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

Rescoring Keyword Search Confidence Estimates with Graph-based Re-ranking Using Acoustic Word Embeddings

Anna Piunova, Eugen Beck, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department,
RWTH Aachen University, 52074 Aachen, Germany

anna.piunova@i6.informatik.rwth-aachen.de, {beck, schlueter, ney}@cs.rwth-aachen.de

Abstract

Postprocessing of confidence scores in keyword search (KWS) task is known to be an efficient way of improving retrieval performance. In this paper, we extend the existing graph-based re-ranking algorithm proposed for KWS score calibration. We replace the originally used Dynamic Time Warping (DTW) distance measure between prospective hits with distances between their Acoustic Word Embeddings (AWEs) learned from Neural Networks. We argue that AWEs trained to discriminate between the same and different words should improve the graph-based re-ranking performance. Experimental results on two languages from IARPA Babel program show that our approach outperforms the DTW and improves the baseline KWS result between 3.0 - 7.5% relative on the Maximum Term Weighted Value (MTWV) measure. It was previously shown, that enhancing detection lists with keyword exemplars given high confidence, improved the algorithm performance. We additionally expanded the detection lists with negative query exemplars and observed further improvements in MTWV.

Index Terms: keyword search, postprocessing, graph-based re-ranking, acoustic word embedding

1. Introduction

A large piece of information consumed by internet users is multimedia content, to a significant part consisting of speech data. Such multimedia data includes video lecture recordings, broadcast news, voice messages, etc. That is why developing technologies for browsing, along with fast and accurate retrieving of speech data from large audio archives is in high demand. Searching for specific concepts, names, discussion topics can be approached via KWS, sometimes also referred to as Spoken Term Detection (STD). KWS is an information retrieval task aimed at detecting all occurrences of a query in a spoken archive. A query is a single word or a sequence of words given to the system in orthographic form $t = \{w_1, ..., w_n\}$.

A typical KWS pipeline consists of an Automatic Speech Recognition (ASR) system, that transcribes the target audio archive into word or subword-based lattices. Lattices are generated and indexed offline, providing a mapping between words in the lexicon and their hypothesized occurrences in the corpus. In our system lattices are converted into time marked word lists [1] (TMWLs) prior to indexing. Given an input query, the retrieval engine performs a search for its occurrences over the generated index and produces a list of prospective hits ranked by confidence scores. The hit confidence score is computed from the arc posterior probabilities of the lattice and denotes the confidence of the KWS system that the query occurrence is met at this time interval. Finally, the system performance is evaluated by adjusting the global decision threshold on development set, so that the KWS evaluation metric is optimized. The hit is accepted if its score is higher than the threshold.

The Term Weighted Value (TWV) [2] is a standard evaluation metric of KWS performance in IARPA Babel program [3]. Given a list of keywords $\tau = \{t_1, ..., t_m\}$ the resulting TWV score is the average of TWV values for each search query. The better the keyword hypotheses found match the actual occurrences, the higher the TWV. As far as all queries equally contribute to the final score regardless of their occurrence, TWV is highly influenced by rare terms. The TWV optimization is also biased towards reducing the number of missed detections (increasing the recall) rather than reducing the number of false alarms.

One of the major problems in KWS is high variability of hit confidence scores across queries, such that the optimal acceptance threshold is different for different hitlists. Moreover, low-resource conditions result in poor recognition performance, which makes the detection scores less reliable in identifying a hit as a correct one or a FA. A low-resource scenario also leads to an increased out-of-vocabulary (OOV) rate. Postprocessing is applied to calibrate confidence scores to overcome the presented limitations.

