OVERCOMING NAWES FIXED-WINDOW CONSTRAINTS FOR SPOKEN TERM DETECTION AND LOCALIZATION

Ramesh Gundluru, Sreekanth Sankala, Venkatesh Parvathala, Sri Rama Murty K

Speech Information Processing Lab
Indian Institute of Technology Hyderabad, India
{ee22m24p000001@, ee20resch11011@, ee22resch01005@, ksrm@ee}.iith.ac.in

ABSTRACT

Neural Acoustic Word Embeddings (NAWEs) have emerged as an efficient alternative to traditional DTW-based spoken term detection (STD) by enabling fast similarity search through fixed-dimensional representations. However, NAWEs suffer from a critical limitation: the collapse of temporal structure into a single vector leads to coarse, window-level decisions, making them ineffective for precise localization. These representations lack the DTW granularity needed to resolve fine phonetic differences and often raise false alarms when encountering acoustically confused terms. To address this, we propose a novel STD framework that extracts frame-level representations of both query and search utterances using an acoustic encoder and aligns them using a cross-attention mechanism. This allows for accurate localization of the query within the utterance and provides a confidence score for detection. Experimental evaluations on both clean and noisy datasets demonstrate significant improvements, with our method achieving an EER as low as 9.02% for in-vocabulary and 6.42% for out-of-vocabulary keywords on clean test data, and 12.95% and 6.74% respectively on noisy test data. Our framework offers a robust and scalable solution for open-vocabulary STD under realistic acoustic conditions.

***Index Terms*—** DTW, Acoustic Word Embeddings, Spoken Term Detection, Cross-attention

1. INTRODUCTION

The rise of multimedia content has led to a surge in unstructured data, making fast and accurate content search essential, especially when no textual metadata is available. Voice search addresses this by detecting and localizing a spoken keyword or phrase directly in audio. Such systems are classified as Keyword Spotting (KWS) or Spoken Term Detection (STD). This work focuses on STD (or Query-by-Example STD), which must handle speaker variability, speaking rate differences, and out-of-domain generalization.

Dynamic Time Warping (DTW) is widely used for aligning the query with the reference and provides timestamps [1, 2]. Its performance depends on the chosen acoustic features, such as spectral, bottleneck, or ASR/GMM-derived posterior features [2,3]. Variants like Segmental DTW [3], Subsequence DTW [4], and Non-segmental DTW [5] have been proposed. However, DTW-based methods remain computationally expensive, lack trainable parameters, are sensitive to noise and speaker variability, and require careful threshold tuning.

An alternative faster solution to DTW-based methods for spoken term detection involves learning the fixed-dimensional vector representations of variable-length spoken word segments, referred to as Neural Acoustic Word Embeddings (NAWEs). Detection is performed by sliding a window over the reference signal by comparing the embeddings of segments to the query by using euclidean or cosine distance as shown in Fig. 1. Several Works have been published in this area to improve the word embeddings [6–10]. Shane Settle et al. trained Long Short Term Memory (LSTM) encoder using a siamese loss for an isolated-word discrimination task [9]. while Wanjia et al. [11] introduced multi-view contrastive loss that maps them to common latent space. It leverages acoustic segment and its corresponding character label sequence in multiple ways of generating triplets to bring the representation into common latent space. This idea was extended to KWS in subsequent work [12–15]. Additionally Kamper et al. and Jacobs et al explored unsupervised methods to get word embeddings. [16, 17].

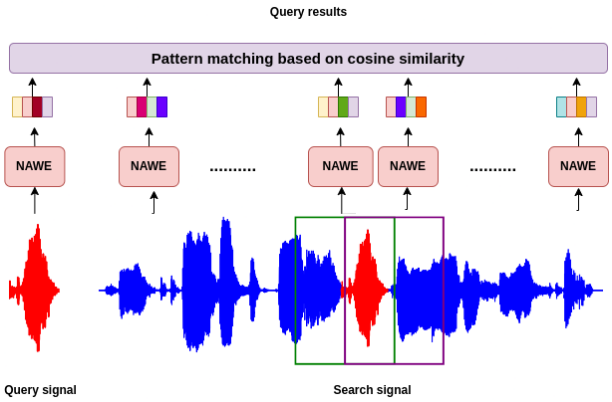

Fig. 1. NAWE framework for STD

NAWEs enable fast comparison, indexing, and capture of semantic and acoustic patterns, making them suitable for low-resource and multilingual data. However, sliding-window search may miss terms crossing window boundaries and produce false alarms due to irrelevant content, requiring careful window-size selection. As a result, NAWEs make coarse window-level decisions and lack precise localization. To address this, multi-hop dot-product attention is used for detection, with attention weights serving as implicit query locations [18]. Still, fixed-dimensional query representations often trigger false alarms and fail to capture fine-grained phonetic differences, motivating the need for phonetic- or frame-level resolution. Motivated by these limitations, we propose a novel approach to STD that combines the granularity of DTW, the generalization capabilities of embeddings, and precise query localization within a single coherent framework.

