Interspeech 2018
2-6 September 2018, Hyderabad

Fast ASR-free and almost zero-resource keyword spotting using DTW and CNNs for humanitarian monitoring

Raghav Menon[1], *Herman Kamper*[1], *John Quinn*[2], *Thomas Niesler*[1]

[1]Department of Electrical and Electronic Engineering, Stellenbosch University, South Africa
[2]UN Global Pulse, Kampala, Uganda

rmenon@sun.ac.za, kamperh@sun.ac.za, john.quinn@unglobalpulse.org, trn@sun.ac.za

Abstract

We use dynamic time warping (DTW) as supervision for training a convolutional neural network (CNN) based keyword spotting system using a small set of spoken isolated keywords. The aim is to allow rapid deployment of a keyword spotting system in a new language to support urgent United Nations (UN) relief programmes in parts of Africa where languages are extremely under-resourced and the development of annotated speech resources is infeasible. First, we use 1920 recorded keywords (40 keyword types, 34 minutes of speech) as exemplars in a DTW-based template matching system and apply it to untranscribed broadcast speech. Then, we use the resulting DTW scores as targets to train a CNN on the same unlabelled speech. In this way we use just 34 minutes of labelled speech, but leverage a large amount of unlabelled data for training. While the resulting CNN keyword spotter cannot match the performance of the DTW-based system, it substantially outperforms a CNN classifier trained only on the keywords, improving the area under the ROC curve from 0.54 to 0.64. Because our CNN system is several orders of magnitude faster at runtime than the DTW system, it represents the most viable keyword spotter on this extremely limited dataset.

Index Terms: relief and developmental monitoring, keyword spotting, convolutional neural networks, dynamic time warping, under-resourced, zero-resource speech processing

1. Introduction

In societies with well-developed internet infrastructure, social media has become a dominant medium for voicing views and concerns about various social issues [1, 2, 3]. In countries like Uganda, where internet availability is limited, phone-in talk shows on local community radio stations are used in a similar way. A United Nations (UN) piloted project has developed radio-browsing systems for monitoring such radio shows in order to inform relief and developmental purposes. These systems have been very successful and are in active use.

In previous work [4, 5], we assumed the availability of small amounts of transcribed data (initially about 9 hours and then just 12 minutes) for the development and deployment of such systems in two regional languages, Luganda and Acholi. However, even the preparation of a 12 minute transcribed corpus requires the availability of annotators in the new language with appropriate skills. This has proved a serious impediment to the development of a radio browsing system in a new language. Here we therefore turn our attention to the development of a keyword spotter that can be set up using resources that are even easier to obtain: a small set of isolated spoken keywords.

Recent advances in automatic speech recognition (ASR) technology have mostly been restricted to scenarios where very large transcribed speech resources are available [6, 7]. For keyword spotting, where the goal is to search a speech signal for occurrences of a keyword provided as text, most systems employ ASR to produce lattices which are subsequently searched [8, 9]. This is not feasible without at least a minimal corpus of transcribed speech in the target domain.

In settings where transcribed data is not available, researchers have attempted to achieve ASR-free keyword spotting by adopting a query-by-example (QbyE) retrieval procedure. In QbyE, the search query is provided not as text but as audio. Typically, dynamic time warping (DTW) is used to match the acoustic features from the search query to acoustic features from speech in the search collection [10, 11]. Alternatively, queries and search utterances can be mapped into a joint fixed-dimensional space [12, 13], allowing for efficient retrieval using vector comparisons. Such fixed-dimensional vector representations can be obtained using recurrent neural networks [14, 15] or, when matching word pairs are known, a Siamese convolutional neural network (CNN) [16]. In [17], an ASR-free keyword spotter is described which maps textual and acoustic input into a shared fixed-dimensional space where text queries can be compared directly to search utterances. However, all these ASR-free neural QbyE approaches rely on large amounts of training data.

