IMPROVED SYSTEM FUSION FOR KEYWORD SEARCH

Zhiqiang Lv, Meng Cai, Cheng Lu, Jian Kang, Like Hui, Wei-Qiang Zhang and Jia Liu

Tsinghua National Laboratory for Information Science and Technology
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
{lv-zq12,cai-m10}@mails.tsinghua.edu.cn, lucheng1983@163.com,
{kangj13,hlk14}@mails.tsinghua.edu.cn, {wqzhang,liuj}@tsinghua.edu.cn

ABSTRACT

It has been demonstrated that system fusion can significantly improve the performance of keyword search. In this paper, we compare the performance of several widely-used arithmetic-based fusion methods using different normalization pipeline and try to find the best pipeline. A novel arithmetic-based fusion method is proposed in this work. The method supplies a more effective way to incorporate the number of systems which have non-zero scores for a detection. When tested on the development test dataset of the OpenKWS15 Evaluation, the proposed method achieves the highest maximum term-weighted value (MTWV) and actual term-weighted value (ATWV) among all other arithmetic-based fusion methods. Usually, discriminative fusion methods employing classifiers can outperform arithmetic-based fusion methods. A DNN-based fusion method is explored in this work. After word-burst information is added, the DNN-based fusion method outperforms all other methods. In addition, it is notable that our arithmetic-based method achieves the same MTWV as the DNN-based method.

Index Terms— system fusion, keyword search, score normalization, DNN

1. INTRODUCTION

Keyword search (KWS) is to find all the occurrences of given keywords in untranscribed speech. A typical KWS system consists of two phases : indexing and searching. In the indexing phase, every audio of speech is indexed after being processed by a large vocabulary continuous speech recognition system. In the searching phase, keywords are searched on the index to produce the final list of all detections. With the rapid development of computer hardwares, it is possible to build more than one KWS systems for the same KWS task. By fusing KWS results from diverse systems, we can usually get a much better KWS result.

This work is supported by National Natural Science Foundation of China under Grant No. 61273268, No. 61370034, No. 61403224 and No. 61005017.

For fusing results of different systems, arithmetic-based fusion methods such as CombSum [1, 2], CombMNZ [1, 2], CombGMNZ [1], WCombMNZ [2] have been proved to be quite effective. Pham et al. [3] proposed the system and keyword dependent fusion method SKDWCombMNZ in 2014, which ourperformed other arithmetic-based methods. Discriminative system fusion methods employing classifiers have been explored in [3, 4, 5]. With large number of features from lattices and detection lists, discriminative fusion can often achieve inspiring performance.

In this paper, the actual term-weighted value (ATWV) [6] and the maximum term-weighted value (MTWV) [6] are used as the measures for KWS performance. For the two measures, score normalization [2, 7, 8] has been proved to be essential. Keyword specific threshold (KST) normalization [9] and sum-to-one (STO) normalization [2] are the two mainstream score normalization methods. In our work, we compare the performance of the two methods when they are applied both before and after system fusion. We also explore the best normalization pipeline when dealing with fusion of up to 11 systems and some quite different conclusions are presented. In addition, we propose a novel arithmetic-based fusion method which is similar to SKDWCombMNZ but more effective and simpler. For discriminative fusion methods, we extend the MLP-based classifier used in [3] to a DNN-based one and some more effective features are extracted to get better performance.

This paper is organized as below: Section 2 is the description of the keyword search task. Fusion methods and normalization methods are introduced in Section 3. Experiments are presented in Section 4. Section 5 contains conclusions.

2. TASK DESCRIPTION

The task of KWS defind by NIST for the OpenKWS15 Evaluation is to find all the exact matches of given queries in a corpus of un-segmented speech data. A query, which can also be called “keyword”, can be a sequence of one or more words. The result of this task is a list of all the detections of keywords found by KWS systems.