Graph-based re-ranking was proposed as a score calibration method in [4, 5] and has shown a considerable improvement in KWS retrieval performance on Babel data. Based on the extensions presented in [6, 7], we apply the graph-based re-ranking algorithm to detection lists expanded by training exemplars. The novelty of our approach is a substitution of the DTW matching between detections by distances between their corresponding acoustic embeddings. To the best of our knowledge, the effect of using distances between AWEs instead of DTW has not been studied before. We compared the performance of embedding models trained with different degrees of supervision. We experimented with embedding vectors extracted from the bottleneck layer of a classifier model, a weakly-supervised metric learning model, and a completely unsupervised autoencoder approach. The basic assumption of our approach states that AWEs should better discriminate between instances of the same and different words as far as embeddings are more robust to acoustic variations.

2. Related work

There are multiple approaches of KWS result postprocessing. The goal of system combination [8, 9, 10] is to merge KWS results obtained on lattices generated by different underlying ASR systems. Another group is represented by score normalization methods, such as sum-to-one [9], query length normalization [10], keyword-specific thresholding [11]. Verification methods are aimed at classifying detections as correct ones or FAs and are mostly focused on engineering features for hit classification

http://dx.doi.org/10.21437/Interspeech.2019-1817

models [12, 13, 14, 15]. Finally, another group of postprocessing methods calibrates scores via optimizing the KWS metric [16, 17, 18, 8]. The graph-based re-ranking is an algorithm inspired from Information Retrieval (IR). It was introduced in [4, 5] and applied to the Babel data. The algorithm is aimed at rescoring hits, assuming that hits similar to other hits with high confidences should also be given higher scores. It was proposed in [7] to apply the algorithm on the hitlists enhanced with ground truth detections of the keyword sampled from the training data.

Acoustic Word Embedding is a speech processing task aimed at extracting fixed-dimensional vector representations $v \in \mathbb{R}^D$ of acoustic words (utterances) using their acoustic features $x_1^T = \{x_1, ..., x_T\}$. The goal of the acoustic embedding model is to learn such a function $f_\theta(x_1^T) : \mathbb{R}^{F \times T} \rightarrow \mathbb{R}^D$, that provides a mapping from a variable length feature sequence to a vector. The embedding model is trained to map acoustically similar words into embedding vectors close together in the embedding space. In the context of the AWE task earlier works studied dimensionality reduction methods, such as downsampling and Laplacian eigenmaps[19, 20]. Classifier neural networks for AWEs were deeply investigated in [21, 22]. The authors proposed using CNN [23] and RNN [24] models, trained to predict word identity of the input acoustic sequence. The effect of siamese training [25] was studied there as well. The unsupervised approach using a sequence-to-sequence autoencoder (S2S-AE) was investigated in [26, 27].

3. Re-ranking using Acoustic Word Embeddings

In this section we provide a brief overview of the graph-based re-ranking algorithm and focus more on describing the AWE models as far as applying embeddings in the graph-based re-ranking framework is the main contribution of this work.

3.1. Graph-based re-ranking

The more detailed explanation of the algorithm can be found in [6, 7]. Given a set of search queries the KWS system produces lists of prospective hits for each query. Detections are described by the location (recording), start time, duration and ranked by a confidence score. According to [7] we expanded the hitlists with ground truth detections sampled from the training corpus, given high confidence scores. We only sampled occurrences of single-word queries. Therefore, the number of training hits varies depending on the number of keyword instances in the training data. The detection lists of multi-word queries were re-ranked by only using hypotheses from the hitlist, as was proposed by the original algorithm design in [4, 5].

A similarity graph is built for a hitlist, so that nodes are detections (both retrieved hits and ground truth detections), while edges denote the acoustic similarity between detections. Acoustic similarity is computed between pairs of nodes as

$$S(h_i, h_j; t) = 1 - \frac{d(h_i, h_j) - d_{min}}{d_{max} - d_{min}} \quad (1)$$

where $d(h_i, h_j)$ is a DTW cost of aligning hits h_i and h_j, or a *cosine* distance between corresponding AWEs. Distances d_{min} and d_{max} are minimum and maximum distances within the hitlist. A bidirectional connection is added from each node h_i to its K-nearest neighbours weighted by the acoustic similarity. Edge weight is normalized over the sum of outgoing edges' weights of the source node. The propagation scores $R_g^k(h_i, t)$ are iteratively updated until convergence using the equation

$$\begin{aligned} R_g^k(h_i,t) &= (1-\alpha-\beta)R(h_i,t) \\ &+ \alpha \sum_{h_j \in B(D)_i} R_g^{k-1}(h_j,t)\hat{S}(h_j,h_i) \\ &+ \beta \sum_{h_j \in B(E)_i} R_g^{k-1}(h_j,t)\hat{S}(h_j,h_i) \end{aligned} \quad (2)$$

