*INTERSPEECH 2021*
30 August – 3 September, 2021, Brno, Czechia

# Emitting Word Timings with HMM-free End-to-End System in Automatic Speech Recognition

*Xianzhao Chen*[1*], *Hao Ni*[2], *Yi He*[2], *Kang Wang*[2], *Zejun Ma*[2], *Zongxia Xie*[1†]

[1]College of Intelligence and Computing, Tianjin University, China
[2]ByteDance AI Lab, China

chenxianzhao@tju.edu.cn, {nihao, heyi.hy, wangkang, mazejun}@bytedance.com, caddiexie@hotmail.com

## Abstract

Word timings, which mark the start and end times of each word in ASR results, play an important part in many applications, such as computer assisted language learning. To date, end-to-end (E2E) systems outperform conventional DNN-HMM hybrid systems in ASR accuracy but have challenges to obtain accurate word timings. In this paper, we propose a two-pass method to estimate word timings under an E2E-based LAS modeling framework, which is completely free of using the DNN-HMM ASR system. Specifically, we first employ the LAS system to obtain word-piece transcripts of the input audio, we then compute forced-alignments with a frame-level-based word-piece classifier. In order to make the classifier yield accurate word-piece timing results, we propose a novel objective function to learn the classifier, utilizing the spike timings of the connectionist temporal classification (CTC) model. On Librispeech data, our E2E-based LAS system achieves 2.8%/7.0% WERs, while its word timing (start/end) accuracy are 99.0%/95.3% and 98.6%/93.7% on `test-clean` and `test-other` two test sets respectively. Compared with a DNN-HMM hybrid ASR system (here, TDNN), the LAS system is better in ASR performance, and the generated word timings are close to what the TDNN ASR system presents.

**Index Terms**: ASR, End-to-end, Listen Attend and Spell, connectionist temporal classification, forced alignment

## 1. Introduction

E2E systems/models have produced favorable results on ASR tasks [1, 2, 3, 4, 5, 6]. However, as these E2E systems are usually incapable of emitting word timings, they are not yet ready to replace conventional hybrid systems in transcription applications such as computer-assisted language learning and video subtitles. Most E2E systems are trained simply for aligning input and output sequences. For example, based on the attention mechanism, LAS learned the soft alignments between character outputs and audio signal [2]. Then hard monotonic attention [7] and monotonic chunkwise attention [8] were proposed for more reasonable alignments compared with soft alignments. The CTC loss [9] and RNN-T loss [5] introduced blank labels for alignments. Further, alignments generated from an HMM-based hybrid system were introduced to the RNN-T loss for better sentence alignments [10] and word alignments [11]. However, word timings have more stringent requirements for alignments, that is, each word corresponds to a continuous and monotonic audio signal. So there remains a gap between alignments and word timings.

Recently, an E2E system is reported to emit word timings better than an HMM-based hybrid system *on-device* [12]. In this work, the attention probabilities of LAS are constrained by word alignments from the HMM-based hybrid system to achieve a promising accuracy of word timings.

In other words, systems that are trained with alignments from HMMs, emit word timings well, but the training of an HMM is tedious and time-consuming. Inspired by the spikes generated from CTC models [13], we assume that alignments can also be generated from CTC models instead of HMMs. We consider LAS for word timings, but CTC is more sensitive to the time axis than LAS when decoding. In that way, as the training of E2E systems free of HMMs is relatively labor-saving, it is probable that using E2E systems for emitting word timings will be ready to be applied to many more languages as soon as possible.

In this paper, we propose a novel objective function to train the frame-level-based word-piece classifier with alignments from CTC models instead of HMMs. In consideration of the difference between word timings and speech recognition, different word-piece sets are used with LAS and CTC.

Our experiments are conducted on the LibriSpeech corpus. We find that spike timings generated from LAS decrease the spike accuracy. Spike timings extended automatically are more robust than those extended artificially. Using different word-piece sets in LAS and CTC results in better word timing metrics and identical word accuracy. Compared with the TDNN ASR system, our HMM-free E2E system improves word accuracy and predicts the start time of words comparably.

