LANGUAGE INDEPENDENT QUERY-BY-EXAMPLE SPOKEN TERM DETECTION USING N-BEST PHONE SEQUENCES AND PARTIAL MATCHING

Haihua Xu[1], *Peng Yang*[2], *Xiong Xiao*[1], *Lei Xie*[2], *Cheung-Chi Leung*[4], *Hongjie Chen*[2], *Jia Yu*[2], *Hang Lv*[2], *Lei Wang*[4], *Su Jun Leow*[1], *Bin Ma*[4], *Eng Siong Chng*[1,3], *Haizhou Li*[1,3,4]

[1]Temasek Lab@NTU, Nanyang Technological University, Singapore
[2]School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[3]School of Computer Engineering, Nanyang Technological University, Singapore
[4]Institute for Infocomm Research, A*STAR, Singapore

ABSTRACT

In this paper, we propose a partial sequence matching based symbolic search (SS) method for the task of language independent query-by-example spoken term detection. One main drawback of conventional SS approach is the high miss rate for long queries. This is due to high variations in symbol representation of query and search audios, especially in language independent scenario. The successful matching of a query with its instances in search audio becomes exponentially more difficult as the query grows longer. To reduce miss rate, we propose a partial matching strategy, in which all partial phone sequences of a query are used to search for query instances. The partial matching is also suitable for real life applications where exact match is usually not necessary and word prefix, suffix, and order should not affect the search result. When applied to the QUESST 2014 task, results show the partial matching of phone sequences is able to reduce miss rate of long queries significantly compared with conventional full matching method. In addition, for the most challenging inexact matching queries (type 3), it also shows clear advantage over DTW-based methods.

Index Terms— spoken term detection, keyword search, query-by-example, phone tokenizer, partial matching.

1. INTRODUCTION

Spoken term detection (STD) is the task of finding all occurrences of a query in an audio database. There are two types of STD tasks depending on whether the query is text or audio. The text query STD task is also called keyword search (KWS) [1, 2] and is the focus of ongoing IARPA Babel program [3–6]. In KWS, large vocabulary continuous speech recognition (LVCSR) systems are usually used to convert speech into word or subword lattices and then indexing is performed to achieve fast search. If the query is audio, the problem is called query-by-example STD (QbE-STD). From 2011 to 2014, there have been 4 QbE-STD evaluations [7], which are part of the MediaEval Benchmarking Initiative for Multimedia Evaluation[1]. As QbE-STD involves multilingual search with little or no resources, LVCSR techniques cannot be applied. In this paper, we will focus on the QbE-STD task, especially the Query-by-Example Search on Speech task (QUESST) within the Mediaeval 2014 evaluation [7].

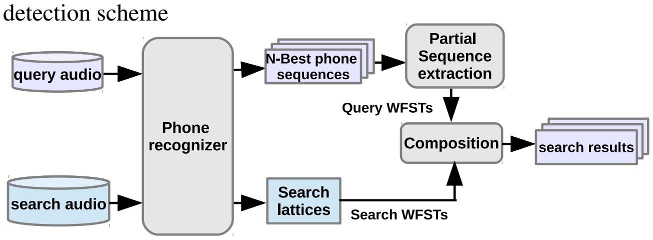
Several approaches have been proposed to solve the QbE-STD problem [8]. The most popular approach is based on pattern matching and usually implemented by derivatives of dynamic time warping (DTW) [9–14]. To deal with feature variation, various robust features are employed such as phone posteriorgram [9, 10, 13, 15, 16]. Besides, as DTW is computationally expensive, speedup methods have also been proposed [11, 17–21]. Another approach for QbE-STD task is the acoustic keyword spotting (AKWS) method [8], where query phone models and background models are employed in decoding. The third approach, called symbolic search (SS) in this paper, first converts query and search audios into symbolic representations such as phone sequence/lattice and then performs symbolic matching. Weighted finite state transducer (WFST) is used for indexing the phone lattices of audio database and search is performed by composing the query and audio WFSTs [22]. Although the SS approach performs the worst among the 3 approaches in [8], it allows indexing of the search audio database and offers a solution for fast search.

