INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

# A Neural Time Alignment Module for End-to-End Automatic Speech Recognition

*Dongcheng Jiang*[1], *Chao Zhang*[2], *Philip C. Woodland*[1]

[1]Department of Engineering, University of Cambridge, Trumpington St., Cambridge, CB2 1PZ UK.
[2]Department of Electrical Engineering, Tsinghua University, Beijing, 100084, P. R. China

dj346@eng.cam.ac.uk, cz277@tsinghua.edu.cn, pcw@eng.cam.ac.uk

## Abstract

End-to-end trainable (E2E) automatic speech recognition (ASR) systems have low word error rates, but they do not model timings or silence by default unlike hidden Markov model (HMM)-based systems. In this paper, an extra neural aligner module is proposed for E2E ASR models, which labels the word timings in a post-processing stage. Pre-trained neural transducer and attention-based encoder-decoder models are adopted as the ASR backbones for experiments. The aligner module uses self-attention and cross-attention and takes the hidden representations from the backbone to predict the durations of each word and the possible silences. A novel loss is proposed for aligner training with the backbone frozen. Experimental results showed that when trained using the references from an existing HMM-based forced aligner, the proposed methods can make time predictions at accuracy about 95% for matched recognised words, and about 99% for utterances up to 10 s with reference text, with 200 ms tolerance.

**Index Terms**: End-to-end speech recognition, forced alignment, duration modelling, normalised duration

## 1. Introduction

Automatic speech recognition (ASR) systems that use deep neural networks have achieved low word error rates (WERs) and are widely used. Conventional ASR systems combine hidden Markov models (HMMs) and neural networks as acoustic models, along with a separate language model, and are known as hybrid systems. In recent years, there has been a significant amount of research into end-to-end trainable (E2E) ASR systems, which come in two main types: neural transducer (NT) models [1] and attention-based encoder-decoder (AED) models. NT models consist of an encoder network for acoustic features, a prediction network for text, and a joint network that takes both acoustic and text information to predict the next symbol. AED models, such as the listen, attend and spell model [2], also have an encoder for acoustic input and a decoder with an attention mechanism to generate the output text. Recurrent neural networks or Transformer networks [3], and similar structures, are typically used as network components.

Although E2E systems can achieve low WERs, they lack the ability to identify non-speech regions such as silence by default, and there is no consensus on the definition of when a recognised unit should begin or end. In hybrid systems, usually phones are the basic modelling units which are represented by HMMs, and then combined according to dictionaries to form words. Hence an HMM network can be constructed for an utterance using a reference transcription along with the dictionary, and optional silence states can be inserted between words. The Viterbi algorithm is then used to identify word boundaries and possible silences, which is known as forced alignment. Accurate timing information for ASR output is crucial for various applications, including keyword spotting and voice editing, as well as downstream tasks based on ASR systems. Although a trained hybrid system can perform forced alignment for E2E system outputs, this approach can be costly to deploy [4]. Therefore, there is a growing interest in enabling E2E systems to predict the timing of recognised units using neural methods. Moreover, to make time prediction an integral part of any E2E ASR pipeline, it is desired to apply it without requiring any changes to the existing ASR backbone model.

This paper introduces a neural aligner module as a post-processing step for E2E ASR systems. The aligner utilises a non-autoregressive (NAR) Transformer decoder structure to segment the utterance into units of interest, with the starting and ending times of each word predicted simultaneously. Non-speech regions such as silence can be labelled by considering extra non-word units with possibly zero duration between words. The aligner does not require a separate HMM-based system or Viterbi decoding algorithm during testing, and it does not affect the training or testing of existing E2E ASR systems. However, relying on regression-based duration prediction for each unit may not ensure that the total predicted duration matches the actual utterance time. To address this issue, our paper proposes modelling the normalised segment lengths for an utterance as a distribution, which can be trained using the cross-entropy (CE) loss. Experimental results on the Librispeech [5] data set show that the proposed methods can accurately model duration and generate satisfactory time alignment, even with ASR errors.