The rest of this paper is organized as follows. The HMM-free E2E system is presented in Section 2. Section 3 describes the detail in emitting word timings. Experiments and results are presented in Section 4 and Section 5, respectively. Finally, Section 6 concludes the paper and discusses future work.

## 2. System Overview

The E2E system used in this work is illustrated in Figure 1. The system is divided into LAS and CTC parts. The LAS part is an attention-based encoder-decoder architecture, and the CTC part contains two dense layers as frame-level-based classifiers. The input is denoted as $\mathbf{x} = (x_1, ..., x_T)$, where $x_t \in \mathbb{R}^{80}$ is a vector of Mel-filter banks and $T$ is the length of the input sequence. The same ground truth transcript is separately tokenized into $\mathbf{y}_{\text{LAS}} = (y_1, ..., y_U)$ for LAS and $\mathbf{y}_{\text{CTC}} = (y_1, ..., y_{U'})$ for CTC, where $y_u \in \mathcal{W}_{\text{LAS}}$ and $y_{u'} \in \mathcal{W}_{\text{CTC}}$. $U$ and $U'$ are the length of $\mathbf{y}_{\text{LAS}}$ and $\mathbf{y}_{\text{CTC}}$, respectively. $\mathcal{W}_{\text{LAS}}$ and $\mathcal{W}_{\text{CTC}}$ are the sets of word-piece [14] units, where a word-piece starting with a meta symbol (U+2581) indicates the beginning of a word. The hidden state is denoted as $\mathbf{h}^{\text{enc}} = (h_1, ..., h_T)$, which is generated from the encoder. The LAS part produces $P(y_u|x, y_{i<u})$

[*] work done during internship at ByteDance
[†] corresponding author



http://dx.doi.org/10.21437/Interspeech.2021-894

for each output step $u$ where $y_{i<u}$ is the ground truth of previous word-pieces. The CTC and CE dense layers transform $\mathbf{h}^{\text{enc}}$ into posterior probabilities $p$ and $p_{\text{guided}}$ using the softmax function, respectively. The $p$ and $p_{\text{guided}}$ belongs to $\mathbb{R}^{|\mathcal{W}_{\text{CTC}} \cup \{-\}| \times T}$, and $-$ is blank label.

Training procedures are shown as follows:

1. The LAS part is trained to maximize $P(y_u|x, y_{i<u})$.
2. The LAS part is trained to minimize the expected word error rate $\mathcal{L}_{\text{werr}}^{N\text{-}best}(\mathbf{x}, \mathbf{y}_{\text{LAS}})$ [15].
3. LAS is frozen, and the CTC dense layer is trained to minimize the CTC loss $\mathcal{L}_{\text{CTC}}(p, \mathbf{y}_{\text{CTC}})$ [9]
4. LAS and the CTC dense layer are frozen. The CE dense layer is trained to minimize the objective function $\mathcal{L}_{\text{CE}}(p, p_{\text{guided}}, \mathbf{y}_{\text{CTC}})$ mentioned in section 3.

There are two stages in decoding. First, LAS makes the top-1 hypothesis $\mathbf{y}_{\text{LAS}}^*$ which will be detokenize into text. Second, the text is tokenized into labels for CTC $\mathbf{y}_{\text{CTC}}^*$, and then word timings are emitted by a forced alignment on $\mathbf{y}_{\text{CTC}}^*$ and $p_{\text{guided}}$. Note that the CTC dense layer does not participate in the decoding.

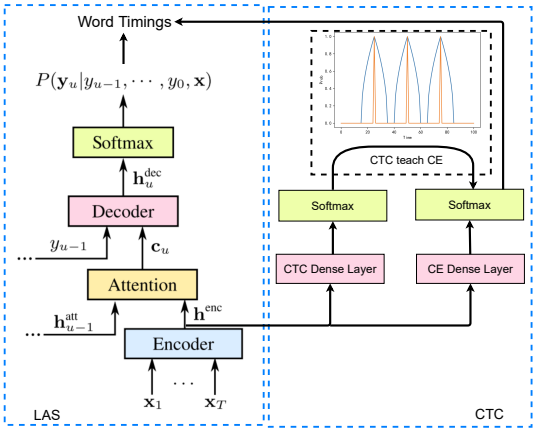


Figure 1: *HMM-free E2E architecture*

## 3. Word Timing System

Each word is tokenized into word-pieces because word-pieces bring better word accuracy, faster decoding [16], and contextual biasing [17]. After tokenizing, emitting word timings is transformed into emitting word-piece timings.