To evaluate the performance, term-weighted value (TWV) [6] is adopted:

$$TWV(\theta) = 1 - \frac{1}{K}\sum_{w=1}^{K}(\frac{\#miss(w,\theta)}{\#ref(w)} + \beta\frac{\#fa(w,\theta)}{T - \#ref(w)}) \tag{1}$$

where θ is the decision threshold and K is the number of keywords. $\#miss(w,\theta)$ is the number of true tokens of keyword w that are missed at threshold θ. $\#fa(w,\theta)$ is the number of false detections of keyword w at threshold θ. $\#ref(w)$ is the number of reference tokens of w. T is the total amount of the evaluated speech. β is a constant. As we can see, TWV is a function of the decision threshold θ. ATWV is the TWV at a specific θ. MTWV is the maximum TWV over all possible values of θ.

3. SYSTEM FUSION

3.1. Arithmetic-based system fusion

As is mentioned above, several arithmetic-based system fusion methods from document retrieval have been applied successfully in KWS. Here we only introduce WCombSum, WCombMNZ and WCombGMNZ. WCombSum is a quite straightforward method:

$$s(h) = \sum_{i=1}^{N} w_i \cdot s_i \tag{2}$$

where w_i is proportional to MTWV achieved by system i [2] and N is the number of fused systems. WCombMNZ incorporates the number of systems which have non-zero scores for a detection into the fusion procedure, and we denote the number as $m(h)$:

$$s(h) = m(h) \times \sum_{i=1}^{N} w_i \cdot s_i \tag{3}$$

WCombGMNZ is a generalizaiton form of WCombMNZ and its formula is:

$$s(h) = m(h)^{\gamma} \times \sum_{i=1}^{N} w_i \cdot s_i, (\gamma \geq 0) \tag{4}$$

When γ is set to 1, WCombGMNZ is equivalent to WCombMNZ. When γ is set to 0, WCombGMNZ is equivalent to WCombSum. SKDWCombMNZ is another extension of WCombMNZ:

$$s(h) = m(h) \cdot (\sum_{i=1}^{N} w_i \cdot s_i)^{\frac{1}{\gamma + \alpha \cdot n(h)}}, (0 < \gamma \leq 1, 0 \leq \alpha \leq 1) \tag{5}$$

where $n(h)$ is the number of systems which accept the detection h as a true one.

Compared with WCombSum, other three methods tend to believe that detections found or accepted by more systems are more reliable. The linear multiplication of $m(h)$ makes big gaps between detections with different $m(h)$ and overemphasizes the importance of $m(h)$. This may restrict the potential improvement in the region of high scores where most detections are true. That is to say, for detections with relatively high scores, we want to incorporate $m(h)$ more smoothly. Therefore, a new fusion method is proposed:

$$s(h) = (\sum_{i=1}^{N} w_i \cdot s_i)^{\frac{1}{m(h)^{\gamma}}}, (\gamma \geq 0) \tag{6}$$

where γ is a parameter for adjusting the boost of different $m(h)$. We denote $\frac{1}{m(h)^{\gamma}}$ as $IDF(\gamma)$. IDF is short for "Inverse Document Frequency" [10], which has been widely used in document retrieval. Here we use $IDF(\gamma)$ to measure how much information is provided by the number of systems that have non-zero scores for a detection. Then the method can be rewritten as:

$$\begin{aligned} log(s(h)) &= \frac{1}{m(h)^{\gamma}} log(\sum_{i=1}^{N} w_i \cdot s_i) \\ &= IDF(\gamma) \cdot log(s(h)_{WCombSum}) \end{aligned} \tag{7}$$

where $s(h)_{WCombSum}$ is the fused score of the method WCombSum. We denote the new proposed method as IDFWCombSum. Similarily, an IDFWCombMNZ method can be written as:

$$log(s(h)) = IDF(\gamma) \cdot log(s(h)_{WCombMNZ}) \tag{8}$$

As we can see, SKDWCombMNZ uses two paramaters for adjusting the boost of different $n(h)$, while IDFWCombSum and IDFWCombMNZ only have one, which makes it easier for our methods to optimize parameters. Furthermore, IDFWCombSum discards the simple linear multiplication of $m(h)$ and is indeed a completely different method to incorporate $m(h)$.