In this paper, we combine DTW and CNNs to develop an ASR-free keyword spotter that is trained on an easy-to-obtain small number of isolated keyword utterances. Specifically, we use a small seed corpus and DTW to calculate training targets for a much larger unannotated corpus, that can subsequently be used to train a CNN-based keyword spotter. The CNN model is much faster than the DTW-based system (since alignment is not required), making this a viable option for real-time monitoring.

2. Radio Browsing System Configuration

To give the broader context for our work, we first describe our complete radio browsing system, shown in Figure 1. This figure also shows our previous ASR-based implementation, in which pre-processed speech from a live audio stream is passed to an ASR system which generates lattices. These lattices are then indexed and searched for the desired keywords. Our new system replaces the ASR components with a CNN-DTW keyword spotter. The detected keywords and their meta-data are passed to human analysts who filter the information and format it into a structured, categorised and searchable format appropriate for humanitarian decision making and situational awareness. In this scenario, high false positive rates can be tolerated because the human analysts can discard false detections. The overall approach allows the analysis of a large amount of audio data while maintaining a high confidence in the final output. A more detailed discussion on the role of human analysts and examples

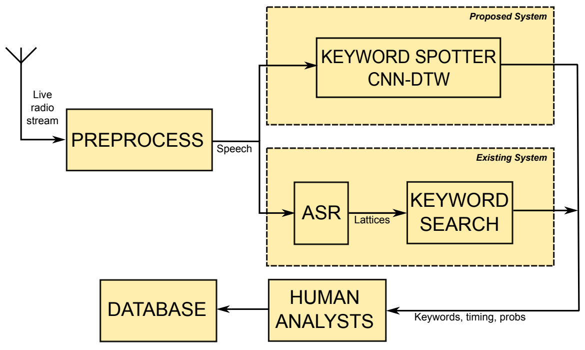


Figure 1: *Radio browsing system showing the existing and the proposed system.*

of detected topics of interest are presented in [5].[1]

3. Data

In this first work we use a corpus of South African Broadcast News (SABN) for experimental analysis. Since transcriptions are available for this data, it allows system performance to be experimentally evaluated. However in all other respects we consider the SABN data as untranscribed. Ultimately our goal is to apply this system to languages such as Somali, Rutooro and Lugbara, for which no language resources are available at all.

The SABN corpus consists of 23 hours of speech from news bulletins broadcast between 1996 and 2006 by one of the South Africa's main radio news channels, SAFM [18]. The corpus contains a mix of newsreader speech, interviews, and crossings to reporters. About 80% of the speakers can be considered native English speakers. The division of the corpus into training, development and test sets is shown in Table 1.

Table 1: *The South African Broadcast News (SABN) dataset.*

	Utterances	Speech (h)
Train	5231	7.94
Dev	2988	5.37
Test	5226	10.33
Total	13445	23.64

For the purpose of training our keyword spotter, we recorded isolated utterances of 40 keywords, each spoken twice by 24 speakers (12 male and 12 female), leading to a set of 1920 labeled isolated keywords. Keywords were displayed to the speakers sequentially, individually and in no particular order. Keywords were recorded in a quiet room under very different audio conditions and by different speakers than those in the SABN data. This is representative of the intended operational setting of our system, where isolated utterances of the required keywords will be recorded on location from native speakers using smart phones or audio recorders in conditions that will differ from the radio broadcasts that must be monitored.

4. Keyword Spotting

Here we describe our proposed approach of using DTW and a small set of individually recorded keywords to train a CNN on a a much larger, untranscribed corpus. We briefly describe DTW- and CNN-based keyword spotting approaches before describing our combined procedure.

[1]Examples available at `http://radio.unglobalpulse.net`.

