SWAN: SubWord Alignment Network for HMM-free word timing estimation in end-to-end automatic speech recognition

Woo Hyun Kang[*] [†], *Srikanth Vishnubhotla*[*], *Rudolf Braun, Yogesh Virkar, Raghuveer Peri, Kyu J. Han*

AWS AI Labs

{whkang, srikvish, ruseni, yvvirkar, raghperi, kyujhan}@amazon.com

Abstract

End-to-end (E2E) automatic speech recognition (ASR) systems often exploited pre-trained hidden Markov model (HMM) systems for word timing estimation (WTE), due to their inability to predict word boundaries. However, training an HMM is difficult for low-resource languages due to the lack of phonetic transcriptions, leading to a high demand for HMM-free WTE methods, particularly for multilingual ASR systems. In this paper, we propose a novel framework for performing WTE without the need for any HMM or phonetic labels. Specifically, the proposed method trains an alignment network using the outputs of the E2E ASR encoder and a voice activity detection module to generate the frame-level subword labels. In our experiments, the proposed method outperforms previous HMM-free WTE methods in a multilingual scenario. Notably, in the Fleurs dataset, we obtain a relative improvement of 57% over previous work in terms of accumulated averaging shift across 5 languages.

Index Terms: End-to-end speech recognition, word alignment, subword classification, voice activity detection

1. Introduction

Classical hybrid ASR systems adopted the Hidden Markov Model (HMM) framework, where the uttered phone of each speech frame is estimated in a probabilistic manner [1, 2]. The frame-wise training objective inherent to the HMM-based ASR system provides accurate word time stamps, having enabled HMM-based ASR systems to be seamlessly leveraged for various timing sensitive tasks such as keyword spotting and speech segmentation for several decades [3, 4, 5, 6]. However, one of the major limitations of the HMM-based framework is that it requires a phonetically transcribed speech corpus, which makes it harder for ASR models to be trained for low-resource languages where phonetic transcription for audio recordings are not likely available. Moreover, optimizing the HMM-based ASR system is known to be challenging, as it necessitates separate training for multiple individual components (e.g., acoustic model, pronunciation model, language model, etc.) [7].

In recent years, various research projects were sparked on adapting the end-to-end (E2E) framework to the ASR task [8]. Owing to its simplified training pipeline, large model capacity, and ability to emit words (or subwords) directly instead of going through component-wise conversions (e.g., phone sequences to word sequences), the E2E framework overcame the aforementioned limitations of the HMM-based ASR systems and significantly outperformed them in terms of ASR performance. The E2E ASR system generally follows the encoder-decoder approach, where the encoder network extracts a frame-level latent embedding from the input acoustic features and the decoder network generates the text output given the encoder representations [9, 10, 11]. One of the most widely used training methods for the E2E ASR system is connectionist-temporal-classification (CTC)/attention training, which jointly optimizes the CTC objective of the encoder and the maximum likelihood criterion of the attention-based decoder [12, 13, 14].

One crucial downside of the E2E ASR is that it cannot innately estimate the word boundary accurately. Therefore, many conventional E2E systems tried to emulate the HMM-based timings by training an E2E system with the output of an HMM-based ASR system [15] or by training an auxiliary network with phonetic transcriptions [16, 17]. These methods are known to provide reliable word timing estimation (WTE) performance, but it is often difficult to find a pre-trained HMM system or phonetically labeled training corpus, especially for low-resource languages. Since these limitations make it difficult to build E2E ASR systems for low-resource languages or multilingual scenarios, there has been a growing interest in the ASR community to develop an HMM-free WTE method. In [18], the authors proposed to take the CTC layer output probability and directly apply forced alignment there to compute the subword-level boundaries. A more recent work [19] also used this CTC alignment method to obtain word boundaries for multiple languages. The authors in [20] proposed to add a silence token on the CTC alignment results based on the frame-level energy of the features. On the other hand, [21, 22, 23] tried to obtain the word boundaries by manipulating the CTC peaks directly. Although the CTC-based WTE methods can scale across multiple languages attributed to their independence from the phonetic labels, they often fall short, particularly in terms of word end time estimation, as the CTC does not have any indicator for between-word silence.

