INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Similarity Learning Based Query Modeling for Keyword Search

Batuhan Gundogdu[1,2], *Murat Saraclar*[1]

[1]Bogazici University, Turkey
[2]National Defense University Naval Academy, Turkey
batuhan.gundogdu@boun.edu.tr, murat.saraclar@boun.edu.tr

Abstract

In this paper, we propose a novel approach for query modeling using neural networks for posteriorgram based keyword search (KWS). We aim to help the conventional large vocabulary continuous speech recognition (LVCSR) based KWS systems, especially on out-of-vocabulary (OOV) terms by converting the task into a template matching problem, just like the query-by-example retrieval tasks. For this, we use a dynamic time warping (DTW) based similarity search on the speaker independent posteriorgram space. In order to model the text queries as posteriorgrams, we propose a non-symmetric Siamese neural network structure which both learns a distance measure to be used in DTW and the frame representations for this specific measure. We compare this new technique with similar DTW based systems using other distance measures and query modeling techniques. We also apply system fusion of the proposed system with the LVCSR based baseline KWS system. We show that, the proposed system works significantly better than other similar systems. Furthermore, when combined with the LVSCR based baseline, the proposed system provides up to 37.9% improvement on OOV terms and 9.8% improvement on all terms.

Index Terms: keyword search, distance metric learning, query modeling, out-of-vocabulary terms

1. Introduction

KWS is defined as the task of retrieving speech content from an untranscribed audio archive through a text query provided by a user. As the amount of unlabeled speech data increases throughout the multimedia platforms such as conference and meeting talks, lecture videos, radio communications and even telephone conversations, today KWS is considered as not only a necessity to retrieve the speech of interest, but also a means of labeling them towards the goal of zero resource speech recognition systems. The contemporary approach to KWS is using an LVCSR system to produce stochastic structures from speech (such as lattices), a procedure called indexing, and to use weighted finite state transducers (WFST) to retrieve the term from these indexes by combination operations of transducers [1, 2, 3].

These systems provide successful precision and recall rates under good recording conditions and when there are enough resources to train the LVCSR systems. However, for low resource languages where the amount of transcribed speech is scarce, the performance of the LVCSR based systems drop down drastically, especially for the out-of-vocabulary (OOV) terms. Recently, the OpenKWS project [4] primarily directed the focus onto these low resource languages where only a very limited amount of labeled data were to be used for system development. Another caveat that the LVCSR based systems have is their dependency on the vocabulary. The retrieval of the in-vocabulary (IV) terms is straightforward. The combination of the speech index transducer and the query transducer obtained from the lexicon gives the hit hypotheses along with their score. On the other hand, the OOV terms need some discretion. One of the most common approaches to OOV handling is using proxy keywords, i.e. finding acoustically similar IV terms for the retrieval of OOV terms [5]. In [6], confusion models were used in a similar manner to retrieve OOV terms. In [7], authors adopted an LVCSR-free approach similar to this work, to handle OOV terms by modeling keywords as point process models (PPM). Recently, Kartik et.al [8] proposed a recurrent neural network (RNN) based approach to create query embeddings and conducted the KWS by means of a feed forward neural network.

The essence of this work depends on modeling the text query as posteriorgrams and conducting the search by means of a version of DTW algorithm. In [9], authors modeled the text queries as posteriorgrams by concatenating either one-hot vectors or average posterior vectors learned from the training alignment. In [10], usage of different versions of cosine distance were compared with each of these query models. In [11] authors proposed a neural network based distance metric learning (DML) system to learn the distance measure to be used in DTW. It was shown that the new distance measure which they called the *'sigma distance'* not only provided a better frame level separation than the cosine distance, but also yielded a better KWS performance.

In this paper, on the other hand, we incorporate the query modeling into the DML algorithm. We do this by adding one more layer on the query side and using the first layer as the look-up table for phonemes. In Section 2, we introduce the template matching based KWS scheme along with the existing query modeling techniques of [10] and in Section 3, we provide mathematical and intuitive proofs and explanations of the proposed model by approaching the subject from 3 different perspectives. Finally in Sections 4 and 5, we present the experimental results conducted on IARPA Babel Program's Turkish development dataset.