In this work, we emit word-piece timings in two steps. The first step is to emit the spike timing, which is the most likely position of each word-piece, and the second step is to extend the spike timing, which is the duration of each word-piece. Given word-piece timings, the start and end times of each word equal the start time of the first word-piece and the end time of the last word-piece, respectively.

### 3.1. Emitting spike timings

Two methods are used to emit spike timings $\mathbf{t}_{\text{spike}}$. The first method is based on the attention probabilities of LAS, and the second method is based on the posterior probabilities of CTC.

LAS learns the soft alignments explicitly through attention probabilities [2]. LAS generates a $U \times T$ matrix $a(u, t)$ across all labels $\mathbf{y}_{\text{LAS}} = (y_1, ..., y_U)$ and the hidden state $\mathbf{h}^{\text{enc}} = (h_1, ..., h_T)$. According to the attention mechanism, $y_u$ pays more attention to $h_t$ with a larger $a(u, t)$. The spike timings $\mathbf{t}_{\text{spike}}$ for each output step $u$ are defined as

$$t_{\text{spike}}^u = \underset{t}{\text{argmax}}\, a(u, t) \tag{1}$$

Note that there is no constraint on $a(u, t)$ when decoding. Therefore, for monotonic spike timings, post-processing is used to make $t_{\text{spike}}^u = t_{\text{spike}}^{u-1}$ when $t_{\text{spike}}^u < t_{\text{spike}}^{u-1}$.

Inspired by the CTC-attention model [4], an additional CTC loss is introduced into the system for spikes. The CTC dense layer generates a $S \times T$ matrix $p(s, t)$ across all word-pieces $w = (w_1, ..., w_S)$ and the hidden state $\mathbf{h}^{\text{enc}} = (h_1, ..., h_T)$, where $w_s$ belongs to $\mathcal{W}_{\text{CTC}} \cup \{-\}$. In order to emit spike timings through $p(s, t)$, a function $F$ is proposed to remove blank labels in the input sequence, i.e.

$$F(--y_1 - y_2 - -y_3 -) = y_1 y_2 y_3 \tag{2}$$

According to the conditional independence assumption in the CTC loss, the intermediate sequence $\boldsymbol{\pi} = (\pi_1, ..., \pi_T)$ is computed by Equation 3 with the Viterbi algorithm, where $\pi_t$ belongs to $\mathcal{W}_{\text{CTC}} \cup \{-\}$.

$$\boldsymbol{\pi} = \underset{F(\boldsymbol{\pi})=\mathbf{y}_{\text{CTC}}}{\text{argmax}}\, P(\boldsymbol{\pi}|\mathbf{x}) \approx \underset{F(\boldsymbol{\pi})=\mathbf{y}_{\text{CTC}}}{\text{argmax}} \sum_t \log p(\pi_t, t) \tag{3}$$

Finally, each spike timing $t_{\text{spike}}^{u'}$ is generated from the index of the intermediate sequence $\boldsymbol{\pi}$ (i.e., $\mathbf{t}_{\text{spike}} = (3, 5, 8)$ in Equation 2).

### 3.2. Extending spike timings

It is unreasonable to give each spike a fixed duration because word-pieces have various lengths. We assume that each extended spike timing is related to its neighboring spikes. Two methods are proposed to extend spike timings.

The start and end times of each artificially extended spike is defined as follows:

$$t_{\text{start}}^{u'} = t_{\text{spike}}^{u'} - \alpha_{\text{left}} \times (t_{\text{spike}}^{u'} - t_{\text{spike}}^{u'-1}) \tag{4}$$

$$t_{\text{end}}^{u'} = t_{\text{spike}}^{u'} + \alpha_{\text{right}} \times (t_{\text{spike}}^{u'+1} - t_{\text{spike}}^{u'}) \tag{5}$$

where $\alpha_{\text{left}}$ and $\alpha_{\text{right}}$ are hyperparameters that control the weight for the distance between continuous spikes. Especially, $t_{\text{spike}}^0 = 0$ and $t_{\text{spike}}^{U'+1} = T$.