The rest of the paper is organised as follows: Sec. 2 summarises related studies on alignments in E2E ASR systems and allowing word timing prediction. Sec. 3 describes the methods to extract duration and model alignments. Sec. 4 gives the detailed setup for the experiments. The results are presented in Sec. 5 with discussions, and conclusions are drawn in Sec. 6.

## 2. Related work

Recently there has been several studies on the aligning mechanisms in E2E ASR systems. Effort has been paid to reduce the latency of streaming models [6, 7, 8, 9, 10], including training with restrictions based on forced alignment for streaming NT models [7]. Although reducing latency is important for streaming applications, the time instant of symbol emission still gives

Dongcheng Jiang is funded by Trinity College, University of Cambridge. This work has been performed using resources provided by the Cambridge Tier-2 system operated by the University of Cambridge Research Computing Service (www.hpc.cam.ac.uk) funded by EPSRC Tier 2 capital grant EP/T022159/1.

10.21437/Interspeech.2023-1071

no direct indication of start and end times of a word, as inter-word silences cannot be detected.

Methods to enable E2E models to predict word times have also been studied [4, 11, 12, 13, 14, 15] and some of them interact with ASR training. HMM-based time alignments were often used to restrict time alignments in E2E systems in training. In [4, 6] it was found that such restrictions may increase the recognition WER for an NT model, but the WER and time accuracy could be improved using a second-pass ASR where an AED decoder is trained and alignment restriction applied to one of the attention heads [4]. Phone-based CE layers, which are similar to the setup used in hybrid systems and trained using forced alignment, were also investigated for the encoders in NT models [12] and AED models [11] so that Viterbi alignment can be performed. There has also been research [11, 16] aimed at reducing the need for HMM-based forced alignments, for example, obtaining time from CTC [17] alignments.

Duration modelling can be important in speech processing, including segmenting units in ASR [18, 19, 20, 21] and duration prediction in text-to-speech (e.g. [22]), since duration is closely related to the matching between acoustic and text.

## 3. Duration modelling and time prediction

In this paper, an additional neural aligner is considered for an E2E ASR system, whose training is independent of ASR training, while the aligner takes the representations from the base ASR model. The aligner can be trained using duration information from forced alignments, so that it can predict the duration of each word or silence. From word and silence durations, the time alignment can then be inferred after the first-pass ASR decoding without relying on HMM-based search or affecting the original ASR system performance.

### 3.1. Model and methodology

As shown in Figure 1, the aligner is a neural module that takes the hidden representations from the ASR system encoder and decoder (or prediction network), and predicts the duration of each unit. Here the E2E system is assumed to be already trained. After the ASR output has been generated, the aligner takes the recognised text as input and predicts the time alignment with respect to its acoustic input representations.

An NAR version of the Transformer decoder block[1] is used to build the aligner, which was also used in [23]. Therefore self-attention can take the whole input information into account (not only the past context) which is important for later operations. Cross-attention uses the encoder representation, which is necessary to predict duration given the current acoustic input. The output vectors of the final aligner block are projected into scalars for duration prediction. The structure of the aligner is shown in Figure 2.

Conventionally, the Viterbi algorithm with path tracking ensures that each frame is assigned to an HMM state hence the total time is matched exactly, which was found to be important in preliminary experiments. Local regression between the aligner output scalars and the duration from the forced alignments cannot enforce this since the model will not be error-free and the total duration can be incorrect. Therefore the proportion of time (or normalised duration) for each unit is predicted via a softmax so that the total time will be matched.

