# Generative RNNs for OOV Keyword Search

Batuhan Gundogdu, Bolaji Yusuf, *Student Member, IEEE*, and Murat Saraclar, *Member, IEEE*

***Abstract*—The modeling of text queries as sequences of embeddings for conducting similarity matching based search within speech features has been recently shown to improve keyword search (KWS) performance, especially for the out-of-vocabulary (OOV) terms. This technique uses a dynamic time warping based search methodology, converting the KWS problem into a pattern search problem by artificially modeling the text queries as pronunciation-based embedding sequences. This query modeling is done by concatenating and repeating frame representations for each phoneme in the keyword's pronunciation. In this letter, we propose a query model that incorporates temporal context information using recurrent neural networks (RNN) trained to generate realistic query representations. With experiments conducted on the IARPA Babel Program's Turkish and Zulu datasets, we show that the proposed RNN-based query generation yields significant improvements over the statistical query models of earlier work, and yields a comparable performance to the state-of-the-art techniques for OOV KWS.**

***Index Terms*—Keyword search, out of vocabulary terms, query modeling, recurrent neural networks.**

## I. Introduction

RETRIEVAL of spoken content is an important task not only for finding the parts of interest in spoken archives, but also for facilitating automated speech mining for better large vocabulary continuous speech recognition (LVCSR). In particular, KWS systems aim to achieve these objectives by locating the specific parts of a spoken document where a user-provided keyword is uttered. The most intuitive and convenient method for keyword search is to transcribe the document into text (in the form of hypotheses lattices) using LVCSR systems, and then conduct a text-based search on the LVCSR output [1]–[3]. However, the paucity of labeled speech training data in low resource languages hinders the development of reliable KWS systems, resulting in error-prone KWS systems. Furthermore, if a term of interest contains words which are not in the training vocabulary of the LVCSR system, it cannot be be found in the word level transcriptions from that system and so cannot be in the search index. Such terms, referred to as out-of-vocabulary (OOV) terms, constitute one of the main challenges of KWS in low resource languages. *Retrieval of OOV terms is the main focus of this letter*.

In KWS, the search term, called *the query*, is provided in text form and KWS systems are required to find where the query is uttered in a speech corpus. Once the speech document is indexed with a LVCSR system, the text query is easily retrieved within this index [4]–[6]. However, when the available training data size is limited, many query words fall outside the coverage of the LVCSR system's training lexicon. It has been estimated that only about $1\%$ of the 7000 languages spoken in the world have enough linguistic resources to build reliable speech to text systems [7]. Therefore, it is necessary to develop alternative approaches that are able to accurately retrieve OOV terms. One such OOV retrieval method is the use of sub-word units for lattice and index generation [8]–[11]. Another line of work involves extending the language model and lexicon by automatic word synthesis [12] and automatic text crawling from web sources [13]; these methods attempt to convert OOV words into in-vocabulary (IV) ones ensuring that those words can occur on the lattice. The most widely practiced approach makes use of *proxy keywords*; by utilizing automatically generated pronunciations of the OOV keywords, similar sounding IV words can be found which are then searched for instead of the actual OOV ones [14], [15]. Apart from the LVCSR related techniques described above, another method has been proposed which models the keywords with their phonetic indexes as point process models (PPM) and conducts the search on the document posteriorgram [16].

The utilization of Query by Example Spoken Term Detection (QbE-STD) techniques for retrieval of OOV terms has recently been shown to provide state-of-the art performance [17]. This technique involves converting the text queries into sequences of frames whose representations are jointly learned with a distance metric to be used in the dynamic time warping (DTW)-based search. The OOV KWS problem is thus converted into a QbE-STD task. This method yields superior performance compared to the soft indexing and proxy keyword-based approaches on OOV terms since the search audio is not indexed to belong to word/sub-word level states, for which the decision would not be confident, due to the OOV nature of the search. Rather, the document is represented in a *softer* form (than the soft-indexing), by just the frame representations and the problem is converted to a pattern-matching task. With this method, two major advantages are achieved: (1) A similarity score for any subsequence in audio could be obtained, enabling fruitful normalization techniques rather than just sum-to-one (STO) or keyword specific thresholding (KST) [18], [19] and, (2) the number of misses is reduced considerably, at the cost of slightly increased false alarm rates, yielding a significant term weighted value improvement; moreover, the normalized cross entropy based on the hits'