3.2. Score normalization before and after system fusion

For the TWV metrics, normalization has been proved to be essential. KST normalization first computes a specific threshold for every keyword. At the specific threshold, the expectation value of ATWV contributed by the keyword is zero. The threshold is:

$$thr(w) = \frac{N_{true}(w)}{T/\beta + \frac{\beta - 1}{\beta} N_{true}(w)} \tag{9}$$

where $N_{true}(w)$ is the number of reference tokens of keyword w. T is the total amount of the evaluated speech. β is a constant of 999.9. $N_{true}(w)$ is unknown and is estimated using the following formula:

$$N_{true}(w) = \sum_{j} s(w)_j \tag{10}$$

where $s(w)_j$ is the j-th detection's posterior probability of keyword w.

Then the specific threshold is mapped to a fixed value (e.g. 0.5). A non-linear function is utilized to map the origional score to the normalized score. Here we adopt the function from Kaldi [11]:

$$KST(s(w)) = \frac{(1 - thr(w)) \cdot s(w)}{(1 - thr(w)) \cdot s(w) + (1 - s(w)) \cdot thr(w)} \quad (11)$$

STO normalization is rather simple:

$$STO(s(w)_i) = \frac{s(w)_i}{\sum_j s(w)_j} \quad (12)$$

It is very straightforward that normlization should be done after system fusion, just as what has been demonstrated on individual systems. Though it has been suggested that normalization should be done before system fusion as well [8], we are still very interested in what normalization should be adopted and whether the normalization is indeed needed, especially when fusing many systems (e.g. more than 3).

3.3. Discriminative system fusion

It has been demonstrated that discriminative system fusion can achieve the best performance compared with arithmetic-based fusion methods such as WCombMNZ, SKDWCombMNZ [3]. The discriminative system fusion method in [3] employed an MLP as the classifier. In our work, we replace the MLP classifier with a DNN classifier. Features are extracted only from detection lists of every system. Lattice-based features such as ranking-score and relative-to-max [3, 12] are not used because extracting these features from lattices can be very time-consuming, especially for fusion of quite many systems. Features from detection lists are as below:

1. Original scores of the detection from every system and their mean value and variance.

2. STO scores of the detection from every system and their mean value and variance.

3. The WCombSum score.

4. The distance in time of the detection relative to the start time and the end time of the segment to which the detection belongs.

5. The number of vowels and consonants of the keyword [3].

6. The number of systems which have non-zeros scores for the detection and the number of systems which accept the detection [3].

7. The duration of the detection and the average duration of every phoneme.

Besides, Richards et al. [13] introduced word-burst information in KWS and consistent improvement was observed. Similar word-burst features are extracted:

8. $\frac{score(w_j)}{dist(w_i,w_j)}$, $\frac{STO(score(w_j))}{dist(w_i,w_j)}$, where w_j is the closest detection of keyword w to the target detection w_i in time. $dist(w_i, w_j)$ is the distance in time between detection w_i and w_j.

9. $\sum_j \frac{score(w_j)}{dist(w_i,w_j)}$, $\sum_j \frac{STO(score(w_j))}{dist(w_i,w_j)}$.

10. The maximum, minimum, mean values of $\frac{score(w_j)}{dist(w_i,w_j)}$ and $\frac{STO(score(w_j))}{dist(w_i,w_j)}$. Here w_j is any repetition of keyword w in the same audio of speech as the target detection w_i.

4. EXPERIMENTS

4.1. Data

All the KWS experiments are conducted using the datasets from the NIST OpenKWS15 Evaluation of Swahili. The training data used for building KWS systems includes the Very Limited Language Pack (VLLP) of the OpenKWS15 Evaluation (denoted as 202VLLP) , the full language packs of 6 languages under the Babel program (denoted as BP&204FullLP) and the web data of the OpenKWS15 Evaluation (denoted as 202Web). 202VLLP consists of 3 hours' transcribed speech of Swahili, while BP&204FullLP consists of about 528 hours' transcribed speech of Cantonese, Pashto, Turkish, Tagalog, Vietnamese and Tamil. 202Web consists of plenty of raw web text.

