



Improving Frame-level Classifier for Word Timings with Non-peaky CTC in End-to-End Automatic Speech Recognition

Xianzhao Chen, Yist Y. Lin, Kang Wang, Yi He, Zejun Ma

ByteDance

{chenxianzhao, yist.lin0, wangkang, heyi.hy, mazejun}@bytedance.com

Abstract

End-to-end (E2E) systems have shown comparable performance to hybrid systems for automatic speech recognition (ASR). Word timings, as a by-product of ASR, are essential in many applications, especially for subtitling and computer-aided pronunciation training. In this paper, we improve the frame-level classifier for word timings in E2E system by introducing label priors in connectionist temporal classification (CTC) loss, which is adopted from prior works, and combining low-level Mel-scale filter banks with high-level ASR encoder output as input feature. On the internal Chinese corpus, the proposed method achieves 95.68%/94.18% compared to the hybrid system 93.0%/90.22% on the word timing accuracy metrics. It also surpasses a previous E2E approach with an absolute increase of 4.80%/8.02% on the metrics on 7 languages. In addition, we further improve word timing accuracy by delaying CTC peaks with frame-wise knowledge distillation, though only experimenting on LibriSpeech.

Index Terms: automatic speech recognition, connectionist temporal classification, word timing, end-to-end

1. Introduction

Word timing is the start and end time of a word. It is widely used in many ASR applications, such as computer-aided pronunciation training, subtitling and filler word removal. Word timings can also be converted into frame-level classifications for pre-training acoustic model in ASR systems [1]. Correspondingly, the word timing metric can be used to measure the performance of the frame-level classifier [2].

Recently, E2E systems adopting Deep Neural Networks (DNN) have demonstrated excellent ASR performance [3, 4]. Unlike traditional hybrid systems, E2E systems directly align inputs and outputs without modeling phonetic alignment. In E2E systems, Listen, Attend and Spell (LAS) [5] uses attention mechanisms, CTC [6] and RNN Transducer (RNN-T) [7] introduce blank token to align variable-length inputs and outputs. Though E2E systems trained with soft alignments achieve superior ASR performance compared to traditional hybrid systems, the estimated word timing in such systems is usually not comparable with that in traditional ones.

There are several methods for estimating word timings in hybrid systems and E2E systems. In a hybrid system, such as DNN-Hidden Markov Models (HMM) system, a DNN model is trained with frame-level classification using alignments from a pretrained GMM-HMM system, and the HMMs are trained using the posterior probabilities of acoustic features provided by DNN [8]. Accurate word timings are obtained from forced alignment of the recognition results. However, a hybrid system requires many steps of training, making it less desirable.

In E2E systems, word timings can be estimated by the forced alignment results of character-level CTC models, where the CTC peak of the first character indicate the word start time and the CTC peak of the last character indicate the word end time [9]. The CTC model cannot estimate word timings well when the duration of the modeling unit is relatively long, e.g., Chinese characters. Because the blank probability of CTC model is dominant in almost all frames, and the non-blank probability is only relatively high in few frames. This is called the peaky behavior [10]. CTC-based alignments for word timings can be improved by alleviating the peaky behavior [11, 12], but these methods have complicated regularization terms which require modifications of the model architecture or CTC loss topology respectively. A two-pass model trained with frame-level classification, where alignments are from traditional hybrid systems, predicts word timings by constrained attention weights [2]. The method achieves state-of-the-art performance on word timing metrics, but its dependence on hybrid systems makes it less favored. A purely E2E system was proposed where alignments from a hybrid system are not needed [13]. But it requires training a frame-level classifier with CTC loss, manually expanding CTC peaks to word timings, and training another classifier using the word timings with cross-entropy (CE) loss. Continuous integrate-and-fire (CIF) is a soft and monotonic alignment well suited for estimating word timings in E2E non-autoregressive ASR systems. The CIF-based system obtains good acoustic boundaries by accumulating weights of frames, until exceeding a threshold value, and firing tokens. With several steps of post-processing, its performance is comparable to HMM-based system and only need to be verified in languages other than Chinese [14].