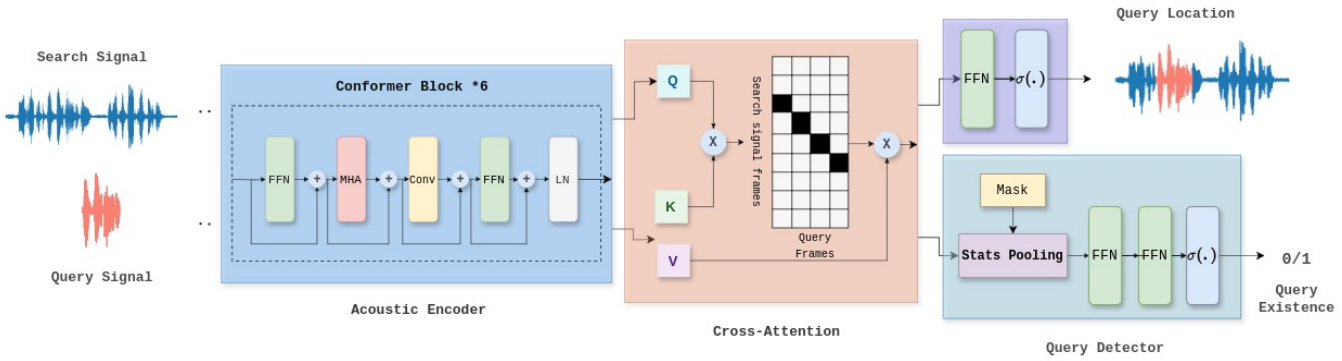

Fig. 2. Architecture of the proposed method.

2. METHODOLOGY

The proposed method consists of 4 modules i.e., Acoustic Encoder, Cross-Attention, Query Localizer, and Query Detector. The below subsections explain all the modules involved in the proposed method.

2.1. Acoustic Encoder

The goal of the acoustic encoder is to extract frame-level embeddings. Self-supervised models like wav2vec 2.0 and HuBERT gives better representations, they are computationally heavy and less suitable for real-time. To keep the system lightweight and deployable. We chose not to compare with large self-supervised models for STD [19–21] to ensure fairness with prior works using similar supervision. Our acoustic encoder is a Conformer encoder and its detailed description is presented in [22]. The acoustic encoder is shared between both the search and query signals. It takes both search and query signal's mel spectrograms and generates the frame level embeddings $\mathbf{S} = \{s_1, s_2, \cdots, s_M\}$ and $\mathbf{E} = \{e_1, e_2, \cdots, e_N\}$ respectively. Where M and N represent the number of frames in the search and query signals respectively. The resultant embeddings are further passed to the cross-attention module to match the query in the search signal.

2.2. Cross Attention

Cross-attention learns the alignment between query embeddings and search signal embeddings. This alignment is achieved by computing attention scores [23]. Let **Q** represent the Queries matrix of size $M \times d_k$, extracted from the search signal embeddings, and **K** and **V** represent the Keys and Values matrices of size $N \times d_k$, extracted from the query embeddings. Here, d_k is the dimension of the Keys. The similarity matrix denoted as **A** is generated by equation (1). The attention output embeddings are generated by multiplying **A** with the values matrix **V** and are given by

$$\mathbf{A} = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}) \tag{1}$$

$$\mathbf{W} = \mathbf{A}\mathbf{V} \tag{2}$$

Multi-head attention with 2 heads was used in the cross-attention module to allow the model to capture contextual information from different perspectives.