2. Sequence Matching Based KWS

The search algorithm of this paper is inspired from the query by example (QbyE) retrieval tasks where the query is also provided in audio form [12]. After the acoustic feature extraction, the per-frame phone level posterior vectors were obtained using the Kaldi speech recognition toolkit trained with the HMM-DNN recipe [13]. The concatenation of these vectors is called

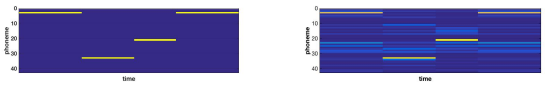
This study uses the IARPA Babel Program base period language collection release babel105b-v0.4, supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Defense U.S. Army Research Laboratory (DoD/ARL) contract number W911NF- 12-C-0012. This work was supported in part by The Scientific and Technological Research Council of Turkey (TUBITAK) under Project 116E076.

http://dx.doi.org/10.21437/Interspeech.2017-1273

posteriorgram which is simply a phone vs. time matrix.

2.1. The Role of Query Modeling

Since the queries are given in text form, they need to be modeled as posteriorgrams to conduct the DTW based similarity search within the document posteriorgram. In [9] and [10] they are modeled by concatenation of one-hot vectors depicting the labels for the pertinent phoneme (*binary modeling*) or by concatenating the means of the posterior vectors for the pertinent phoneme obtained from the training alignment (*average modeling*). For the modeling of the phoneme durations, the one-hot or average vectors were repeated as many times as the average phoneme durations which are estimated from the training alignment. The pseudo posteriorgram models of a sample word '*arma*' obtained using the binary and average query modeling techniques can be seen on Figure-1.



(a) *Binary Model* (b) *Average Model*

Figure 1: *Binary and Average Query Modeling*

In this work, on the other hand, we propose learning the pseudo query frame representations that would have better discriminative and representative features, rather than using one-hot vectors or only first order means. The proposed system's flowchart can be seen on Figure-2.

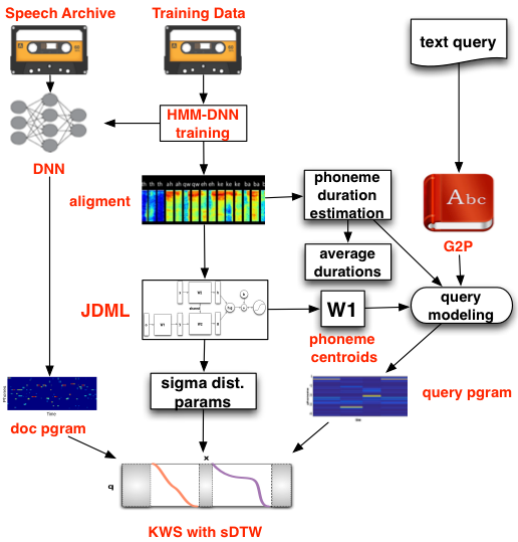


Figure 2: *Flowchart of the Proposed System*

2.2. Subsequence DTW and Similarity Score

Once the document and query posteriorgrams are obtained, the subsequence DTW (sDTW) algorithm [14] is used to obtain the alignments of the query with subsequences of the speech. If we call the query, $\mathcal{Q} = \{\mathbf{q}_1, \cdots, \mathbf{q}_M\}$, the speech utterance $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$, and the optimal alignment path between $\mathcal{Q}$ and any subsequence of $\mathcal{X}$ as Φ; the detection score for this specific subsequence is found from the average of the accumulated distance through the path using the frame-level distance measure of the sDTW algorithm.

$$\text{score} = 1 - \frac{1}{\text{length}(\Phi)} \sum_{(i,j)\in\Phi} d(\mathbf{q}_i, \mathbf{x}_j) \tag{1}$$

It is obvious from (1) that not only the representation power of the frames of query model ($\mathbf{q}_i$), but also the distance metric $d(\mathbf{q}, \mathbf{x})$ posses a great importance in the success of the KWS. In the end, the detection score is all that we have to decide if a subsequence is a match or not.

2.3. Choice of Distance Metric and DML

As the distance metric of the sDTW algorithm, several measures can be used. In [15] authors used euclidean distance measure for MFCC features. In [10] versions of cosine distance were compared with respect to different query models. In [11] *sigma distance* metric, learned through a siamese neural network, was used and shown to provide better frame level separations by modeling inter-phoneme confusions and yielded a better KWS performance than cosine distance. The sigma distance is defined as

$$d_\sigma(\mathbf{x}, \mathbf{y}) = 1 - \sigma(\mathbf{x}^T \mathbf{W}^T \mathbf{W} \mathbf{y} + b) \tag{2}$$

where

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

in which, the parameters $\mathbf{W}$ and b are learned through on-line gradient descent. Although the sigma distance is compared with other distance metrics,it does not fully conform to the axioms of metric spaces yet it only provides a discriminative measure to serve the goals of KWS.