The spoken web search (SWS) task since 2011 has become a major benchmarking task for QbE-STD techniques. In the first 3 SWS tasks (2011-2013), the objective was to find exact match of the query in the audio database. DTW-based techniques comprised the top performance systems [23]. In 2014, the SWS task was renamed to QUESST and the task is redefined towards practical applications of QbE-STD. There are 3 types of queries in QUESST 2014. Type 1 query is a single word or multiple word sequence and requires exact match, similar to previous SWS tasks. Type 2 query allows for small difference between the query and its instances at the beginning and ending of the query due to prefix, suffix, or inflectional variations. Type 3 query also allows for changed word order in the query and its instances. In addition, short “filler” may exist between words in the query instances. To perform well in type 2 and 3 queries, the partial matching technique is needed.

In this paper, we improve the SS approach by using partial matching for language independent QbE-STD task. Motivated by the text-based STD/KWS tasks [4], we adopt the WFST-based KWS system for QbE-STD task, similar to the work in [8]. However, our preliminary study shows that the conventional SS approach suffers from high miss rate problem, especially for long queries. This is due to the inconsistency between the symbolic representation of the query and search audio. The exact matching of query and search audio becomes exponentially more difficult as query becomes longer. To address this problem, we propose to use partial phone sequence matching, where all partial phone sequences longer than a predefined length are used for searching. Such an approach is also useful for inexact matching task. For example, in practical use, the prefix/suffix or the word order in the query usually should not affect the

[1] http://www.multimediaeval.org

Fig. 1. Finite state transducer based query-by-example spoken term detection scheme



user's intention significantly. Partial matching provides a feasible way to find query instances in such scenarios.

The rest of the paper is organized as follows. The QUESST 2014 task is first described in section 2. Then the analysis of symbolic search on the QUESST task is presented and the partial sequence search is proposed in section 3. Experimental results are presented in section 4 and finally, we conclude in section 5.

2. QUESST 2014 TASK DESCRIPTION

The QUESST 2014 task is to find out all utterances that contain exact or approximate matches of the 3 types of query audio – as described in the introduction –from the search audio database. Three data sets are provided to the participants of the task, i.e. the 23-hour search database (12,492 sentences), 560 development audio queries and 555 evaluation audio queries. Both the search and query audios contain 6 languages, including Slovak, Romanian, Albanian, Czech, Basque, and non-native English. The language identity of each audio file is not known during the evaluation.

Two evaluation metrics are employed to measure the performance of systems. The first metric is actual/minimum normalized cross entropy cost (Cnxe, minCnxe) [24, 25], and the second metric is actual/maximum term weighted value (ATWV/MTWV) [2]. Though defined differently, these two metrics are highly correlated. For more details of the QUESST 2014 task, please refer to [25].

3. SYMBOLIC SEARCH WITH PARTIAL MATCHING

The proposed symbolic search system is illustrated in Fig. 1. The query and search audio datasets are both recognized into phone sequences using the same phone recognizer. For search audio, phone lattice is converted into a timed factor transducer [22]. For query audio, we use N-best phone sequence for easy manipulation. The partial phone sequences of each query are extracted and converted to WFST format. The search is performed by the composition of query and search audio WFSTs. In the following subsections, we will describe each step in detail.

3.1. Phone Recognizers Used for Tokenization

Tokenization is the process of converting the audio signal into discrete symbols to facilitate search. As we assume that there is not enough resources, or no resources at all, to build LVCSR or phone recognizer from the search audio itself, it is necessary to borrow language independent tokenizers, such as phone recognizers, trained from known languages and resources. In this paper, we used two sources of phone recognizers for tokenization as shown in Table 1. The 3 Brno University of Technology (BUT) phone recognizers trained from Czech, Hungarian, and Russian are used [26]. In addition, we trained three phone recognizers from Switchboard (SWB), one using stacked bottleneck features (SBN) and Gaussian mixture models (GMM), and two (monophone and triphone) using deep neural network (DNN) acoustic models. We also trained two phone recognizers from a Malay broadcast news corpus [27, 28], one of them using SBN and GMM model and the other using DNN acoustic models.

