INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Bayes Risk Transducer: Transducer with Controllable Alignment Prediction

Jinchuan Tian[1], *Jianwei Yu*[1,2], *Hangting Chen*[1], *Brian Yan*[3], *Chao Weng*[1,2], *Dong Yu*[1], *Shinji Watanabe*[3]

[1]Tencent AI LAB, [2]Tencent ASR Oteam,
[3]Language Technologies Institute, Carnegie Mellon University

tomasyu@tencent.com, swatanab@andrew.cmu.edu

Abstract

Automatic speech recognition (ASR) based on transducers is widely used. In training, a transducer maximizes the summed posteriors of all paths. The path with the highest posterior is commonly defined as the predicted alignment between the speech and the transcription. While the vanilla transducer does not have a prior preference for any of the valid paths, this work intends to enforce the preferred paths and achieve controllable alignment prediction. Specifically, this work proposes Bayes Risk Transducer (BRT), which uses a Bayes risk function to set lower risk values to the preferred paths so that the predicted alignment is more likely to satisfy specific desired properties. We further demonstrate that these predicted alignments with intentionally designed properties can provide practical advantages over the vanilla transducer. Experimentally, the proposed BRT saves inference cost by up to 46% for non-streaming ASR and reduces overall system latency by 41% for streaming ASR.[1] [2]

Index Terms: speech recognition, transducer, alignment

1. Introduction

Automatic speech recognition (ASR) based on transducers [1] is one of the most popular frameworks [2, 3, 4]. In the past few years, a series of approaches have been proposed as extensions of the transducer with the goals of optimizing its recognition accuracy [5, 6], language model integration [7], flexibility [8], decoding efficiency [9] and simplicity [10, 11], memory efficiency during training [12, 13, 14] and overall system latency during streaming decoding [15, 16, 17, 18, 19, 20]. During training, the vanilla transducer, as well as its extensions [9, 12, 13, 15, 16, 17, 20], maximizes the summed posterior of all potential aligning sequences (a.k.a, *paths*) between the speech and the transcription. In particular, these extensions achieve their goals by manipulating the transducer paths, such as allowing multi-frame big skips [9], pruning paths with minor posteriors with [13, 17, 20] or without [12] reference alignment labels, discouraging blank emissions [15] and encouraging non-blank emissions [16]. This work provides another extension of the transducer, which also conducts manipulation over paths. Specifically, as a follow-up of the previous work [21] which attempts to achieve controllable alignment prediction in CTC criterion [22], this work extends this controllability to the transducer model by taking its distinctive forward-backward process into consideration.

The alignment prediction of the transducer is commonly defined as the path with the highest posterior. In vanilla transducer formulation, there is no prior preference among the paths since predicting each valid path will yield the correct textual transcription. Currently, the alignment selection among the paths (i.e., which path will become the predicted alignment) can hardly be affected by human intervention during training. To achieve controllable alignment prediction in the transducer is exactly to intentionally choose paths with specific desired properties as the alignment prediction. With this motivation, this work proposes an extension of the transducer called Bayes Risk Transducer (BRT), which adopts a Bayes risk function to intentionally enforce a preference for paths with the desired properties, so that the predicted alignments are more likely to be characterized by these properties. Particularly, the original forward-backward algorithm of the transducer is revised into a divide-and-conquer approach: all paths are firstly divided into multiple exclusive groups and the groups with more favored properties are enforced by receiving lower risk values than the others.

This work further demonstrates that BRT with controllable alignment prediction has practical advantages over vanilla transducers. By designing various Bayes risk functions, we can obtain alignment predictions with desired properties that are specific to different task setups, which subsequently helps to offer novel solutions for two practical challenges in ASR: inference cost for non-streaming ASR and overall system latency for streaming ASR. In the non-streaming setup, a Bayes risk function is designed to enforce the paths that emit the last non-blank predictions earlier. As a benefit, the last non-blank prediction occurs at an early time stamp so the inference cost can be reduced by terminating the decoding loop early without exploring all frames. In the streaming setup, another Bayes risk function is designed to encourage early emissions for all non-blank tokens. Thus, the model emits before waiting too long context and the latency for each non-blank token is reduced. Experimentally, the former case accelerates non-streaming inference by up to 46% and the latter case reduces the overall system latency of streaming ASR system by 41%.