For more robust, spike timings are extended automatically with an objective function. The posterior probabilities $p_{\text{guided}}$ and $p$ have a same size, but $p_{\text{guided}}$ learns the additional duration of word-pieces.

In the training step, the CE dense layer is optimized using a cross-entropy criterion.

$$\mathcal{L}_{\text{CE}}(p, p_{\text{guided}}, \mathbf{y}_{\text{CTC}}) = \sum_s \sum_t l(s, t) \log p_{\text{guided}}(s, t) \tag{6}$$

The soft labels $l$, which are generated from the CTC dense layer instead of HMMs, teach the CE dense layer to automatically extend spike timings. We assume that the further away from the spike, the smaller the non-blank label $l(y_{u'}, t)$ and the larger the blank label $l(-, t)$. Based on this assumption, the non-blank labels $l(y_{u'}, t)$ are computed by Equation 7 for each output step $u'$, where $\beta$ is hyperparameter control the exponent

for non-blank labels, and the rest of non-blank labels is set to zero.

$$l(y_{u'}, t) = \begin{cases} (\frac{t - t^{u'}_{\text{start}}}{t^{u'}_{\text{spike}} - t^{u'}_{\text{start}}})^{\beta} & t^{u'}_{\text{start}} \leq t < t^{u'}_{\text{spike}} \\ (\frac{t^{u'}_{\text{end}} - t}{t^{u'}_{\text{end}} - t^{u'}_{\text{spike}}})^{\beta} & t^{u'}_{\text{spike}} \leq t \leq t^{u'}_{\text{end}} \end{cases} \quad (7)$$

The blank label $l(-, t)$ for each time $t$, which is related to non-blank labels, is computed from Equation 8 as:

$$l(-, t) = \max(\gamma \times (1 - \sum_{w \in \mathcal{W}_{\text{CTC}}} l(w, t)), 0) \quad (8)$$

where $\gamma$ is hyperparameters that control the weight for blank labels.

In the decoding step, word-piece timings are emitted directly with the $p_{\text{guided}}$ instead of $p$. First, the top-1 hypothesis from LAS $\mathbf{y}^*_{\text{LAS}}$, which will be converted to $\mathbf{y}^*_{\text{CTC}}$. A function $F'$ mentioned in [9] removes consecutive duplicate word-pieces and blank labels in the input sequence, i.e.

$$F'(--y_1y_1 - y_2 - -y_3-) = y_1y_2y_3 \quad (9)$$

Similar to Equation 3, the new intermediate sequence $\boldsymbol{\pi}' = (\pi'_1, ..., \pi'_T)$ is computed as

$$\boldsymbol{\pi}' = \underset{F'(\boldsymbol{\pi}')=\mathbf{y}^*_{\text{CTC}}}{\text{argmax}} P(\boldsymbol{\pi}'|\mathbf{x}) \approx \underset{F'(\boldsymbol{\pi}')=\mathbf{y}^*_{\text{CTC}}}{\text{argmax}} \sum_t \log p_{\text{guided}}(\pi_t, t) \quad (10)$$

The new $\mathbf{t}'_{\text{start}}$ and $\mathbf{t}'_{\text{end}}$ are generated from the index of $\boldsymbol{\pi}'$ (i.e., $\mathbf{t}'_{\text{start}} = (3, 6, 9)$ and $\mathbf{t}'_{\text{end}} = (4, 6, 9)$ in Equation 9). Note that word-piece timings are estimated with Viterbi algorithm through $p$ and $p_{\text{guided}}$ in the training and decoding, respectively.

### 3.3. Developing different word-piece sets

There is no need to keep the same word-piece set for both LAS and CTC, as long as the label of LAS $\mathbf{y}_{\text{LAS}}$ can be converted to the label of CTC $\mathbf{y}_{\text{CTC}}$. For a lower time variance among word-pieces, the length of word-pieces is limited to $L_{\text{max}}$ in CTC.