4.1. Keyword spotting using dynamic time warping

When only few isolated spoken keywords are available, dynamic time warping (DTW) is an appropriate technique for their detection since this technique requires as little as a single audio template. DTW aligns two sequences of feature vectors by warping their time axes to achieve an optimal match. The associated alignment cost can be used as a metric of similarity between sequences. For keyword spotting with DTW, features are extracted for both the keyword and the search utterances. To determine whether a keyword is present in an utterance, a naive method is to compute the DTW cost of aligning the keyword with the speech in a sliding window over the search utterance. More advanced approaches that find subsequences have been proposed [19, 20], however in this work we restrict ourselves to this simpler implementation. We use cosine distance for frame-wise comparison, normalize to obtain per-frame cost, and use a frame-skip of 3 frames. The resulting cost $c \in [0, 2]$ has a value of 0 when the keyword matches a portion of the utterance exactly and a value of 2 when they are dissimilar. By choosing an appropriate threshold for c, it is possible to take a decision regarding the presence of the keyword in unlabeled speech. Although useful in low-resource settings, a major disadvantage of DTW-based search is that it requires alignment between keywords and search utterances, which can be prohibitively slow.

4.2. Convolutional neural network keyword spotting

In fully supervised settings where a large number of labelled keywords are available, an end-to-end keyword spotter could be trained to directly classify whether a keyword is present in a search utterance. Although we have only a limited set of labelled spoken keyword, we nevertheless attempt to train a supervised CNN classifier in this way. This baseline CNN classifier is trained directly in a supervised fashion on the 1920 recorded isolated keywords, as well as negative samples drawn randomly from utterances in the SABN training set. At test time, a sliding window is applied and keyword presence is predicted based on a threshold. Here we used a fixed window of 60 frames. Similar QbyE and keyword spotting systems based on neural networks have been developed using much larger labeled datasets in previous work [16, 21, 22].

4.3. CNN-DTW keyword spotting

CNNs require large amounts of data for training, but are computationally efficient to apply. DTW-based keyword spotting, on the other hand, can be applied with only a few keyword exemplars, but is computationally costly. Our proposal is to employ DTW during training to address the challenge of data scarcity while taking advantage of the speed benefits of CNNs at runtime. We achieve this by using DTW to calculate similarity scores between our small set of isolated keywords and a much larger untranscribed dataset and then use this set of similarity scores as targets to train a CNN. This strategy is shown in Figure 2. In the upper half of the figure, each repetition of a keyword type (48 in our case) is aligned with an utterance from the untranscribed data using DTW. Subsequently, the lowest cost among the 48 repetitions is determined. This procedure is repeated for all keyword types (40 in our case). The result is a vector of scores for each utterance in the untranscribed corpus. Each dimension of this vector gives an indication of whether the corresponding keyword is present in the particular utterance. These scores are the targets used to train the CNN, as shown in the lower half of Figure 2. The overall approach therefore relies only on a small set of labeled keywords and a large corpus of untranscribed speech.

We now state our approach more formally. Consider a keyword type $\mathcal{K}$ of which we have N repetitions:

$$\mathcal{K} = (k_1, \ldots, k_i, \ldots, k_N) \tag{1}$$

where each k_i is the sequence of speech features for the i^{th} exemplar of keyword $\mathcal{K}$. To obtain the DTW-based score indicating how likely it is that a particular utterance $\mathcal{U}$ contains an instance of keyword $\mathcal{K}$, we calculate:

$$c = \min_{i \in 1 \ldots N} \left[\min_{u_p \in \mathcal{U}} \text{DTW}\{k_i, u_p\} \right] \tag{2}$$