Manuscript received August 26, 2018; revised November 11, 2018; accepted November 12, 2018. Date of publication November 15, 2018; date of current version December 3, 2018. This work was supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Project 116E076. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hakan Erdogan. *(Corresponding author: Batuhan Gundogdu.)*

B. Gundogdu is with the Department of Electrical and Electronics Engineering, National Defense University, Naval Academy, Istanbul 34940, Turkey (e-mail: mbgundogdu@dho.edu.tr).

B. Yusuf and M. Saraclar are with the Department of Electrical and Electronics Engineering, Boğaziçi University, Istanbul 34342, Turkey (e-mail: bolaji.yusuf@boun.edu.tr; murat.saraclar@boun.edu.tr).

Digital Object Identifier 10.1109/LSP.2018.2881610

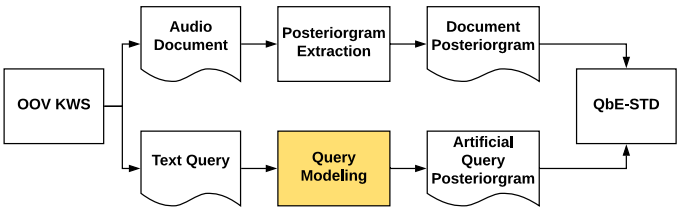


Fig. 1. Flowchart of the posteriorgram-based KWS. Focus of this letter is the query modeling phase.

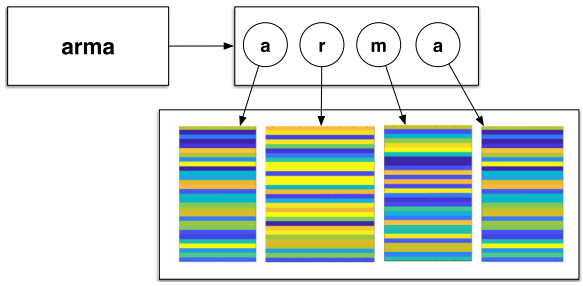


Fig. 2. Pseudo query modeling for template matching-based KWS.

scores reduce [17]. This letter follows this methodology with a further study of the inclusion of temporal context information to the artificial query modeling.

The main contribution of this letter can be summarized as follows: The query modeling phase of the DTW-based search is improved by inclusion of temporal information via generative RNNs. Instead of the statistical frame-based modeling and concatenation of certain representations, we propose learning whole *pseudo example queries*. In this aspect, this work differs significantly from the earlier work. Since the main goal of this work is to address modeling of OOV terms, we work towards generating a realistic posteriorgram that would serve as a template for search in the document posteriorgram. Experiments conducted on the IARPA Babel Program's [20] Turkish[1] and Zulu[2] datasets show that the generative modeling of query posteriorgrams using RNNs provides a 9% relative improvement on OOV retrieval performance when compared to the baseline that uses statistical frame-based query modeling (average ATWV: $0.1574 \rightarrow 0.1716$). The main scope of this work is OOV terms. However, it should be noted that the performance of the QbE-based technique is independent of vocabulary, hence it performs just as well on IV terms. Although the IV performance of the proposed system is modest compared to LVCSR-based techniques, both systems can be further improved by combining their results [17].

## II. Methodology

As mentioned in Section I, we aim to generate more realistic query posteriorgrams for KWS by incorporating temporal context information into query modeling. Posteriorgram-based KWS converts the OOV KWS problem into a QbE-STD task. A brief flowchart is provided in Fig. 1.