## 4. Experiments

### 4.1. Datasets

We use the LibriSpeech 960h [18] for training. We test the system on `test-clean` and `test-other`. The 80-dimensional Mel-filter bank features are used as input. The transcripts are tokenized as labels using the SentencePiece Model [19], which is an implementation of word-piece. The word-piece set is constructed using the training set transcripts. The ground truth word timings are generated by the forced alignment method with a HMM-based hybrid system.

### 4.2. Modeling

In this paper, we employ LAS, a recurrent neural network encoder-decoder architecture with the attention mechanism. Two additional dense layers are used as the CTC part. The Mel-filter bank is computed with a 25ms window and shifted every 10ms. For more speedy training, the input sequences are batched according to the length of utterances. In the encoder, two CNN kernels (3, 3, 1, 32) and (3, 3, 32, 32) with (2,2) stride are used to extract high-level features and downsampling by 4 to a 40ms frame rate. Then, the output of CNN is fed into 6 BLSTM layers, which has 1024 hidden units in each direction (2048 per layer) followed by a 768-dimensional projection layer. The LAS decoder consists of multi-head attention [20] with two attention heads where each head has 512 hidden units, computing context vectors which are fed into 4 LSTM layers of 1024 hidden units without projection. The LAS decoder has an embedding layer of 96 units and is trained to predict word-pieces in $|\mathcal{W}_{\text{LAS}}|$. In the CTC and CE dense layers, two fully connected layers use RELU as an activation function, which has 1024 hidden units and output size of $|\mathcal{W}_{\text{CTC}} \cup \{-\}|$.

The HMM-free E2E system is trained in Tensorflow [21] using the Lingvo [22] toolkit on 8 Tesla V100 GPUs. The DNN-HMM hybrid system is trained in Kaldi [23] using the TDNN model[1].

The parameters $(\alpha_{\text{left}}, \alpha_{\text{right}}, \beta, \gamma)$ in Equations 4, 5, 7, and 8 are all tuned. We find $(0.2, 0.7, 0.5, 0.5)$ to be an optimal value. Beam search is conducted with a beam width of 8. The behavior when changing the word-piece sets (i.e., $\mathcal{W}_{\text{LAS}}$ and $\mathcal{W}_{\text{CTC}}$) will be explored in Section 5.3.

### 4.3. Training configuration

The learning rate has three periods, which are increasing linearly from the initial learning rate, holding the value of peak learning rate, then exponentially decaying by half every 100k steps. This schedule is parameterized by five time stamps $(s_{\text{hold}}, s_{\text{decay}}, s_{\text{stop}}, s_{\text{specaug}}, s_{\text{noise}},)$ -the step $s_{\text{hold}}$ where the ramp-up is complete, the step $s_{\text{decay}}$ where the learning rate begins descending, the step $s_{\text{stop}}$ where the training stops, the step $s_{\text{specaug}}$ where the SpecAugment [24] is turned on $(F, m_F, T, p, m_T) = (27, 1, 100, 0.1, 1)$, the step $s_{\text{noise}}$ where the variational weight noise [25] is turned on of standard deviation 0.075. In LAS training, using $(5k, 50k, 200k, 2k, 10k)$ as the schedule, the initial learning rate is 2.5e-4, and the peak is 7.5e-4. In MWER, CTC, and CE training, the schedules are $(0, 30k, 30k, 2k, 10k)$, $(0, 30k, 30k, 2k, 10k)$, and $(0, 10k, 10k, 2k, 10k)$, and the learning rates are fixed to 1e-6, 2.5e-4 and 2.5e-4, respectively. In LAS and MWER training, the uniform label smoothing [26] is introduced with uncertainty 0.01. The top-4 hypotheses are used when MWER training.

### 4.4. Measuring word timings

The word timings metrics come from [12]. Average Start Time Delta (Ave. ST$\Delta$) is the average offset on the start time of words between the ground truth and the prediction. Percentage of Word Start Times Less than 200ms (% WS$<$200ms) means that the offset of the start time is less than 200ms. It is similar for the end time of words to compute Ave. ET$\Delta$ and %WE$<$200ms.