Each u_p is a successive segment of utterance $\mathcal{U}$, and DTW$\{k_i, u_p\}$ is the DTW alignment cost between the speech features of exemplar k_i and the segment u_p. Thus, we determine the relevance of a keyword according to the lowest cost encountered when sweeping each of the exemplars over the current utterance. Assuming we have L keyword types, we calculate equation (2) separately for each of the L keywords. Hence for utterance $\mathcal{U}$ we obtain costs $[c_1, \ldots, c_j, \ldots c_L]$. Since we use the averaged cosine distance in our DTW implementation, the costs $c_j \in [0, 2]$. In order to interpret these costs as probabilities, we apply the normalization $y_j = -\frac{1}{2}c_j + 1$ so that $y_j \in [0, 1]$, with 1 indicating a perfect match and 0 indicating maximum dissimilarity. Finally, we combine all the normalized scores into a single target vector $\boldsymbol{y} = [y_1, \ldots, y_L]$ for utterance $\mathcal{U}$.[2]

Given this target vector, we train a CNN to take $\mathcal{U}$ as input and predict this target vector, as illustrated in the lower part of Figure 2. Our CNN consists of a number of convolutional layers, a global temporal max-pooling layer, and a number of fully connected layers. The global max-pooling layer takes the maximum of the activations over the time dimension, and therefore gives a fixed-dimensional output independent of the length of the input sequence. The intuition is that this would extract the dominant features that are necessary for detecting the presence of a keyword in the utterance. We use leaky ReLU ($\alpha = 1/3$) for all activations [23] except the final feedforward layer which uses a sigmoid activation. Since our scores are normalized to resemble probabilities, we train using the summed binary cross-entropy loss:

$$\ell = -\sum_{j=1}^{L} \{y_j \log \hat{y}_j + (1 - y_j) \log [1 - \hat{y}_j]\} \tag{3}$$

This is the loss for a single training utterance, with $\hat{y}_j$ the prediction of our model for the j^{th} keyword type (this would be a single dimension from the model output). We sum this loss over all M training utterances in the untranscribed corpus. Our CNN model can thus be interpreted as L binary classifiers, one for each keyword, with shared input layers. Finally, the trained CNN can be applied to unseen test utterances, using an appropriate threshold to determined the presence or absence of a keyword.

5. Experiments

5.1. Experimental setup

Three baseline systems are used for comparative evaluation. Firstly, DTW is performed for each exemplar of a keyword and the resulting scores averaged. This corresponds to an established

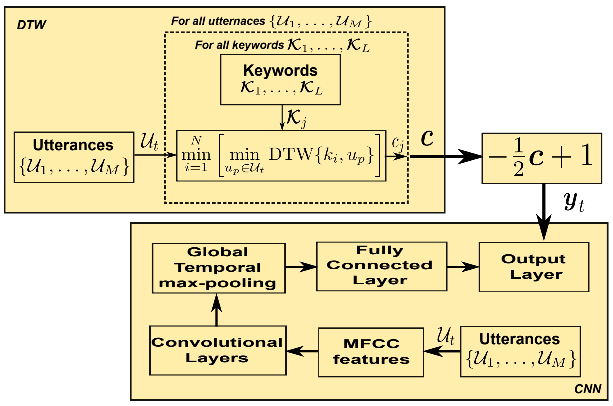

DTW
For all utternaces $\{\mathcal{U}_1, \ldots, \mathcal{U}_M\}$
For all keywords $\mathcal{K}_1, \ldots, \mathcal{K}_L$
Keywords $\mathcal{K}_1, \ldots, \mathcal{K}_L$
$\mathcal{K}_j$
Utterances $\{\mathcal{U}_1, \ldots, \mathcal{U}_M\}$
$\mathcal{U}_t$
$\min_{i=1}^{N} \left[\min_{u_p \in \mathcal{U}_t} \text{DTW}\{k_i, u_p\} \right]$
c_j
$\boldsymbol{c}$
$-\frac{1}{2}\boldsymbol{c} + 1$
$\boldsymbol{y}_t$
Global Temporal max-pooling
Fully Connected Layer
Output Layer
Convolutional Layers
MFCC features
$\mathcal{U}_t$
Utterances $\{\mathcal{U}_1, \ldots, \mathcal{U}_M\}$
CNN

Figure 2: *The CNN-DTW keyword spotter training approach. The top shows how the supervisory signal is obtained and the bottom how this signal is used to train the CNN.*

QbyE method and will therefore be indicated as DTW-QbyE. Second, the minimum (best) score over all exemplars of a keyword is used instead of the average. This will be referred to as DTW-KS. Third, we consider the direct application of a CNN classifier trained only on the isolated words, as described in Section 4.2. These three baseline systems are compared with our proposed CNN-DTW approach.

