APPROXIMATE SEARCH OF AUDIO QUERIES BY USING DTW WITH PHONE TIME BOUNDARY AND DATA AUGMENTATION

Haihua Xu[1], *Jingyong Hou*[3], *Xiong Xiao*[1], *Van Tung Pham*[2], *Cheung-Chi Leung*[4], *Lei Wang*[4], *Van Hai Do*[1]
Hang Lv[3], *Lei Xie*[3], *Bin Ma*[4], *Eng Siong Chng*[1,2], *Haizhou Li*[1,2,4]

[1]Temasek Laboratories, Nanyang Technological University, Singapore
[2]School of Computer Engineering, Nanyang Technological University, Singapore
[3]School of Computer Science, Northwestern Polytechnical University(NWPU), Xi'an, China
[4]Institute for Infocomm Research(I^{2}R), A*STAR, Singapore

ABSTRACT

Dynamic Time Warping (DTW) is widely used in language independent query-by-example (QbE) spoken term detection (STD) tasks due to its high performance. However, there are two limitations of DTW based template matching, 1) it is not straightforward to perform approximate match of audio queries; 2) DTW is sensitive to the mismatch of signal conditions between the query and the speech search data. To allow approximate search, we propose a partial template matching strategy using phone time boundary information generated by a phone recognizer. To have more invariant representation of audio signals, we use bottleneck features (BNF) as the input of DTW. The BNF network is trained from augmented data, which is generated by adding reverberation and additive noises to the clean training data. Experimental results on QUESST 2015 task shows the effectiveness of the proposed methods for QbE-STD when the queries and search data are both distorted by reverberation and noises.

Index Terms— DTW, Query-by-example, spoken term detection, partial matching, data augmentation

1. INTRODUCTION

Query-by-Example (QbE) Spoken Term Detection (STD) is a special STD task where the queries are also audio signals. In the past few years, the QbE-STD research has been mainly driven by the QUESST task [1, 2] in the MediaEval workshops. In the recent QUESST tasks, the problem becomes more and more realistic. For example, conversational speech data and queries distorted by additive noises and reverberation are used in QUESST 2015 task [2]. Approximate matching of queries to speech database is also introduced to the task as it is a common phenomenon in real life applications.

There are several approaches for QbE-STD task, including template matching based on subsequence dynamic time warping (DTW) [3], acoustic keyword spotting [4], and symbolic search that relies on a phone recognizer [4, 5, 6]. DTW based template matching is the most popular approach due to its high performance. However, the DTW approach has several limitations. First, the DTW system is difficult to scale up due to the difficulty to index. Second, DTW is sensitive to variations of the same word spoken in different context, by different speakers, and in different environments. Third, it is not straightforward to perform approximate search in DTW. In this paper, we will mainly address the second and third limitations.

Haihua Xu, Van Tung Pham and Xiong Xiao are supproted by the DSO funded project MAISON DSOCL14045, Singapore.

Various features have been proposed in the past to enhance the robustness of the DTW based template matching. In [7, 8], phone posterior features are used as the input features of DTW for the QbE-STD task. In [9], manifold based features using longer temporal contexts were proposed, alleviating the effect of both speaker and phonetic context variations. In [10], a multiple feature stream fusion method was proposed to compute the DTW distance. More recently, it has been shown in [11] that bottleneck features (BNF) produce better performance than the phone posterior features. However, from our preliminary experiments, BNF features trained from relatively clean speech data do not perform well on noisy and reverberant speech data.

To perform approximate search, several modified DTW methods [12] are proposed to handle the mismatch of prefix and suffix in the query and the instances in the speech data. A new DTW method is also proposed for word-reordering queries based on heuristics. In [6], a partial matching based symbolic search (SS) system is proposed to deal with approximate match. Subsequences of phone sequences of the queries are also used to search for the speech database. However, symbolic systems generally performs worse than DTW based systems.