The posteriorgram representation, which is a class vs time matrix that represents *the probability of a speech frame belonging to one of the finite set of classes*, has been shown to provide better results than other feature representations in QbE-STD tasks, due to its speaker independence [21], [22]. In QbE-STD, given the document and query acoustic features ($A_x = \{a_{x_1}, a_{x_2}, \ldots, a_{x_T}\}$) and ($A_q = \{a_{q_1}, a_{q_2}, \ldots, a_{q_T}\}$), the posteriorgram representations can be expressed for the document ($\mathcal{X} \in [0,1]^{S \text{ x } x_T}; \mathcal{X} = \{\mathbf{x}_1 \ldots \mathbf{x}_{x_T}\}$ where $\mathcal{X}(i,j) = p(s_i|a_{x_j})$) and for the query ($\mathcal{Q} \in [0,1]^{S \text{ x } q_T}; \mathcal{Q} = \{\mathbf{q}_1 \ldots \mathbf{q}_{q_T}\}; \quad \mathcal{Q}(i,j) = p(s_i|a_{q_j})$),\break where $S$ is the number of phones. The phone posteriors are obtained by summing the probabilities of the context-dependent states corresponding to each phoneme. Since in KWS, the query sequence ($\mathcal{Q}$) is significantly shorter than the document sequence ($\mathcal{X}$), the search is employed via the subsequence DTW (sDTW) algorithm [23]. In sDTW, the boundary constraints are altered such that the short sequence is allowed to align with any subsequence of the long sequence. The similarity score between $\mathcal{Q}$ and any subsequence of document, $\mathcal{X}^{(s)}$ is calculated via the accumulated distance along the alignment path $\Phi$:

$$\text{score}(\mathcal{Q}, \mathcal{X}^{(s)}) = 1 - \frac{1}{\text{length}(\Phi)} \sum_{(r,k)\in\Phi} d(\mathbf{q}_r, \mathbf{x}_k^{(s)}) \qquad (1)$$

Since the frames of posteriorgrams $\mathcal{Q}$ and $\mathcal{X}^{(s)}$, i.e. $\mathbf{q}_t$ and $\mathbf{x}_t^{(s)}$ are probability mass functions, the logarithmic cosine distance is used as the local distance function in (1) and is defined as:

$$d_{\text{log-cos}}(\mathbf{q}, \mathbf{x}) = -\log\left(\frac{\mathbf{q}^T\mathbf{x}}{\|\mathbf{q}\|\|\mathbf{x}\|}\right) \qquad (2)$$

In KWS, however, the query is not spoken ($A_q$), but given in text form. To make a QbE-STD-like search feasible for KWS, the query posteriorgrams are artificially modeled using their (estimated) phonetic transcriptions. The synthetic query modeling method, first proposed in [24] uses the grapheme to phoneme (G2P) indexes [25] of the keyword's pronunciation and concatenates frame representations for each phoneme in the estimated pronunciation. While in [24], the phoneme posteriorgrams are modeled as repetitions of average posterior vectors for the corresponding phoneme, in [17], each phone posteriorgram is a learned representation that serves as a centroid in the jointly learned distance space. Since in these methods, one frame representation is learned for each phoneme, keyword pseudo posteriorgrams are chunks of repetitions of feature vectors. An illustration for the artificial query modeling is given in Fig. 2.

In this letter, we propose training a recurrent neural network that would 'generate' the artificial query, given its (estimated) pronunciation. The RNN is trained with the Viterbi alignment labels of the training audio as input, and the training posteriorgram as the output. The recurrent structure of the neural network uses the temporal information in generating the posteriorgram frames. The cross entropy cost is minimized comparing the actual posteriorgrams with the generated ones. With this methodology, we expect to get the following contributions to the posteriorgram-based KWS scheme: (1) The inter-phoneme confusions are modeled by non-recurrent weights. Since labels to the generator network are output activations of the decoder that is used to obtain the document posteriorgram, different ground truth posterior vectors are provided for the same input phoneme.

[1]babel105b-v0.4 (dev:kwlist, evalpart1:kwlist2)
[2]babel206b-v0.1e (dev:kwlist3, evalpart1:kwlist4)

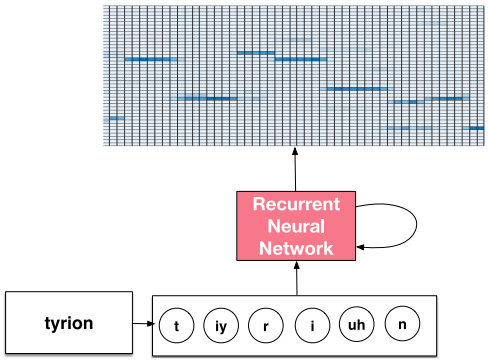


Fig. 3. Proposed query generation flowchart.