In light of this, we propose a new WTE method, namely SWAN, that trains an HMM-free subword alignment network (SWAN) to generate frame-level subword labels accurately. More precisely, SWAN is a frame-level classification model that is trained using the frame-level alignment labels generated from the CTC layer and voice activity detection (VAD) outputs. Thus, SWAN is optimized to identify whether each frame belongs to a specific subword or contains no speech (i.e., silence). Moreover, contrary to the CTC outputs containing an ambiguous *blank token*, which can either be the same token as the one in the preceding frames or silence, the proposed SWAN employs an explicit *silence token* to remove such uncertainty. Unlike the conventional approach [20] which computes the *CTC tokens* and *silence token* labels separately, our proposed method estimates all labels in a jointly manner to

[*]These authors contributed equally to this work

[†]Corresponding author: whkang@amazon.com

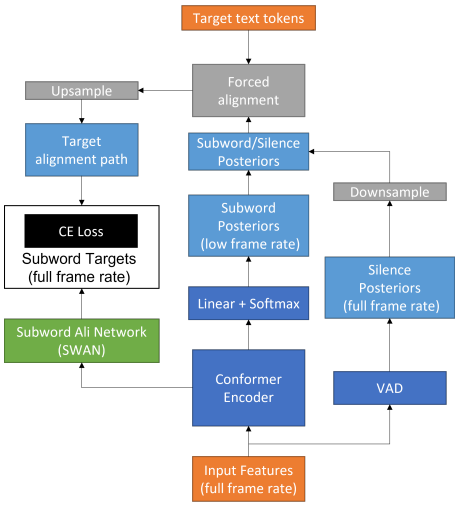


Figure 1: *SWAN training process.*

generate a more natural word duration. To evaluate the WTE capability of the proposed method, we conducted a set of experiments using the VoxPopuli [24] and Fleurs [25] datasets, and the proposed SWAN generated more accurate word boundaries than other HMM-free WTE methods. Specifically, in the Fleurs evaluation, the proposed method was able to outperform the CTC alignment [18] with a relative improvement of 57% in terms of accumulated averaging shift (AAS), which will be defined in Section 4.1.

2. Background

2.1. Limitations of CTC timings for WTE

The CTC loss is formulated to optimize the E2E ASR encoder to emit the correct tokens in the right order [26]. Although this is often sufficient for achieving optimal word accuracy, it introduces various limitations in terms of WTE. One of the most crucial issues is the spiky behavior of the CTC probability. Since the CTC layer is trained on the target π, a blank-inserted token sequence [26], the CTC output tends to exhibit spiky behavior, with series of subword token spikes separated by a large number of predicted blank tokens. As the blank token can be either a subword or silence, which specifically makes end timing estimation confusing, it is difficult to infer the subword duration from the CTC probability [23].

2.2. CTC forced alignment algorithm

The most straightforward way to obtain the word boundary from a CTC-based E2E model is to perform forced alignment on the CTC probability directly [18]. To do this, the CTC probabilities are first mapped into a trellis diagram which ensures that the trellis path takes either a blank token (i.e., staying in the same token as before) or the next subword token. Once the trellis diagram is computed, subword-level alignment is performed by backtracking on the trellis.

Since the CTC subword alignment algorithm assigns a subword token label to every frame excluding blanks, it provides word timing without the spiky behavior of the CTC probability. However, since this does not account for the silence between words, its accuracy is still limited in terms of subword duration estimation.

Algorithm 1 SWAN training process

Freeze ASR model Θ_{ASR} and VAD model Θ_{VAD}
Initialize SWAN parameter Θ_{SWAN}
for i **in** $(1, ..., ep_{max})$ **do**
  Generate SWAN output $\hat{y}$
  Generate CTC probability p_{ctc} from Θ_{ASR}
  Generate VAD silence probability p_{vad} from Θ_{VAD}
  Obtain frame location of CTC spikes $z_1, ..., z_M$ from ground truth token list $L = [l_1, ..., l_M]$
  for m **in** $(1, ..., M-1)$ **do**
    if $p_{vad}(n) > \tau_{sil}$ for any $n \sim (z_m, ..., z_{m+1})$ **do**
      Insert "silence token" between l_m and l_{m+1} in L
    end if
  end for
  Concatenate p_{ctc} and p_{vad} to create p_{comb}
  Perform DP on p_{comb} given L to create alignment y_{low}
  Upscale y_{low} to create y_{high}
  Compute cross-entropy loss between y_{high} and $\hat{y}$
  Update Θ_{SWAN}
end for