In this paper, we investigate two methods to improve the performance of DTW based template matching. To improve the robustness of the BNF, we train the BNF extractor network by using a large amount of diverse training data that are generated by simulation. This is a data augmentation approach. The distorted speech data are generated by convolving the clean speech data with simulated room impulse responses (RIR) and then corrupted by various additive noises. To perform more flexible and meaningful approximate matching, we further improve the DTW partial matching method that we proposed in [13]. We use phone time boundaries to produce partial segments of the queries for search. We also study the interactions of data augmentation and partial DTW matching in the QUESST 2015 task where the speech data are corrupted by both reverberation and noises.

2. QUESST TASK DESCRIPTION

2.1. Query type definition

The QUESST (2014/2015) task is to find out all utterances that contain exact or approximate matches of a given query audio. Previously in QUESST 2014, three types of query are defined [6, 14]. Type 1 query is a kind of exact match query that has one or multiple words uttered in the audio. For instance, a "researcher" query exactly occurs in the utterance "I want to be a researcher". Type 2 query is

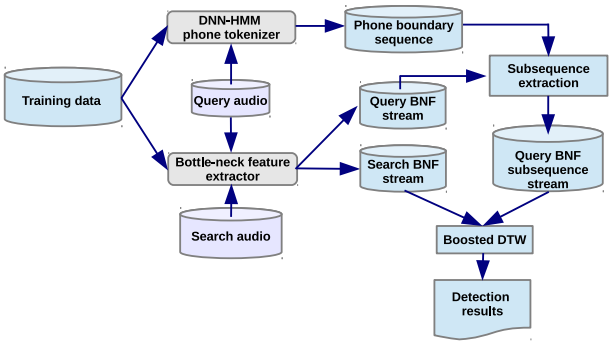

Fig. 1. *Illustration of the proposed DTW approach for approximate matching QbE-STD*

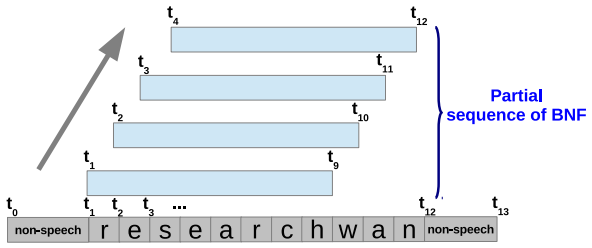

Fig. 2. *Illustration of the partial sequence based DTW search approach.*

an approximate matching query, where word morphological variations between the query and the search utterance are allowed. For instance, a "research" is allowed to be matched with "researching" or "researcher" etc. Type 3 query covers type 2 query, and allows word-reordering. For example, "a researcher is careful" might be allowed to match with "a careful researcher".

In QUESST 2015, the 3 categories of query are defined differently. Type 1 query allows word morphological variations as well as exact match. Type 2 query covers the type 1 query, plus word-reordering and filler words are allowed. Type 3 query contains all the features of the type 2 query, and the speech of the type 3 query is spontaneous, which implies silence, human hesitations and coarticulations may be contained. Another change of the QUESST 2015 is that all the spoken queries are part of a longer utterance, and hence start-end boundaries are provided to the participants [1].

2.2. Database and evaluation metrics

The whole data set for the QUESST task can be divided into 3 parts. They are `dev` and `eval` sets of queries, and a search database. In this work, all our experiments are conducted on the `dev` data. The data is from several languages and the language identity of utterances are not known. Overall, there are 555 queries with 2.18 seconds duration on average in the QUESST 2014 `dev` query set, and 445 queries with 1.39 seconds on average for the QUESST 2015 `dev` query set. For the search data, there are 23.1 and 19.4 hours for the QUEST 2014 and QUEST 2015 respectively.

Two metrics we used are normalized cross entropy (Cnxe) [15] and Term Weighted Value (TWV) [16] respectively. Specifically, they are Minimum Cnxe (minCnxe) and Maximum TWV (MTWV), both are attainable with an optimal global threshold [14].