To the best of our knowledge, we are the first to adopt label priors to eliminate the CTC peaky behavior [15] for pure E2E word timing estimation and the proposed system surpasses the performance of HMM-based system on Chinese language. Our contributions in this paper are:

1. We adopt label priors to eliminate the peaky behavior of CTC and train frame-level classifiers to predict word timings, which can be easily extended to other languages.
2. We introduce Mel features as low-level local information and combine with Transformer encoder outputs as high-level utterance-wise information, which can be seen as feature fusion and is found to be beneficial for word timing estimation.
3. We further improve word timing accuracy by a frame-level knowledge distillation method to shift word boundaries adaptively.

E2E speech recognition and word timing estimation systems are introduced in Section 2. In Section 3, methods for optimizing the frame-level classifier used in this paper are presented. Sec-

tion 4 is the experimental setup. Results and analyses are presented in Section 5, and a conclusion is given in Section 6.

2. System Overview

The E2E system used in this paper is illustrated in Figure 1. The ASR system employs LAS [5] which is an attention-based encoder-decoder architecture. The word timing system employs a frame-level classifier following the encoder and the Mel feature is introduced optionally. The Mel filter bank sequence is given as $\mathbf{x} = (x_1, \dots, x_T)$ and the wordpiece sequence is given as $\mathbf{y} = (y_1, \dots, y_U)$, where y_u is a wordpiece from a vocabulary of the size V . The output of the decoder is denoted as $P(y_u|\mathbf{x}, y_{<u})$ at the step u . The output of the encoder is denoted as $\mathbf{h} = (h_1, \dots, h_T)$ and the encoder output with Mel feature denoted as $\mathbf{h}' = (h_1; x_1, \dots, h_T; x_T)$ (represented by +Mel). The output of the frame-level classifier is denoted as $P(v, t|\mathbf{x})$ of the wordpiece v at the time t . The training stage consists of two steps. First, the encoder-decoder is trained to maximize $P(y_u|\mathbf{x}, y_{<u})$. Second, the parameters in the LAS are fixed and the frame-level classifier is trained using different methods which are described in the next section. In the decoding stage, the encoder-decoder produces ASR results \mathbf{y}' and the word timings come from the forced alignment between \mathbf{y}' and $P(v, t|\mathbf{x})$.

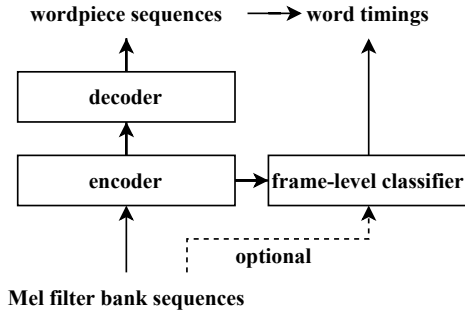


Figure 1: system workflow

3. Methodology

The frame-level classifier trained with CTC loss has peaky behavior where the peak roughly falls within the ground truth range of a token which can be character, wordpiece, or word. The exact start and end times of a token can be obtained by estimating the left and right boundaries based on CTC peak position. In this section, word timing estimation including peak position estimation and boundary estimation will be presented separately. In addition, we also propose two methods to further polish word timing accuracy.

3.1. Peak position estimation

This subsection presents two methods combined to estimate the CTC peak position.

CTC loss is proposed for training frame-level classifiers in E2E systems [6]. In order to align arbitrary frame-level and token-level sequences, the CTC model introduces a blank token (represented by ϕ). The CTC loss is computed by Equation 1, where $P(\pi_t|\mathbf{x})$ is the posterior probability of the token π_t from the frame-level classifier at the time t , and the function \mathcal{B} removes consecutive equivalent tokens as well as blank tokens so that $\mathcal{B}(\boldsymbol{\pi}) = \mathbf{y}$. There are many valid paths $\boldsymbol{\pi}$ that satisfy $\mathcal{B}(\boldsymbol{\pi}) = \mathbf{y}$, so the forward-backward algorithm is used