3. SWAN: SubWord Alignment Network

In order to overcome the limitations of the CTC-based word timings noted in Sections 2.1 and 2.2, we propose a learnable subword timing module called SWAN. The proposed SWAN is a network that takes the last layer representation from an E2E ASR encoder as input and outputs the frame-level probability of each subword token. More specifically, SWAN is composed of transposed convolutional layers to produce alignments with high temporal resolution. Note that during the SWAN model training and fine-tuning, the ASR model parameters are frozen. The general framework for SWAN training is depicted in Figure 1 and Algorithm 1.

To ensure that the word alignment accounts for the silence regions, we use the CTC probability p_{ctc} in combination with the silence probability p_{vad} obtained from a voice activity detection (VAD) model while training the SWAN model. The two probabilities are concatenated to create p_{comb}, which has a size of $[N \times (C+1)]$ where N is the number of frames and C is the number of subword tokens. The p_{comb} essentially introduces an additional "silence token" to the CTC output, where its probability is obtained from the VAD.

Since the original ground truth tokens $L = [l_1, ..., l_M]$ only consist of "speech" tokens appearing in the CTC layer, we need to introduce the silence token in between them at appropriate locations. To determine these locations, we check the CTC spiking frame location z_m for each token l_m, and insert the silence token between l_m and l_{m+1} if $p_{vad}(n) > \tau_{sil}$ for any n between z_m and z_{m+1}.

Given p_{comb} and the silence-augmented L, the frame-level subword alignment y_{low} is generated by performing the dynamic programming process described in Section 2.2. Alignment y_{low} has the same frame rate as the CTC output of the ASR encoder, which typically yields WTE resolution lower than desired due to the encoder front-end downsampling process. Therefore, y_{low} is upsampled to match the framerate of the SWAN output and the input acoustic feature. Finally, the SWAN model is trained by minimizing the cross-entropy loss between y_{high} and the predicted SWAN output $\hat{y}$, given the last layer representation from the ASR encoder as input.

4. Experiments

4.1. Experimental setup

We validated the WTE performance of SWAN on multiple languages through a set of experiments conducted using the VoxPopuli [24] dataset. The training was done on 1,566 hours of the

Table 1: *AAS performance comparison between the proposed method and other CTC-based WTE techniques on the VoxPopuli and Fleurs eval sets. (ms)*

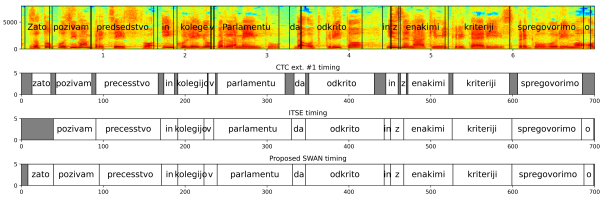
VoxPopuli test						
Languages	*de*	*es*	*it*	*en*	*fr*	*avg.*
CTC align [18]	77.8	67.6	71.3	81.6	64.1	72.5
ITSE [20]	51.6	58.1	44.9	53.6	53.0	52.2
CTC ext. #1 [21]	57.5	49.2	54.3	59.3	46.9	53.4
CTC ext. #2 [21]	96.3	91.6	97.1	97.4	96.9	95.8
SWAN (Proposed)	47.2	57.7	42.5	54.1	52.4	**50.8**
Fleurs test						
Languages	*de*	*es*	*it*	*en*	*fr*	*avg.*
CTC align [18]	67.0	138.5	128.5	76.1	175.1	117.0
ITSE [20]	67.6	52.6	53.7	77.3	44.8	59.2
CTC ext. #1 [21]	82.1	115.9	61.2	53.5	162.4	95.0
SWAN (Proposed)	54.0	44.7	38.4	66.9	45.0	**49.8**