Table 1. List of phone tokenizers. CTS and BN stand for conversational telephone speech and broadcast news, respectively.

Recognizer	Language	#phones	Training Data	Type
BUT CZ	Czech	45	12 hours	CTS
BUT HU	Hungarian	61	10 hours	CTS
BUT RU	Russian	52	18 hours	CTS
SWB-bn	English	42	100 hours	CTS
SWB-dnn	English	42	100 hours	CTS
SWB-mono	English	42	100 hours	CTS
MY-bn	Malay	33	78 hours	BN
MY-dnn	Malay	33	78 hours	BN

Some of the phone recognizers will also be used for DTW-based systems in the experimental section. Note that the use of phone recognizers in SS and DTW-based systems is different. In the SS approach, phone recognizers are used to generate discrete phone sequences/lattices, while in the DTW approach, phone posteriorgrams are generated and used as features.

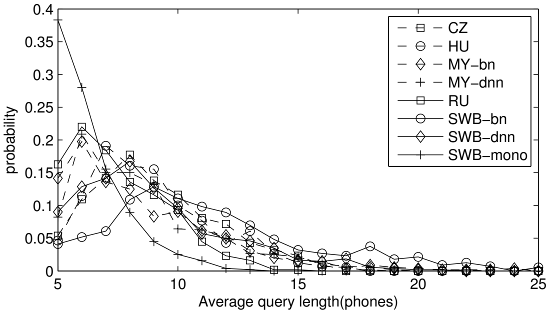
3.2. WFST-based Symbolic Search

In text-based STD tasks, WFST is widely used for indexing and search [22, 29]. Word or phone lattices of the search audio are converted to a special type of WFST which allows the query to match from any starting point to any ending point of the search audio. The query WFST can be a simple sequence of phones, or a lattice. A highly efficient operation called composition will find all the common paths of the query and search WFSTs.

Approximate matches can be achieved by using phone lattice [8] or N-best phone hypotheses in both query and search WFSTs. In this work, we use phone lattice for search audio and N-best hypotheses for query audio for easy manipulation of the query phone sequences and efficient searching. Furthermore, it is straightforward to remove short hypotheses and extract partial phone sequences from N-best hypotheses, but it is not as easy to do so on lattices. In addition, phone lattices of queries may contain a huge number of paths, so it is not easy to control search complexity [29].

We use a heuristic rule to decide the number of hypotheses used for each query: $N_i = 2^{L_i}$ where N_i is the number of hypotheses used for query i, L_i is the average number of phones in the phone hypotheses of query i. The base 2 is decided empirically from data and works reasonably well. The purpose of such rule is to use more hypotheses for longer queries to sufficiently represent the variations of hypotheses. Another heuristic is to remove all hypotheses that have less than 5 phones to avoid too many false alarms. After these operations, the distribution of average query length in terms of phones are shown in Fig. 2. It can be observed that while the median of query length is around 10 phones, the distributions have long tails on the right side. The phone sequence of queries can be as long as 25 phones, which will be very difficult to match in practice. We will discuss solutions to this problem in next section.

Fig. 2. Distribution of average query length (in terms of phones) in N-best hypotheses ($N = 1000$) using various tokenizers.