The CNN's parameters such as the learning rate, number of convolutional and fully connected layers, number of filters in the convolutional layers, number of neurons in the fully connected layers and the dropout probability for regularization were optimized on the development set; this optimisation was performed in terms of the summed binary cross-entropy loss of the DTW targets on the development set, meaning that transcriptions of the SABN data are not used for either training of validation.[3] We train using the Adam optimiser [24] along with an early stopping criterion. Performance is reported in terms of the area under the curve (AUC) of the receiver operating characteristic (ROC). The ROC is a plot of the false positive rate against the true positive rate, as the detection threshold is varied. AUC therefore indicates the performance of the model independent of a threshold, with higher AUC indicating a better model. We also report equal error rate (EER), the point at which the false positive rate equals the false negative rate (thus, lower EER is better).

5.2. Results and Discussions

Table 2 shows the performance for the different systems in terms of AUC and EER averaged over the 40 keywords. From Table 2 we find that the best result is obtained by the DTW-KS configuration which uses the best scores obtained among all exemplars of a keyword. The performance of our proposed algorithm combining CNN and DTW is close to the average result of the DTW-QbyE approach. The performance of the CNN classifier trained only on isolated words is much worse. We found that including a Gaussian noise layer (GNL) between the input and the convolution layers improved the AUC by approximately 1% absolute, showing that this noise layer aids in generalisation.

In a qualitative analysis we found that performance differed significantly between different keyword types.Table 3 therefore analyses the development set performance of the CNN-DTW system on selected keywords. Examples (a), (c) and (e) are

[2]We also considered applying a threshold to obtain hard targets (1 indicating the presence and 0 indicating the absence of the keyword), but this did not improve performance.

[3]Final model: 10 convolutional layers with between 80 and 512 filters per layer, two 3000-unit fully connected layers trained with a dropout of 0.5, and the learning rate is linearly changed from 10^{-4} to 10^{-5}.

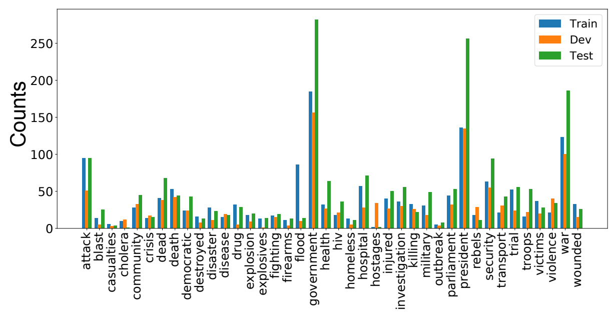


Figure 3: *Keyword occurrence distribution in the SABN corpus.*

Table 2: *Keyword spotting performance on development and test sets with the execution time on the test set in minutes.*

	AUC		**EER**		**Time**
	dev	***test***	***dev***	***test***	**(min)**
CNN	0.5698	0.5448	0.4435	0.4771	55
DTW-QbyE	0.6639	0.6612	0.3864	0.3885	900
DTW-KS	0.7556	0.7515	0.3092	0.3162	900
CNN-DTW	0.636	0.6285	0.4073	0.4161	5
CNN-DTW inc. GNL	0.6443	0.6357	0.4036	0.4092	5

