is used, the PRF is selected as the baseline. When graph random walk is used, we compare the graph-based rescoring with sample (denoted as GBRWS) with the graph-based rescoring without samples (denoted as GBR) [20, 21]. The performance of raw ASR score is also reported as another baseline for comparison.

The KWS results, i.e. MTWV and MAP@100, are presented in Table 1. It can be seen that both techniques in section 2.2 provide similar performance. This is consistent with the observation in section 4.1. Another point is that rescoring methods using the proposed acoustic similarities significantly outperform all baselines. When using the acoustic similarities estimated in section 2.2.1, the simple interpolation method provides 2.1% MTWV and 5.3% MAP@100 absolute improvement over the baseline PRF. The proposed GBRWS also outperforms the baseline GBR on both MTWV and MAP@100 metrics: although the improvement on MTWV is marginal, the performance improvement on MAP@100 is significant, i.e. 4.4% absolute, when using the similarities estimated in section 2.2.1. This can be explained that in many keywords, the rescoring process changes the orders (i.e. changes the MAP@100) in the detection list but does not change the YES/NO decisions of the detections (i.e. MTWV remained the same).

## 5. Conclusion

We proposed two techniques to estimate the acoustic similarities between detections and subword's samples of OOV keywords, so that rescoring methods can be applicable for such keywords. The first technique concatenates samples of their constituent subwords to form samples of the entire keywords. The second technique splits hypothesized detections into segments, then estimates the similarities between the detections and subword samples using similarities between each segments and these samples. We showed that the proposed similarity scores are helpful for rescoring the list of detections. Specifically, when interpolating with ASR scores, the similarities help to separate effeciently correct detections out of false alarm detections. Experimental results also show that rescoring methods using the similarity scores outperform significantly other methods that do not use such similarity scores.

## 6. Acknowledgment

This work is supported by the DSO funded project MAISON DSOCL14045, Singapore.

# 7. References


[1] NIST, "The spoken term detection (std) 2006 evaluation plan," in *http://www.nist.gov/speech/tests/std*, 2006.

[2] NIST, "The open keyword search (std) 2013 evaluation," in *http://www.nist.gov/itl/iad/mig/upload/OpenKWS13-evalplan-v4.pdf*, 2013.

[3] N. F. Chen, S. Sivadas, B. P. Lim, H. G. Ngo, H. Xu, V. T. Pham, B. Ma, and H. Li, "Strategies for vietnamese keyword search," in *Proceedings of ICASSP*, 2014.

[4] V. T. Pham, N. F. Chen, S. Sivadas, H. Xu, I. Chen, C. Ni, E. S. Chng and H. Li, "System and keyword dependent fusion for spoken term detection," in *Proceedings of SLT*, 2014.

[5] N. Chen, C. Ni, S. Sivadas, V. T. Pham, H. Xu, X. Xiao, T. S. Lau, S. J. Leow, B. P. Lim, C. C. Leung, L. Wang, C. H. Lee, A. Goh, E. S. Chng, B. Ma, and H. Li, "Low-resource keyword search strategies for tamil," in *Proceedings of ICASSP*, 2015.

[6] N. F. Chen, V. T. Pham, H. Xu, X. Xiao, V. H. Do, C. Ni, I. Chen, S. Sivadas, C. H. Lee, E. S. Chng, B. Ma, and H. Li, "Exemplar-inspired strategies for low-resource spoken keyword search in swahili," in *Proceedings of ICASSP*, 2016.

[7] C. Allauzen, M. Mohri, and M. Saraclar, "General Indexation of Weighted Automata Application to Spoken Utterance Retrieval," in *Proceedings of HLT*, 2004.

[8] D. Can and M. Saraclar, "Lattice Indexing for Spoken Term Detection," *IEEE Trans. Speech Audio Process*, vol. 19, no. 8, 2011.

[9] S. Parlak and M. Saraclar, "Spoken term detection for turkish broadcast news," in *Proceedings of ICASSP*, 2008.

[10] M. Saraclar and R. Sproat, "Lattice-Based Search for Spoken Utterance Retrieval," in *Proceedings of HLT-NAACL*, 2004.