3.3. Partial Phone Sequence Matching

One major challenge in applying symbolic search approach to a language independent QbE-STD task is that the phone sequence of a query is not stable and varies from one instance to another. There are several reasons for such instability. For example, the phone recognizer may be trained with language different from the search audio and hence not sharp enough to produce consistent phone sequences. Other reasons may be due to different recording channels, speaking styles, background noises, etc. among the 3 types of data involved: 1) the phone recognizer's training data; 2) query audio; 3) search audio database. As a result, the phone representations of query and search audio data are highly variable, and this is challenging to the WFST-based symbolic search approach which relies on exact sequence matching.

The instable phone representation problem inherent with SS can be alleviated by using N-best hypotheses to represent query audio and phone lattice to represent search audio as described in section 3.2. However, preliminary experimental results on the QUESST 2014 data show that such a solution is far from enough to obtain reasonable results as illustrated in Fig. 3. From the figure, it can be seen that the missing probability of full phone sequence match, pMiss(Full), is approaching 100% as the query length becomes longer than 11 phones. At the same time, the false alarm rate pFA(Full) and ATWV(Full) are both approaching 0 for long queries. This observation indicates that for long queries, the system simply returns nothing, since it is exponentially more difficult to match the full phone sequence of queries with search audio as the query length increases. Hence, to improve the performance of the SS approach, the first priority is to reduce the miss rate.

To make it possible to detect long queries, we take a simple but effective approach. That is, instead of using the full phone sequence of a query to match search lattices, we only use a partial segment of the query phone sequence. The rationale is that we first let the system return some matches by using shorter phone sequence and then filter out invalid matches. In implementation, all partial phone sequences of length M of a query are used to search for its instances . For example, if a query has a phone sequence of length $L = 8$ phones and $M = 6$, then there will be $L - M + 1 = 3$ partial phone sequences, i.e. the 1st to the 6th, the 2nd to the 7th, and the 3rd to the 8th phones, respectively. This method is applied to all hypotheses of a query longer than a specified threshold ($L \geq 7$). As most of the partial phone sequences will be identical, the unique number of partial phone sequences for a query will not increase significantly with length. Usually, several hundreds of partial sequences are used to represent a long query. From Fig. 3, with partial sequence matching ($M = 6$), the miss rate pMiss(Partial) is significantly reduced for long queries. On the other hand, the false alarm rate pFA(Partial) is also increased. However, the ATWV (Partial) is still boosted significantly for queries longer than 7. For short queries $L \leq 6$, the ATWV is unaffected as the partial matching is not applied. The results in Fig. 3 show that partial matching is a practical solution to the high miss rate problem of the symbolic search.

Fig. 3. Comparison of pMiss, pFA, and ATWV with and without partial matching using SWB-dnn tokenizer.

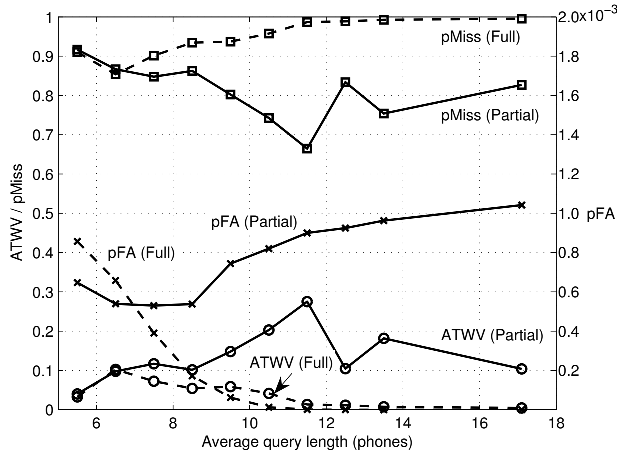
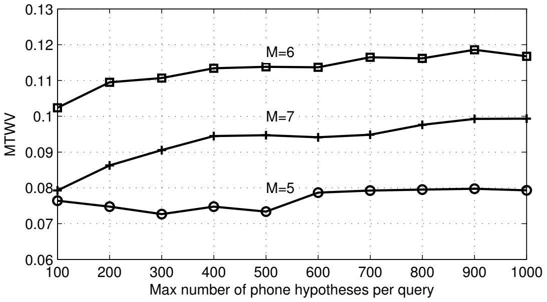


Fig. 4. MTWV obtained with partial matching SS approach and SWB-dnn tokenizer for different partial phone segment length M and maximum number of of phone hypotheses per query.