The acoustic model is trained using 202VLLP. The language model is trained using 202VLLP and part of 202Web. All the results are reported on the 10-hour development test data of Swahili from the OpenKWS15 Evaluation datasets. Parameters are tuned on the tuning set released by NIST for the development of OpenKWS15.

The keyword list is the one for development from the "IndusDB" of the OpenKWS15 Evaluation. It consists of 2480 keywords.

4.2. KWS systems

For the fusion experiments, up to 11 diverse systems are built. More than half of the systems utilize the multilingual bottleneck (MBN) features trained with BP&204FullLP.

The baseline system S1 uses convolutional maxout neural network acoustic model [14, 15] with filter-bank features.

S2 uses RNN acoustic model with MBN features.

S3 uses DNN acoustic model with speaker adapted MBN features.

S4 uses P-norm maxout neural network acoustic model [16] with MBN features.

S5 uses DNN acoustic model with filter-bank plus pitch features.

S6 uses DNN acoustic model with MBN features.

S7 uses DNN acoustic model with PLP plus pitch features.

S8 uses convolutional recurrent neural network acoustic model with filter-bank features.

S9 uses LSTM RNN acoustic model [17] with sMBR sequence training and MBN features.

S10 uses subspace GMM acoustic model [18] with speaker adapted MBN features.

S11 uses DNN acoustic model with speaker adapted MBN features based on HTK.

Among them, S1-S10 are based on Kaldi, while S11 is based on HTK. The language model of S1-10 is a word trigram language model, while S11 utilizes a feed-forward neural network language model with variance regularizations [19]. Besids, S11 employs our own decoder [19] while other systems employ the Kaldi decoder. The TWV results of our KWS systems after KST normalization are listed in Table 1.

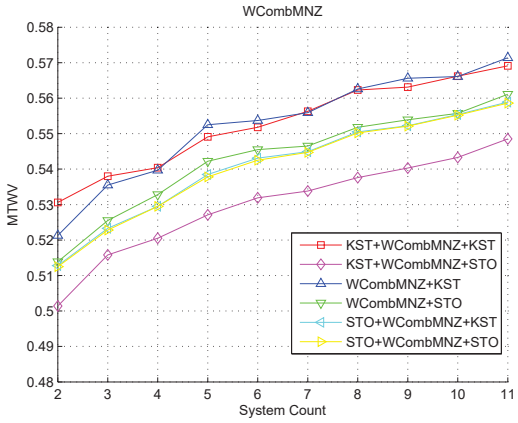
4.3. Results of score normalization before and after system fusion

In this section, fusion experiments from 2 systems to 11 systems using WCombSum and WCombMNZ are conducted. We want to explore the performance of system fusion on different number of systems using different fusion pipeline. KST normalization, STO normalization and no normalization are done before fusion separately. After fusion, normalization is usually essential. Therefore, KST normalization and STO normalization are chosen after fusion. From experiments in this section, we try to find out whether normalization is needed before fusion and what kind of normalization can lead to a better result. The results of pipelines using different normalization methods are shown in Figure 1.

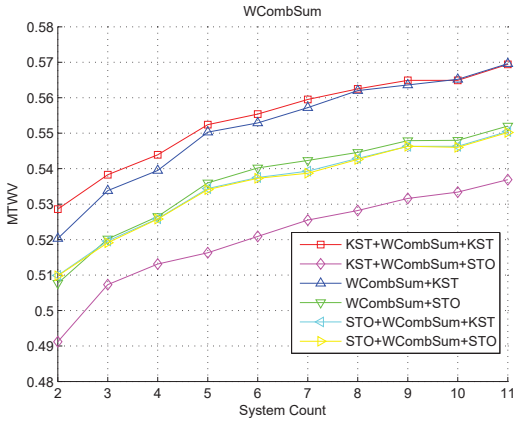
Fusion of different number of systems is incrementally done from S1 to S11, in the decreasing order of MTWV. From the results, we can obviously see that KST normalization after fusion outperforms that of STO normalization. The best performance is achieved by either KST normalization or no normalization before fusion. KST normalization before fusion performs best for fusing a few systems, while no normalization before fusion performs best when fusing more systems.