Hence an averaged activation is expected for each phoneme in the language. (2) The intra-phoneme variation as well as the confusion caused by context is modeled by the recurrent layers. The generator network takes the indexes of the pronunciation for the text keyword as input, and outputs a posteriorgram that would emulate an actual posteriorgram for an utterance of that keyword. The proposed scheme is illustrated in the flowchart in Fig. 3.

The pronunciation of the query term is obtained by a G2P system and represented as sequences of indexes. Network inputs are concatenations of one-hot vectors each with a one at the index of the corresponding phoneme in the pronunciation. To facilitate a realistic temporal flow and model the duration of the keyword, each phoneme frame is repeated as many times as its average duration estimated from the training alignment. The estimated query model ($\hat{\mathcal{Q}}$) is a function of the one-hot vectors, $\mathbf{o}_t$, which is modeled by RNN operations [26], [27].

$$\hat{\mathcal{Q}}(k,t) = f(\mathbf{o}_t, \mathbf{o}_{t-1} \dots \mathbf{o}_{t-\tau}) = p(s_k|\mathbf{o}_t, \mathbf{o}_{t-1} \dots \mathbf{o}_{t-\tau}) \quad (3)$$

During our system set-up, we experimented with various RNN models, to alter the function, $f$, as well as the memory, $\tau$, used in query modeling given in (3). In the vanilla RNN structure, where the hidden layer is a linear combination of recurrent and non-recurrent mappings, specifically

$$\mathbf{h}_t = g(\mathbf{W}_{oh}\mathbf{o}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1}), \quad \text{and} \quad \hat{\mathbf{q}}_t = \text{softmax}(\mathbf{h}_t), \quad (4)$$

where $g(.)$ is the nonlinearity applied to the hidden layer. The desired inter-phoneme confusions are modeled by $\mathbf{W}_{oh}$ and the temporal information is carried with $\mathbf{W}_{hh}$. Although not directly observable in weights as in the vanilla RNN networks, gated RNNs such as gated recurrent units (GRU) and long short-term memory (LSTM) RNNs also model temporal behavior of the speech and are used for query generation.

The document and training posteriorgrams are obtained from DNN acoustic models (with perceptual linear prediction (PLP) feature as input) trained with the Kaldi speech recognition toolkit [28]. The RNNs for query modeling are implemented with the Keras Toolkit [29] using the Theano back-end. Finally, the scores for each keyword are normalized by considering their distributions over the set of different returned hit hypotheses, following the recipe explained in [30].

## III. Experiments

We experiment on two low resource languages of the IARPA Babel Program: Turkish and Zulu, each with only 10 hours of transcribed training data.

### A. Experimental Set-up

In our experiments, each language has two test speech documents each with its set of keywords. The first set is called the *dev-set* and comprises a 10-hour spoken document. The Turkish and Zulu dev-sets contain 88 and 806 OOV terms for Turkish and Zulu respectively. We use the term weighted value (TWV) as the main KWS evaluation metric [31]. TWV is a linear combination of the precision and recall at a predefined global threshold. For development experiments, we report two TWV scores: the maximum TWV (MTWV) and the optimum TWV (OTWV). MTWV is defined as the TWV obtained at the best global threshold, whereas OTWV is the MTWV obtained with a separate, optimal, threshold values for each keyword.

We use the Turkish dev-set as the main development set to decide certain parameters of the proposed model. We train several different RNN architectures, including vanilla RNNs, GRUs and LSTMs. We also experiment on how much temporal context ($\tau$) to be included in the query generation. From these experiments, we pick the best performing systems, not by direct observation of training or validation loss, but by observing the actual KWS performance, evaluated by the MTWV.