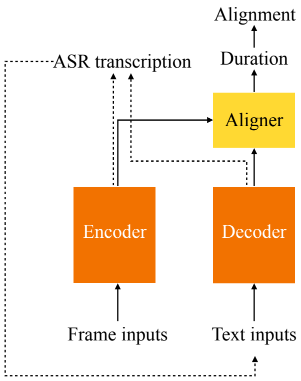

Figure 1: *The overall model. The aligner is connected to the encoder and the decoder of the original ASR system and trained alone. It can use ASR output (dashed lines), or act as a forced aligner if the reference text is given.*

For an utterance $u$ with $N$ words and no silence, in practice, E2E ASR decoders use a padding token (<pad>) as the first token to be fed forward before the first word piece is given, e.g. blank for NT models or start-of-sentence (<sos>) for AED models. These tokens can be treated as pseudo-words with zero duration so there are a total of ($N$+1) words. Although due to word piece tokenisation there are more decoder input steps, ($N$+1) scalars can be selected from the aligner output that correspond to word boundary tokens in $u$ ($T_0$, $T_1$, $T_2$ in Figure 2), and applying a softmax to get the $N$-dimensional ($N$-d) normalised duration vector $\mathbf{t}^u_{\text{pred}}$ ignoring those corresponding to a zero duration ($T_0$ in Figure 2). Each selected output still takes account of the full utterance text information due to the self-attention. If the training set is $\mathcal{U}$, then the training loss $L$ for the aligner can be written as follows:

$$L = \sum_{u \in \mathcal{U}} \text{CE}(\mathbf{t}^u_{\text{pred}}, \mathbf{t}^u_{\text{label}}) \tag{1}$$

Notice that the CE in Eq. 1 is different to the usual setup using 1-hot label vector for classification. Here the label $\mathbf{t}^u_{\text{label}}$ refers to the normalised duration from forced alignment.

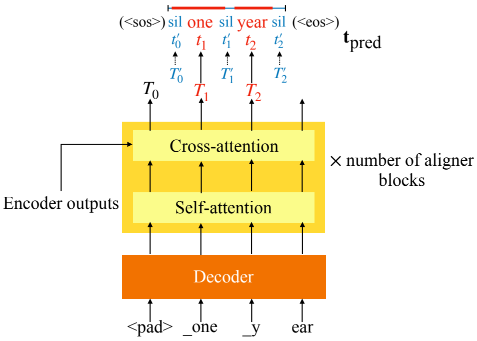
(<sos>) sil one sil year sil (<eos>)
$t'_0$ $t_1$ $t'_1$ $t_2$ $t'_2$
$\mathbf{t}_{\text{pred}}$
$T'_0$ $T'_1$ $T'_2$
$T_0$ $T_1$ $T_2$
Cross-attention
× number of aligner blocks
Encoder outputs
Self-attention
Decoder
<pad> _one _y ear

Figure 2: *Structure of the aligner and training with the utterance "one year", only attention-related parts are shown. Optional silences (sil) can be inserted before and after words.*

### 3.2. Dealing with silence

To find word start or end times, apart from modelling the word durations, silences are also important as they can occur before

[1] A standard Transformer decoder block only allows access to past context. However this restriction is not needed in this setting.

and after each word with a zero minimum duration. Assume the utterance has $N$ words, then $(2N+1)$ durations will be needed. With <pad> there will be $(N+1)$ output scalars for word duration. By introducing another projection layer at the aligner output vectors to generate another $(N+1)$ scalars (e.g. $T_0'$, $T_1'$, $T_2'$ in Figure 2) to model the silence after each word, the $(2N+1)$-d vector for softmax can be obtained (<pad> has zero duration and is not considered). This method will be referred to as **joint modelling**. Notice that the increase in the dimension of the normalised duration could be problematic for long utterances.

Modelling the word start times and word end times can be separated, by considering pseudo-words start-of-sentence (<sos>) and end-of-sentence (<eos>) of zero duration. For an $N$-word utterance, to model the word end times, each possible silence can be combined with its following word and the combined duration is modelled with the final silence combined with <eos>, hence a $(N+1)$-d normalised duration vector is used. Similarly the start times can be modelled using another $(N+1)$-d vector by combining silences with the preceding word, and the two corresponding CE losses are added together. By comparing the two $(N+1)$-d normalised durations, potential silences can be found assuming the model is well trained. If the two neighbouring words overlap then there is no silence in between, and the midpoint of the overlap is chosen as the boundary between the words. Although the modelling targets are changed, the required model structure is the same as for joint modelling. This method will be referred to as **separate modelling**. By merging silence, duration modelling may be poorer but the alignment may be better especially for small amounts of silence as it reduces the number of units for each softmax.