2.3. Query Localizer

The output of the cross-attention layer is passed through a logistic regression layer. Each frame-level representation is categorized as either 0 or 1 to predict the precise temporal location of the query. The frame-level Binary Cross Entropy (BCE) is employed at the final layer and is given by

$$L_{loc} = -\frac{1}{LM}\sum_{i=1}^{L}\sum_{t=1}^{M}\Big((1-y_t^i)\log(1-p_t^i) + \gamma \cdot y_t^i \log(p_t^i)\Big) \tag{3}$$

where $y_t^i = \begin{cases} 1 & \text{if query is present in the } t^{th} \text{ frame of } i^{th} \text{ example} \\ 0 & \text{otherwise} \end{cases}$

p_t^i = predicted posterior probability of the t^{th} frame in i^{th} example. L is total number of examples in a batch and M is number of max frames in a batch. and γ is the relative importance that controls the balance between misses and false detections in the loss function. As γ increases, frames labeled 1 contribute more to loss, prioritizing reduction of misses. In contrast, a lower γ reduces false detections by emphasizing frames labeled 0.

2.4. Query Detector

The query localizer produces posterior probabilities for each frame. Direct estimation of query's estimation probability is not straightforward from them. we use an utterance-level classifier to predict whether a query would appear in the search signal or not. We construct masks that focus only on frames corresponding to the query location. During training, these masks are derived from ground-truth query alignments, and the unmasked frame embeddings are averaged to produce a utterance-level representation. This representation is then projected to a single dimension to predict the query's existence probability. At inference time, we generate masks by thresholding the predicted frame-level posteriors, which are used to pool the frame level representations. The loss employed to detect the query is BCE and is given by

$$L_{det} = -\frac{1}{L}\sum_{i=1}^{L}\Big[y^i \log(p^i) + (1-y^i)\log(1-p^i)\Big] \tag{4}$$

where $y^i = \begin{cases} 1 & \text{if query is present in the } i^{th} \text{ example} \\ 0 & \text{otherwise} \end{cases}$

p^i = Predicted posterior probability for i^{th} example.

$$L_{total} = \lambda_1 L_{loc} + \lambda_2 L_{det} \quad (5)$$

The total loss function given by equation (5) includes both localization and detection losses, with weighting factors $\lambda_1 = 1$ and $\lambda_2 = 0.01$ respectively.

3. RESULTS AND ANALYSIS

3.1. Datasets

We conducted our experiments using the LibriSpeech "train-clean-100" dataset for training and the "test-clean" and "test-other" dataset for testing [24]. Precise word-level alignments were extracted using the Montreal Forced Aligner [25]. We evaluate the model under in-vocabulary (IV) and out-of-vocabulary (OOV) scenarios using distinct query sets:

- IV Queries: Keywords appear in both the training and test sets, but the query audio samples are taken from the test set only and were not seen during training. This ensures the model is assessed on its ability to generalize to new acoustic realizations of known terms.
- OOV Queries: Keywords absent from the training set, with their audio samples never seen during training. This ensures the model's open-vocabulary capabilities on completely unseen keywords.

For evaluation, we selected 10 keywords from each category to assess the performances.

3.2. Experimental Setup

The acoustic encoder takes sequence of 128-dimensional mel-filter bank coefficients extracted with 25ms window size at 10ms frame shift. The acoustic encoder comprises six conformer blocks, each with a 512-dimensional feed-forward layer and four self-attention heads. The encoder is shared across both the search signal and query signal. Our training strategy involves randomly selecting a positive query from its occurrences in other training sentences to learn speaker-independent search and generalization. To create negative examples, we randomly choose keywords from the lexicon that are absent in the given utterance, then identify their occurrences in other samples and extract only the corresponding speech query segments. The ratio of positive to negative examples in each batch is maintained at 3:1. All the components in the model are trained from the scratch. We used the Adam optimizer with an initial learning rate of 1e-3, employing a scheduler that reduced the rate to a maximum of 5e-4 within the first 0.2% of 100 epochs and then decayed it further.

3.3. Detection Performance on test-clean dataset

We evaluate our proposed STD system against NAWE baselines using Equal Error Rate (EER) as the metric. Baselines follow a sliding-window approach, segmenting each search utterance into overlapping windows (0.2–0.6s, 50% overlap), computing cosine similarity with query embeddings, and thresholding the maximum similarity for detection. Our method processes the entire utterance in a single pass, avoiding windowing. We reported the results for IV and OOV queries to assess generalization.

As shown in Table 1, NAWE baselines are sensitive to window size: very short windows (0.2s) miss full keywords, while longer windows introduce irrelevant content, degrading performance beyond 0.3s. OOV queries further amplify this effect due to their longer duration, leading to higher error rates. In contrast, our proposed method achieves substantially lower EERs (9.02% IV, 6.42% OOV), consistently outperforming all baselines and demonstrating robust, segmentation-free detection.