3. PROPOSED DTW APPROACH

3.1. System overview

Figure 1 illustrates our overall system diagram. We first use the same training data set to train a BNF feature extractor and a DNN-HMM based phone tokenizer separately. The former is used to generate a compact and robust representation for the incoming query and search data, while the phone tokenizer is employed to produce phone sequences with time boundary information. The phone boundary information will be used in the DTW method to exclude silence frames in the beginning and ending of the query audio and also extracting meaningful partial feature sequences.

3.2. DTW partial matching with phone boundary

For each audio query, its N-best phone sequences are generated by using the phone tokenizer. From the N-best sequences, a single sequence is chosen that has the largest number of content phones. A content phone is the phone other than silence phone and phones representing non-speech sound in the audio, e.g. noise. The selected phone sequence will be used to provide information for partial feature sequence extraction as illustrated in Figure 2.

For phones at the beginning and ending of a query audio, if it is a non-content phone such as silence or non-speech phone, it will be removed as it is not useful in the search of queries. If a non-content phone occurs between content phones, it can be retained as we found that it makes no difference whether it is removed or not. The removal of non-speech frames using phone boundary information has similar effects as using a SAD.

The subsequence of feature representation of a query is extracted with the help of the phone boundary information. For subsequence length L, we extract all the feature subsequences of the query which spans L number of consecutive phones as illustrated in Figure 2. For example, if there are 11 phones in the query after removing preceding and ending non-speech phones, and $L = 8$, there will be 4 subsequences to be extracted. These subsequences of BNF feature vectors will be used as queries for DTW search. The use of subsequences for searching is similar to the query expansion approach [17] widely used in text-based information retrieval.

3.3. Data augmentation

In practical applications, the speech signal may be corrupted by additive background noise and room reverberation. The BNF extractor and phone tokenizers trained from clean data will be less effective on such distorted data. In order to improve the robustness of these modules, we create more training data by adding noise and reverberation to the clean training data. This is called data augmentation and widely used in the acoustic model training of ASR systems.

We first convolve the clean utterances with room impulse responses (RIR) simulated by the image method [18]. There are 3 room sizes considered, including small, medium, and large rooms. Two speaker-to-microphone distances are used, near (1.5m) and far (3m). The reverberation time T60 is randomly selected from 0.1s to 1.0s. After a clean utterance is convolved with a RIR, additive noises are added with signal to noise ratio (SNR) randomly selected from 0dB to 50dB. Totally 15 types of additive noises samples from

[1] Many thanks to the reviewer's supplementary comments.

NOISEX-92 database [19] are used in corrupting the data. For each clean utterance, the RIR and noise samples are randomly selected.

4. EXPERIMENTAL SETUP

We use the `Kaldi` toolkit [20] to build the experimental platform. For the DTW method, BNF features are used. In practice, we trained a stacked BNF extractor (two-stacked NN) as advocated in [21], yielding 30 dimensional BNF. The training features are filter bank plus pitch features [22], with 25 dimensions each frame, of which 22 is for the filter bank features, and 3 is for the pitch features. The input features to the first neural network (NN) are 11 (5-1-5) frames concatenated. The first NN topology is 1500-1500-80-1500-senones (context-dependent tied-states), while the second NN topology is 1500-1500-30-1500-senones. Different from [21], we used the senones, instead of mono-phone states, to train the BNF DNN. Once trained, the two NNs are stacked to generate 30 dimensional BNF. To run the approximate matching DTW method, we extract the phone boundary sequences from the 3 N-best phone sequences, and each subsequence contains 6 phones for the partial matching in all cases.

For the phone tokenizer, a 6 hidden-layer DNN is trained, with 2048 neurons for each hidden-layer, also taking the filter bank features as input, but 21(10-1-10) frames concatenated. The DNN uses the same senones as used in the BNF extractor training. For comparison, we also perform partial matching SS proposed in [6]. To do SS search, we generate 2000 N-best phone sequences for each query, with each partial phone sequence containing 6 phones for the partial matching.