# 4. Experimental setup

## 4.1. Data and model

The experiments were conducted using the Librispeech [5] data set. The backbone ASR model and the aligner were trained using the train-clean-100 subset, while dev-clean and dev-other were merged to form the validation set. Testing results using test-clean and test-other are reported. The word-level forced alignment for Librispeech in [24] was used as the reference time, which was generated by the Montreal Forced Aligner [25]. 80-d filter bank features together with 3-d pitch information were extracted at 10 ms frame shift. SpecAugment [26] with parameters $(W, F, m_F, T, p, m_T) = (40, 27, 2, 40, 1.0, 2)$ was used for data augmentation. A unigram word piece model was used to generate a set of 600 word pieces as targets.

The implementation was based on the ESPnet toolkit [27]. An NT model with limited decoder history [28] and an AED model were tested in this paper as ASR backbones. The encoder for both models is the Conformer (S) in [29], which has 16 encoder blocks, 4 attention heads, and encoder dimension 144. The NT model has a feed-forward predictor taking 3 steps of word piece history and a joiner, both with 320-d hidden units, and was trained with the standard transducer loss. The AED model has an 1-layer uni-directional LSTM [30] decoder and location-sensitive single head attention, both with dimension 320, and was trained without CTC. Both backbones have roughly 10M parameters. The aligner has 12 blocks with 4 attention heads and the remaining dimensions matched to the encoder and decoder hidden dimensions so that it can directly use the corresponding representations. Apart from the encoder output, for the NT model, the aligner takes the decoder output, and for the AED model the aligner takes the output of the embedding layer of word pieces. The aligner has roughly 4M parameters. The Noam scheduler [3] and empirical hyperparameters (drop-out rate 0.1, 25k warm-up steps) were adopted. The NT model final WERs on test-clean and test-other were 10.4% and 26.7%, and for the AED model they were 10.1% and 26.1%. Then the aligner was added to the backbone and trained using the proposed loss for 120 epochs (roughly took 43 h for using NT and 60 h for using AED on single NVIDIA A100 GPU). The backbones were frozen during aligner training.

## 4.2. Evaluation metrics

To measure the quality of the time alignment, the mismatch in start time $|\Delta t_s|$ and end time $|\Delta t_e|$ for words was measured as suggested in [4, 11]. The average (Avg) mismatch was calculated in $\mathrm{ms}$, and with a 200 ms tolerance window the accuracy was also calculated. A small average mismatch and a high accuracy are preferred. The aligner aligns the reference text or the decoded text after ASR (-D for using decoded text). To measure the alignment quality for ASR output, only matched words in the decoded text and the reference were considered. Since the duration is the basic quantity being modelled, the average mismatch of word duration ($|\Delta d_w|$) and silence duration ($|\Delta d_s|$) were also calculated. Durations were found by assigning the Conformer outputs (with a 40 ms stride) to each word, so they were quantised into Conformer steps first to count the mismatch and then converted back to deviations in $\mathrm{ms}$.

# 5. Results and discussion

The aligners based on the NT model using joint modelling (system A1) and separate modelling (system A2) mentioned in Section 3.2 were tested. Table 1 summarises the performance of A1 and A2 for duration modelling with reference texts. It can be seen on average the duration mismatch is slightly larger than half of one Conformer step (20 ms) and more than 99% of the predictions are within 200 ms tolerance, which indicates the aligner could learn to predict duration accurately. In Table 1, A1 has smaller duration mismatch on average and has more predictions within the tolerance compared to A2, which indicates that joint modelling has, on average, better performance for duration prediction compared to separate modelling.