The partial sequence matching also allows the inexact search of type 2 and 3 queries in QUESST 2014 task. For example, if partial matching is used, the mismatch in prefix or suffix of a word may not affect the search results significantly. Even if the word order of the query and its instances are different, partial matching may still be able to find the instances as far as part of the query is matched.

We note that although partial matching of phone sequences alleviates the high miss rate problem of symbolic search and provides a solution to complex queries, it also introduces extra false alarms. Therefore, it is necessary to reduce false alarm rate in future research.

4. EXPERIMENTS

The proposed partial matching SS approach is implemented on the KALDI [30] platform. We built one SS system from each of the 7

Table 2. Comparison of MTWV on QUESST 2014 eval data obtained by full and partial matching SS systems, and subsequence DTW-based systems. CZ and HU refer to BUT Czech and Hungarian phone recognizers, respectively.

Systems	Type-1	Type-2	Type-3	Overall
SS-CZ full	0.0392	0.0084	0.0031	0.0141
SS-CZ partial	0.2057	0.0924	**0.0527**	0.1069
DTW-CZ	**0.3510**	**0.1369**	0.0301	**0.1839**
SS-HU full	0.0188	0.0027	0.0062	0.0079
SS-HU partial	0.1537	0.0509	**0.0415**	0.0801
DTW-HU	**0.2852**	**0.1111**	0.0255	**0.1514**

Table 3. MTWV and minCnxe obtained by all partial matching SS systems and their fusion on eval data.

Systems	Type-1	Type-2	Type-3	Overall
MTWV				
SS-CZ	**0.2058**	0.0924	0.0528	0.1069
SS-HU	0.1538	0.0510	0.0416	0.0802
SS-MY-bn	0.0158	0.0116	0.0446	0.0222
SS-MY-dnn	0.1334	0.0902	0.1062	0.1040
SS-SWB-bn	0.1010	0.0681	0.0681	0.0777
SS-SWB-dnn	0.1888	**0.0975**	**0.1269**	**0.1373**
SS-SWB-mono	0.1093	0.0404	0.0285	0.0575
Fusion	**0.3603**	**0.2357**	**0.2237**	**0.2717**
minCnxe				
SS-CZ	0.8183	0.8778	0.9163	0.8700
SS-HU	0.8547	0.9037	0.9247	0.8949
SS-MY-bn	0.9550	0.9528	0.9463	0.9554
SS-MY-dnn	0.8718	0.9016	0.9204	0.8958
SS-SWB-bn	0.9252	0.9330	0.9438	0.9338
SS-SWB-dnn	0.8675	0.8994	0.9184	0.8965
SS-SWB-mono	0.9285	0.9559	0.9647	0.9507
Fusion	**0.6715**	**0.7338**	**0.7950**	**0.7293**

phone recognizers in Table 1 except for Russian. The scores of the SS systems are normalized by keyword specific threshold normalization [1]. We also built 9 subsequence DTW [19] based systems for comparison and fusion purposes. The DTW systems make use of the 5 tokenizers, including the 3 BUT phone recognizers, the SBN features of SWB-bn (not the posteriors), and a 1024-component Gaussian posteriors. We also implemented a different version of partial matching in 4 of the DTW systems, where a shifting fixed length window of feature vectors are extracted from the query and used for matching. For more details of the DTW systems, please refer to [31].

Fig. 4 shows the tuning of partial phone sequence length M. The results show that $M = 5$ is too short and generates too many false alarms, and $M = 7$ is too long and results in too many missing keywords. $M = 6$ achieves a good tradeoff of missing and false alarm errors and obtained the best MTWV. This is true for most of the tokenizers used in our study.