The training data comes from the SWBD English Conversational Telephony Speech (CTS) corpus as used in [23]. Since we also aim at better performance on the QUESST 2015 task in addition to the QUESST 2014 task, two data sets are employed separately. The first has 318-hour pure CTS data while the second has 194-hour mixed data after data augmentation. The second contains both 97-hour data randomly selected from the first set and the corresponding data that is noise contaminated. We do data augmentation following the procedure as described in Section 3.3.

5. RESULTS

We report the overall experimental results in this section. In addition to the results of the proposed DTW method, we also report the results from the phone tokenizer based SS systems trained on the same data set. Our DTW method is dependent on the phone tokenizer that is also used in the SS system, allowing us to compare the performance of partial matching DTW and partial matching SS systems closely.

5.1. Results on the QUESST 2014 task

Table 1 reports the results of the SS system, of which the phone tokenizer is trained on the 318-hour CTS data. From Table 1, the partial matching of the SS search method is very effective. It consistently makes significant improvement on all 3 query types, particularly on Type_2 and Type_3 which belong to the approximated matching case. Take Type_3 query results for instance, the partial matching method attains 324% improvement for the MTWV result, compared with the full matching method. We note that the results in Table 1 are significantly better than our previous results in [6]. This is due to we employed different features and longer feature window as DNN input, as well as larger N-best sequence set here.

Table 1. Symbolic system performance on the QUESST 2014 task. The "Full" stands for using the entire phone sequence to match; the "Partial" refers to the partial phone sequence matching.

Metric	Method	Type_1	Type_2	Type_3	Overall
MTWV	Full	0.1122	0.0742	0.0280	0.0822
	Partial	0.1978	0.1465	0.1187	0.1693
minCxne	Full	0.9211	0.9657	0.9626	0.9467
	Partial	0.8697	0.9276	0.8770	0.8906

Table 2. DTW system results on the QUESST 2014 task. The "Naive" stands for the method using the entire feature sequence matching, without SAD; the "Bounded" is similar to the "Naive" but it uses the endpoint phone boundaries from the phone tokenizer. The "Partial" stands for that the phone boundaries are employed to do the DTW based partial matching.

Metric	Method	Type_1	Type_2	Type_3	Overall
MTWV	Naive	0.0620	0.0582	0.0203	0.0578
	Bounded	0.3441	0.1862	0.0509	0.2211
	Partial	0.4542	0.2859	0.2040	0.3392
minCxne	Naive	0.8955	0.9168	0.9125	0.09080
	Bounded	0.7062	0.8631	0.8562	0.8000
	Partial	0.6146	0.8138	0.7761	0.7257

Table 2 presents the DTW system results with different setups. Several points in Table 2 are worth a notice. First, the DTW based method is highly dependent on the SAD. It yields much worse results without SAD in the "Naive" setup. Secondly, endpoint phone boundary detection is very effective; the "Bounded" method produces much improved results compared with the "Naive" method. Thirdly, the DTW based method does not do well in inexact/approximated query search, particularly in Type_3 queries, as evidenced with the contrast between Table 1 and Table 2. Finally, the phone time boundary boosted DTW method makes significant improvement, attributing to the partial matching method employed[2].

5.2. Results on the QUESST 2105 task

We are now applying the systems in Section 5.1 to the QUESST 2015 task, of which both the query and search data have been corrupted with various kinds of noise by the organizer. Based on this reason, we also report the results from our boosted systems after data augmentation applied.

5.2.1. No data augmentation used

Table 3 shows the results of the Section 5.1 SS system applied to the QUESST 2015 task. From Table 3, the SS system of the phone tokenizer trained with the "clean" CTS data gives much worse results compared with those in Table 1 of the QUESST 2014 task. This is because the data in the QUESST 2015 task are quite noisy. Various additive noise as well as reverberant noise are added to the data from our perception.