Table 1: *Quality of duration prediction for NT joint modelling (A1) and NT separate modelling (A2) on the test sets.*

| Set | test-clean | | test-other | |
|---|---|---|---|---|
| System | A1 | A2 | A1 | A2 |
| Avg($\|\Delta d_w\|$) (ms) | 21.7 | 25.1 | 25.2 | 27.3 |
| Avg($\|\Delta d_s\|$) (ms) | 21.6 | 25.1 | 24.7 | 26.7 |
| % $\|\Delta d_w\|$ ≤200 ms | 99.7 | 99.3 | 99.5 | 99.3 |
| % $\|\Delta d_s\|$ ≤200 ms | 99.7 | 99.3 | 99.5 | 99.4 |

For the NT backbone model, with the strict monotonicity applied on the transducer loss trellis, an alignment can be found by tracking the best path with the largest total likelihood. As silence is not modelled, the word end time was defined as the time of its last word piece in the best-path alignment. This approach was used here as a baseline for word end times for both reference text (A0) and recognised text (A0-D) using the NT ASR backbone, and the results are given in Table 2. The baseline word end time average mismatch is about 3 Conformer steps (120 ms) and the accuracy within 200 ms tolerance is about 86%. In Table 2, for the NT model, alignment quality using recognised text is lower compared to using reference text.

Table 2: *Quality of word end time given by NT backbone using the reference text (A0) and using the recognised text (A0-D).*

| Set | test-clean | | test-other | |
|---|---|---|---|---|
| System | A0 | A0-D | A0 | A0-D |
| Avg($\|\Delta t_e\|$) (ms) | 116 | 119 | 113 | 118 |
| % $\|\Delta t_e\|$ $\leq$200 ms | 86.3 | 85.8 | 86.8 | 85.9 |

The time alignment quality using reference texts for A1 and A2 are presented in Table 3. Due to duration error accumulation, the accuracy of word timings was reduced compared to duration prediction, which is in line with the differences between Tables 1 and 3. On average the time mismatch in Table 3 went up to nearly 2 Conformer steps (80 ms). With the 200 ms tolerance, A1 could approach 94% accuracy and A2 could obtain a better accuracy of 95.5%, both are much better than A0 in Table 2 for word end timings. Hence the proposed methods are effective, and the alignment quality could be improved by reducing the vector lengths for the final softmax via separate modelling. On the other hand, A1 is slightly better than A2 in duration modelling. So the experiments also showed that the effect of duration errors in separate modelling are smoothed and more localised. Since only the train-clean-100 subset was used for training, in Table 1 and Table 3, the performance on test-clean is slightly better compared to test-other. Overall, both A1 and A2 can still predict good time alignments.

Table 3: *Alignment quality using the reference text for NT joint modelling (A1) and NT separate modelling (A2) on the test sets.*

| Set | test-clean | | test-other | |
|---|---|---|---|---|
| System | A1 | A2 | A1 | A2 |
| Avg($\|\Delta t_s\|$) (ms) | 79.4 | 74.5 | 81.5 | 74.3 |
| Avg($\|\Delta t_e\|$) (ms) | 79.7 | 74.6 | 82.2 | 75.3 |
| % $\|\Delta t_s\|$ $\leq$200 ms | 94.0 | 95.5 | 93.9 | 95.5 |
| % $\|\Delta t_e\|$ $\leq$200 ms | 94.0 | 95.6 | 93.8 | 95.4 |

Instead of using reference text, the recognised text from the NT model were adopted for testing A1 and A2 and the results are in Table 4. Compared to A0-D in Table 2, A1-D and A2-D still has much smaller mismatch (about 2 Conformer steps) and higher accuracy (>90%). Compared to Table 3, in Table 4 the average mismatch becomes slightly larger and the accuracy is reduced. The degradation could be caused by word errors in ASR, while A2-D is more affected compared to A1-D (0.3% vs. 0.8% absolute difference on test-clean). Therefore, with the NT backbone, the separate modelling A2-D seems to be more sensitive to the ASR quality by merging silences during modelling, although it still has better overall performance compared to A1-D. The results on test-other degrades more compared to test-clean as it has (many) more errors in ASR.