to optimize the calculation of the CTC valid paths. Since the E2E system does not rely on the frame-level classifier to output ASR results, it must use forced alignment to obtain word timings. Forced alignment, computed by Equation 2, aims to find the most likely valid path $\boldsymbol{\pi}$. The start and end time of each token y_u is obtained through the path $\boldsymbol{\pi}$. Many experiments have shown that frame-level classifiers trained with CTC loss always predict blank tokens, known as the peaky behavior [12, 16, 17]. As shown in Figure 2, forced alignment of non-blank tokens last for few frames and peaks appear near the ground truths. To summarize, the peaks can be obtained as anchor points for the position estimation by forced alignment from the CTC model.

$$\mathcal{L}_{\text{CTC}} = -\log P(\mathbf{y}|\mathbf{x}) = -\log \sum_{\boldsymbol{\pi} \in \mathcal{B}^{-1}(\mathbf{y})} \prod_t P(\pi_t|\mathbf{x}) \quad (1)$$

$$\boldsymbol{\pi} = \underset{\mathcal{B}(\boldsymbol{\pi})=\mathbf{y}}{\operatorname{argmax}} P(\boldsymbol{\pi}|\mathbf{x}) = \underset{\mathcal{B}(\boldsymbol{\pi})=\mathbf{y}}{\operatorname{argmax}} \prod_t P(\pi_t|\mathbf{x}) \quad (2)$$

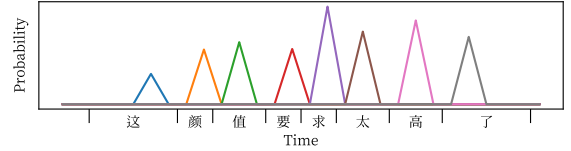


Figure 2: CTC peaks

3.2. Boundary estimation

This subsection introduces two methods of peak expansion to estimate boundaries.

A previous method to estimate boundaries is based on neighboring CTC peaks, and teaches the frame-level classifier with a new posterior probability using cross-entropy [13]. For convenience, this method is abbreviated to CETC. The position of CTC peak of the token y_u is denoted as t_{peak}^u . The start and end times are computed by Equation 3. The probability of the token y_u at the time t is $P(y_u, t|\mathbf{x})$ from the classifier trained with $\mathcal{L}_{\text{CETC}} = \text{Cross-Entropy}(P, P_{\text{guided}})$, where P_{guided} is computed by Equation 4. The hyperparameters α_{left} and α_{right} control the left and right boundaries, and β controls that the further away from the CTC peak, the lower the probability. According to the CETC paper, the $(\alpha_{\text{left}}, \alpha_{\text{right}}, \beta)$ are set to $(0.2, 0.7, 0.5)$. The CETC method estimating boundaries by neighboring peaks is used to estimate word timings.

$$\begin{aligned} t_{\text{start}}^u &= t_{\text{peak}}^u - \alpha_{\text{left}} \cdot (t_{\text{peak}}^u - t_{\text{peak}}^{u-1}) \\ t_{\text{end}}^u &= t_{\text{peak}}^u + \alpha_{\text{right}} \cdot (t_{\text{peak}}^{u+1} - t_{\text{peak}}^u) \end{aligned} \quad (3)$$

$$P_{\text{guided}}(y_u, t|\mathbf{x}) = \begin{cases} \left(\frac{t - t_{\text{peak}}^u}{t_{\text{peak}}^u - t_{\text{start}}^u} \right)^\beta, & t_{\text{start}}^u \leq t < t_{\text{spike}}^u \\ \left(\frac{t_{\text{end}}^u - t}{t_{\text{end}}^u - t_{\text{peak}}^u} \right)^\beta, & t_{\text{spike}}^u \leq t \leq t_{\text{end}}^u \end{cases} \quad (4)$$

$$P_{\text{guided}}(\phi, t|\mathbf{x}) = 1 - \sum P(\neq \phi, t|\mathbf{x})$$

A CTC model without peaky behavior is well suited for boundary estimation. A recent paper explains and elegantly solves the CTC peaky behavior by introducing the label prior, which is abbreviated to the non-peaky CTC (NPC) method here [18]. The label prior must be calculated using logits before softmax function, computed by Equation 5. The modified logits are computed by Equation 6 and the NPC paper suggests using different γ in training and inference, which will be reported in the following section. The CTC loss is calculated on

the modified logits. The NPC method extending a peak to a span by reducing the probability of blank token is employed to estimate the token boundaries.