2. Bayes Risk Transducer

2.1. Vanilla Transducer

In training, vanilla transducer maximizes the posterior of the transcription $\mathbf{l} = [l_1, ..., l_U]$ with the given acoustic feature sequence $\mathbf{x} = [\mathbf{x}_1, ..., \mathbf{x}_T]$. Instead of maximizing $P(\mathbf{l}|\mathbf{x})$ directly, the transducer maximizes the summed posterior of all *paths* in the transducer *lattice* (see Fig.1.a). Note $\varnothing$ as the blank symbol and extend the vocabulary $\hat{\mathcal{V}} = \mathcal{V} \cup \{\varnothing\}$, each symbol sequence $\pi = [\pi_1, ..., \pi_{T+U}]$ is a valid path if all entries of π are in $\hat{\mathcal{V}}$ and $\mathcal{B}(\pi) = \mathbf{l}$. Here $\mathcal{B}$ is a mapping that removes all $\varnothing$. So the vanilla transducer objective to minimize is defined as:

$$\mathbf{J}_{\text{transducer}}(\mathbf{l}, \mathbf{x}) \triangleq -\log P(\mathbf{l}|\mathbf{x}) = -\log \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} P(\pi|\mathbf{x}) \quad (1)$$

where $\mathcal{B}^{-1}(\mathbf{l})$ is the set of all valid paths. Next, the posterior of each path $P(\pi|\mathbf{x})$ is computed as:

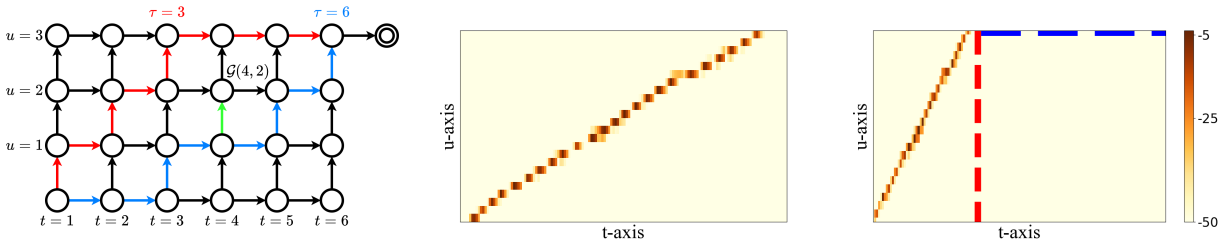
$$P(\pi|\mathbf{x}) = \prod_{i=1}^{T+U} p(\pi_i|\mathbf{x}_{1:t}, \mathbf{l}_{1:u}) \quad (2)$$

where the condition $(\mathbf{x}_{1:t}, \mathbf{l}_{1:u})$ specifies the node (t, u) on the transducer lattice s.t. $\mathcal{B}(\pi_{1:i}) = \mathbf{l}_{1:u}$ and $t + u = i - 1$. Instead of enumerating all paths and summing their

[1]We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.

[2]Code available: https://github.com/espnet/espnet

10.21437/Interspeech.2023-1342



(a) Transducer Lattice (b) $\log \mathcal{G}(t,u)$ - Vanilla Transducer (c) $\log \mathcal{G}(t,u)$ - BRT

Figure 1: *(a): Transducer lattice. $\mathcal{G}(4,2)$ is the summed posterior of all paths that go through the vertical arrow (in green) from node $(4,1)$ to node $(4,2)$ and emit 2nd token at the 4th frame. The red path ends all non-blank predictions at $\tau = 3$ while the blue path ends at $\tau = 6$. The red path is preferred in Sec.3.2. (b) & (c): The heat maps for* $\log \mathcal{G}(t,u)$.

posteriors, the transducer objective is computed efficiently by *forward-backward algorithm* [23], which recursively computes the forward-backward variables $\alpha(t,u)$ and $\beta(t,u)$ for each node (t,u) in the transducer lattice:

$$\alpha(t,u) = \sum_{\pi \in \mathcal{V}'^{(T+U)}, \mathcal{B}(\pi_{1:t+u}) = \mathbf{l}_{1:u}} P(\pi|\mathbf{x}) \quad (3)$$

$$\beta(t,u) = \sum_{\pi \in \mathcal{V}'^{(T+U)}, \mathcal{B}(\pi_{t+u+1:T+U}) = \mathbf{l}_{u+1:U}} P(\pi|\mathbf{x}) \quad (4)$$

Subsequently, by decomposing each path π into partial paths $\pi_{1:t+u}$ and $\pi_{t+u+1:T+U}$ and using Eq.{3, 4}, the transducer objective is derived as[3]:

$$\begin{aligned}\mathbf{J}_{\text{transducer}}(\mathbf{l},\mathbf{x}) &= -\log \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} P(\pi|\mathbf{x}) \\ &= -\log \sum_{(t,u):t+u=n} \sum_{\substack{\mathcal{B}(\pi_{1:t+u}) = \mathbf{l}_{1:u} \\ \mathcal{B}(\pi_{t+u+1:T+U}) = \mathbf{l}_{u+1:U}}} P(\pi|\mathbf{x}) \\ &= -\log \sum_{(t,u):t+u=n} \alpha(t,u) \cdot \beta(t,u) \quad (5)\end{aligned}$$

where n is any known integer s.t. $n \in [0, T+U]$. Finally, the path with the highest posterior is usually considered as the alignment prediction between $\mathbf{x}$ and $\mathbf{l}$: $\text{ali}(\mathbf{x},\mathbf{l}) = \arg\max_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} P(\pi|\mathbf{x})$.

