*INTERSPEECH 2019*
September 15–19, 2019, Graz, Austria

# An Empirical Evaluation of DTW Subsampling Methods for Keyword Search

*Bolaji Yusuf, Murat Saraclar*

Bogazici University

bolaji.yusuf@boun.edu.tr, murat.saraclar@boun.edu.tr

## Abstract

State of the art vocabulary-independent spoken term detection methods are typically based on variants of the dynamic time warping (DTW) algorithm since DTW, being based on acoustic sequence matching, allows robust retrieval in settings with scarcity of linguistic resources. However, the DTW comes with a high computational cost which limits its practicality in a deployed server. To this end, we investigate the efficacy of subsampling and propose a neural network architecture to reduce the computational load of DTW-based keyword search. We use a time-subsampled RNN to reduce the frame rate of the document as well as the dimensionality of representation while training it to maintain the cost incurred along the DTW alignment path, thus allowing us to reduce the computational complexity (both space and time) of the search algorithm.

Experiments on the Turkish and Zulu limited language packs of the IARPA Babel program show that the proposed methods allow considerable reduction in CPU time (88 times) and memory usage (18 times) without significant loss in search accuracy (0.0270 ATWV). Moreover, even at very high compression levels with lower search precision, high recall rates are maintained, allowing the potential of multi-resolution search.

**Index Terms**: keyword search, dynamic time warping, multiresolution signal processing

## 1. Introduction

Keyword search (KWS) involves the retrieval of a written query within a spoken archive. Given a query provided as a string of words in orthographic form and a collection of spoken utterances, the task is to determine if any of the utterances contain the query, the temporal location (within 500 ms) of any such detections, and a confidence score for each detection. A query-independent threshold below which detections are disregarded as false alarms is also required from a KWS system.

KWS is typically conducted by using an LVCSR system to transcribe the document into compact representations such as lattices [1] or confusion networks [2], which are then converted into an inverted index [3] on which the query can be efficiently searched. One drawback of this approach, especially in low resource settings, is the vocabulary constraint that comes with it. Out-of-vocabulary (OOV) terms such as neologisms, rare names and morphological inflections that are not in the training vocabulary are ordinarily irretrievable.

OOV terms are usually handled by using subword units such as syllables, morphs, multigrams or even phones instead of whole words and leveraging the much larger vocabulary coverage of such units to retrieve the OOV words [4, 5, 6]. Another method of OOV term retrieval involves searching for similar-sounding in-vocabulary (IV) "proxy" words instead of the original OOV ones [7, 8]. The proxy approach has an advantage of not requiring the generation of a second, subword, index. Lexicon expansion methods [9, 10] have also been applied, although the effectiveness of such methods naturally depends on how well they are able to preemptively cover the OOV terms without knowing them. DTW has also been used been used for vocabulary independent term retrieval in the context of both query-by-example (QBE) [11, 12, 13] and keyword search [14, 15].

Using DTW for KWS involves representing the document in a vector space (e.g. posteriorgrams from a DNN), learning a mapping from phonemes to vectors in that space, and then aligning such representations so that, in effect, the search becomes QBE. While this method outperforms proxy keywords [14] and, as we'll show, subword OOV retrieval, its practicality is impeded by the computational cost of the DTW algorithm, which is linear (memory and CPU) in the lengths of both the query and document sequences.

Several methods have been proposed for reducing the computational load of DTW. One approach involves computing a cheaper lower bound to the DTW [16, 17]. Another technique involves computing coarse DTW on segments obtained using hierarchical agglomerative clustering [18] or posteriorgram averages [19]. Another framework uses randomized bit signatures to reduce the local computation has [20]. Overall, these frameworks leverage the properties of the document subspace to do a cheap search and using it to obtain candidates for a second-pass, finer DTW search.