*1) RNN Memory:* The first important parameter to decide in the generator development is the size of the memory to be used in posteriorgram generation, that is the number of past frames that the error is back propagated to. In the Turkish dev-set experiments we evaluate 5 memory settings : 10 frames (*remember the phone*), 40 frames (*the syllable*), 100 frames (*term*), 150 and 200 frames (*longer terms or phrases*). It should be noted that each frame is 10 ms and the number of frames are chosen to have the corresponding meanings in the parentheses on average. Although the terms in the keyword list are never directly seen in the generator training, we observe that inclusion of more memory up to 100 frames (*one term length*) improved the KWS performance, where longer memory caused high fluctuations in performance and a lower averaged performance. The results of these experiments are shown in Fig. 4.

*2) Architecture Selection:* Once we decide on the best memory parameter (100 frames), we then decide the architecture on which to continue the experiments. The development experiments are conducted on 13 different models on Turkish dev-set. The list of experiments and their performance with respect to the MTWV metric can be seen on Table I.

There are a number of observations on the initial generator experiments on the Turkish dev-set. Gated RNNs perform better than vanilla RNNs as they are less susceptible to the vanishing gradient problem for long temporal memory (up to a second). Making the networks wider or deeper does not seem help with the MTWV performance; we speculate that given the limited amount of training data, as the number of parameters increase, generalization to previously unseen speech parts (OOV terms) decreases. It should also be noted that the proper early stopping criteria are applied for each of the models. For Turkish,

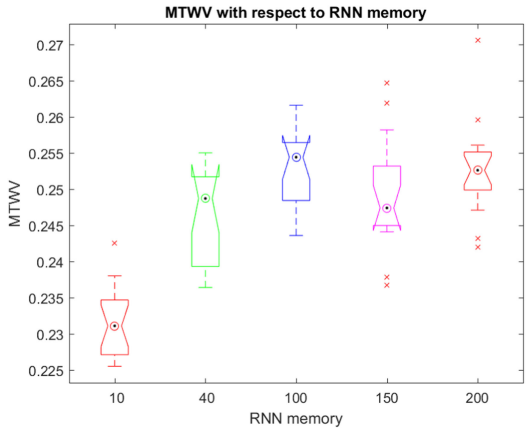


Fig. 4. Boxplot of the MTWV values obtained by several models and various memory settings ($\tau$).

TABLE I
PERFORMANCE OF VARIOUS GENERATOR MODELS IN THE TURKISH DEV-SET EXPERIMENTS

| MODEL NAME | Recurrent Structure | Layer Depth | Hidden Width | Turkish-Dev MTWV |
|---|---|---|---|---|
| RNN-1-S | Vanilla | 1 | $S$ | 0.2403 |
| RNN-1-128 | Vanilla | 1 | 128 | 0.2509 |
| RNN-1-256 | Vanilla | 1 | 256 | 0.2513 |
| RNN-2-128 | Vanilla | 2 | 128 | 0.2385 |
| GRU-1-S | GRU | 1 | $S$ | 0.2572 |
| GRU-1-128 | GRU | 1 | 128 | 0.2479 |
| GRU-1-256 | GRU | 1 | 256 | 0.2555 |
| GRU-2-S | GRU | 2 | $S$ | 0.2544 |
| **LSTM-1-S** | **LSTM** | **1** | **S** | **0.2606** |
| LSTM-1-64 | LSTM | 1 | 64 | 0.2530 |
| LSTM-1-128 | LSTM | 1 | 128 | 0.2506 |
| LSTM-1-128 | LSTM | 1 | 256 | 0.2443 |
| LSTM-2-64 | LSTM | 2 | 64 | 0.2550 |

the model named LSTM-1-S with one hidden layer and square weight matrices has the best performance.

### B. System Evaluation

After the development and system set-up experiments on the Turkish dev-set, we select the top five best performing architectures to experiment on Zulu. We test on different models since the two languages inevitably possess structural differences which could affect generator experiments. For Zulu, GRU-1-S was the best performing model, which is not unlike LSTM-1-S in that it also has square weight matrices and a gated structure [32].