"train" partition of the VoxPopuli dataset, which consists of 16 different languages. The evaluation was done on the "test" partitions of the VoxPopuli (in-domain) and Fleurs (out-of-domain) [25] datasets. Among the 16 languages, for 5 high-resource languages (i.e., *en*, *de*, *es*, *it*, *fr*), we took the word timestamps generated from their respective monolingual HMM systems and used them as ground truth word timing. Given these HMM timings, we computed the accumulated averaging shift (AAS) [27] for overall WTE performance comparison, which is the average absolute delta between ground truth and the hypothesized word start and end times. In addition, we also computed the separate absolute start time deltas (ASTD) and the absolute end time deltas (AETD) for analyzing the effect of *silence token* on the word boundaries. Lower AAS indicates that the estimated word boundaries are similar to the HMM timestamps, while lower ASTD and AETD indicate closer word start and end times to the HMM, respectively. Additionally, we have selected 3 languages with less than 10 hours of training data (i.e., *sl*, *et*, *lt*) for a qualitative inspection of the word boundaries.

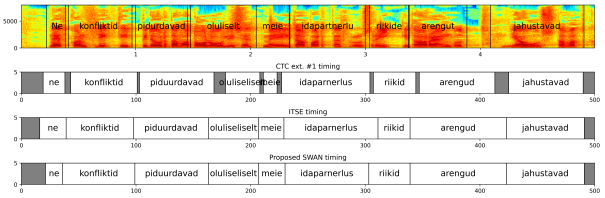
The encoder of the ASR model used in our experiments consists of a convolutional layer with a downsample scale of 4, followed by 16 conformer blocks [28] with 8 attention heads and a dimensionality of 512, while the decoder was composed of a single conformer block. On top of the encoder network, CTC and language classifier modules are placed, where each module consists of a single linear layer. The ASR model is optimized with CTC, attention decoder, and language classification loss, and a total of 4,416 subword tokens were used as the target classes for the ASR model, which is generated using byte-pair encoding (BPE) [29, 30]. For the SWAN, we used two 1D transposed convolution layers with stride of 2, followed by a linear layer and trained the alignment network with $\tau_{sil} = 0.5$ for silence insertion. The ASR model and alignment network were trained using the same training set for 40 and 10 epochs, respectively.

4.2. Comparison with other CTC-based WTE methods

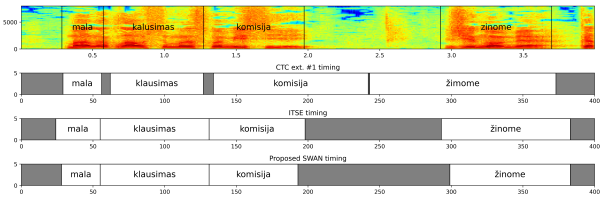
In this section, we compare the WTE performance of the proposed SWAN model and other conventional CTC-based WTE techniques. More precisely, we compare the CTC spike extend method [21, 22] (CTC ext.) and the CTC alignment method [18] (CTC align), along with ITSE [20], which is another learnable method that employs $silence\ token$. The CTC align method is described in Section 2.2 and was used by popular ASR toolkits such as ESPnet [31], NeMo [32], Speechbrain [33], or multilingual ASR systems like MMS [19]. The ITSE method [20]



(a) *Slovenian (sl)*



(b) *Estonian (et)*



(c) *Lithuanian (lt)*

Figure 2: *Word boundary plots of 3 lowest-resource languages, generated by CTC extension, ITSE, and SWAN. (In each figure, first: spectrogram of the audio and the human-transcribed word timestamps, second: word boundaries from CTC ext. #1, third: word boundaries from ITSE, fouth: word boundaries from SWAN). Grey regions indicate silence.*