2.2. Bayes Risk Transducer

As suggested in Eq.1, the formulation of the vanilla transducer has no prior preference among paths. This work intentionally selects the predicted alignment among the paths and thus attempts to achieve controllable alignment prediction. With this motivation, this work proposes Bayes Risk Transducer (BRT), which adopts a customizable Bayes risk function to express the preference for specific paths with desired properties. To preserve a similar format like Eq.1, we sets the risk function for each path as $-r(\pi)$ so that minimizing the expected risk is equivalent to minimizing the BRT objective [4]:

$$\mathbf{J}_{\text{BRT}}(\mathbf{l},\mathbf{x}) \triangleq -\log \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} [P(\pi|\mathbf{x}) \cdot r(\pi)] \quad (6)$$

The computation of the proposed BRT objective still adopts the forward-backward variables and works in a divide-and-conquer approach. To express the preference for specific property of the paths, the Bayes risk function specifies 1) what property is concerned and 2) what values of the concerned property are preferred. To answer these questions, the concerned property of each path is defined by $f(\pi)$. Then, all paths are divided into multiple exclusive groups s.t. paths with the identical concerned property value τ are in the same groups. For paths in one identical group, it is reasonable to assign the same risk value as the concerned property is the same. Thus, the Bayes risk function $r(\pi)$ is replaced by a group-level risk function $r_g(\tau)$, which only depends on the group-level concerned property τ rather than the path π. Formally, this process is written as:

$$\begin{aligned}\mathbf{J}_{\text{BRT}}(\mathbf{l},\mathbf{x}) &= -\log \sum_{\tau} \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l}), f(\pi)=\tau} [P(\pi|\mathbf{x}) \cdot r(\pi)] \\ &= -\log \sum_{\tau} \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l}), f(\pi)=\tau} [P(\pi|\mathbf{x}) \cdot r_g(\tau)] \\ &= -\log \sum_{\tau} [r_g(\tau) \cdot \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l}), f(\pi)=\tau} P(\pi|\mathbf{x})] \quad (7)\end{aligned}$$

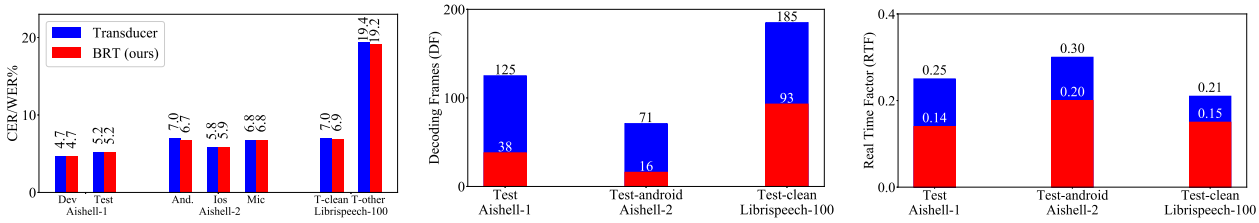
Please be aware of: 1) when splitting the path into groups, the groups are supposed to be mutually exclusive so that each path is considered once and only once; 2) by adopting the group-level risk function, we can avoid the complex weighted summation over all paths within each group. 3) the summed posterior of each path group should be fully tractable by the forward-backward variables so the computation remains efficient; 4) by pursuing the desired properties in the predicted alignment between $\mathbf{x}$ and $\mathbf{l}$ during training, these properties are expected to be preserved in the predicted alignments between the test speech and the textual hypotheses during decoding.

Provided the general formulation of BRT in Eq.7, a naive example is in Eq.5, where the concerned property τ represents the pair (t,u) and $r_g(\tau) = 1$. Under this setting, the vanilla transducer is a special case of the proposed BRT. Alternatively, given the u-th non-blank token l_u in the transcription, another useful example is to set the concerned property τ as the time stamp when l_u is emitted, a.k.a, $\pi_{\tau+u} = l_u$. With a similar factorization like Eq.5 and considering Eq.{2, 3, 4}, the BRT objective is further revised as:

$$\begin{aligned}\mathbf{J}_{\text{BRT}}(\mathbf{l},\mathbf{x},u) &= -\log \sum_{\tau} [r_g(\tau) \cdot \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l}), \pi_{\tau+u}=l_u} P(\pi|\mathbf{x})] \\ &= -\log \sum_{\tau} [r_g(\tau) \cdot \sum_{\substack{\mathcal{B}(\pi_{1:\tau+u-1}) = \mathbf{l}_{1:u-1} \\ \pi_{\tau+u}=l_u \\ \mathcal{B}(\pi_{\tau+u+1:T+U}) = \mathbf{l}_{u+1:U}}} P(\pi|\mathbf{x})] \\ &= -\log \sum_{\tau} r_g(\tau) \cdot \underbrace{[\alpha(\tau,u-1) \cdot p(l_u|\mathbf{x}_{1:\tau}, \mathbf{l}_{1:u-1}) \cdot \beta(\tau,u)]}_{\triangleq\ \mathcal{G}(\tau,u)} \quad (8)\end{aligned}$$