The KWS performance evaluation is employed on the following systems:

- *Proxy:* The work in [14], retrieving acoustically similar proxy keywords for OOV keywords.
- *Statistical Frame-Based Query Modeling (SFQM):* The work in [24], posteriorgram-based keyword search with frame level query modeling using averages of posteriors.
- *Generative Query Modeling (GQM):* The proposed, posteriorgram-based keyword search with RNN-generated query models.

The dev-set performance metrics can be seen in Table II, which is also compared with the similar work of [17]. The SFQM numbers in this letter are higher than the original work reported in [24] since we use a distribution-based normalization technique (b2-norm) introduced in [30] whereas [24] uses STO normalization [33].

TABLE II
DEV-SET EXPERIMENTS

| Language | Metric | Proxy | SFQM | GQM | JDML [17] |
|---|---|---|---|---|---|
| Turkish | MTWV | 0.1880 | 0.2365 | **0.2606** | 0.2421 |
| | OTWV | 0.2317 | 0.4088 | 0.4099 | **0.4110** |
| Zulu | MTWV | 0.0915 | 0.2277 | 0.2542 | **0.2566** |
| | OTWV | 0.1552 | 0.4078 | **0.4408** | 0.3780 |
| Average | MTWV | 0.1398 | 0.2321 | **0.2574** | 0.2493 |
| | OTWV | 0.1935 | 0.4083 | **0.4254** | 0.3945 |

TABLE III
EVAL-SET EXPERIMENTS

| Language | Metric | Proxy | SFQM | GQM | JDML [17] |
|---|---|---|---|---|---|
| Turkish | ATWV | 0.0820 | 0.1554 | **0.1814** | 0.1551 |
| | MTWV | 0.0913 | 0.1657 | **0.1897** | 0.1608 |
| | OTWV | 0.2258 | 0.3382 | **0.3629** | 0.3370 |
| Zulu | ATWV | 0.0720 | 0.1594 | 0.1618 | **0.1693** |
| | MTWV | 0.0730 | 0.1750 | **0.1775** | 0.1775 |
| | OTWV | 0.1469 | 0.3298 | 0.3449 | **0.3516** |
| Average | ATWV | 0.0770 | 0.1574 | **0.1716** | 0.1622 |
| | MTWV | 0.0822 | 0.1703 | **0.1836** | 0.1691 |
| | OTWV | 0.1864 | 0.3340 | **0.3539** | 0.3443 |

The dev-set experiments serve as a system selection and tuning phase. Having learned the KWS parameters such as optimal decision threshold values, network architecture, score pruning threshold etc, the actual evaluation is done on a separate dataset, which is called the *evalpart1*, also provided by the Babel Program. The evaluation audio for each language is about 5 hours long and the keyword sets have 1216 and 1112 OOV terms for Turkish and Zulu, respectively. For the eval-set experiments we also report the actual TWV (ATWV) that is the system performance result for a set global threshold, learned in the dev-set experiments. The eval-set results are given in Table III.

## IV. CONCLUSION AND DISCUSSION

In this work, we proposed a method of modeling queries for QbE-based OOV term retrieval. By modeling queries as pseudo-posteriorgrams, we are able to reframe the keyword search task as a QbE one. Previously, the use of phoneme average vectors as models has been shown to outperform competing OOV retrieval methods. Using that as a baseline, we propose using RNNs to generate pseudo-posteriorgrams which account for phonemic context. The resulting query models are thus dynamic, incorporating context-dependent decoder confusions, unlike the baseline which represent each query phoneme with an unchanging vector throughout its evolution regardless of context. We show the superiority of the RNN-generated query model to the average model baseline. Moreover, we find that even the models obtained from the less optimal RNN parameters outperform the baseline (compare Tables I and II). Further work may involve developing a unified approach that eschews the DTW completely, as well as normalization methods that close the gap between the ATWV/MTWV and the OTWV.

## REFERENCES


[1] C. Allauzen, M. Mohri, and M. Saraclar, "General indexation of weighted automata: Application to spoken utterance retrieval," in *Proc. Workshop Interdiscip. Approaches Speech Indexing Retrieval HLT-NAACL*, 2004, pp. 33–40.