We compare SS systems using full or partial matching and the DTW systems using the same tokenizers in Table 2. Results show that the full matching systems (SS-CZ-full and SS-HU-full) return close to 0 MTWV, and partial matching systems (SS-CZ-partial and SS-HU-partial) increases the MTWV several folds. Compared to DTW systems, the partial matching SS systems perform worse for type 1 query (exact matching), but better for type 3 query where the word order may be different in query and its instances.

Table 4. Performance of partial matching SS and DTW on eval data. Results are separated by query types.

Methods	Cnxe	MinCnxe	ATWV	MTWV
Type 1 Queries				
DTW	0.5733	0.5971	0.4448	0.4465
Symbolic	0.6787	0.6715	0.3526	0.3603
Fusion	0.5248	0.5088	0.5115	0.5136
Type 2 Queries				
DTW	0.7300	0.7191	0.2306	0.2408
Symbolic	0.7405	0.7338	0.2294	0.2357
Fusion	0.6386	0.6290	0.3158	0.3324
Type 3 Queries				
DTW	0.8029	0.7925	0.1465	0.1673
Symbolic	0.8035	0.7950	0.2134	0.2237
Fusion	0.7210	0.7140	0.3061	0.3102
All Queries				
DTW	0.6925	0.6816	0.2918	0.2974
Symbolic	0.7322	0.7293	0.2696	0.2717
Fusion	0.6125	0.6062	0.3896	0.3952

The performance of all partial matching SS systems and their fusion is listed in Table 3. It can be observed that the performance varies significantly from tokenizer to tokenizer. The fusion of the systems using the FoCal [32] produces a big gain for all query types.

Finally, the fusion of 7 partial matching SS systems and 9 DTW systems are shown in Table 4. From the table, the DTW systems show advantage for type 1 exact match queries, while SS systems are stronger for ATWV and MTWV for type 3 complex queries. For type 2 query, the two types of systems perform almost equally. When all the 9 DTW and 7 SS systems are fused, significant improvement is observed in all query types. This result shows that the SS system is highly complementary to the DTW systems.

One advantage of the partial matching SS approach is its fast searching speed. For example, a typical SS system takes about 50 CPU hours to finish the searching of 555 eval queries on the 23 hours search audio, i.e. a searching speed factor (SSF) of 0.0012 [24].). For comparison, a 150D phone posterior features based DTW system uses about 360 CPU hours to do the same search (SSF=0.016). The SS system needs another 80 CPU hours to index the 23 hours of search audio (ISF=3.5), which is performed offline.

5. CONCLUSION

In this paper, we studied a partial matching based symbolic search approach for language independent QbE-STD task. To reduce the high miss rate due to exact matching nature of the WFST-based symbolic search, and to allow inexact matching searches, we proposed to use partial phone phone sequence matching. When evaluated on the QUESST 2014 task, the proposed partial matching SS method shows promising results, especially on the complex queries. In the future, we will focus on verification of the matching candidates to reduce false alarms. We will also apply the partial matching strategy to the text-based keyword search task.

6. REFERENCES

[1] D. R. H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colthurst, S. A Lowe, R. M Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proceedings of INTERSPEECH*, 2007.

[2] N. F. Chen and J. G. Fiscus, "Overview of the NIST open keyword search 2013 evaluation workshop," in *SLTC Newsletters*. IEEE, 2013.

[3] Intelligence Advanced Research Projects Activity (IARPA), "Babel program," in *http://www.iarpa.gov/index.php/research-programs/babel*.

[4] NIST, "OpenKWS keyword search evaluation plan," in *http://nist.gov/itl/iad/mig/upload/KWS14-evalplan-v11.pdf*.

[5] N. Chen et al., "Strategies for Vietnamese keyword search," in *ICASSP*, 2014.

[6] N. Chen et al., "Low-resource keyword search strategies for Tamil," in *ICASSP*, 2015.