Here $R(h_i, t)$ is the initial confidence score of hit h_i, $\alpha, \beta \in [0, 1]$ are interpolation weights defining the contribution from query detections and training exemplars respectively, $R_g^0(h_j, t)$ is randomly initialized. Finally, the graph scores are defined as

$$R_g(h_i, t) = R(h_i, t)^{1-\gamma} R_g^k(h_i, t)^\gamma \quad (3)$$

where $\gamma \in [0, 1]$ is an exponentiation parameter, that defines how re-ranking score will contribute to the final confidence.

3.2. Acoustic Word Embedding models

We experimented with AWE models trained with different degrees of supervision considering the low-resource condition. The Classifier model requires the most supervision as it is trained to classify the word identity from a given acoustic feature sequence. The Siamese network does not need word identities, it only needs to know whether two sequences represent the same or different words. The Encoder-decoder models are trained in the style of sequence-to-sequence auto-encoders.

3.2.1. Classifier model

The RNN classifier used in our work is inspired by [22]. The network reads a sequence of acoustic features $x_1^T = \{x_1, x_2, x_3, ..., x_T\}$ corresponding to an acoustic word w frame by frame and outputs a prediction of its word label. The model consists of S bidirectional LSTM [24, 28] (biLSTM) layers as depicted in Figure 1. We used 3-layer biLSTM (800 units per direction) in Mongolian setup and 2-layer biLSTM (2048 units per direction) in Pashto. The concatenation of the last hidden states of the top recurrent layer in both directions $v = [\overrightarrow{h}_T^S, \overleftarrow{h}_T^S]$ is used as the embedding of the input acoustic sequence. It is then forwarded through a feed-forward layer using the softmax activation function to predict the most probable word. Cross-Entropy is used as the loss function. According to [22], increasing the number of stacked recurrent layers along with the feed-forward layers on top leads to improvements in the discriminative power of the embeddings which is supported by the results on word discrimination task [29]. However, in our setups, linear layers on top only decreased both the validation score during training and the graph-based re-ranking results.

3.2.2. Weakly-supervised training using Siamese networks

As far as the desired property of AWEs in the graph-based re-ranking framework is the suitability to discriminate between correct detections and incorrectly retrieved occurrences, we investigated metric learning methods. Siamese network [25] denotes a pair/triplet of tied neural networks, which share parameters. These networks are jointly trained to optimize a common objective. Siamese training allows the use of word pair side information instead of word identities. It is explicitly trained to maximize the distance between embeddings of different words while minimizing the embedding distance of different instances of the same word. The triplet model consists of three jointly trained networks and gets a tuple of inputs (x^a, x^s, x^d) in each training iteration, where the *anchor* sample x^a and *same* sample x^s are sequences with the same word label while the third

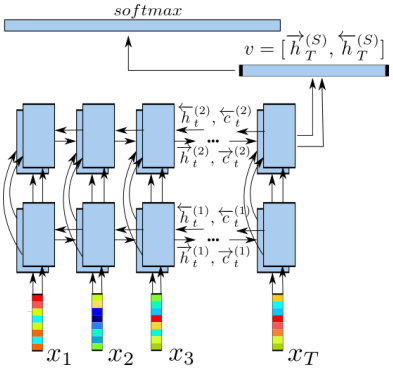

Figure 1: *Classifier network*

input x^d is an instance of a different word. Triplets are built at the beginning of a training epoch so that each word example in a dataset is used as an anchor once.