among the best-detected keywords, while (b), (d) and (f) are among the worst. A keyword specific threshold (at the EER) was used for this analysis. As an example, Table 3(a) shows that for the keyword *Government*, at the EER, 93 of the 156 occurrences are correctly detected and 1683 of the 2832 negatives have also been correctly classified. As described in Section 2, we intend to incorporate a human analyst in our overall radio-browsing approach; here we picked a threshold at the EER for the purpose of analysis, but a different threshold could be used depending on how the human analyst would want to balance correct detections and false positives. The ROC plots for each of the keywords in Table 3(a-f) are shown in Figures 4(a-f), and are compared to those of the DTW-KS system. We notice from the plots that the performance of the CNN-DTW system is closer to the DTW baseline for keywords occurring more often in the SABN training set (shown in brackets in Table 3) and for keywords with distinct pronunciations, such as *HIV*. The distribution of the keywords in the SABN train, development and test sets are shown in Figure 3, and we see that *Government* and *War* are among the most frequent words. Note, again, that we never use the transcriptions of these sets during training or validation.

In a practical setting, the computational complexity of running the keyword spotter on live audio is an extremely important consideration. In this regard, our CNN-DTW keyword spotter shows a clear advantage over its DTW-KS counterpart. As indicated in Table 2, the application of the DTW baseline systems for all 40 keywords and over all utterances in the 10-hour test (Table 1) set was approximately 15 hours on a 20-core machine. The CNN-DTW system, on the other hand, can process the same data in approximately 5 minutes on a conventional desktop PC with a single GeForce GTX 1080 GPU. Thus, we are able to process the audio 180 times faster using the CNN-DTW system, making it highly attractive for the cost-effective continuous monitoring of live audio streams.

6. Conclusions

We have shown that, by combining CNNs and DTW, it is possible to obtain an ASR-free keyword spotting system that is

Table 3: *Analysis of the 3 best performing and the 3 worst performing keywords. The number of occurrences of each keyword in the SABN corpus is shown in brackets. The absolute number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) are shown.*

Government (156)

TP: 93	FP: 1149
FN: 63	TN: 1683

(a)

Attack (51)

TP: 25	FP: 1481
FN: 26	TN: 1456

(b)

HIV (21)

TP: 14	FP: 1032
FN: 7	TN: 1935

(c)

Health (27)

TP: 14	FP: 1422
FN: 13	TN: 1539

(d)

War (100)

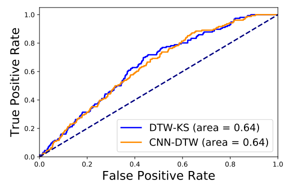
TP: 58	FP: 1222
FN: 42	TN: 1666

(e)

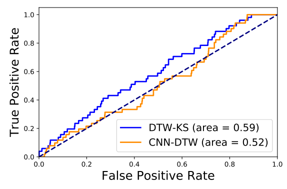
Wounded (15)

TP: 8	FP: 1452
FN: 7	TN: 1521

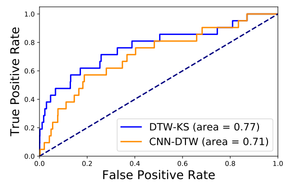
(f)



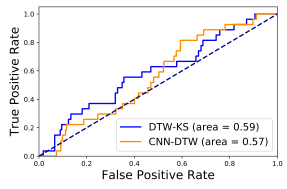
(a) *Keyword: Government*



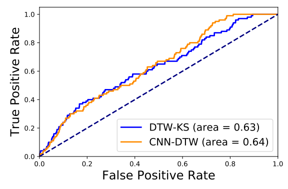
(b) *Keyword: Attack*



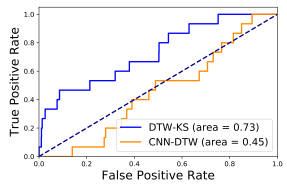
(c) *Keyword: HIV*



(d) *Keyword: Health*



(e) *Keyword: War*



(f) *Keyword: Wounded*

Figure 4: *Receiver operating characteristic plots for selected keywords for the DTW baseline and the proposed systems.*