As for the DTW method, the first group (MTWV/minCnxe) of Table 5 reports the results using the Section 5.1 DTW system on

[2]Technically, we should compare the results between the "Bounded" and the SAD method in Table 2. However, we have no SAD module available, and we believe the results of the "Bounded" method in Table 2 are comparable with what have been reported in [11, 6], where the SAD has been applied.

Table 3. Results of the SS system in Section 5.1 on the QUESST 2015 task

Metric	Method	$Type_1$	$Type_2$	$Type_3$	Overall
MTWV	Full	0.0204	0.0020	0.0101	0.0090
	Partial	0.0430	0.0134	0.0214	0.0245
minCxne	Full	0.9650	0.9855	0.9857	0.9826
	Partial	0.9616	0.9755	0.9855	0.9779

Table 4. Results of the updated SS system on the QUESST 2015 task after the tokenizer trained with the augmented data

Metric	Method	$Type_1$	$Type_2$	$Type_3$	Overall
MTWV	Full	0.0694	0.0220	0.0074	0.0312
	Partial	0.1077	0.0295	0.0306	0.0550
minCxne	Full	0.9371	0.9712	0.9853	0.9709
	Partial	0.9025	0.9248	0.9485	0.9311

the QUESST 2015 task. We can see despite that the DTW method achieves better performance than the SS method, its results are still not good enough as compared to the QUESST 2014 results, indicating the challenge of the data in the QUESST 2015. Besides, we can also see that the "Bounded" method does not make improvement over the "Naive" method, which means the boundary information provided by the "clean" data trained tokenizer is inaccurate in this scenario.

5.2.2. Using data augmentation

After the tokenizer being trained with the augmented data, Table 4 reports the results of the updated SS system on the QUESST 2015 task. Compared with what were revealed in Table 3, the results in Table 4 are significantly improved in all circumstances. However, compared with the results of the BNF based DTW system as shown in the first group of Table 5 (no data augmentation), the updated SS system is still much worse. This implies the SS system transcribes the query and corresponding search data inconsistently, due to the data mismatch issue. That is, different noise is added between the query and search data, and the SS system is more vulnerable to this data mismatched condition.

The fourth group of Table 5 reports the corresponding DTW results after data augmentations which are applied to both BNF extractor and phone tokenizer training respectively. Comparing the first group of Table 5 where no data augmentation is employed, we can observe the data augmentation is very effective for the DTW method, which yields remarkable improvement in all cases. Additionally, several conclusions can be drawn. First, the updated phone tokenizer performs well in endpoint speech detection, which helps the DTW system make remarkable improvement, as is shown by comparing the "Bounded" and the "Naive" results. Secondly, phone time boundary information is still useful for the DTW based partial matching. This is shown in the "Partial" rows in the group.

Observing the results in the fourth group of Table 5, we know the data augmentation method is effective for the DTW method, in terms of providing augmented features (augmented BNF) and augmented boundaries (from the augmented tokenizer). We are curious about which one of these two factors is more important. Comparing the results of the second and third pair groups in Table 5, we found the augmented boundaries are more important. This indicates the accuracy of the phone time boundary is critical to the DTW method.

Table 5. Results of the different DTW methods on the QUESST 2015 task with various settings: BNF with/without data augmentation, and we use "A" to represent them; boundary generated by the tokenizer with/without data augmentation, and we use "B" to stand for them. "✓" stands for data augmentation applied, while "✗" stands for no data augmentation applied. Besides, both MTWV and minCnxe results are divided into four pair groups.