In our proposed method, the performance on OOV keywords surpasses that on IV keywords. This seemingly counterintuitive result can be explained by the characteristics of the two sets. The IV set consists of high-frequency keywords with greater variability in their occurrences, making them harder to model consistently. In contrast, the OOV set contains rarer and generally longer keywords, which tend to exhibit less variation and are therefore easier to detect, resulting in higher performance.

Table 1. Performance in terms of Equal Error Rates (EER %) of different systems across vocabulary types and segment sizes for both STD and KWS.

System	**In-Vocabulary (IV)**				**Out-of-Vocabulary (OOV)**			
Segment Size (s)	0.2	0.3	0.4	0.6	0.2	0.3	0.4	0.6
CAE-RNN [16]	28.01	26.96	34.67	53.60	44.71	41.40	33.75	34.52
Siamese-RNN [9]	22.70	21.71	30.00	46.82	39.05	31.01	22.88	27.71
Contrastive-RNN [17]	20.64	20.24	29.11	45.62	38.70	32.56	34.57	30.10
Multi-view-RNN [11]	22.11	22.72	26.78	37.77	42.92	35.51	30.49	32.17
A2E-DWD [10]	37.67	25.45	37.81	48.81	35.21	37.62	32.34	31.42
Proposed	**9.02**				**6.42**			

3.4. Detection Performance on test-other noisy dataset

The table summarizes the EER of various NAWE baseline systems on the noisy test-other dataset, evaluated across in-vocabulary (IV) and out-of-vocabulary (OOV) keywords. The baseline NAWE models exhibit relatively high EERs ranging from approximately 25% to over 46% depending on vocabulary type and segment duration. In contrast, the proposed method significantly outperforms all baselines with a remarkably lower EER of 12.95% for IV keywords and 6.74% for OOV keywords, indicating superior detection accuracy under noisy conditions.

Table 2. Performance in terms of Equal Error Rates (EER %) of different systems across vocabulary types and segment sizes on test other noisy dataset.

System	**In-Vocabulary (IV)**				**Out-of-Vocabulary (OOV)**			
Segment Size (s)	0.2	0.3	0.4	0.6	0.2	0.3	0.4	0.6
CAE-RNN [16]	39.42	36.34	41.38	46.44	35.76	31.66	29.94	40.30
Siamese-RNN [9]	34.34	28.83	36.75	46.04	37.33	29.86	24.59	44.80
Contrastive-RNN [17]	30.35	27.21	25.86	34.00	34.76	35.59	37.69	43.46
Multi-view-RNN [11]	32.73	31.04	29.57	36.20	34.70	28.14	27.51	35.64
A2E-DWD [10]	40.62	29.64	35.27	42.61	47.49	42.25	39.65	48.05
Proposed	**12.95**				**6.74**			

3.5. Visulisation of Cosine Similarity Scores

To analyze the behavior of acoustic word embeddings across an utterance, we plot cosine similarity scores between the query "HERSELF" and each segment of a sentence using two segmentation strategies (Fig. 3). The orange curve shows scores from fixed-size sliding windows (0.4s window, 0.2s hop), while the blue curve uses ground-truth phoneme-aligned word segments from forced alignments. The pink region marks the keyword location, with the blue

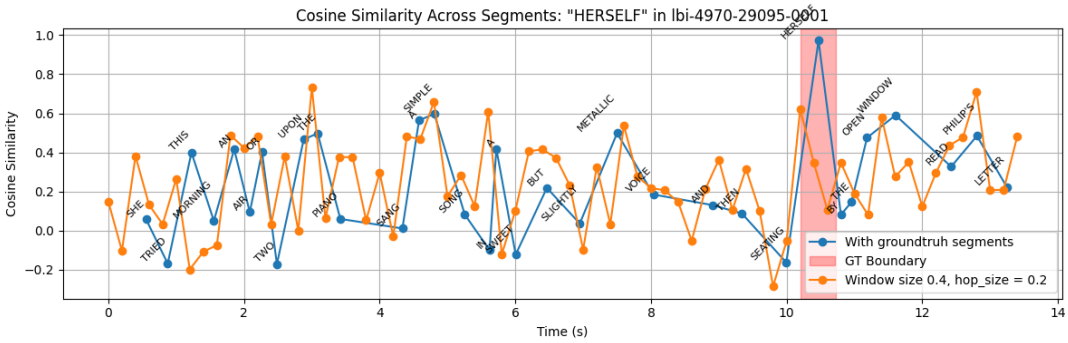

Fig. 3. Visualisation of obtained cosine similarities with fixed size window and ground truth boundaries