Motivated by the work in [22] we built 3-layer biLSTM models in both language setups. The network processes input sequences and extracts their AWEs (v_a, v_s, v_d) as concatenated last hidden states of the top recurrent layer. Extracted AWEs are used to optimize the loss function proposed in [21]

$$\mathcal{L}_{tripl} = max\{0, m + d_{cos}(v_a, v_s) - d_{cos}(v_a, v_d)\} \quad (4)$$

where $d_{cos}(v_i, v_j)$ is a cosine distance between embeddings v_i and v_j. However, the large vocabulary resulted in a situation where many word identities never appear to be sampled together in a triplet. The model is not trained to discriminate between them. Based on the ideas from [30] and [31] we implemented a minibatch triplet loss. We concatenate three streams of samples in batch dimension and build all possible same pairs $P_s = \{(v_i, v_j) : v_i, v_j \in B, w(x^i) = w(x^j)\}$ and different pairs $P_d = \{(v_i, v_j) : v_i, v_j \in B, w(x^i) \neq w(x^j)\}$ within a minibatch to maximize the number of training pairs.

$$\begin{aligned} \mathcal{L}_{tripl\_batch} &= \textstyle\sum_{(v_i,v_j)\in P_s} d_{cos}(v_i, v_j) + \\ &+ \textstyle\sum_{(v_i,v_j)\in P_d} max\{0, m - d_{cos}(v_i, v_j)\} \end{aligned} \quad (5)$$

In addition it is possible to combine the Classifier and Siamese models, such that the AWE of the anchor sample is also processed by a feed-forward layer with a softmax at the end.

3.2.3. Sequence-to-sequence autoencoder

All the methods presented so far use some form of information about the word identity, which requires some supervision. Alternatively, it is possible to extract a word representation only relying on the information from the signal.

[26, 27] proposed using a sequence-to-sequence autoencoder to extract representations of audio signals in an unsupervised fashion. The model consists of the RNN encoder $\mathbf{h} = f(x_1^T)$, whose last hidden states $v = [\overrightarrow{h}_T^S, \overleftarrow{h}_T^S]$ are taken as the embedding of the input utterance. This representation v is then used to initialize the decoder hidden state. The decoder $\hat{y} = g(\mathbf{h})$ is an RNN network trained to reconstruct the input feature sequence via optimizing the Mean Squared Error loss.

$$\mathcal{L} = \sum_{t=1}^{T} ||x_t - \hat{y}_t||^2 \quad (6)$$

Motivated by the reported results we also applied sequence-to-sequence autoencoder to extract AWEs. Our model consists of a 2-layer biLSTM encoder (1024 units per direction) and a decoder consisting of LSTM (2048 units) followed by 2-layer biLSTM (1024 units) and a feed-forward layer mapping the output to the feature size.

3.2.4. Encoder-decoder for state sequence classification

Along with reconstructing the sequence of acoustic features we also experimented with an encoder-decoder model, trained to predict the sequence of context dependent phone states for the input acoustic sequence. Our model consists of the encoder, similar to the one in the sequence-to-sequence autoencoder, whose last hidden states are used to initialize the decoder. In contrast to sequence-to-sequence autoencoder, the decoder part here predicts the sequence of tied triphone states. We decided on the model having a 2-layer biLSTM encoder (1500 units per direction) and the decoder consisting of LSTM (3000 units) followed by 2-layer biLSTM (1500 units per direction). The model is trained using the frame-level *cross-entropy* loss.

4. Experiments

4.1. Experimental setup

The Babel Program provided audio corpora for a set of low-resource languages. For KWS postprocessing experiments, we used the Pashto (IARPA-babel104b-v0.4bY) and Mongolian (IARPA-babel401b-v2.0b) Full Language Pack. The training corpora used for training AWEs consist of $\approx 50 - 60$ hours of transcribed telephone conversational speech. The training vocabulary size is 14436 for Pashto and 24054 for Mongolian. The KWS performance and obtained improvements were evaluated on a 10 hour development set with 2065 and 2404 search queries for Pashto and Mongolian.