Separate modelling was then tested for the AED backbone (system B2). Results using both the reference text and the recognised text from the AED model were included in Table 5. It can be observed that B2 and B2-D have alignment quality similar to A2 and A2-D by comparing Table 5 with Tables 3 and 4, which shows the effectiveness of the proposed methods for E2E models. In Table 5, unlike for the NT setup, the alignment is better when considering the matched words after ASR. By taking the decoder embeddings and the encoder outputs of the AED model, the aligner's time alignment prediction seems to be more robust to the backbone ASR errors compared to the NT model's setup, where each decoder output takes history information, but is also affected by ASR errors in the history.

Table 4: *Alignment quality using the NT recognised text for NT joint modelling (A1-D) and NT separate modelling (A2-D).*

| Set | test-clean | | test-other | |
|---|---|---|---|---|
| System | A1-D | A2-D | A1-D | A2-D |
| Avg($\|\Delta t_s\|$) (ms) | 81.1 | 78.0 | 87.5 | 80.7 |
| Avg($\|\Delta t_e\|$) (ms) | 81.0 | 78.0 | 87.6 | 80.7 |
| % $\|\Delta t_s\|$ $\leq$200 ms | 93.7 | 94.7 | 92.4 | 93.7 |
| % $\|\Delta t_e\|$ $\leq$200 ms | 93.7 | 94.8 | 92.4 | 93.7 |

Table 5: *Alignment quality for AED separate modelling using the reference text (B2) and AED recognised text (B2-D).*

| Set | test-clean | | test-other | |
|---|---|---|---|---|
| System | B2 | B2-D | B2 | B2-D |
| Avg($\|\Delta t_s\|$) (ms) | 77.8 | 76.9 | 80.2 | 79.7 |
| Avg($\|\Delta t_e\|$) (ms) | 77.8 | 76.8 | 81.8 | 79.7 |
| % $\|\Delta t_s\|$ $\leq$200 ms | 95.0 | 95.1 | 94.7 | 95.2 |
| % $\|\Delta t_e\|$ $\leq$200 ms | 95.0 | 95.1 | 94.5 | 95.2 |

In [4], training used relatively short utterances, and the accuracy over 99% with the 200 ms tolerance was considered as good performance. Here utterances less than 10 s in the test sets were selected to form subsets test-clean-short and test-other-short. Alignment quality evaluated using the subsets and reference text are shown in Table 6, both A2 and B2 were able to approach the 99% accuracy, which is much better compared to the results for the original sets. So the methods are very accurate for short to moderate-length utterances, and long utterances could be split to obtain better quality alignments.

Table 6: *Alignment quality using the reference text and separate modelling for NT setup (A2) and AED setup (B2) on utterances with moderate lengths (<10 s).*

| Set | test-clean-short | | test-other-short | |
|---|---|---|---|---|
| System | A2 | B2 | A2 | B2 |
| Avg($\|\Delta t_s\|$) (ms) | 52.5 | 56.1 | 57.9 | 61.6 |
| Avg($\|\Delta t_e\|$) (ms) | 53.2 | 56.7 | 58.8 | 63.1 |
| % $\|\Delta t_s\|$ $\leq$200 ms | 99.2 | 99.0 | 98.8 | 98.4 |
| % $\|\Delta t_e\|$ $\leq$200 ms | 99.2 | 99.1 | 98.7 | 98.4 |

## 6. Conclusions

In this paper, a novel way of enabling a trained E2E ASR system to predict the start time and end time of each word after ASR has been proposed by using a neural aligner based on an NAR Transformer decoder. The aligner takes the ASR system representations to model normalised word and silence durations, and produce a time alignment. Using the NT backbone, durations could be modelled well. The alignment quality was improved by trading off against duration acuracy. For both the NT and the AED backbone, alignment accuracy for matched words could approach 95% using reference or decoded text with 200 ms tolerance, and accuracy of aligning reference texts for utterances less than 10 s was about 99%. The proposed methods are effective in duration modelling and time alignment prediction for E2E ASR models, and can be robust to ASR errors.

## 7. References

[1] A. Graves, “Sequence transduction with recurrent neural networks,” in *Proc. ICML Workshop on Representation Learning*, 2012.

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

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Proc. NeurIPS*, 2017.