3. The Proposed System : Simultaneous Query Modeling and DML

Having built up the usage of the distance metric and the query modeling in the posteriorgram based KWS, now we can discuss the aim of this study better: *Learning a distance metric and the query model at the same time*. DML has been studied in image processing literature, in search for better embeddings to work in euclidean distance using similar siamese structures [16, 17]. In another face verification task, triplet costs were used to find the embedding, again on euclidean space [18]. In this work, we follow the recipe in [11]; work on posteriorgram space and use cross entropy cost function. To incorporate the query modeling into DML, we add one more layer at the query side and call it joint DML (JDML) since it jointly learns the distance metric parameters and query frame representations. The proposed JDML model is shown on Figure-3.

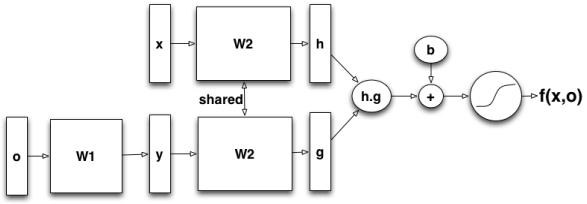


Figure 3: *JDML non-symmetric Siamese Network Model*

3.1. JDML Training

The goal of DML, proposed in [11], is to learn a distance measure that models the frame level phone confusions and to make the frames belonging to the same class (called *friends*) have smaller average distance between them while increasing the average distance between frames belonging to different classes (called *foes*). In other words, making *friends closer to each other and foes farther from each other*. JDML, on the other hand, is optimized for KWS purposes. The goal is to simultaneously learn proper representations for each class (phoneme) and learn distance measure parameters to get friends closer to their representation and farther from foes' representations. Hence, the input of the network is a pair, composed of the posterior vectors of the training alignment and a one-hot vector $(\mathbf{x}, \mathbf{o})$. The output labels are the desired sigma similarities, given as $r_t = \mathbf{o}_t[\text{class}(\mathbf{x}_t)]$. For any $\mathbf{x}$ in training, the prior of the pertinent class will be non-uniform due to the difference of frequencies of the phonemes in speech. Also, for each $\mathbf{x}$, the number of foe centroids are going to be much higher than the friend centroids. So, there is a risk of the system learning to favor 0 outputs. To overcome these two problems, we propose the prior equalization and JDML algorithm given in Algorithm-1.

Algorithm 1 JDML with Prior Equalization

1: Separate the dataset into subset of classes: $Set - class(i) = \{\mathbf{x}_t | , \mathbf{x}_t \in C_i\}_{i=1}^{K}$
2: Initialize network parameters $\mathbf{W}_1$,$\mathbf{W}_2$ and b, set the learning rates μ_1, μ_2, η
3: **repeat**
4: sample a class C_i randomly $1 \leq i \leq K$
5: sample $\mathbf{x}$ from $Set - class(i)$ randomly, set $\mathbf{o} = I(:, i)$
6: Calculate $f(\mathbf{x}, \mathbf{o}) = \sigma(\mathbf{x}^T\mathbf{W}_2^T\mathbf{W}_2\mathbf{W}_1\mathbf{o} + b)$ and the gradients (Figure-3)
7: Update network parameters for this pair of friends $\mathbf{W}_1 \leftarrow \mathbf{W}_1 + \mu_1\triangle\mathbf{W}_1$, $\mathbf{W}_2 \leftarrow \mathbf{W}_2 + \mu_2\triangle\mathbf{W}_2$ and $b \leftarrow b + \eta\triangle b$
8: sample a class C_i randomly $1 \leq i \leq K$
9: sample $\mathbf{x}$ from $Set - class(i)$ randomly
10: sample a class C_j $1 \leq j \leq K, j \neq i$, (foes of C_i)
11: set $\mathbf{o} = I(:, j)$
12: Calculate $f(\mathbf{x}, \mathbf{o}) = \sigma(\mathbf{x}^T\mathbf{W}_2^T\mathbf{W}_2\mathbf{W}_1\mathbf{o} + b)$ and the gradients
13: Update network parameters for this pair of foes $\mathbf{W}_1 \leftarrow \mathbf{W}_1 + \mu_1\triangle\mathbf{W}_1$, $\mathbf{W}_2 \leftarrow \mathbf{W}_2 + \mu_2\triangle\mathbf{W}_2$ and $b \leftarrow b + \eta\triangle b$
14: **until** the change in cost is less than ϵ

3.2. Interpretations of JDML

In this section, we provide mathematical and intuitive interpretations and justifications for the proposed JDML recipe.