$$O_{\text{prior}}(y_u|\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T O(y_u, t|\mathbf{x}) \quad (5)$$

$$O'(y_u, t|\mathbf{x}) = O(y_u, t|\mathbf{x}) - \gamma \cdot O_{\text{prior}}(y_u|\mathbf{x}) \quad (6)$$

3.3. Polishing word timing accuracy

In our experiments, we found that the NPC method tends to estimate word boundaries earlier in time, which leaves room for improvement in word timing accuracy. Having the manually labelled development set, the best shifting for predicted boundaries can be grid searched and applied in post-processing. On the other hand, to shift word timings adaptively, the CTC peaks are shifted using a frame-by-frame knowledge distillation method called Peak First Regularization (PFR) [10]. The PFR loss is computed by Equation 7, where $\mu = -1$ delays CTC peaks, $\mu = 1$ advances CTC peaks and P_{PFR} is the processed classifier probability. A temperature coefficient τ in Equation 8 smooths the output of the frame-level classifier $O(v, t|\mathbf{x})$ of the token v at the time t . According to the independence assumption of CTC, the PFR loss must be employed before freezing the LAS and the loss is computed by Equation 9. It is necessary to prevent the gradient from passing through the teacher frame $P_{\text{PFR}}(v, t + \mu|\mathbf{x})$. The default $(\mu, \tau, \lambda_{\text{CE}})$ is $(-1, 10.0, 0.95)$ when using the PFR loss.

$$\mathcal{L}_{\text{PFR}} = \sum_t \sum_v P_{\text{PFR}}(v, t + \mu|\mathbf{x}) \log \frac{P_{\text{PFR}}(v, t + \mu|\mathbf{x})}{P_{\text{PFR}}(v, t|\mathbf{x})} \quad (7)$$

$$P_{\text{PFR}}(v, t|\mathbf{x}) = \text{softmax}(O(v, t|\mathbf{x})/\tau) \quad (8)$$

$$\mathcal{L} = \lambda_{\text{CE}} \cdot \mathcal{L}_{\text{CE}} + (1 - \lambda_{\text{CE}}) \cdot (\mathcal{L}_{\text{CTC}} + \lambda_{\text{PFR}} \cdot \mathcal{L}_{\text{PFR}}) \quad (9)$$

4. Experiments

4.1. Datasets

We use the LibriSpeech 960h for base experiment and apply the word timing estimation method to the internal data in 7 languages including Chinese (zh), English (en), German (de), Russian (ru), Spanish (es), Indonesian (id) and Vietnamese (vi) where the total durations are 100k, 40k, 9k, 10k, 20k, 6k and 10k hours respectively. The word timing test set for LibriSpeech is the forced alignment result from HMM-based hybrid system with script¹ and the internal test sets are annotated by the experts. The E2E system employs 80-dimensional Mel-filter bank features as input and wordpiece sequences obtained with SentencePiece tokenizer [19] as output.

4.2. Modeling

In this paper, an encoder-decoder model outputs ASR results and a frame-level classifier of 2 dense layers with 2048 hidden units following the encoder outputs word timings. The 80-dimensional Mel-filter bank is computed on a 25 ms window with a 10 ms shift. In the encoder, two CNN kernels of width, height, input channel size, and output channel size $(3, 3, 1, 64)$ and $(3, 3, 64, 128)$ with $(2, 2)$ stride are used to extract high-level feature. Then the CNN feature projected to 512-dimension is fed into 18 transformer blocks. The transformer blocks have a

¹<https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/run.sh>

multi-head self-attention layer with 8 heads and a feed-forward layer adopting GLU activation function with 2048 hidden units. The output of the last transformer block is projected from 512 to 2048. The decoder has a multi-head attention layer with 2 heads, 4 LSTM layers with 1024 hidden units and an embedding layer with 512 hidden units. The encoder output optionally combines 4 consecutive frames of the Mel features into 2368-dimensional features, which are fed to the frame-level classifier as acoustic features. Overall the model has 168M parameters.