[4] T. N. Sainath, R. Pang, D. Rybach, B. García, and T. Strohman, “Emitting word timings with end-to-end models,” in *Proc. Interspeech*, 2020.

[5] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. ICASSP*, 2015.

[6] B. Li, S.-y. Chang, T. N. Sainath, R. Pang, Y. He, T. Strohman, and Y. Wu, “Towards fast and accurate streaming end-to-end ASR,” in *Proc. ICASSP*, 2020.

[7] 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.

[8] J. Yu, C.-C. Chiu, B. Li, S.-y. Chang, T. N. Sainath, Y. He, A. Narayanan, W. Han, A. Gulati, Y. Wu, and R. Pang, “Fastemit: Low-latency streaming ASR with sequence-level emission regularization,” in *Proc. ICASSP*, 2021.

[9] J. Kim, H. Lu, A. Tripathi, Q. Zhang, and H. Sak, “Reducing streaming ASR model delay with self alignment,” in *Proc. Interspeech*, 2021.

[10] H. Inaguma and T. Kawahara, “Alignment knowledge distillation for online streaming attention-based speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 1371–1385, 2023.

[11] 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.

[12] R. Zhao, J. Xue, J. Li, W. Wei, L. He, and Y. Gong, “On addressing practical challenges for RNN-transducer,” in *Proc. ASRU*, 2021.

[13] R. Yang, G. Cheng, H. Miao, T. Li, P. Zhang, and Y. Yan, “Keyword search using attention-based end-to-end ASR and frame-synchronous phoneme alignments,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3202–3215, 2021.

[14] Y. Luo, C.-C. Chiu, N. Jaitly, and I. Sutskever, “Learning online alignments with continuous rewards policy gradient,” in *Proc. ICASSP*, 2017.

[15] L. Dong and B. Xu, “CIF: Continuous integrate-and-fire for end-to-end speech recognition,” in *Proc. ICASSP*, 2020.

[16] L. Huang, J. Sun, Y. Tang, J. Hou, J. Chen, J. Zhang, and Z. Ma, “HMM-free encoder pre-training for streaming RNN transducer,” in *Proc. Interspeech*, 2021.

[17] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML*, 2006.

[18] A. Zeyer, R. Schmitt, W. Zhou, R. Schlüter, and H. Ney, “Monotonic segmental attention for automatic speech recognition,” in *Proc. SLT*, 2023.

[19] D. Jiang, C. Zhang, and P. C. Woodland, “Variable frame rate acoustic models using minimum error reinforcement learning,” in *Proc. Interspeech*, 2021.

[20] Z. Chen, Y. Zhang, A. Rosenberg, B. Ramabhadran, P. J. Moreno, A. Bapna, and H. Zen, “MAESTRO: Matched speech text representations through modality matching,” in *Proc. Interspeech*, 2022.

[21] Z. Chen, A. Bapna, A. Rosenberg, Y. Zhang, B. Ramabhadran, P. Moreno, and N. Chen, “Maestro-U: Leveraging joint speech-text representation learning for zero supervised speech ASR,” in *Proc. SLT*, 2023.

[22] H. Chung, S.-H. Lee, and S.-W. Lee, “Reinforce-aligner: Reinforcement alignment search for robust end-to-end text-to-speech,” in *Proc. Interspeech*, 2021.

[23] E. A. Chi, J. Salazar, and K. Kirchhoff, “Align-refine: Non-autoregressive speech recognition via iterative realignment,” in *Proc. NAACL*, 2021.

[24] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, “Speech model pre-training for end-to-end spoken language understanding,” in *Proc. Interspeech*, 2019.

[25] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, “Montreal forced aligner: Trainable text-speech alignment using Kaldi,” in *Proc. Interspeech*, 2017.

[26] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Proc. Interspeech*, 2019.

[27] 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.

[28] R. Prabhavalkar, Y. He, D. Rybach, S. Campbell, A. Narayanan, T. Strohman, and T. N. Sainath, “Less is more: Improved RNN-T decoding using limited label context and path merging,” in *Proc. ICASSP*, 2021.

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

[30] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.