In this paper, we investigate the use of document subsampling to improve the computational efficiency of DTW in the context of keyword search, and show that it reduces the computational load considerably without much loss in actual term weighted value (ATWV). Furthermore, we propose an RNN-based subsampling framework from which we obtain a low dimensional document vector representative of a sequence of higher dimensional ones, thus allowing us to further increase the efficiency of the search. Although the RNN architecture does not improve the simple subsampling in terms of ATWV, it does in terms of both memory usage and CPU efficiency. Experiments on the IARPA Babel Turkish and Zulu limited language packs (LLP) show that compared to a baseline without any subsampling, the proposed framework results in an 88-times speedup in search time and 18 times average memory usage reduction at the expense of 0.0270 reduction in average ATWV. Moreover, after compression, we maintain a high recall rate as evidenced by the average supremum term weighted value (STWV) of 0.7774, leaving room for search at finer resolution if needed.

## 2. Methodology

Spoken term detection with DTW involves matching a query sequence $W = [\mathbf{w}_1, \mathbf{w}_2, \dots \mathbf{w}_L]$ with a subsequence of the document $X = [\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_T]$. In text-based KWS, following the method of [14], the vectors of the query sequence come from a finite alphabet, $\mathcal{Q} = \{\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{q}^C\}$, whose elements correspond to linguistic units such as phones, and which are concatenated to obtain the query sequence. The subsequence dynamic

http://dx.doi.org/10.21437/Interspeech.2019-2413

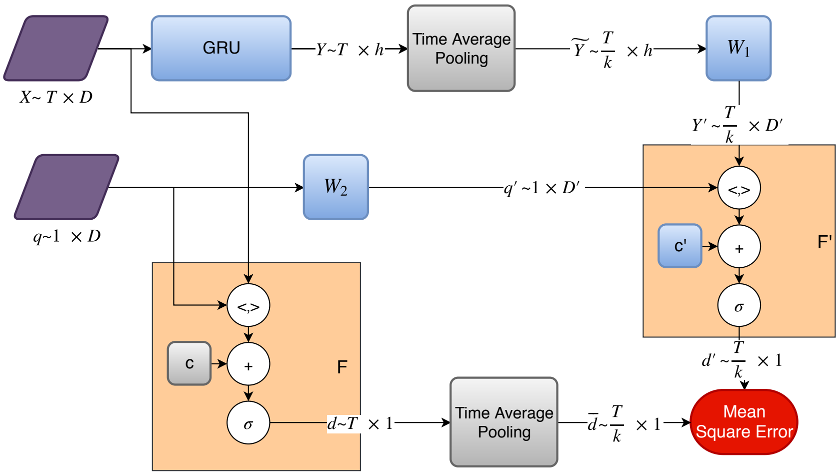


Figure 1: *Training the subsampling network to minimize the difference between full-rate and subsampled distortions. Note that only the light blue boxes are trained.*

time warping (sDTW) algorithm [21] is used for matching. The algorithm uses a frame distance function $F(\cdot,\cdot)$ to compute an optimal alignment path, $\mathbf{\Pi}$, between $W$ and subsequences of $X$ and returns a score:

$$\text{score} = 1 - \frac{1}{\text{length}(\mathbf{\Pi})} \sum_{n_i, m_i \epsilon \mathbf{\Pi}} F(\mathbf{x}_{n_i}, \mathbf{w}_{m_i}). \tag{1}$$

We use the extended distance metric learning (EDML) network of [22] to learn the document representation $X$ and the query alphabet $\mathcal{Q}$ with a distance function defined as

$$F(\mathbf{x}_i, \mathbf{w}_j) = \sigma(\mathbf{x}_i^T \mathbf{w}_j + c) \tag{2}$$

where $c$ is a bias term that is learned during the original metric learning framework. Note that the proposed framework is not bound to this representation. In fact, all we require are the query alphabet, the document representation and the distance function between them, all of which are already required for sDTW-based KWS anyway.