A detailed description of the KWS system and underlying ASR systems is presented in [32]. All ASR systems use a tandem acoustic model (AM), based on multilingual features, generated by a DNN trained on data from the Babel Program [33] covering 28 languages. The ASR engines primarily differ in the pronunciation lexicon. A bigram language model (LM) was used for lattice generation. The first system uses a lexicon build from the provided training data, we will designate this system as **transcription lexicon**. Another system uses additional textual data extracted by a web crawler, it will be denoted as **web lexicon**. Lattices generated by the **web lexicon** system were rescored with an LSTM LM [34]. These results are referenced in the experiments as **web lexicon with lstm lm**. The retrieval results are postprocessed with sum-to-one normalization [9] before re-ranking.

4.2. Results and Discussion

Results in Table 1 indicate that despite the large vocabulary size AWEs learned by the Classifier model outperform other approaches on IV queries in both Pashto and Mongolian setups. Nonetheless, there is no single method, that consistently outperforms on OOVs for both languages. It is worth to mention, that the Siamese-classifier improvements on transcription lexicon on OOVs are obtained by AWEs extracted by an earlier training epoch of the model, while web lexicon OOVs improvements are obtained by using AWEs extracted by a deeper model (4-layer biLSTM with 800 units per direction). Embeddings learned by a sequence-to-sequence autoencoder result in poor IV performance while they outperform other methods on OOVs in web lexicon and web lexicon lstm lm on Pashto. Training

Table 1: *MTWV improvements obtained by applying the graph-based re-ranking. Detection lists of IV queries are expanded with exemplars sampled from the training data.*

Lexicon	GBRR setup	MTWV IV	MTWV OOV	MTWV Full
Pashto				
transcript.	Baseline	0.4440	0.3005	0.4277
	DTW	0.4743	0.3225	0.4570
	Siamese	0.4646	0.3163	0.4477
	Siamese class.	0.4748	**0.3269**	0.4580
	Class. net.	**0.4769**	0.3260	**0.4597**
	S2S-AE	0.4648	0.3101	0.4477
	Enc-dec	0.4684	0.3172	0.4512
web	Baseline	0.4403	0.2983	0.4323
	DTW	0.4699	0.3177	**0.4613**
	Siamese	0.4590	0.3106	0.4506
	Siamese class.	0.4671	**0.3265**	0.4591
	Class. net.	**0.4700**	0.3159	**0.4613**
	S2S-AE	0.4581	**0.3265**	0.4506
	Enc-dec	0.4644	0.3123	0.4556
web	Baseline	0.4756	0.2665	0.4638
+ lstm lm	DTW	0.4989	0.2718	0.4860
	Siamese	0.4910	**0.2771**	0.4789
	Siamese class.	0.4999	0.2736	0.4871
	Class. net.	**0.5015**	0.2753	**0.4887**
	S2S-AE	0.4902	**0.2771**	0.4781
	Enc-dec	0.4952	0.2753	0.4827
Mongolian				
web	Baseline	0.4832	0.2561	0.4731
	DTW	0.4958	0.2719	0.4858
	Siamese class.	0.4973	0.2696	0.4863
	Class. net.	**0.4974**	**0.2741**	**0.4875**
	S2S-AE	0.4924	0.2696	0.4825
	Enc-dec	0.4965	0.2696	0.4864

setups of all considered AWE models can be requested from the main author.

We observe that DTW distances generally outperform AWEs generated by a sequence-to-sequence autoencoder, Encoder-decoder, and Siamese networks on Pashto. However, graph-based re-ranking using Classifier embeddings resulted in the highest improvements in our experiments. The maximum relative improvements over the first-pass KWS result are 7.5% for the system using transcription lexicon, 6.7% for web lexicon and 5.4% for web lexicon lstm lm. The relative improvement of the setup using Classifier AWEs over the DTW-based rescoring reaches 0.5% on IVs and 2.8% on OOVs. The use of distances between Classifier AWEs in Mongolian setup resulted in 3.0% relative MTWV increase over the first-pass KWS result, while it reached 0.3% improvement over DTW on IVs and 0.8% on OOVs. Considering this, we conclude that Classifier AWEs and DTW show competitive performance when used in the graph-based re-ranking framework. The possible way of improvement is a combination of different models for IV and OOV queries.