The E2E system is trained in Tensorflow [20] using the Lingvo [21] toolkit and RETURNN [22] toolkit (for calculating the modified CTC loss) on 16 Tesla A100 GPUs for around 5 days. The hybrid system is trained in Kaldi [8].

4.3. Configuration

Using Adam optimizer to train the LAS model, the highest learning rate is $7.5e-4$ at the end of the warm-up at 4k steps. The learning rate starts decaying at 60k steps decaying by half every 50k steps. A model is trained 200k steps, and models for high-resource languages are trained up to 300k steps. Audios of similar length are grouped into same batches, and each step of batch size multiplied by the longest audio in the batch is approximately 1224 seconds. The training technique is from the LAS paper [5], using Specaugment [23] where $(F, m_F, T, p, m_T) = (27, 2, 100, 0.1, 1)$. The variational weight noise [24] is turned on optionally where standard deviation is 0.075 and the uniform label smoothing [25] is introduced with uncertainty 0.1. In addition MWER [26] and CTC fine tuning with fixed learning rates $1e-6$ are trained for 20k and 10k steps respectively.

4.4. Measuring word timings

The word timing metrics are adopted from [2], which are the average offset (ms) of the predicted word start to the ground truth (Ave. ST Δ) and the percentage of those word start offsets smaller than 200 ms (%WS<200ms). Similarly, the Ave. ED Δ and %WE<200ms for word end. In our experiments, word timing metrics are only calculated on the matched token pairs according to the alignment of minimum edit distance. For higher granularity we employ %WS<80ms and %WE<80ms in our experiments.

5. Results

5.1. Analyzing the label prior

Table 1 shows the word timing metrics with different label priors used in training and inference on LibriSpeech test-clean. The frame-level classifier which is trained with CTC loss predicts mostly blanks and rarely the other tokens, known as peaky behavior. The model with peaky behavior has bad word timing metrics even though inference with the label prior when $\gamma_{\text{train}} = 0, \gamma_{\text{inf}} \neq 0$. It shows that the word timing metrics have a significant improvement because of training with the label prior as a normalization term when $\gamma_{\text{train}} \neq 0$. In the following experiments, the default $(\gamma_{\text{train}}, \gamma_{\text{inf}})$ are $(0.25, 1.0)$ because the summation of the %WS<80ms and the %WE<80ms is the largest.

5.2. Improving frame-level classifiers in 7 languages

Table 2 shows the performance of 4 frame-level classifiers in Chinese and compares the proposed method NPC+Mel to the previous work CETC in 6 other languages. HMM denotes the frame-level classifier from HMM-based hybrid system, and Offset denotes the offset in ms applied to all timestamps. It

Table 1: Effect of the label prior (γ) in training and inference on word timing metrics on LibriSpeech

γ_{train}	γ_{inf}	Ave. ST Δ	Ave. ED Δ	%WS<80ms	%WE<80ms
0.0	0.0	117.51	158.06	60.57	44.59
0.0	1.0	103.21	148.25	69.05	50.31
0.25	0.0	41.36	52.92	96.71	90.19
0.25	1.0	40.28	50.54	96.75	91.18
0.5	0.0	42.4	50.35	95.48	91.35
0.5	1.0	42.54	49.18	95.14	91.87
0.75	0.0	47.36	51.4	93.01	90.51
0.75	1.0	48.11	50.77	92.4	90.63
1.0	0.0	49.89	52.61	91.66	89.57
1.0	1.0	50.49	51.55	91.0	89.64

shows that NPC outperforms HMM in Chinese when 40 ms offset is applied. Further, the introduction of Mel feature is to provide accurate local information to reduce the amount of offset needed. Nevertheless it doesn't meet the purpose, it improves performance. Hence, we suspect the cause of the word timing shift is not from feature extraction but the use of CTC loss. NPC+Mel obtains an average of 86.79%/78.99% for %WS<80ms and %WE<80ms in all 7 languages, greatly outperforms CETC's 82.14%/70.94% and slightly surpasses NPC's 86.53%/78.10%².