With ordinary document subsampling, we would take one out of every $k$ samples to obtain a new document representation, $\bar{X} = [\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \ldots \bar{\mathbf{x}}_{\frac{T}{k}}]$ where $\bar{\mathbf{x}}_t = \mathbf{x}_{tk}$. In the proposed subsampling neural network (sNN), shown in Figure 1, we use a gated recurrent unit (GRU) RNN with temporal pooling to reduce the frame rate and a pair of linear transformations to further reduce the dimensionalities of the document and query representations. We train the entire network with an objective to minimize the difference between the average DTW cost that each query frame incurs before and after the subsampling.

### 2.1. sNN Training

For a given training segment, $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_T]$, of the document and an element, $\mathbf{q} \in \mathcal{Q}$ from the query alphabet, the pairwise distance between $\mathbf{q}$, and every element of $X$ is a sequence $\mathbf{d}(\mathbf{q}) \in \mathcal{R}^T$ such that $\mathbf{d}_i(\mathbf{q}) = F(\mathbf{x}_i, \mathbf{q})$. First, we form another sequence $\bar{\mathbf{d}}(\mathbf{q}) \in \mathcal{R}^{T/k}$, of average distortions such that:

$$\bar{\mathbf{d}}_i(\mathbf{q}) = \frac{1}{k} \sum_{j=1}^{k} \mathbf{d}_{(i-1)k+j} \tag{3}$$

and train the network to predict this sequence.

We use a GRU (the default flavor described in [23]) with one hidden layer which takes $X$ as its input and outputs another sequence, $Y$, of the same length. We use average pooling across time to reduce the length to $T/k$. With two linear transformations, $W_1$ and $W_2$ respectively, $Y$ and $\mathbf{q}$ are transformed to have the same dimensionality. A function $F'(\cdot,\cdot)$ is used to compute the pairwise distances between $\mathbf{q}'$ and the elements of $Y'$:

$$\mathbf{d}'_i(\mathbf{q}') = F'(Y'_i, \mathbf{q}'). \tag{4}$$

The objective function that we then minimize is:

$$J = \frac{1}{|\mathcal{Q}|} \sum_{\mathbf{q} \in \mathcal{Q}} ||\mathbf{d}'(\mathbf{q}') - \bar{\mathbf{d}}(\mathbf{q})||^2 \tag{5}$$

Although we choose $F'(\cdot,\cdot)$ to have the same form as $F(\cdot,\cdot)$ in Equation (2), note that the only limitations on $F'(\cdot,\cdot)$ are that it be differentiable and have a range that covers that of $F(\cdot,\cdot)$. For instance, we could have chosen another $F''(Y_i, \mathbf{w}_j) = \frac{Y_i^T \mathbf{w}_j}{||Y_i|| \cdot ||\mathbf{w}_j||}$.

The training is done using Adam [24] with an initial learning rate of $10^{-3}$, which is reduced by half whenever the validation loss stalls for more than six training epochs.

### 2.2. sNN pre-search processing

Before search, we pre-compute $Y'$ and the set of $\{\mathbf{q}'\}$. This is done offline and ensures that at search time, the only computations we have to deal with are strictly DTW computations. Note that with this framework, we have control over the frame rate (by choice of $k$) and the dimensionality of the stored representations (by choice of the shapes of $W_1$ and $W_2$).

## 3. Experiments and Discussion

In this section, we describe the experiments conducted to evaluate the performance of the subsampling method described in the previous section. First we describe the dataset and features used to conduct the experiments as well as the metrics used to measure the system performance. Then we provide a desciption of the baseline and compare between it to other OOV-handling methods. Finally, we show the performance of the subsampling methods under various frame rates and feature dimensionalities.