We additionally enhanced the detection lists of single-word IV queries with negative exemplars given low confidence scores. Our goal was to sample words, that are acoustically similar to the target keyword, in order to match possible recognition errors. Therefore, if some hit is similar to these negative detections, then it is probably a FA. We randomly sampled instances of words from the training corpus with the smallest character edit distance to the keyword (we used K-nearest words). We varied the number of negative hits from 10 to 300, while optimal number is different for different setups. Table 2 shows the effect of negative keyword samples in detection lists. We observe the consistent improvement on MTWV for all setups except sequence-to-sequence autoencoder evaluated on transcription lexicon system.

Table 2: *MTWV improvements on IV queries obtained by applying the graph-based re-ranking to the results enhanced by both positive and negative exemplars.*

Lexicon	GBRR setup	MTWV Only pos.	MTWV Pos. + Neg.
Pashto			
transcript. lexicon	DTW	0.4743	0.4771
	Siamese class.	0.4748	0.4762
	Class. net.	**0.4769**	**0.4778**
	S2S-AE	0.4648	0.4643
	Enc-dec	0.4684	0.4692
web lexicon	DTW	0.4699	0.4709
	Siamese class.	0.4671	0.4688
	Class. net.	**0.4700**	**0.4715**
	S2S-AE	0.4581	0.4588
	Enc-dec	0.4644	0.4657
web lexicon lstm lm	DTW	0.4989	0.5014
	Siamese class.	0.4999	0.5008
	Class. net.	**0.5015**	**0.5029**
	S2S-AE	0.4902	0.4923
	Enc-dec	0.4952	0.4962
Mongolian			
web lexicon	DTW	0.4958	0.4961
	Siamese class.	0.4973	**0.4987**
	Class. net	**0.4974**	**0.4987**
	S2S-AE	0.4924	0.4930
	Enc-dec	0.4965	0.4975

5. Conclusion

In this work, we propose utilizing distances based on AWEs as a measure of acoustic similarity in graph-based re-ranking framework. The experimental results indicate that the proposed approach relying on Classifier AWEs outperforms the DTW-based counterpart. We also showed that including negative query examples yields further MTWV gains. The total relative improvements on Pashto over the first-pass result vary between 5.6% and 7.6%, while the improvements over the DTW-based result vary between 0.1% and 0.3%. The relative improvement for Mongolian reached 3.3% and 0.5% compared to the first-pass KWS result and to the DTW-based rescoring respectively.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 694537, project "SEQCLAS") and funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 644283. The work reflects only the authors' views and the European Research Council Executive Agency (ERCEA) is not responsible for any use that may be made of the information it contains. Eugen Beck was partly funded by the 2018 Google PhD Fellowship for North America, Europe and the Middle East.

References

[1] L. Mangu, G. Saon, M. Picheny, and B. Kingsbury, "Order-free spoken term detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5331–5335.

[2] J. G. Fiscus, J. Ajot, J. S. Garofolo, and G. Doddingtion, "Results of the 2006 spoken term detection evaluation," in *Proc. sigir*, vol. 7, 2007, pp. 51–57.

[3] "Babel: US IARPA Project," http://www.example.com, 2012-2016.

[4] Y.-N. Chen, C.-P. Chen, H.-Y. Lee, C.-A. Chan, and L.-S. Lee, "Improved spoken term detection with graph-based re-ranking in feature space," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5644–5647.

[5] H.-y. Lee, Y. Zhang, E. Chuangsuwanich, and J. R. Glass, "Graph-based re-ranking using acoustic feature similarity between search results for spoken term detection on low-resource languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[6] N. F. Chen, H. Xu, X. Xiao, C. Ni, I.-F. Chen, S. Sivadas, C.-H. Lee, E. S. Chng, B. Ma, H. Li *et al.*, "Exemplar-inspired strategies for low-resource spoken keyword search in swahili," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 6040–6044.