Table 1. TWV results of our baseline KWS systems after KST normalization

System	ATWV	MTWV
S1: CMNN, fbank, CE	**0.4785**	**0.4829**
S2: RNN, MBN, sMBR	0.4741	0.4778
S3: DNN, SAT, sMBR	0.4667	0.4712
S4: pnorm, MBN, CE	0.4666	0.4712
S5: DNN, fbank+pitch, sMBR	0.4586	0.4675
S6: DNN, MBN, sMBR	0.4620	0.4666
S7: DNN, PLP+pitch, sMBR	0.4347	0.4562
S8: CRNN, fbank, sMBR	0.4482	0.4419
S9: LSTM, MBN, sMBR	0.4261	0.4333
S10: SGMM, MBN, BMMI	0.4263	0.4331
S11: DNN, SAT, CE, NNLM	0.4302	0.4308



(a) WCombMNZ



(b) WCombSum

Fig. 1. MTWV results using different normalization methods before and after system fusion.

This can be explained by the central limit theorem. Normalization before fusion tries to get rid of the impact of system-specific biases. Given more scores (posterior probabilities) from different systems, the expectation of the total bias introduced by each system tends to be zero and therefore has little impact on the final result.

Besides, we compare the performance of the two widely-used arithmetic-based methods WCombSum and WCombMNZ, using the best normalization pipeline demonstrated above. The results are shown in Figure 2.

For the MTWV metric, WCombSum with KST normalization both before and after fusion performs best almost for all the count that is less than 8, while WCombMNZ with KST normalization after fusion performs best when the system count is greater than 8. Though the best pipeline for the MTWV metric is not consistent, WCombMNZ with KST nor-

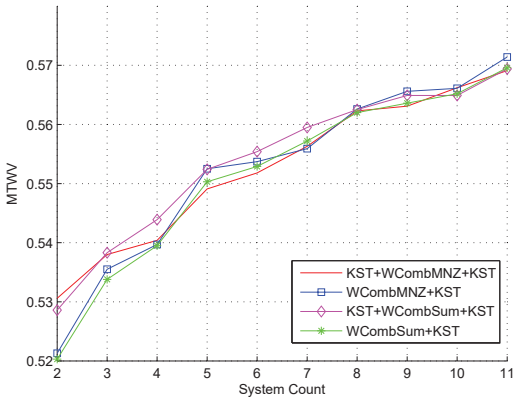


Fig. 2. MTWV results of WCombMNZ and WCombSum using the best normalization pipeline.

malization after fusion outperforms other methods on all conditions for the ATWV metric. For the final fusion of all the 11 systems, WCombMNZ with KST normalization after fusion achieves the best MTWV and ATWV.

4.4. Comparison of arithmetic-based system fusion methods

In Section 4.3, we have found that for fusing different number of systems, different normalization pipeline should be adopted. Here we conduct experiments using more arithmetic-based methods to fuse all the 11 systems, including WCombGMNZ, SKDWCombMNZ, IDFWCombMNZ and IDFWCombSum. The normalization pipeline adopted here is KST normalization after fusion, which has been demonstrated best above for fusing 11 systems. Results are presented in Table 2.