### 3.1. Dataset and feature description

We experiment on the limited language pack (LLP) data from the IARPA Babel Program [25] which contains ten hours of conversational telephone speech for training in each language. Specifically, we use the Turkish[1] and Zulu[2] as our test languages and 19 of the other languages for training a multilingual neural network for feature extraction. The multilingual network is then finetuned on each test language and used to obtain posteriorgrams. Using the EDML network described in [22], we obtain 200-dimensional embeddings for DTW-based keyword search from the posteriorgrams.

In addition to the training set, each language also has a 10-hour development (Dev) set for tuning, as well as a 5-hour evaluation (Eval) set on which search is conducted. All the results provided except for those in Section 3.6 are on the Dev set.

### 3.2. Evaluation metrics

We use the term weighted value (TWV) to evaluate the goodness each system's search results. Given a set of terms, $\mathcal{Q}$ and a threshold, $\theta$, the TWV is defined thus:

$$TWV(\theta, \mathcal{Q}) = 1 - \frac{1}{|\mathcal{Q}|} \sum_{q \epsilon \mathcal{Q}} (P_{miss}(q, \theta) + \beta P_{fa}(q, \theta)) \quad (6)$$

where $P_{miss}(q, \theta)$ and $P_{fa}(q, \theta))$ are the probabilities of misses and false alarms respectively, and $\beta = 999.9$ is a parameter that controls the relative costs of false alarms and misses.

On the Dev set, we report the maximum term weighted value (MTWV) which is the TWV evaluated at the value of $\theta$ that maximizes it, and for the Eval set, we report the ATWV which is computed using the threshold learned on the Dev set. Additionally, we report the Eval set STWV, which is the MTWV computed with the cost of false alarms set to zero, and hence provides a measure of recall. Further discussion of the TWV metrics can be found in [26] and [27].

In addition to these accuracy measures, we also report two efficiency measures for the DTW-based systems compared to tge baseline system, namely: the speedup in search time, and the reduction in the memory required to store the document.

### 3.3. Baseline system

The baseline system uses the 200-dimensional features obtained from the distance metric learner for DTW-based search. Figure 2 shows the MTWV obtained for the baseline on OOV terms when compared with other OOV handling methods. We see that it provides average MTWV improvements of about 0.06 and 0.20 on the subword and proxy methods respectively. However, the other two methods involve constructing an (FST-based) index offline, thus ensuring that the actual search is quite fast.

Figure 3 shows the impact of duration statistics on query construction for the baseline system (note that this figure, unlike Figure 2, includes all terms, and not just OOV terms). The method described in [14] requires that the query sequence be generated by repeating each phoneme representation before concatenating them. This ensures that the constructed queries are approximately as long as they are in the document. The repetitions are done using duration statistics obtained from the training alignments. From Figure 3, we see that using these repetitions is crucial for good KWS performance, with an MTWV drop-off of about 0.13 when they are ignored in favor of a query representation that uses one frame per phone.