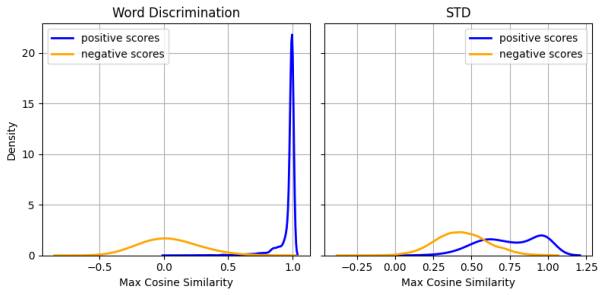

Fig. 4. Kernel Density Estimation (KDE) plots of maximum cosine similarity scores for positive (matching) and negative (non-matching) utterances across using fixed window and ground truth locations

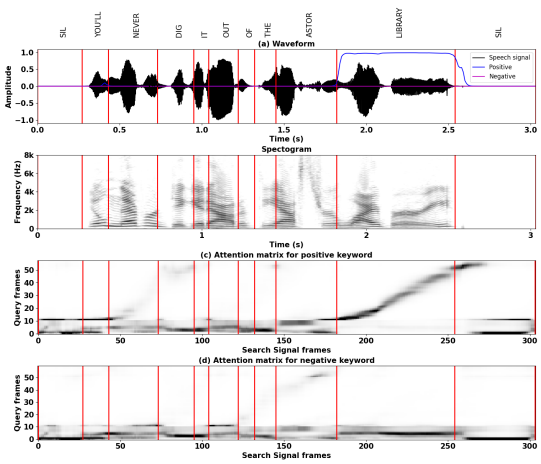

Fig. 5. Visualisation of attention matrices and predicted boundaries for positive and negative examples

dot indicating its midpoint. Several sliding windows overlap with neighboring words, introducing irrelevant acoustic content that suppresses similarity around 3s. In some cases, such as at 10.8s, the true query location is missed entirely, showing that mismatches between fixed window size and query duration can limit detection.

Figure 4 presents the distribution of cosine similarity scores between queries and reference segments for two tasks: Word Discrimination (left) and STD (right). In the Word Discrimination task, reference segments are cropped using ground-truth boundaries, enabling clean word-level comparisons. Positive pairs show high, tightly clustered similarity scores near 1.0, while negative pairs are lower and widely spread, indicating strong discrimination. In the STD setting, search utterances are segmented into overlapping fixed-size windows, which often misalign with the true keyword duration. This causes negative scores to shift right and overlap with positive scores due to surrounding words or noise. Even windows containing the keyword may yield lower similarity from boundary mismatches. Our approach removes windowing by processing the full utterance, enabling direct detection and localization while improving robustness. For a 10-second test utterance, the inference latency on an NVIDIA RTX A4000 is 61.30 ms for the proposed method, compared to 120.1 ms for the Contrastive RNN baseline, demonstrating a significant improvement in runtime efficiency.

3.6. Visualisation of the Attention Matrices.

Fig. 5 (a) illustrates the localization of the speech query 'LIBRARY' within the utterance 'lbi-4970-29093-0000'. The query is picked randomly from its positive instances. Red vertical lines indicate the ground truth word boundaries. The predicted boundary for the positive query 'LIBRARY' is marked in blue, while the predicted boundary for the negative query 'OCCUPATION' is shown in magenta. The model accurately identifies the location of the positive query, while the predicted posteriors for the negative query remain close to zero, confirming its absence. Fig. 5 (b) presents the spectrogram of the search signal. Fig. 5 (c) displays the attention matrix extracted from the cross-attention layer for a positive instance, where a monotonic pattern indicates alignment between the query and the search signals. In contrast, the absence of a diagonal pattern in Fig. 5 (d) further confirms that the query is not present in the utterance.