Method	A	B	$Type_1$	$Type_2$	$Type_3$	Overall
MTWV						
Naive	✗	-	0.1947	0.0684	0.1064	0.1231
Bounded	✗	✗	0.2033	0.0586	0.0915	0.1211
Partial	✗	✗	0.2051	0.0640	0.1110	0.1290
Bounded	✓	✗	0.2341	0.0817	0.0966	0.1403
Partial	✓	✗	0.2401	0.0852	0.1225	0.1520
Bounded	✗	✓	0.2854	0.1081	0.1111	0.1708
Partial	✗	✓	0.2762	0.1305	0.1303	0.1814
Naive	✓	-	0.2560	0.0886	0.1179	0.1534
Bounded	✓	✓	0.3366	0.1504	0.1200	0.2048
Partial	✓	✓	0.3295	0.1925	0.1493	0.2245
minCnxe						
Naive	✗	-	0.8094	0.8951	0.8665	0.8668
Bounded	✗	✗	0.8303	0.9100	0.9127	0.8931
Partial	✗	✗	0.8335	0.9040	0.9100	0.8906
Bounded	✓	✗	0.7918	0.8887	0.9083	0.8749
Partial	✓	✗	0.7937	0.8816	0.9053	0.8715
Bounded	✗	✓	0.7642	0.8476	0.8703	0.8389
Partial	✗	✓	0.7674	0.8244	0.8620	0.8280
Naive	✓	-	0.7641	0.8656	0.8420	0.8366
Bounded	✓	✓	0.7000	0.8092	0.8470	0.8020
Partial	✓	✓	0.7014	0.7806	0.8378	0.7875

Finally, we note that our "NNI" team[3] has achieved the best results in this year contest, and the final results are 0.2864/0.7572 (MTWV/minCnxe) on the `dev` data by fusing 50+ component systems. However, we are able to achieve 0.2314/0.7834 (MTWV/minCnxe) results by only fusing the partial SS (from Table 4) and the partial DTW (from the fourth group of Table 5) systems [24]. This demonstrates the effectiveness of the proposed method.

6. CONCLUSION

We proposed an approximate template matching method based on the DTW for the QbE-STD task. Phone time boundary information provided by a phone tokenizer is used in the proposed DTW system. We also used data augmentation to train robust BNF extractor and phone tokenzier to improve the robustness of the DTW system. Experimental results on QUESST 2014/2015 benchmarking tasks show the effectiveness of the proposed methods. We also found that in noisy and reverberant cases, accurate phone time boundary and robust feature representation are both critical for achieving high QbE-STD performance.

[3]The whole team contains three sides that are NWPU, NTU, and I^2R respectively.

7. REFERENCES

[1] Xavier Anguera, Luis Javier Rodriguez-Fuentes, Igor Szöke, Andi Buzo, and Florian Metze, "Query-by-example search on speech at mediaeval 2014," in *Working Notes Proc. of the Mediaeval 2014 Workshop*, 2014.

[2] Igor Szöke, Luis Javier Rodriguez-Fuentes, Andi Buzo, Xavier Anguera, Florian Metze, Jorge Proenca, Martin Lojka, and Xiong Xiao, "Query by example search on speech at MediaEval 2015," in *Working notes Proc. of the Mediaeval 2015 Workshop*, 2015.

[3] Meinard Müller, *Dynamic Time Warping, chapter 4, Information Retrieval for Music and Motion*, pp. 69–84, Springer-Verlag, Berlin, Germany, 2007.

[4] Javier Tejedor, Michal Fapšo Igor Szöke Jan Honza Černocký, and František Grézl, "Comparison of methods for language-dependent and language-independent query-by-example spoken term detection," *ACM Transactions on Information Systems*, 2012.

[5] Peng Yang, Haihua Xu, Xiong Xiao, Lei Xie, Cheung-Chi Leung, Hongjie Chen, Jia Yu, Hang Lv, Lei Wang, Su Jun Leow, Bin Ma, Eng Siong Chng, and Haizhou Li, "The NNI Query-by-Example system for MediaEval 2014," in *Working notes Proc. of the Mediaeval 2014 Workshop*, 2014.

[6] Haihua Xu, Peng Yang, Xiong Xiao, Lei Xie, Cheung-Chi Leung, Hongjie Chen, Jia Yu, Hang Lv, Lei Wang, Su Jun Leow, Bin Ma, Eng Siong Chng, and Haigzhou Li, "Language independent query-by-example spoken term detection using N-best phone sequences and partial matching," in *Proc. of ICASSP 2015*. IEEE, 2015.