is a technique that introduces $silence\ token$ to the CTC-based alignment framework, similarly to the proposed method. However, there is one critical difference between ITSE and SWAN in how the $silence\ token$ is inserted into the alignment network target label. Unlike the proposed SWAN that performs alignment on the combined silence and CTC probabilities, the ITSE creates the targets for training the alignment model in a two-stage fashion, which first performs CTC align [18] and inserts the silence labels on the CTC align results. While the original ITSE uses a simple feature energy thresholding method to insert silence, for a fair comparison with the proposed method, we utilized the VAD that was used to train the SWAN for inserting silence in our ITSE experiments. The CTC ext. [21, 22] heuristically estimates the word timing by extending the subword boundaries from the CTC spike with fixed hyperparameters α_{left} and α_{right} which control the weight for the distance between CTC spikes (e.g., larger α_{left} indicates earlier start time and larger α_{right} indicates later end time). For our experiments, we used 2 different CTC ext. hyperparameters: 1) CTC ext. #1 using $\alpha_{left} = 0.2$ and $\alpha_{right} = 0.7$, 2) CTC ext. #2 using $\alpha_{left} = 0.7$ and $\alpha_{right} = 0.2$.

Table 1 shows the AAS performance of the conventional CTC-based methods and the proposed SWAN. Static methods like CTC align or CTC ext., which rely entirely on the artifacts created from the pre-trained ASR encoder (e.g., CTC probability), showed generally worse performance compared to the learnable techniques (i.e., ITSE or SWAN). For example, on the VoxPopuli test set, ITSE outperformed CTC align by 28% relative improvement. The CTC ext. method was able to yield good performance with $\alpha_{left} = 0.2$ and $\alpha_{right} = 0.7$ (CTC

Table 2: *WTE performance of SWAN trained with and without VAD probability (ASTD, AETD, AAS in ms) on the VoxPopuli eval sets.*

	Languages	*de*	*es*	*it*	*en*	*fr*	*avg.*
ASTD	CTC align [18]	82.4	77.3	65.2	89.4	66.1	76.1
	SWAN CTC	48.1	60.3	42.3	53.7	51.8	51.2
	+ VAD	45.7	58.6	40.5	52.9	49.4	**49.4**
AETD	CTC align [18]	73.2	57.9	77.4	73.7	62.0	67.8
	SWAN CTC	75.2	62.6	77.9	73.0	67.6	71.3
	+ VAD	48.7	56.7	44.5	55.2	55.3	**52.1**
AAS	CTC align [18]	77.8	67.6	71.3	81.6	64.1	72.5
	SWAN CTC	61.7	61.5	60.1	63.4	59.7	61.3
	+ VAD	47.2	57.7	42.5	54.1	52.4	**50.8**

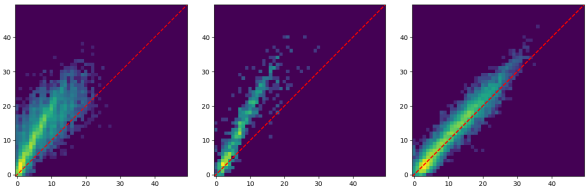
ext. #1) achieving the best AAS in *fr* and *es*, but this did not scale well to other languages like *de* or *en*. Moreover, changing the α_{left} and α_{right} values (CTC ext. #2) resulted in catastrophic degradation, highlighting the instability of static WTE methods like CTC ext. Such instability issue was more evident in out-of-domain evaluation, as AAS degraded heavily on the Fleurs test set. The AAS metrics of the static methods (i.e., CTC align, CTC ext.) on the Fleurs test set were much worse than on the VoxPopuli test set. Specifically, CTC ext. #1 experienced a relative degradation of 77% on Fleurs compared to VoxPopuli in terms of average AAS. On the other hand, learnable methods (i.e., ITSE or SWAN) were able to show similar AAS performance between the VoxPopuli and Fleurs test sets.

Such performance gap can also be seen from the lower resource languages. Figure 2 depicts the word boundaries obtained by CTC ext. #1, ITSE, and the proposed SWAN for the 3 lowest resource languages during our training (i.e., sl, et, lt). From the figures, we can see that the CTC ext. #1 method suffers from over-fragmentation, where silence appears too often between two word timestamps. This is mainly due to the static nature of the CTC ext. method, as it simply extends the right and left side of the CTC spike without knowing if there is a silence region between two spikes. In the Estonian example of Fig. 2 for instance, we can observe that there exist numerous spaces between words in the CTC ext. result, while there are no silence regions visible from the spectrogram except for the beginning and the end of the audio. On the other hand the learnable methods (i.e., ITSE or SWAN) did not experience this issue, as they were trained to determine the silence regions via the $silence\ token$. Particularly, the ITSE and SWAN did not insert unnecessary silence regions to the alignment in the Estonian example of Fig. 2.