4. CONCLUSION

Motivated by the limitations of NAWEs for STD, particularly their reliance on fixed sliding windows and suboptimal performance in real-world STD scenarios. We proposed a unified framework that combines the fine-grained alignment capabilities of DTW, the robustness of learned embeddings, and the precision of query localization. Our approach leverages an acoustic encoder and cross-attention mechanism to align frame-level representations of the query and search utterance, enabling end-to-end prediction of both

query presence and exact time boundaries. This eliminates the need for handcrafted windowing and enhances flexibility across varying query lengths and acoustic conditions. Experimental results demonstrate significant improvements in detection accuracy over traditional NAWE-based methods. Our framework presents a promising direction for future research in open-vocabulary STD and offers a scalable solution for real-time speech retrieval applications.

5. REFERENCES

[1] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[2] Timothy J. Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *IEEE Workshop on Automatic Speech Recognition & Understanding*, 2009, pp. 421–426.

[3] Yaodong Zhang and James R. Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *IEEE Workshop on Automatic Speech Recognition & Understanding*, 2009, pp. 398–403.

[4] Meinard Müller, *Information Retrieval for Music and Motion*, Springer, 2007.

[5] Luis J. Rodriguez-Fuentes, Amparo Varona, Mikel Penagarikano, Germán Bordel, and Mireia Diez, "High-performance query-by-example spoken term detection on the sws 2013 evaluation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7819–7823.

[6] Herman Kamper, Weiran Wang, and Karen Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2016, pp. 4950–4954.

[7] Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-Yi Lee, and Lin-Shan Lee, "Audio Word2Vec: Unsupervised Learning of Audio Segment Representations Using Sequence-to-Sequence Autoencoder," in *Interspeech*, 2016, pp. 765–769.

[8] Guoguo Chen, Carolina Parada, and Tara N Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5236–5240.

[9] Shane Settle, Keith Levin, Herman Kamper, and Karen Livescu, "Query-by-Example Search with Discriminative Neural Acoustic Word Embeddings," in *Interspeech*, 2017, pp. 2874–2878.

[10] Pantid Chantangphol, Theerat Sakdejayont, and Tawunrat Chalothorn, "Enhancing word discrimination and matching in query-by-example spoken term detection with acoustic word embeddings," in *International Conference on Natural Language and Speech Processing (ICNLSP)*, 2023, pp. 293–302.

[11] Wanjia He, Weiran Wang, and Karen Livescu, "Multi-view recurrent neural acoustic word embeddings," *CoRR*, vol. abs/1611.04496, 2016.

[12] Niccolò Sacchi, Alexandre Nanchen, Martin Jaggi, and Milos Cernak, "Open-Vocabulary Keyword Spotting with Audio and Text Embeddings," in *Interspeech*, 2019, pp. 3362–3366.

[13] Hyeon-Kyeong Shin, Hyewon Han, Doyeon Kim, Soo-Whan Chung, and Hong-Goo Kang, "Learning Audio-Text Agreement for Open-vocabulary Keyword Spotting," in *Interspeech*, 2022, pp. 1871–1875.

[14] Kumari Nishu, Minsik Cho, and Devang Naik, "Matching Latent Encoding for Audio-Text based Keyword Spotting," in *Interspeech*, 2023, pp. 1613–1617.

[15] Yong-Hyeok Lee and Namhyun Cho, "PhonMatchNet: Phoneme-Guided Zero-Shot Keyword Spotting for User-Defined Keywords," in *Interspeech*, 2023, pp. 3964–3968.

[16] Herman Kamper, "Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6535–3539.

[17] Christiaan Jacobs, Yevgen Matusevych, and Herman Kamper, "Acoustic word embeddings for zero-resource languages using self-supervised contrastive learning and multilingual adaptation," in *IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 919–926.

[18] Chia-Wei Ao and Hung-yi Lee, "Query-by-example spoken term detection using attention-based multi-hop networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6264–6268.

[19] Amit Meghanani and Thomas Hain, "Improving acoustic word embeddings through correspondence training of self-supervised speech representations," in *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics*, 2024, pp. 1959–1967.

[20] Ramon Sanabria, Hao Tang, and Sharon Goldwater, "Analyzing acoustic word embeddings from pre-trained self-supervised speech models," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[21] Jingru Lin, Xianghu Yue, Junyi Ao, and Haizhou Li, "Self-Supervised Acoustic Word Embedding Learning via Correspondence Transformer Encoder," in *Interspeech*, 2023, pp. 2988–2992.

[22] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech*, 2020, pp. 5036–5040.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[24] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2015, pp. 5206–5210.

[25] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, "Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi," in *Interspeech*, 2017, pp. 498–502.