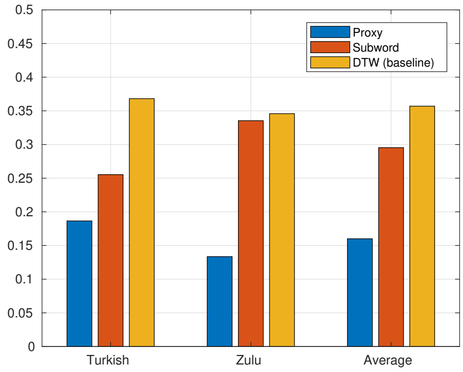


Figure 2: *MTWV comparison of various OOV-handling techniques on OOV terms.*

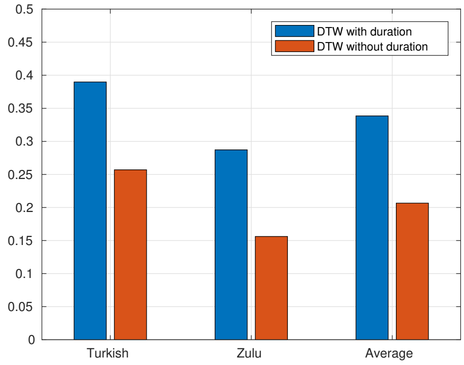


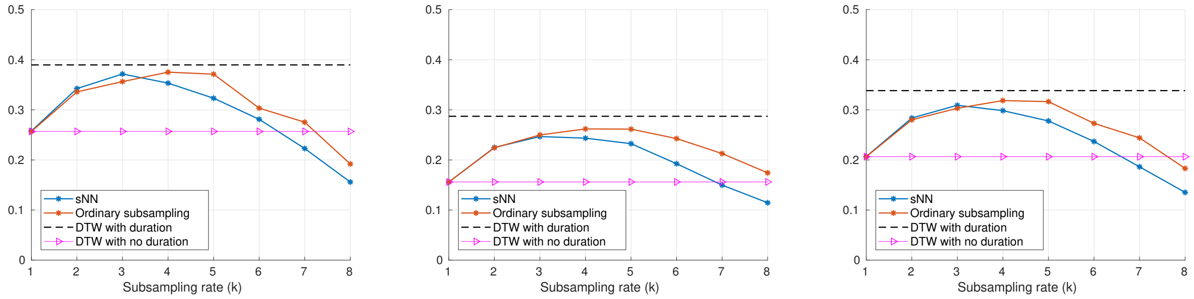
Figure 3: *Impact of duration statistics on the MTWV performance of baseline system over all keywords.*

Note that the exclusion of duration statistics, although detrimental to MTWV, is good for runtime since the DTW algorithm is linear in both document and query length. In the remaining experiments, we use the one-frame-per-phone setup to take advantage of this efficiency, and show that subsampling the document makes up for the omission of duration statistics.

### 3.4. Impact of frame rate reduction

With the document (and query) dimensionality kept at the original 200, and without using duration statistics for query modeling, we investigate the impact of subsampling at various rates. Figure 4 shows an MTWV comparison between ordinary subsampling and subsampling with the sNN approach as well as the baseline (with and without duration statistics). We observe that both methods initially give MTWV values similar to the baseline with no duration statistics. As the subsampling rate is increased (and the queries' durations approach their durations in the document), they both improve and are only slightly worse than the DTW with query duration statistics despite the considerable improvement in efficiency. As the rate is further increased, the minimum (DTW alignment) duration of each query frame becomes too large, and the MTWV deteriorates, although the sNN peaks earlier than ordinary subsampling ($k = 3$) vs.

[1] `babel105b-v0.4 (dev:kwlist, eval:kwlist2)`
[2] `babel206b-v0.1e (dev:kwlist3, eval:kwlist4)`



(a) *Turkish*

(b) *Zulu*

(c) *Average*

Figure 4: *MTWV progression of the with increase in subsampling rate for both ordinary and neural netword based subsampling.*

($k = 5$) and hence provides less computational improvement.

### 3.5. Impact of dimensionality reduction

With the subsampling rate set to $k = 3$, we attempt to further improve the sNN DTW efficiency by reducing the dimensionality of the stored representations, thus reducing the memory cost as well as the CPU cost of the inner product–based distances. Note that we are able to control this by choosing the shapes of $W_1$ and $W_2$ (see Figure 1) accordingly.

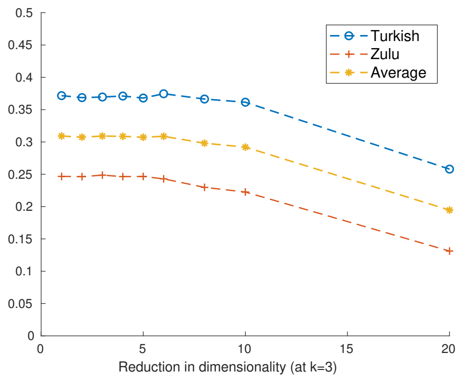


Figure 5: *MTWV progession as dimensionality of representation is reduced. The x-axis denotes $\frac{200}{D'}$, i.e. ratio of original dimensionality to the test dimensionality.*

Figure 5 shows the MTWV as the dimensionality is reduced from its original 200. We see that further memory and CPU improvements are achievable without loss in MTWV. For the Eval set experiments reported below, we choose $D' = 33$ (6 times further improvement in memory efficiency).