Table 2: Word timing metrics in 7 languages in internal datasets

Method	Lang	Ave. ST Δ	Ave. ED Δ	%WS<80ms	%WE<80ms	Offset
HMM	zh	41.47	47.05	93.0	90.22	0
CETC	zh	39.62	44.53	91.86	89.72	0
NPC	zh	32.69	35.39	95.5	93.07	40
NPC+Mel	zh	31.93	33.78	95.68	94.18	40
CETC	en	55.66	105.43	78.72	58.33	0
NPC+Mel	en	44.05	70.35	88.64	77.91	60
CETC	ru	53.6	92.63	82.18	61.62	0
NPC+Mel	ru	48.49	80.49	85.2	69.68	60
CETC	es	180.58	189.14	59.72	55.5	0
NPC+Mel	es	173.7	181.4	62.53	60.62	70
CETC	de	44.78	63.0	86.01	75.94	0
NPC+Mel	de	33.62	52.64	93.61	83.7	60
CETC	id	49.3	80.85	85.53	69.98	0
NPC+Mel	id	46.23	80.62	88.11	76.28	60
CETC	vi	38.48	47.92	90.99	85.49	0
NPC+Mel	vi	30.09	39.62	94.85	90.36	40

5.3. Analyzing CTC peak positions

The introduction of CTC loss brings the peaky behavior, so we first analyze the CTC peak positions. As shown in Figure 3, the number of CTC peaks is measured as y-axis and the relative position of the peak time to the ground truth word timings is measured as x-axis. $x = 0$ means that CTC peaks are emitted on the word start time and $x = -1$ means that CTC peaks appear a whole word duration early before the start time. The red dotted dashed line in the figure shows the average relative position of the peak to the ground truth is 0.249. The grey dashed lines indicate the average word start and end times of the NPC+Mel prediction, and the CTC peaks are located on average at 0.406 of

²Due to space limitation, the details of NPC results for 6 languages are not listed in Table 2

the relative position of the prediction. The reason that applying positive offset to the predicted timestamps increases accuracy is because the predicted word start and end times are usually earlier than they should be, as shown in Figure 3, the grey dashed lines do not lie exactly on $x = 0$ and $x = 1$. Table 3 shows that the average word duration of the prediction is very close to the ground truth. In conclusion, the NPC+Mel method accurately predicts the word duration, and by delaying the emission time of CTC peaks we might reduce the amount of offset needed for increasing accuracy.

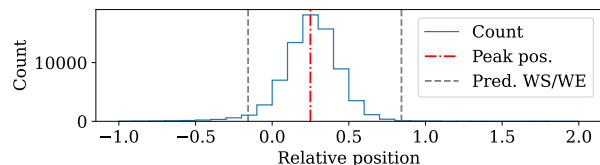


Figure 3: Distribution of the relative position of CTC peaks

Table 3: Average word duration of the ground truth and prediction (ms) in internal datasets

	zh	en	ru	es	de	id	vi
Ground truth	163.3	287.1	376.6	287.7	289.5	368.0	201.0
Prediction	162.8	257.2	331.6	288.7	279.3	337.5	195.8

5.4. Delaying CTC peaks and timestamps

We conduct the experiment in LibriSpeech. The CTC peak is delayed by introducing PFR loss during training. Table 4 shows the performance with different λ_{PFR} . The E2E system is trained from scratch with CE, CTC and PFR loss. Because of the independence assumption in CTC loss, PFR loss cannot change the position of CTC peaks in the case of fixed encoder parameters. Without fixing the encoder parameters, the ASR result is affected and we observe WER increases greatly as λ_{PFR} surpasses 2.5. The result shows that setting $\lambda_{\text{PFR}} = 1.5$ not only reduces offset to 0 ms but also achieves best word timing metrics.