3.2.1. JDML as a new kernel

Majority of the common distance metrics are based on inner product similarities. The distance value, reversely related to similarity, is obtained by applying a kernel function to the inner product similarity. These kernel functions, corresponding to some known metrics are shown on Figure-4. The smooth and symmetric behavior of the sigmoid kernel function may be desirable for KWS. We see that for low similarity values, log-cosine distance is quite aggressive and gives high distance values. This may prevent us finding keywords when there is a lot of pronunciation variability and phone confusion. Furthermore, for sigma distance, the inner product of two vectors may be increased or decreased with the $\mathbf{W}_2$ matrix to meet the desired distance value, whereas this is not an option for other distance measures.

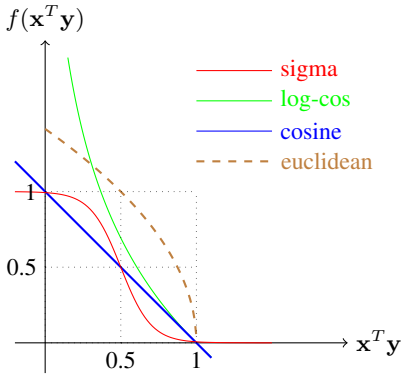


Figure 4: *Analysis of kernel functions for each distance metric.*

3.2.2. JDML as k-means clustering

We can consider JDML as the task of finding the centroids of phoneme classes with respect to sigma distance (2) and simultaneously updating the centroids and distance parameters. Similar to the k-means clustering in euclidean space, we take the derivative of the cost function and find the optimum. But this time the distance measure is sigma distance (instead of euclidean distance) and the cost is total cross entropy (instead of MMSE). If we denote the class centroids as $\mathbf{m}_k$, $k = 1 \cdots K$, and the class labels $r_{t,k} \triangleq \delta(\text{class}(\mathbf{x}_t) = k)$, $k = 1 \cdots K$, $t = 1 \cdots N$.

$$J_{\text{k-means}}(\mathbf{m}_k, r_{t,k}) = -\sum_t \sum_k r_{t,k} log(f_{t,k}) + (1 - r_{t,k}) log(1 - f_{t,k}) \quad (4)$$

where $f_{t,k} = \sigma(\mathbf{x}_t^T\mathbf{W}^T\mathbf{W}\mathbf{m}_k + b)$

If we take the derivative of J with respect to $\mathbf{m}_k$ and equate to zero, we get

$$\begin{aligned} \frac{dJ}{d\mathbf{m}_k} = 0 &= -\sum_t (r_{t,k} - f_{t,k})\mathbf{W}^T\mathbf{W}\mathbf{x}_t \\ \sum_t r_{t,k}\mathbf{x}_t &= \sum_t f_{t,k}\mathbf{x}_t \\ \sum_{\forall \mathbf{x}_t \in C_k} \mathbf{x}_t &= \sum_t f_{t,k}\mathbf{x}_t \end{aligned} \quad (5)$$

This result makes sense, in that the optimal centroids are obtained at locations where $f_{t,k} = 1$ if $\mathbf{x}_t \in C_k$ and zero otherwise. Since there is no closed-form solution, we approach with gradient descent. The iteration of $\mathbf{m}_k$ is

$$\mathbf{m}_k = \mathbf{m}_k + \eta(\sum_t (r_{t,k} - f_{t,k})\mathbf{W}^T\mathbf{W}\mathbf{x}_t) \quad (6)$$

The equation (6) shows us that the class centroids $\mathbf{m}_k$ are the wighted and projected sum of all training samples. Weights are decided by the error, basically *friends are added and foes are subtracted*. This approach differs from euclidean distance in another aspect, that is, while in euclidean k-means, only the

samples of the pertinent class is averaged, here all samples are taken in the calculation.