Table 2. Results of different arithmetic-based methods on system fusion of 11 systems

Methods	ATWV	MTWV
WCombMNZ+KST	0.5711	0.5714
WCombGMNZ+KST ($\gamma = 0.4$)	0.5711	0.5720
SKDWCombMNZ+KST ($\gamma = 1.0, \alpha = 0.1$)	0.5703	0.5712
IDFWCombMNZ+KST ($\gamma = 0.2$)	0.5696	0.5722
IDFWCombSum+KST ($\gamma = 0.2$)	**0.5747**	**0.5759**

We can see that WCombGMNZ achieves slightly better MTWV than WCombMNZ. SKDWCombMNZ doesn't show improvement in our experiments, maybe due to the severe VLLP condition. Our methods IDFWCombMNZ and IDFWCombSum both achieve better MTWV than WCombMNZ. IDFWCombSum outperforms all other methods and gains the maximum improvement of 0.45% for the MTWV metric over the baseline WCombMNZ.

We also test the performance of IDFWCombSum for fusing different number of systems and the MTWV results are shown in Table 3.

Table 3. Results of IDFWCombSum for fusing different number of systems

System Count	Pipeline	MTWV
2	KST+WCombMNZ+KST	**0.5306**
	IDFWCombSum+KST($\gamma = 0.9$)	0.5271
3	KST+WCombSum+KST	0.5383
	IDFWCombSum+KST($\gamma = 0.7$)	**0.5420**
4	KST+WCombSum+KST	0.5439
	IDFWCombSum+KST($\gamma = 0.4$)	**0.5453**
5	KST+WCombSum+KST	0.5524
	IDFWCombSum+KST($\gamma = 0.3$)	**0.5576**
6	KST+WCombSum+KST	0.5554
	IDFWCombSum+KST($\gamma = 0.4$)	**0.5610**
7	KST+WCombSum+KST	0.5595
	IDFWCombSum+KST($\gamma = 0.3$)	**0.5643**
8	WCombMNZ+KST	0.5626
	IDFWCombSum+KST($\gamma = 0.3$)	**0.5680**
9	WCombMNZ+KST	0.5656
	IDFWCombSum+KST($\gamma = 0.2$)	**0.5709**
10	WCombMNZ+KST	0.5661
	IDFWCombSum+KST($\gamma = 0.2$)	**0.5724**
11	WCombMNZ+KST	0.5714
	IDFWCombSum+KST($\gamma = 0.2$)	**0.5759**

For every system count in Table 3, the upper line is the best pipeline of system fusion using WCombSum and WCombMNZ, while the lower line is the performance of IDFWCombSum. For the MTWV metric, consistent improvement has been observed when fusing more than 2 systems. In addition, we can see that the pipeline with KST normalization after IDFWCombSum is a consistent one and can provide us an easier way to obtain the best result.

4.5. Results of discriminative system fusion

For discriminative system fusion, a DNN classifier for binary classification is built. Our DNN classifier consists of 3 hidden layers that have 64, 64, 8 nodes separately. Training data for the DNN classifier is obtained from the tuning dataset, using an augmented keyword list of up to 12,000 keywords. Large number of features including word-burst features are

extracted only from detection lists. We denote the DNN experiment using word-burst information as DNN-wordBurst, and the experiment without word-burst information as DNN-baseline. For comparison, KST normalization is done after the DNN-based fusion. The discriminative system fusion results of fusing 11 systems are presented in Table 4.

Table 4. Results of DNN-based fusion of 11 systems

Methods	ATWV	MTWV
WCombMNZ+KST	0.5711	0.5714
IDFWCombSum+KST ($\gamma = 0.2$)	0.5747	**0.5759**
DNN-baseline+KST	0.5703	0.5712
DNN-wordBurst+KST	**0.5749**	**0.5759**

The DNN-baseline achieves very similar performance to WCombMNZ, while DNN-wordBurst achieves the highest MTWV and ATWV among all the methods. By adding some more features from lattices, the performance of DNN-based fusion may be better. However, our method only extracts features from detection lists and achieves the best performance, which makes it much easier for us to build a state-of-the-art fusion system, especially for fusing a lot of systems. Besides, it is worthwhile to note that our arithmetic-based fusion method IDFWCombSum gets the same MTWV as DNN-wordBurst.