## 5. Results

### 5.1. Emitting spike timings

Considering the word timing metrics, the definition of spike accuracy is that the percentage of words whose all spikes are within the ground truth word timing with 200ms tolerance. Table 1 shows that CTC spikes are more accurate than LAS spikes. This is an important step to show the reason for introducing the additional CTC loss.

### 5.2. Extending spike timings

The word boundaries are not well predicted by spikes because spikes only last one frame. Table 2 shows performance of spikes is extended artificially ($A0$) and automatically ($B0$). The results

[1] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5

Table 1: *Spike accuracy on the test set*

| | CTC spikes | LAS spikes |
|---|---|---|
| clean | **99.18%** | 96.72% |
| other | **98.92%** | 95.24% |

show that the $B0$ has a better %WS/WE<200ms. The following experiments base on the $B0$ for the better accuracy of word boundaries.

Table 2: *Word timing: artificial vs. automatic*

| | clean | | other | |
|---|---|---|---|---|
| ID | $A0$ | $B0$ | $A0$ | $B0$ |
| Ave. ST$\Delta$ | 62.7 | **60.7** | 64.3 | **61.4** |
| Ave. ET$\Delta$ | **68.5** | 78.2 | **72.1** | 84.6 |
| %WS<200ms | **94.6** | **94.6** | 95.0 | **95.3** |
| %WE<200ms | 92.3 | **93.9** | 91.9 | **92.2** |

### 5.3. Introducing different word-piece sets

In this section, we explore both WER and word timing metrics when using three word-piece configurations $(L_{\max}, N_{\text{size}})$ that are $(-, 7000), (5, 7000)$, and $(3, 3972)$, where $-$ means an unlimited length. The $L_{\max}$ is the maximum length of word-pieces, and the $N_{\text{size}}$ is the size of the word-piece set. Table 3 shows that the results of different word-piece sets used in CTC and LAS. The $B0, B1$, and $B2$ show that as the $L_{max}$ decreases, word timing metrics decrease but WERs increases. The $B0, B3$, and $B4$ show that using different $L_{\max}$ in CTC and LAS does not result in the WER degradation but the word timing metrics improvement. The $B4$ shows a better WER and word timing metrics.

Table 3: *Word timing for different word-piece sets*

| ID | | B0 | B1 | B2 | B3 | B4 |
|---|---|---|---|---|---|---|
| LAS | $N_{\text{size}}$ | 7000 | 7000 | 3972 | 7000 | 7000 |
| | $L_{\max}$ | - | 5 | 3 | - | - |
| CTC | $N_{\text{size}}$ | 7000 | 7000 | 3972 | 7000 | 3972 |
| | $L_{\max}$ | - | 5 | 3 | 5 | 3 |
| clean | WER | **2.8** | 2.9 | 3.2 | **2.8** | **2.8** |
| | Ave. ST$\Delta$ | 60.7 | 45.6 | 48.7 | 43.7 | **41.2** |
| | Ave. ET$\Delta$ | 78.2 | 79.3 | 72.2 | **65.4** | 67.1 |
| | %WS<200ms | 94.6 | **99.0** | 98.8 | 98.8 | **99.0** |
| | %WE<200ms | 93.9 | 93.7 | **96.5** | 95.3 | 95.3 |
| other | WER | **7.0** | 7.4 | 8.4 | **7.0** | **7.0** |
| | Ave. ST$\Delta$ | 61.4 | 49.5 | 53.3 | 46.4 | **45.6** |
| | Ave. ET$\Delta$ | 84.6 | 85.3 | 78.1 | **72.7** | 74.5 |
| | %WS<200ms | 95.3 | **98.6** | 98.5 | 98.5 | **98.6** |
| | %WE<200ms | 92.2 | 92.1 | **95.0** | 93.6 | 93.7 |

### 5.4. Analysis

#### 5.4.1. Comparing with the conventional system

We compare the word timing metrics and WER of the E2E system to the TDNN ASR system in Table 4. The table shows that the E2E system has a significant improvement in WER and a comparable %WS<200ms but a decreased %WE<200ms. One hypothesis for the poor prediction of the end time is that the E2E system is incapable of detecting silence. Therefore, the E2E system can not distinguish between silence and words.