fast enough for real-time processing. Because it requires only a small set of easily-obtained isolated keywords for training, it is suitable for a low-resource keyword detection scenario in which no further transcribed speech is available. The performance of the proposed system is comparable with the DTW-based query-by-example (QbyE) approaches which is commonly employed in these situations, achieving an AUC of 0.64. Besides requiring minimal resources to train, the proposed system is also computationally much more efficient than its DTW counterpart, and is suitable for the real-time application required by the United Nation's ongoing efforts in humanitarian monitoring.

7. Acknowledgements

We thank the NVIDIA corporation for the donation of GPU equipment used for this research. We also gratefully acknowledge the support of Telkom South Africa.

8. References

[1] S. Vosoughi and D. Roy, "A human-machine collaborative system for identifying rumors on Twitter," in *Proc. ICDMW*, 2015.

[2] K. Wegrzyn-Wolska, L. Bougueroua, and G. Dziczkowski, "Social media analysis for e-health and medical purposes," in *Proc. CASoN*, 2011.

[3] P. Burnap, G. Colombo, and J. Scourfield, "Machine classification and analysis of suicide related communication on Twitter," in *Proc. ACM-HT*, 2015.

[4] R. Menon, A. Saeb, H. Cameron, W. Kibira, J. Quinn, and T. Niesler, "Radio-browsing for developmental monitoring in Uganda," in *Proc. ICASSP*, 2017.

[5] A. Saeb, R. Menon, H. Cameron, W. Kibira, J. Quinn, and T. Niesler, "Very low resource radio browsing for agile developmental and humanitarian monitoring," in *Proc. INTERSPEECH*, 2017.

[6] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks recognition," in *Proc. ICASSP*, 2015.

[7] Y. Zhang, W. Chan, and N. Jaitly, "Very deep convolutional networks for end-to-end speech recognition," in *Proc. ICASSP*, 2017.

[8] M. Larson and G. J. F. Jones, "Spoken content retrieval: A survey of techniques and technologies," *Foundations and Trends in Information Retrieval*, vol. 5, no. 4-5, pp. 235–422, 2012.

[9] A. Mandal, K. R. P. Kumar, and P. Mitra, "Recent developments in spoken term detection: A survey," *International Jour. of Speech Technology*, vol. 17, no. 2, pp. 183–198, 2014.

[10] T. J. Hazen, W. Shen, and C. White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Proc. ASRU*, 2009.

[11] Y. Zhang and J. R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *Proc. ASRU*, 2009.

[12] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. ASRU*, 2013.

[13] Y. Chung, C. Wu, C. Shen, H. Lee, and L. Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *Proc. INTERSPEECH*, 2016.

[14] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. ICASSP*, 2015.

[15] S. Settle and K. Livescu, "Discriminative acoustic word embeddings: Recurrent neural network-based approaches," in *Proc. SLT*, 2016.

[16] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. ICASSP*, 2016.

[17] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end ASR-free keyword search from speech," in *Proc. ICASSP*, 2017.

[18] H. Kamper, F. D. Wet, T. Hain, and T. Niesler, "Capitalising on North American speech resources for the development of a South African English large vocabulary speech recognition system," *Computer Speech and Language*, vol. 28, no. 6, pp. 1255–1268, 2014.

[19] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Trans. Audio, Speech, Language Process.*, vol. 16, no. 1, pp. 186–197, 2008.

[20] A. Jansen and B. Van Durme, "Indexing raw acoustic features for scalable zero resource search," in *Proc. Interspeech*, 2012.

[21] D. Palaz, G. Synnaeve, and R. Collobert, "Jointly learning to locate and classify words using convolutional networks," in *Proc. Interspeech*, 2016.

[22] T. N. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," in *Proc. INTERSPEECH*, 2015.

[23] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

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