The proposed SWAN was able to achieve the best overall performance across all experimented systems on both VoxPopuli and Fleurs, outperforming the ITSE and CTC align by relative improvement of 15.9% and 57.4%, respectively in terms of average AAS on the Fleurs test sets. Furthermore, while the ITSE showed degraded performance on Fleurs compared to the VoxPopuli dataset, the proposed method did not suffer any degradation. These results tell us that not only the proposed method can yield reliable WTE performance across different languages, but also it can generalize well to unseen domains. From Figure 2, we could see that the proposed SWAN can scale well to low resource languages.

4.3. Effect of VAD on SWAN performance

In this section, we analyze the impact of using VAD probabilities during SWAN training. In Table 2, we report the results from SWAN trained with and without the VAD probability, along with the results from the CTC align method [18] as a baseline.



(a) *CTC align [18]* (b) *SWAN CTC* (c) *SWAN CTC+VAD*

Figure 3: *2D histograms where the x-axis is the ground truth word duration and the y-axis is the estimated duration for the corresponding words in the Italian test set of VoxPopuli. The diagonal red line indicates the ideal case where the estimated and the ground truth duration are identical.*

Both CTC align and SWAN trained with only CTC probabilities (i.e., SWAN CTC) perform word alignment using only the CTC tokens. However, unlike CTC align method that performs alignment directly on the low frame-rate CTC output, the SWAN CTC method performs alignment on the high frame-rate SWAN output, which may yield more fine-grained word boundaries. Attributed to this characteristics, the SWAN CTC was able to yield much better word start times than CTC align, achieving a relative improvement of 32.7% in terms of ASTD. However, no improvement was observed in terms of AETD, likely due to the lack of an indicator for silence regions.

Training the SWAN with both VAD and CTC probabilities (i.e., SWAN CTC + VAD) was able to yield much better word timing than using CTC probability alone (i.e., SWAN CTC), outperforming in both ASTD and AETD. The largest improvement was observed for AETD, which highlights the importance of $silence\ token$ for accurate word end time estimation. More specifically, the SWAN CTC + VAD achieved a relative improvement of 23.2% and 26.9% in terms of AETD compared to CTC align and SWAN CTC, respectively. The impact of VAD probability on the SWAN's capability to estimate accurate word duration can be seen in Figure 3. From the 2D histograms, we can see that the CTC align and SWAN CTC method tend to hypothesize the words to have longer duration than the ground truth. This is likely due to their inability to correctly determine the word ending, as they do not know if an audio region is silence or not. On the other hand, the SWAN CTC+VAD was able to accurately estimate the word duration, closely aligned to the ground truth duration.

5. Conclusion

In this paper, we proposed a novel learnable WTE method for the E2E ASR system that trains an auxiliary SWAN using the probabilities outputted by the VAD and the CTC layer of the ASR model. Since the proposed SWAN does not rely on any phonetic transcriptions, it can be used to predict word boundaries in ASR systems even for low-resource languages. To demonstrate this, we performed a set of experiments on the VoxPopuli dataset, which consists of 16 labeled languages. From our experiments, the proposed SWAN outperformed all other systems by +57% relative in terms of the mean AAS metric.

In our future work, we will explore various methods to reduce SWAN's dependency on the CTC probability. Moreover, we will also assess different types of metrics (e.g., word coverage) to compare SWAN against other word alignment methods.

6. References

[1] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, 1993.

[2] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Proc. ASRU*, 2013, pp. 273–278.

[3] S. Sigtia, R. Haynes, H. B. Richards, E. Marchi, and J. S. Bridle, "Efficient voice trigger detection for low resource hardware," in *Proc. Interspeech*, 2018.

[4] A. Shrivastava, A. Kundu, C. Dhir, D. Naik, and O. Tuzel, "Optimize what matters: Training dnn-hmm keyword spotting model using end metric," in *Proc. ICASSP*, 2021, pp. 4000–4004.