[7] Timothy J. Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Automatic Speech Recognition & Understanding (ASRU), 2009*. IEEE, 2009.

[8] Luis J. Rodriguez-Fuentes, Amparo Varona, Mikel Penagarikano, Germán Bordel, and Mireia Diez, "High-performance query-by-example spoken term detection on the SWS 2013 evaluation," in *Proc. of ICASSP 2015*. IEEE, 2014.

[9] Peng Yang, Cheung Chi Leung, Lei Xie, Bin Ma, and Haizhou Li, "Intinsic spectral analysis based on temporal context features for query-by-example spoken term detection," in *Proc. of INTERSPEECH 2014*. ISCA, 2014.

[10] Haipeng Wang, Tan Lee, Cheung-Chi Leung, Bin Ma, and Haizhou Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," in *Proc. of ICASSP 2013*. IEEE, 2013.

[11] Igor Szöke, Miroslav Skácel, Lukáš Burget, and Jan Honza Cernocký, "Coping with channel mismatch in query-by-example–BUT QUESST 2014," in *Proc. of ICASSP 2015*. IEEE, 2015.

[12] Jorge Proença, Arlindo Veiga, and Fernando Perdigão, "The SPL-IT query by example search on speech system for mediaeval 2014," in *Working notes Proc. of the Mediaeval 2014 Workshop*, 2014.

[13] Jingyong Hou, Lei Xie, Peng Yang, Xiong Xiao, Cheung-Chi Leung, Haihua Xu, Lei Wang, Hang Lv, Bin Ma, Eng Siong Chng, and Haizhou Li, "Spoken Term Detection Technology Based on DTW," in *NCMMSC*, 2015.

[14] Xavier Anguera, Luis-J. Rodriguez-Fuentes, Andi Buzo, Florian Metze, Igor Szöke, and Mikel Penagarikano, "QUESST2014: evaluating query-by-example speech search in a zero-resource setting with real-life queries," in *Proc. of ICASSP 2015*. IEEE, 2015.

[15] Luis J. Rodriguez-Fuentes and Mikel Penagarikano, "MediaEval 2013 Spoken Web Search Task: System Performance Measures," in *Technical Report TR-2013-1*, http://gtts.ehu.es/gtts/NT/fulltext/rodriguezmediaeval13.pdf, 2013.

[16] J.G. Fiscus, J. Ajot, J. S Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *SIGIR*. ACM, 2007.

[17] Jinxi Xu and W Bruce Croft, "Query expansion using local and global document analysis," in *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2012.

[18] Jont B Allen and David A Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[19] A.P. Varga, H.J.M. Steeneken, M. Tomlinson, and D. Jones, "The NOISEX-92 study on the effect of additive noise on automatic speech recognition," in *Technical Report*. DRA Speech Research Unit, 1992.

[20] D. Povey, A. Ghoshal, G.Boulianne, L. Burget, O.Glembek, N. Goel, M. Hannermann, P. Motlíček, Y. Qian, P. Schwartz, J. Silov ský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. of ASRU, 2011*. IEEE, 2011.

[21] F. Greźl and M. Karafiát, "Semi-supervisied bootstrapping approach for neural network feature extractor training," in *Proc. of ASRU, 2013*. IEEE, 2013.

[22] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Jrmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recogniton," in *Proc. of ICASSP, 2014*. IEEE, 2014.

[23] Karel Veselý, Arnab Ghoshal, Lukáš Burget, and Daniel Povey, "Sequence-discriminative training of deep neural networks," in *Proc. of INTERSPEECH, 2013*. ISCA, 2013.

[24] Hou et al, "The NNI Query-by-Example System for MediaEval 2015," in *Working Notes Proc. of the Mediaeval 2015 Workshop*, 2015.