Table 4: Effect of λ_{PFR} on the offset needed to achieve the high-est accuracy on LibriSpeech

λ_{PFR}	Ave. ST Δ	Ave. ED Δ	%WS<80ms	%WE<80ms	Offset	WER
0.0	40.28	50.54	96.75	91.18	40	5.3
0.5	35.06	42.04	97.35	93.84	20	4.1
1.0	34.73	40.85	98.0	95.18	10	4.7
1.5	38.69	42.9	97.99	96.04	0	5.0
2.0	32.57	35.96	97.89	94.82	-10	4.2
2.5	40.36	41.81	97.84	95.59	-20	4.7
3.0	51.62	55.69	97.67	93.96	-30	6.3

6. Conclusions

In this paper, we improve frame-level classifiers with non-peaky CTC loss in a purely E2E system. The proposed method outperforms the HMM-based hybrid system in Chinese and the CETC method in 7 languages. We observe the timestamp accuracy can be further improved, which requires manually labelled development set for gridsearching the offset of timestamp in post-processing. Also we employ PFR method to adaptively shift CTC peaks in training, improving word timing performance. In the future, the proposed approach can be applied to a multilingual E2E ASR system for word timings.

7. References

- [1] 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*, 2021, pp. 1797–1801.
- [2] T. N. Sainath, R. Pang, D. Rybach, B. Garcia, and T. Strohmaier, "Emitting word timings with end-to-end models," *Proc. Interspeech 2020*, pp. 3615–3619, 2020.
- [3] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, "A comparative study on transformer vs rnn in speech applications," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.
- [4] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.
- [5] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.
- [6] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.
- [7] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.
- [8] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.
- [9] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.
- [10] Z. Tian, H. Xiang, M. Li, F. Lin, K. Ding, and G. Wan, "Peak-first ctc: Reducing the peak latency of ctc models by applying peak-first regularization," *arXiv preprint arXiv:2211.03284*, 2022.
- [11] Y. Teytaut, B. Bouvier, and A. Roebel, "A study on constraining connectionist temporal classification for temporal audio alignment," *Proc. Interspeech 2022*, pp. 5015–5019, 2022.
- [12] H. Liu, S. Jin, and C. Zhang, "Connectionist temporal classification with maximum entropy regularization," *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [13] 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*, 2021, pp. 2571–2575.
- [14] X. Shi, Y. Chen, S. Zhang, and Z. Yan, "Achieving timestamp prediction while recognizing with non-autoregressive end-to-end asr model," *arXiv preprint arXiv:2301.12343*, 2023.
- [15] A. Zeyer, R. Schlüter, and H. Ney, "Why does ctc result in peaky behavior?" *arXiv preprint arXiv:2105.14849*, 2021.
- [16] Y. Miao, M. Gowayed, and F. Metzger, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 167–174.
- [17] H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, and J. Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 4280–4284.
- [18] A. Zeyer, R. Schlüter, and H. Ney, "Why does CTC result in peaky behavior?" *CoRR*, vol. abs/2105.14849, 2021. [Online]. Available: <https://arxiv.org/abs/2105.14849>
- [19] T. Kudo and J. Richardson, "SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. [Online]. Available: <https://www.aclweb.org/anthology/D18-2012>
- [20] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [21] J. Shen, P. Nguyen, Y. Wu, Z. Chen *et al.*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," 2019.
- [22] A. Zeyer, T. Alkhouli, and H. Ney, "RETURNN as a generic flexible neural toolkit with application to translation and speech recognition," in *Proceedings of ACL 2018, System Demonstrations*. Association for Computational Linguistics, 2018. [Online]. Available: <https://doi.org/10.18653/v1%2Fp18-4022>
- [23] 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*, 2019, pp. 2613–2617.
- [24] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*. Citeseer, 2011, pp. 2348–2356.
- [25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Re-thinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [26] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4839–4843.