Table 4: *Word timing: HMM-free vs. conventional*

| | clean | | other | |
|---|---|---|---|---|
| System | Conv | HMM-free | Conv | HMM-free |
| WER | 4.0 | 2.8 | 9.9 | 7.0 |
| Ave. ST$\Delta$ | 27.2 | 41.2 | 30.7 | 45.6 |
| Ave. ET$\Delta$ | 28.1 | 67.1 | 33.0 | 74.5 |
| %WS<200ms | 99.0 | 99.0 | 98.7 | 98.6 |
| %WE<200ms | 99.1 | 95.3 | 98.7 | 93.7 |

#### 5.4.2. Visualizing word timings

Figure 2 shows word timings generated from the HMM-free E2E system (red) and the ground truth (black) for a specific utterance on `test-clean`. It is found that the $B4$ (top) emits "GENTLEMAN" timings better than the $B0$ (bottom). When checking the word-piece set, we find that the word "GENTLEMAN" is tokenized into one word-piece when the $L_{\max}$ is unlimited but several word-pieces when the $L_{\max}$ is three.

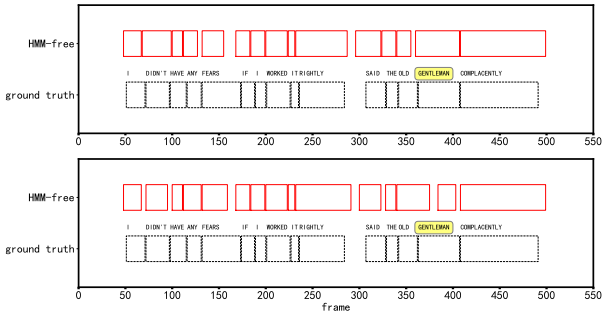


Figure 2: *Word timing for ground truth (black) and E2E systems(red). The B4 (top) and the B0 (bottom).*

## 6. Conclusion

We have introduced a novel, general objective function for emitting word timings based on the CTC-attention-based system without HMMs. Our method predicts a result using LAS and then emits word timings using the automatically extended spikes from CTC. The HMM-free E2E system is comparable to the conventional hybrid system in emitting word start timing and has a better word accuracy. In the future, a Voice activity detection system will be introduced to help the CE dense layer detect silence for better accuracy of word timings, and this work will be applied to more languages.

## 7. Acknowledgements

Thank you to Haihua Xu, Bilei Zhu, Wangqian Zhou, Meng Cai, Zhengkun Tian, Lu Huang, and Jun Zhang for helpful discussions and other members who helped me in ByteDance AI Lab S&A.

# 8. References


[1] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International conference on machine learning*. PMLR, 2014, pp. 1764–1772.

[2] 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)*. IEEE, 2016, pp. 4960–4964.

[3] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[4] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4835–4839.

[5] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[6] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S.-y. Chang, W. Li, R. Alvarez, Z. Chen *et al.*, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6059–6063.

[7] C. Raffel, M.-T. Luong, P. J. Liu, R. J. Weiss, and D. Eck, "Online and linear-time attention by enforcing monotonic alignments," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2837–2846.

[8] C.-C. Chiu and C. Raffel, "Monotonic chunkwise attention," in *International Conference on Learning Representations*, 2018.

[9] 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.

[10] 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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6069–6073.

[11] 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," *arXiv preprint arXiv:2011.03072*, 2020.

[12] T. N. Sainath, R. Pang, D. Rybach, B. Garcıa, and T. Strohman, "Emitting word timings with end-to-end models," *Proc. Interspeech 2020*, pp. 3615–3619, 2020.

[13] 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.

[14] M. Schuster and K. Nakajima, "Japanese and korean voice search," in *International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5149–5152.

[15] 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)*. IEEE, 2018, pp. 4839–4843.

[16] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[17] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-fusion end-to-end contextual biasing." in *Interspeech*, 2019, pp. 1418–1422.

[18] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[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] 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 *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[21] 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/

[22] J. Shen, P. Nguyen, Y. Wu, Z. Chen *et al.*, "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," 2019.

[23] 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.

[24] 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. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2680

[25] A. Graves, "Practical variational inference for neural networks," in *Advances in neural information processing systems*. Citeseer, 2011, pp. 2348–2356.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.