### 3.6. Eval-set results

Using the parameters tuned on the Dev set, we conduct the Eval-set DTW-based search and show the results in Table 1.

We observe that subsampling makes the search about 54 times faster compared to the baseline, at a loss of about 0.0181 ATWV. With neural network–based subsampling, we get average speedup up to 88 times, at about 0.0270 loss in average ATWV. Thus we're able to search in about 0.05 seconds/query/document-hour before parallelization. Furthermore, we notice high recall rates on the subsampled search systems (higher even than the baseline); thus, there is a potential for using these systems as a first pass search mechanism on whose results finer search can be further conducted.

Table 1: *Eval ATWV performance. B. (+dur) and B. (-dur) refer to the baseline with and without duration statistics; Subs. denotes ordinary subsampling and sNN denotes neural network based subsampling. The Mem. column reports the ratio of the baseline's memory requirement to each system's. The feature dimensionality is 200 except where stated otherwise.*

| Language | System | ATWV | STWV | Speedup | Mem. |
|---|---|---|---|---|---|
| Turkish | B. (+dur.) | 0.2646 | 0.6857 | 1 | 1 |
| | B. (-dur.) | 0.1748 | 0.7448 | 9 | 1 |
| | Subs.($k = 5$) | 0.2535 | 0.7819 | 43 | 5 |
| | sNN($k = 3$) | 0.2594 | 0.7977 | 25 | 3 |
| | sNN($k = 3, D' = 33$) | 0.2570 | 0.8026 | 62 | 18 |
| Zulu | B. (+dur.) | 0.2640 | 0.6465 | 1 | 1 |
| | B. (-dur.) | 0.1514 | 0.6945 | 17 | 1 |
| | Subs.($k = 5$) | 0.2390 | 0.7437 | 65 | 5 |
| | sNN($k = 3$) | 0.2252 | 0.7544 | 42 | 3 |
| | sNN($k = 3, D' = 33$) | 0.2176 | 0.7522 | 134 | 18 |
| Average | B. (+dur.) | **0.2643** | 0.6661 | 1 | 1 |
| | B. (-dur.) | 0.1631 | 0.7197 | 13 | 1 |
| | Subs.($k = 5$) | 0.2462 | 0.7628 | 54 | 5 |
| | sNN($k = 3$) | 0.2423 | 0.7760 | 33 | 3 |
| | sNN($k = 3, D' = 33$) | 0.2373 | **0.7774** | **88** | **18** |

## 4. Conclusions

In this work, we investigate the use of subsampling to improve the efficiency of DTW-based KWS and show its feasibility as a technique to significantly improve both CPU and memory efficiency with negligible loss in ATWV. Futhermore, we propose an RNN-based sampling method with which we further reduce the computational cost by learning a low-dimensional, low frame-rate representation that maintains DTW fidelity. In future work, we shall study this method in the context of infinite query alphabet search as in QBE e.g. by subsampling both the query and document as they are, or by learning a finite alphabet for the query and performing the search as in this work.

## 5. Acknowledgements

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) under Project No. 116E076.

# 6. References

[1] C. Chelba, T. J. Hazen, and M. Saraclar, “Retrieval and browsing of spoken content,” *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 39–49, 2008.

[2] L. Mangu, E. Brill, and A. Stolcke, “Finding consensus in speech recognition: word error minimization and other applications of confusion networks,” *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[3] D. Can and M. Saraclar, “Lattice indexing for spoken term detection,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 8, pp. 2338–2347, 2011.