[5] K. Gorman, H. Jonathan, and W. Michael, "Prosodylab-aligner: A tool for forced alignment of laboratory speech," in *Canadian Acoustics*, 2011, pp. 192–193.

[6] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Proc. Interspeech*, 2017.

[7] J. Li *et al.*, "Recent advances in end-to-end automatic speech recognition," *Proc. APSIPA*, vol. 11, no. 1, 2022.

[8] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schluter, and S. Watanabe, "End-to-end speech recognition: A survey," *ArXiv*, vol. abs/2303.03329, 2023.

[9] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016.

[10] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NeurIPS*, 2015, p. 577–585.

[11] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Proc. ICASSP*, 2016, pp. 4945–4949.

[12] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proc. ACL*, Jul. 2017, pp. 518–529.

[13] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[14] H. Miao, G. Cheng, P. Zhang, and Y. Yan, "Online hybrid ctc/attention end-to-end automatic speech recognition architecture," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1452–1465, 2020.

[15] J. Mahadeokar, Y. Shangguan, D. Le, G. Keren, H. Su, T. Le, C.-F. Yeh, C. Fuegen, and M. L. Seltzer, "Alignment restricted streaming recurrent neural network transducer," in *Proc. SLT*, 2021, pp. 52–59.

[16] R. Zhao, J. Xue, J. Li, W. Wei, L. He, and Y. Gong, "On addressing practical challenges for rnn-transducer," in *Proc. ASRU*, 2021, pp. 526–533.

[17] M. Bain, J. Huh, T. Han, and A. Zisserman, "Whisperx: Time-accurate speech transcription of long-form audio," in *Proc. Interspeech*, 2023.

[18] L. Kürzinger, D. Winkelbauer, L. Li, T. Watzel, and G. Rigoll, "Ctc-segmentation of large corpora for german end-to-end speech recognition," in *Proc. Speech and Computer*, 2020, pp. 267–278.

[19] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, A. Baevski, Y. Adi, X. Zhang, W.-N. Hsu, A. Conneau, and M. Auli, "Scaling speech technology to 1,000+ languages," *ArXiv*, vol. abs/2305.13516, 2023.

[20] R. Yang, G. Cheng, P. Zhang, and Y. Yan, "An e2e-asr-based iteratively-trained timestamp estimator," *IEEE Signal Processing Letters*, vol. 29, pp. 1654–1658, 2022.

[21] X. Chen, H. Ni, Y. He, K. Wang, Z. Ma, and Z. Xie, "Emitting Word Timings with HMM-Free End-to-End System in Automatic Speech Recognition," in *Proc. Interspeech*, 2021, pp. 2571–2575.

[22] X. Chen, Y. Y. Lin, K. Wang, Y. He, and Z. Ma, "Improving frame-level classifier for word timings with non-peaky ctc in end-to-end automatic speech recognition," *ArXiv*, vol. abs/2306.07949, 2023.

[23] A. Zeyer, R. Schlüter, and H. Ney, "Why does ctc result in peaky behavior?" *ArXiv*, vol. abs/2105.14849, 2021.

[24] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proc. ACL*, 2021, pp. 993–1003.

[25] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna, "Fleurs: Few-shot learning evaluation of universal representations of speech," *ArXiv*, vol. abs/2205.12446, 2022.

[26] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, p. 369–376.

[27] X. Shi, Y. Chen, S. Zhang, and Z. Yan, "Achieving timestamp prediction while recognizing with non-autoregressive end-to-end asr model," *ArXiv*, vol. abs/2301.12343, 2023.

[28] A. Gulati, C.-C. Chiu, J. Qin, J. Yu, N. Parmar, R. Pang, S. Wang, W. Han, Y. Wu, Y. Zhang, and Z. Zhang, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020.

[29] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proc. ACL*, 2016, pp. 1715–1725.

[30] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proc. EMNLP*, Nov. 2018, pp. 66–71.

[31] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[32] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook, P. Castonguay, M. Popova, J. Huang, and J. M. Cohen, "Nemo: a toolkit for building ai applications using neural modules," *ArXiv*, vol. abs/1909.09577, 2019.

[33] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "Speechbrain: A general-purpose speech toolkit," *ArXiv*, vol. abs/2106.04624, 2021.