[2] D. Can and M. Saraçlar, "Lattice indexing for spoken term detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2338–2347, Nov. 2011.

[3] M. Saraçlar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. HLT-NAACL Main Proc.*, 2004, vol. 51, pp. 129–136.

[4] D. Wang, J. Frankel, J. Tejedor, and S. King, "A comparison of phone and grapheme-based spoken term detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2008, pp. 4969–4972.

[5] A. Garcia and H. Gish, "Keyword spotting of arbitrary words using minimal speech resources," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006, vol. 1, pp. 949–952.

[6] S.-w. Lee, K. Tanaka, and Y. Itoh, "Combining multiple subword representations for open-vocabulary spoken document retrieval," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, vol. 1, pp. 505–508.

[7] Y. Zhang, "Unsupervised speech processing with applications to query-by-example spoken term detection," Ph.D. dissertation, Dept. Elect. Eng. Comp. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2013.

[8] Y. He *et al.*, "Using pronunciation-based morphological subword units to improve OOV handling in keyword search," *IEEE/ACM Trans. Audio, Speech, Lang. Process*, vol. 24, no. 1, pp. 79–92, Jan. 2016.

[9] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2007, pp. 615–622.

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

[11] L. Burget, "Hybrid word-subword decoding for spoken term detection," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2008, pp. 42–48.

[12] G. Chen, S. Khudanpur, D. Povey, J. Trmal, D. Yarowsky, and O. Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in lowresource languages," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8560–8564.

[13] A. Gandhe, L. Qin, F. Metze, A. Rudnicky, I. Lane, and M. Eck, "Using web text to improve keyword spotting in speech," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 428–433.

[14] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for OOV keywords in the keyword search task," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 416–421.

[15] M. Saraçlar *et al.*, "An empirical study of confusion modeling in keyword search for low resource languages," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 464–469.

[16] C. Liu, A. Jansen, G. Chen, K. Kintzley, J. Trmal, and S. Khudanpur, "Low-resource open vocabulary keyword search using point process models," in *Proc. Interspeech*, 2014, pp. 2789–2793.

[17] B. Gündoğdu, B. Yusuf, and M. Saraçlar, "Joint learning of distance metric and query model for posteriorgram-based keyword search," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 8, pp. 1318–1328, Dec. 2017.

[18] D. R. Miller *et al.*, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007, pp. 314–317.

[19] Y. Wang and F. Metze, "An in-depth comparison of keyword specific thresholding and sum-to-one score normalization," in *Proc. Interspeech*, 2014, pp. 2474–2478.

[20] M. Harper, "IARPA Babel program," Accessed: Dec. 2017, 2014. [Online]. Available: https://www.iarpa.gov/index.php/research-programs/babel

[21] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2009, pp. 421–426.

[22] 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 *Proc. Int. Workshop Spoken Lang. Technol. Underresourced Lang.*, 2014, vol. 24, pp. 24–31.

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

[24] L. Sarı, B. Gündoğdu, and M. Saraçlar, "Fusion of LVCSR and posteriorgram based keyword search," in *Proc. Interspeech*, 2015, pp. 824–828.

[25] M. Bisani and H. Ney, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Commun.*, vol. 50, no. 5, pp. 434–451, 2008.

[26] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 338–342.

[27] A. Graves, A.-rahman Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *Acoustics, speech signal process. (icassp), IEEE Int. Confe.* pp. 6645–6649, 2013.

[28] J. Trmal *et al.*, "A keyword search system using open source software," *IEEE Spoken Lang. Techn. Workshop (SLT)*, pp. 530–535, Dec. 2014, doi: 10.1109/SLT.2014.7078630.

[29] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras

[30] B. Gündogdu, "Keyword search for low resource languages," Ph.D. dissertation, Electrical Engineering Dept., Bogaziçi Univ., Istanbul, Turkey, 2017.

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

[32] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, pp. 103–111, 2014.

[33] J. Mamou *et al.*, "System combination and score normalization for spoken term detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8272–8276.