[7] V. T. Pham, H. Xu, X. Xiao, N. F. Chen, E. S. Chng, and H. Li, "Keyword search using query expansion for graph-based rescoring of hypothesized detections," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*. IEEE, 2016, pp. 6035–6039.

[8] D. Karakos, R. Schwartz, S. Tsakalidis, L. Zhang, S. Ranjan, T. T. Ng, R. Hsiao, G. Saikumar, I. Bulyko, L. Nguyen *et al.*, "Score normalization and system combination for improved keyword spotting," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 210–215.

[9] L. Mangu, H. Soltau, H.-K. Kuo, B. Kingsbury, and G. Saon, "Exploiting diversity for spoken term detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8282–8286.

[10] J. Mamou, J. Cui, X. Cui, M. J. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran *et al.*, "System combination and score normalization for spoken term detection," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8272–8276.

[11] D. R. Miller, M. Kleber, C.-L. Kao, O. Kimball, T. Colthurst, S. A. Lowe, R. M. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[12] T.-W. Tu, H.-Y. Lee, and L.-S. Lee, "Improved spoken term detection using support vector machines with acoustic and context features from pseudo-relevance feedback," in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*. IEEE, 2011, pp. 383–388.

[13] H. Xu, N. F. Chen, S. Sivadas, B. P. Lim, E. S. Chng, H. Li *et al.*, "Discriminative score normalization for keyword search decision," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7078–7082.

[14] D. Xu and F. Metze, "Word-based probabilistic phonetic retrieval for low-resource spoken term detection," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[15] J. Richards, M. Ma, and A. Rosenberg, "Using word burst analysis to rescore keyword search candidates on low-resource languages," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7824–7828.

[16] J. van Hout, L. Ferrer, D. Vergyri, N. Scheffer, Y. Lei, V. Mitra, and S. Wegmann, "Calibration and multiple system fusion for spoken term detection using linear logistic regression," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7138–7142.

[17] Z. Chen and J. Wu, "A rescoring approach for keyword search using lattice context information," in *Proceedings of the Interspeech*, 2017, pp. 3592–3596.

[18] K. Audhkhasi, A. Sethy, B. Ramabhadran, and S. S. Narayanan, "Semi-supervised term-weighted value rescoring for keyword search." in *ICASSP*, 2014, pp. 7869–7873.

[19] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Automatic Speech Recognition and Understanding (ASRU)*, vol. Band, 2013, pp. 410–415.

[20] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[21] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4950–4954.

[22] S. Settle and K. Livescu, "Discriminative acoustic word embeddings: Recurrent neural network-based approaches," *arXiv preprint arXiv:1611.02550*, 2016.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, 1994, pp. 737–744.

[26] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," *arXiv preprint arXiv:1603.00982*, 2016.

[27] C.-H. Shen, J. Y. Sung, and H.-Y. Lee, "Language transfer of audio word2vec: Learning audio segment representations without target language data," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 2231–2235.

[28] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional lstm and other neural network architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.

[29] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[30] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[31] J. Huang, Y. Li, J. Tao, Z. Lian *et al.*, "Speech emotion recognition from variable-length inputs with triplet loss function," in *Proc. Interspeech*, 2018, pp. 3673–3677.

[32] P. Golik, Z. Tüske, K. Irie, E. Beck, R. Schlüter, and H. Ney, "The 2016 rwth keyword search system for low-resource languages," in *International Conference on Speech and Computer*. Springer, 2017, pp. 719–730.

[33] Z. Tüske, D. Nolden, R. Schlüter, and H. Ney, "Multilingual mrasta features for low-resource keyword search and speech recognition systems," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 7854–7858.

[34] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth annual conference of the international speech communication association*, 2012.