[4] K. Ng and V. W. Zue, “Subword-based approaches for spoken document retrieval,” *Speech Communication*, vol. 32, no. 3, pp. 157–186, 2000.

[5] I. Szöke, M. Fapšo, and L. Burget, “Hybrid word-subword decoding for spoken term detection,” in *The 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2008, pp. 42–48.

[6] D. Karakos and R. Schwartz, “Subword and phonetic search for detecting out-of-vocabulary keywords,” in *Interspeech*, 2014, pp. 2469–2473.

[7] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, “Using proxies for OOV keywords in the keyword search task,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2013, pp. 416–421.

[8] M. Saraclar, A. Sethy, B. Ramabhadran, L. Mangu, J. Cui, X. Cui, B. Kingsbury, and J. Mamou, “An empirical study of confusion modeling in keyword search for low resource languages,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Dec 2013*, 2013, pp. 464–469.

[9] D. Can, E. Cooper, A. Ghoshal, M. Jansche, S. Khudanpur, B. Ramabhadran, M. Riley, M. Saraclar, A. Sethy, M. Ulinski *et al.*, “Web derived pronunciations for spoken term detection,” in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2009, pp. 83–90.

[10] A. Gandhe, L. Qin, F. Metze, A. Rudnicky, I. Lane, and M. Eck, “Using web text to improve keyword spotting in speech,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 428–433.

[11] Y. Zhang and J. R. Glass, “Unsupervised spoken keyword spotting via segmental DTW on Gaussian posteriorgrams,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009, pp. 398–403.

[12] C. Parada, A. Sethy, and B. Ramabhadran, “Query-by-example spoken term detection for OOV terms,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, Nov 2009, pp. 404–409.

[13] 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 *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 05 2014.

[14] B. Gündoğdu, B. Yusuf, and M. Saraçlar, “Joint learning of distance metric and query model for posteriorgram-based keyword search,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1318–1328, 2017.

[15] B. Gundogdu, B. Yusuf, and M. Saraclar, “Generative rnns for oov keyword search,” *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 124–128, Jan 2019.

[16] Y. Zhang and J. Glass, “A piecewise aggregate approximation lower-bound estimate for posteriorgram-based dynamic time warping,” in *Interspeech*, 2011, pp. 1909–1912.

[17] P. Yang, L. Xie, Q. Luan, and W. Feng, “A tighter lower bound estimate for dynamic time warping,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8525–8529.

[18] C.-a. Chan and L.-s. Lee, “Integrating frame-based and segment-based dynamic time warping for unsupervised spoken term detection with spoken queries,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 5652–5655.

[19] G. Cetinkaya, B. Gundogdu, and M. Saraclar, “Pre-filtered dynamic time warping for posteriorgram based keyword search,” in *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT)*, Dec 2016, pp. 376–382.

[20] A. Jansen and B. Van Durme, “Efficient spoken term discovery using randomized algorithms,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 401–406.

[21] M. Müller, *Information Retrieval for Music and Motion*. Berlin, Heidelberg: Springer-Verlag, 2007.

[22] B. Yusuf, B. Gundogdu, and M. Saraclar, “Low resource keyword search with synthesized crosslingual exemplars,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 7, pp. 1126–1135, July 2019.

[23] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: https://www.aclweb.org/anthology/D14-1179

[24] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[25] M. Harper, *IARPA Babel program*, 2014, accessed at June 2018. [Online]. Available: https://www.iarpa.gov/index. php/research-programs/babel

[26] S. Wegmann, A. Faria, A. Janin, K. Riedhammer, and N. Morgan, “The tao of atwv: Probing the mysteries of keyword search performance,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2013, pp. 192–197.

[27] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, “Results of the 2006 spoken term detection evaluation,” in *Proceedings of the ACM SIGIR Workshop on Searching Spontaneous Conversational Speech*, 2007, pp. 51–57.