5. CONCLUSIONS

In this paper, we compare the performance of two widely-used fusion methods WCombSum and WCombMNZ using different normalization pipeline for keyword search. We find that normalization after fusion is always essential, while normalization before fusion is only needed for fusing not so many systems. WCombMNZ outperforms WCombSum when fusing a lot of systems, while WCombSum performs better for fusing fewer systems.

A novel arithmetic-based fusion method IDFWCombSum is proposed in this work and achieves state-of-the-art performance. For discriminative system fusion, we explore a DNN-based fusion method employing features only from detection lists. When combined with word-burst information, the DNN-based fusion method achieves the hightest MTWV and ATWV.

6. REFERENCES

[1] J. H. Lee, "Analyses of Multiple Evidence Combination," ACM SIGIR Forum Volume 31 Issue SI Pages 267-276.

[2] J. Mamou, J. Cui, X. Cui, M. J. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schluter, A. Sethy, and P. C. Woodland, "System combination and score normalization for spoken term detection," in *Proc. ICASSP*, 2013, pp. 8272–8276.

[3] V. T. Pham, N. F. Chen, S. Sivadas, H. Xu, I. F. Chen, C. Ni, E. S. Chng, and H. Li, "System and keyword dependent fusion for spoken term detection," in *Proc. SLT*, 2014, pp. 430–435.

[4] P. Motlicek, F. Valente, and I. Szoke, "Improving acoustic based keyword spotting using LVCSR lattices," in *Proc. ICASSP*, 2012, pp. 4413–4416.

[5] L. Mangu, H. Soltau, H. K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Proc. ICASSP*, 2013, pp. 8282–8286.

[6] "KWS15 keyword search evaluation plan," `http://www.nist.gov/itl/iad/mig/upload/KWS15-evalplan-v05.pdf`, 2015.

[7] B. Zhang, R. Schwartz, S. Tsakalidis, L. Nguyen, and S. Matsoukas, "White listing and score normalization for keyword spotting of noisy speech," in *Proc. Interspeech*, 2012.

[8] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen, J. Makhoul, F. Grezl, M. Hannemann, M. Karafiat, I. Szoke, K. Vesely, L. Lamel, and V. B. Le, "Score normalization and system combination for improved keyword spotting," in *Proc. ASRU*, 2013, pp. 210–215.

[9] D. R. H. 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 *Proc. Interspeech*, 2007.

[10] K. Sparck Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of documentation*, vol. 28, no. 1, pp. 11–21, 1972.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, Dec. 2011.

[12] 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 *Proc. ICASSP*, 2014, pp. 7078–7082.

[13] J. Richards, *Echolocation: Using Word-Burst Analysis to Rescore Keyword Search Candidates in Low-Resource Languages*, City University of New York, 2014.

[14] M. Cai, Y. Shi, J. Kang, J. Liu, and T. Su, "Convolutional maxout neural networks for low-resource speech recognition," in *Proc. ISCSLP*, 2014, pp. 133–137.

[15] M. Cai, Z. Lv, Y. Shi, W. Wu, W. Q. Zhang, and J. Liu, "The THUEE system for the OpenKWS14 keyword search evaluation," in *Proc. ICASSP*, 2015, pp. 4734–4738.

[16] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in *Proc. ICASSP*, 2014, pp. 215–219.

[17] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, 2014.

[18] D. Povey, L. Burget, M. Agarwal, P. Akyazi, F. Kai, A. Ghoshal, O. Glembek, N. Goel, M. Karafiát, A. Rastrow, R. C. Rose, P. Schwarz, and S. Thomas, "The subspace gaussian mixture model—A structured model for speech recognition," *Computer Speech & Language*, vol. 25, no. 2, pp. 404–439, 2011.

[19] Y. Shi, W. Q. Zhang, M. Cai, and J. Liu, "Efficient one-pass decoding with NNLM for speech recognition," *Signal Processing Letters, IEEE*, vol. 21, no. 4, pp. 377–381, 2014.