[11] C. Chelba and J. Silva, "Soft indexing of speech content for search in spoken documents," *Computer Speech and Language*, vol. 21, no. 3, pp. 458–478, 2007.

[12] J. Mamou, D. Carmel, and R. Hoory, "Spoken document retrieval from call-center conversation," in *Proceedings of SIGIR*, 2006.

[13] K. Thambiratnam and S. Sridharan, "Rapid yet accurate speech indexing using Dynamic Match Lattice Spotting," *IEEE Trans. Speech Audio Process*, vol. 15, no. 1, 2007.

[14] G. Evermann and P. Woodland, "Large vocabulary decoding and confidence estimation using word posterior probabilities," in *Proceedings of ICASSP*, 2000.

[15] V. Soto, L. Mangu, A. Rosenberg, and J. Hirschberg, "A comparison of multiple methods for rescoring keyword search lists for low resource languages," in *Proceedings of Interspeech*, 2014.

[16] J. Tejedor, D. T. Toledano, D. Wang, S. King, and J. Colas, "Feature analysis for discriminative confidence estimation in spoken term detection," *Speech Communication*, vol. 28, no. 5, 2014.

[17] O. Vinyals and S. Wegmann, "Chasing the metric: Smoothing learning algorithms for keyword detection," in *Proceedings of ICASSP*, 2014.

[18] V. T. Pham, H. Xu, N. F. Chen, S. Sivadas, B. P. Lim, E. S. Chng, and H. Li, "Discriminative score normalization for keyword search decision," in *Proceedings of ICASSP*, 2014.

[19] M. Seigel, P. Woodland, and M. Gales, "A confidence-based approach for improving keyword hypothesis scores," in *Proceedings of ICASSP*, 2013.

[20] H. Y. Lee, Y. Zhang, E. Chuangsuwanich, and J. Glass, "Graph-based re-ranking using acoustic feature similarity between search results for spoken term detection on low-resource," in *Proceedings of ICASSP*, 2013.

[21] H. Y. Lee and L. S. Lee, "Improved semantic retrieval of spoken content by document/query expansion with random walk over acoustic similarity graphs," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, 2013, pp. 80–94.

[22] Y. N. Chen, C. P. Chen, H. Y. Lee, C. Chan, and L. S. Lee, "Improved spoken term detection with graph-based re-ranking in feature space," in *Proceedings of ICASSP*, 2011.

[23] A. Norouzian, R. C. Rose, S. H. Ghalehjegh, and A. Jansen, "Zero resource graph-based confidence estimation for open vocabulary spoken term detection," in *Proceedings of ICASSP*, 2013.

[24] V. T. Pham, H. Xu, X. X., N. F. Chen, E. S. Chng, and H. Li, "Keyword search using query expansion for graph-based rescoring of hypothesized detections," in *Proceedings of ICASSP*. IEEE, 2016.

[25] 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 *Proceedings of ICASSP*, 2014.

[26] H. Yi Lee, P. Wei Chou, and L. Shan Lee, "Improved open-vocabulary spoken content retrieval with word and subword lattices using acoustic feature similarity," in *Computer speech and language*, 2014.

[27] Y. He. et.al, "Subword-based modeling for handling oov words inkeyword spotting," in *Proceedings of ICASSP*, 2014.

[28] V. T. Pham, H. Xu, T. Y. Chong, X. Xiao, E. S. Chng, , and H. Li, "On the study of very low-resource language keyword search," in *Proceedings of APSIPA*, 2015.

[29] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *In Proceedings of the Workshop on Morphological and Phonological Learning of ACL-02*, 2002.

[30] D. Povey. et.al, "The kaldi speech recognition toolkit," in *Proceedings of ASRU*, 2011.

[31] T. H. Nguyen, E. Chng, and H. Li, "T-test distance and clustering criterion for speaker diarization," in *Proceedings of Interspeech*, 2008.

[32] B. Mann, R. Whitney, "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other," in *Annals of Mathematical Statistics*, 1947.

[33] D. Karakos et.al.,"Score normalization and system combination for improved keyword spotting," in *Proceedings of ASRU*. IEEE, 2013.