[7] X. Anguera, L. J. Rodriguez-Fuentes, I. Szöke, A. Buzo, and F. Metze, "Query-by-example search on speech at Mediaeval 2014," in *Working Notes Proceedings of the Mediaeval 2014 Workshop*, Barcelona, Spain, Oct. 16-17.

[8] J. Tejedor, M. Fapšo I. Szöke J. H. Černocký, and F. Grézl, "Comparison of methods for language-dependent and language-independent query-by-example spoken term detection," *ACM Transactions on Information Systems*, 2012.

[9] V. Gupta, J. Ajmera, A. Kumar, and A. Verma, "A language independent approach to audio search," in *Proc. INTERSPEECH*, 2011.

[10] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Using parallel tokenizers with DTW matrix combination for low-resource spoken term detection," in *Proc. ICASSP*, 2013.

[11] X. Anguera, "Information retrieval-based dynamic time warping," in *Proc. INTERSPEECH*, 2013.

[12] 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.

[13] G. Mantena and K. Prahallad, "Use of articulatory bottle-neck features for query-by-example spoken term detection in low resource scenarios," in *Proc. ICASSP*, 2014.

[14] B. George and B. Yegnanarayana, "Unsupervised query-by-example spoken term detection using segment-based bag of acoustic words," in *Proc. ICASSP*, 2014.

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

[16] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 23, no. 2, pp. 264–277, 2015.

[17] Y. Zhang and J. R. Glass, "A piecewise aggregate approximation lower-bound estimate for posteriorgram-based dynamic time warping," in *Proc. INTERSPEECH*, 2011.

[18] Y. Zhang and J. R. Glass, "An inner-product lower-bound estimate for dynamic time warping," in *Proc. ICASSP*, 2011.

[19] X. Anguera and M. Ferrarons, "Memory efficient subsequence DTW for query-by-example spoken term detection," in *Proc. ICME*, 2013.

[20] G. Mantena and X. Anguera, "Speed improvements to information retrieval-based dynamic time warping using hierarchical k-means clustering," in *Proc. ICASSP*, 2014.

[21] P. Yang, L. Xie, Q. Luan, and W. Feng, "A tighter lower-bound estimate for dynamic time warping," in *Proc. ICASSP*, 2013.

[22] D. Can and M. Saraclar, "Lattice indexing for spoken term detection," *Audio, Speech ,and Language Processing, IEEE Transactions on*, pp. 2338–2347, 2011.

[23] X. Anguera, L. J. Rodriguez-Fuentes, I. Szöke, A. Buzo, F. Metze, and M. Penagarikano, "Query-by-example spoken term detection on multilingual unconstrained speech," in *Proc. INTERSPEECH*, 2014.

[24] L. J. Rodriguez-Fuentes and M. 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.

[25] X. Anguera, L. Javier, I. Szöke, A. Buzo, and F. Metze, "Query by example search on speech at mediaeval 2014," in *Proc.* MediaEval 2014 Workshop, 2014.

[26] Schwarz et al., "Hierarchical structures of neural networks for phoneme recognition," in *Proc. ICASSP*, 2006.

[27] T. Chong, X. Xiao, H. Xu, T. Tan, C. Pham, D. Ly, E. Chng, and H. Li, "The development and analysis of a Malay broadcast news corpus," in *Proc. O-COCOSDA*, 2013.

[28] H. Xu, V. T. Pham, E. S. Chng, and H. Li, "Towards better keyword search performance on Malay broadcast news data," in *Proc. APSIPA*, 2014.

[29] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. of ASRU*, 2013.

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

[31] P. Yang et al., "The NNI Query-by-Example system for MediaEval 2014," in *Working Notes Proceedings of the Mediaeval 2014 Workshop*, Barcelona, Spain, Oct. 16-17.

[32] N. Brummer, "Focal toolkit," *available online: https://sites.google.com/site/nikobrummer/focal*.