3.2.3. JDML as a Representation Learning Layer

In this approach, we represent the learning of class centroids as an initial layer on the query side and solve for the model in Figure-3. Since the desired values are binary, we consider the problem as a two-class classification task rather than a binary logistic regression, and use the cross entropy cost function.

$$J_{CE}(\mathbf{W}_1, \mathbf{W}_2, b; \mathbf{x}_t, \mathbf{o}_t, r_t) = \\ -r_t \log(f(\mathbf{x}_t, \mathbf{o}_t)) - (1 - r_t)\log(1 - f(\mathbf{x}_t, \mathbf{o}_t)) \tag{7}$$

where, $f(\mathbf{x}_t, \mathbf{o}_t) = \sigma(\mathbf{x}^T \mathbf{W}_2^T \mathbf{W}_2 \mathbf{W}_1 \mathbf{o} + b)$. The gradients are calculated following the on-line gradient descent recipe:

$$\begin{aligned} \triangle \mathbf{W}_2 &= \frac{dJ}{d\mathbf{W}_2} = \frac{dJ}{df}\frac{df}{dz}\frac{dz}{d\mathbf{W}_2} \\ &= (\frac{r-f}{f(1-f)})(f(1-f))\frac{d}{d\mathbf{W}_2}(\mathbf{x}^T \mathbf{W}_2^T \mathbf{W}_2 \mathbf{W}_1 \mathbf{o}) \\ &= (r-f)\mathbf{W}_2(\mathbf{x}\mathbf{o}^T \mathbf{W}_1^T + \mathbf{W}_1 \mathbf{o}\mathbf{x}^T) \end{aligned} \tag{8}$$

$$\begin{aligned} \triangle \mathbf{W}_1 &= \frac{dJ}{d\mathbf{W}_1} = \frac{dJ}{df}\frac{df}{dz}\frac{dz}{d\mathbf{W}_1} \\ &= (\frac{r-f}{f(1-f)})(f(1-f))\frac{d}{d\mathbf{W}_1}(\mathbf{x}^T \mathbf{W}_2^T \mathbf{W}_2 \mathbf{W}_1 \mathbf{o}) \\ &= (r-f)(\mathbf{W}_2^T \mathbf{W}_2 \mathbf{x}\mathbf{o}^T) \end{aligned} \tag{9}$$

where $z = \mathbf{x}^T \mathbf{W}_2^T \mathbf{W}_2 \mathbf{W}_1 \mathbf{o} + b$ and $f(\mathbf{x}_t, \mathbf{o}_t)$ is denoted as f for the sake of simplicity.

The update results obtained by the two approaches (clustering and representation learning) yield the same mathematical result. The equations (6) and (9) denote the same operations since the update on $\mathbf{W}_1$ takes place only on the pertinent column of the matrix (because of the one-hot vector on the outer product). Here the columns of $\mathbf{W}_1$ become the centroids of classes with respect to sigma distance ($\mathbf{m}_k$). One good aspect of this approach is that we can update the sigma distance parameters ($\mathbf{W}_2$ and b) and the class centroids ($\mathbf{W}_1$) at the same time.

4. Experiments

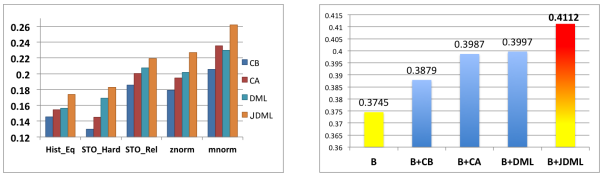
For the experiments, we used IARPA Babel limited language pack (LimitedLP) Turkish conversational telephone speech data (babel105b-v0.4) [4]. In this low-resource set-up, the training set is only 10-hours. Both the search and training posteriorgrams were obtained using Kaldi Speech Recognition Toolkit [13]. The experiments were conducted on 10-hour speech document over 307 keywords provided by the Babel Program. The baseline system uses the LVSCR based KWS setup in Kaldi toolkit. This pipeline uses proxy keywords to handle OOV queries [19].

4.1. Evaluation Metrics

The evaluation metric to KWS is the term weighted value (TWV) which is a linear combination of precision and recall [20]. TWV yields a performance score based on a balanced evaluation between correct detections and false alarms. A system returning all queries with no false alarms will yield a TWV of **1**. Similarly, a system with no outputs will yield a zero TWV and hence it is possible to have negative TWVs for those systems having more false alarms than correct decisions. For system development, we observe the maximum TWV (MTWV), which is the TWV for the optimal global threshold.

4.2. Experimental Results

On the individual systems, we applied several normalization techniques, such as histogram equalization [21], sum-to-one normalization [22], z-norm and m-norm [23]. We see that JDML consistently yields the best result in every normalization method. As an alternative system to state of the art LVCSR based KWS systems, JDML based KWS has a comparable performance to the LVCSR based baseline. We observe the real power of JDML based KWS from its effect on performance upon fusion with the baseline. When we combine the results of JDML based KWS with the LVCSR based KWS, we observe a 9.8% relative improvement over all terms. This improvement is more significant on the OOV terms (37.9%), yet we also see that the proposed system helps the retrieval of IV terms with an increase of 4.8% in MTWV. (Table-1)



(a) *Individual* (b) *Combined*

Figure 5: *Individual and Fused System MTWVs, CB and CA stand for cosine distance with binary and average query modeling respectively, and B denotes the baseline.*

Table 1: *MTWV and OTWV upon system fusion*

		B	B+JDML	Gain (%)
MTWV	all	0.3745	**0.4112**	9.80
	iv	0.4499	**0.4717**	4.85
	oov	0.1886	**0.2601**	37.91
OTWV	all	0.4147	**0.5191**	25.17
	iv	0.4896	**0.5595**	14.28
	oov	0.2240	**0.4162**	85.80

5. Conclusions and Future Work

Experiments conducted on IARPA Babel Program's Turkish data show that, the proposed similarity metric learning based query modeling system outperforms all similar systems. Furthermore, when combined with the LVCSR based baseline, it improves the performance especially on OOV terms. If we observe the OTWV metric (Table-1), where separate thresholds for each keyword are used instead of a global threshold, the improvement is even more significant: The OOV OTWV improvement reaches 85%. This result is a promising sign that further normalization techniques may be studied to reach the OTWV with a single global threshold.

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] 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.

[3] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," *HLT-NAACL 2004: Main Proceedings*, vol. 51, p. 61801, 2004.

[4] "OpenKWS14 keyword search evaluation plan," http://www.nist.gov/itl/iad/mig/upload/KWS14-evalplan-v11.pdf.

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

[6] 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 *2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), Dec 2013*, 2013, pp. 464–469.

[7] C. Liu, A. Jansen, G. Chen, K. Kintzley, J. Trmal, and S. Khudanpur, "Low-resource open vocabulary keyword search using point process models," in *2014 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2789–2793.

[8] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end ASR-free keyword search from speech," in *The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 5-9 March 2017, New Orleans, USA*.

[9] L. Sarı, B. Gündoğdu, and M. Saraçlar, "Fusion of LVCSR and posteriorgram based keyword search," in *2015 Sixteenth Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[10] B. Gundogdu, L. Sari, G. Cetinkaya, and M. Saraclar, "Template-based keyword search with pseudo posteriorgrams," in *2016 IEEE 24th Signal Processing and Communication Application Conference (SIU)*, pp. 973–976.

[11] B. Gundogdu and M. Saraclar, "Distance metric learning for posteriorgram based keyword search," in *The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 5-9 March 2017, New Orleans, USA*.

[12] X. Anguera, L. J. Rodriguez-Fuentes, I. Szoke, A. Buzo, and F. Metze, "Query-by-example spoken term detection evaluation on low-resource languages," in *Proceedings of the International Workshop on Spoken Language Technologies for Underresourced Languages (SLTU)*, vol. 24, p. 31.

[13] 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 *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (ASRU)*.

[14] M. Mueller, "Dynmic Time Warping," in *Information Retrieval for Music and Motion*, 2007.

[15] A. Muscariello, G. Gravier, and F. Bimbot, "Zero-resource audio-only spoken term detection based on a combination of template matching techniques," in *2011 12th Annual Conference of the International Speech Communication Association(INTERSPEECH)*.

[16] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 539–546.

[17] J. Bromley, J. W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Säckinger, and R. Shah, "Signature verification using a "siamese" time delay neural network," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 7, no. 04, pp. 669–688, 1993.

[18] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[19] J. Trmal, G. Chen, D. Povey, S. Khudanpur, P. Ghahremani, X. Zhang, V. Manohar, C. Liu, A. Jansen, D. Klakow *et al.*, "A keyword search system using open source software," in *2014 IEEE Spoken Language Technology Workshop (SLT)*, pp. 530–535.

[20] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proceedings of ACM SIGIR Workshop on Searching Spontaneous Conversational*, 2007, pp. 51–55.

[21] M. Montague and J. A. Aslam, "Relevance score normalization for metasearch," in *Proceedings of the tenth international conference on Information and knowledge management*. ACM, 2001, pp. 427–433.

[22] Y. Wang and F. Metze, "An in-depth comparison of keyword specific thresholding and sum-to-one score normalization," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[23] I. Szoke, L. Burget, F. Grezl, J. H. Cernocky, and L. Ondel, "Calibration and fusion of query-by-example systems—but sws 2013," in *The 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7849–7853.