[3]Note, for a path that go through a known node (t,u) in transducer lattice, the partial path $\pi_{t+u+1:T+U}$ is independent to partial path $\pi_{1:t+u}$ so that the factorization of the path posterior is $P(\pi|\mathbf{x}) = P(\pi_{1:t+u}|\mathbf{x}) \cdot P(\pi_{t+u+1:T+U}|\cancel{\pi_{1:t+u}}, \mathbf{x})$.

[4]In the remained part of this work, $r(\pi)$ is termed as the Bayes risk function even though the real Bayes risk function is $-r(\pi)$. Higher $r(\pi)$ value represents a lower risk, a.k.a., preference.



(a) WER / CER% (b) Decoding Frames (DF) (c) Real Time Factor (RTF)

Figure 2: *Results of non-streaming ASR. (a) Recognition accuracy in CER / WER %; (b) Average decoding frames (DF); (c) Real time factor (RTF). With comparable recognition accuracy, the proposed BRT achieve more efficient decoding by decoding fewer frames.*

Here is $\mathcal{G}(\tau, u)$ the summed posterior of all paths that go through the vertical arrow from node $(\tau, u-1)$ to node (τ, u). Fig.1.a gives a demonstration of $\mathcal{G}(\tau, u)$ in the lattice and Fig.1.{b, c} provides numerical examples of $\mathcal{G}(\tau, u)$. $\mathcal{G}(\tau, u)$ measures the summed probability of all valid paths that l_u is emitted at τ-th frame, which indicates the alignment prediction. So far the group-level risk function $r_g(\tau)$ is not defined. Below we show two applications of Eq.8 with different $r_g(\tau)$ designs.

2.3. Non-streaming Application: Efficient Decoding

For frame-synchronized decoding algorithms of the transducer [1, 24], the inference cost highly depends on T as the decoding loop is conducted frame-by-frame. In Fig.1.b, the whole sequence $\mathbf{l}$ cannot be predicted until all frames are explored. By contrast, in Fig.1.c, all non-blank tokens are emitted before reaching the red line, which allows us to stop decoding at an early time stamp (e.g., the red line) to save computation.

To achieve the heat map like Fig.1.c, the concerned property is exactly the time stamp τ when the last token l_U, as well as the whole sequence, is emitted: $\pi_{\tau+U} = l_U$. In addition, paths with smaller τ are preferred (see Fig.1.a) since fewer frames are consumed to predict all tokens. Set $u = U$ in Eq.8, the objective to minimize is:

$$\mathbf{J}_{\text{BRT}}(\mathbf{l}, \mathbf{x}, U) = -\log \sum_{\tau} \min(e^{-\lambda \cdot (\tau - m \cdot U)/T}, 1) \cdot \mathcal{G}(\tau, U) \tag{9}$$

where the risk function $r_g(\tau) = \min(e^{-\lambda \cdot (\tau - m \cdot U)/T}, 1)$ expresses the preference for $\tau \in [1, m \cdot U]$ and shows exponentially decayed interest in $\tau > m \cdot U$[5]. λ and m are hyper-parameters. m is empirically set to 2 and λ varies according to datasets.

This work further provides an early-stop mechanism to reduce the number of decoding frames of BRT. First, assume we obtain a hypothesis $\hat{\mathbf{l}} = [\hat{l}_1, ..., \hat{l}_u]$ at τ-th frame during decoding. The hypothesis is considered complete if no additional non-blank tokens are expected to be emitted in the search process over the remaining frames after τ. In other words, for any possible path of the complete $\hat{\mathbf{l}}$, its sub-path after the τ-th frame only consists of continuous $\varnothing$ (see the blue line in Fig.1.c). So $\hat{\mathbf{l}}$ is considered complete only if the accumulated probability of the continuous $\varnothing$ since the τ-th frame are with high confidence: $\sum_{t=\tau}^{T} \log p(\varnothing | \mathbf{x}_{1:t}, \hat{\mathbf{l}}_{1:u}) > D$, where $D = -10$ is a threshold value[6]. Secondly, for a search beam that contains multiple hypotheses, we terminate the search when 1) the top $k = 3$ blank-free hypotheses do not change for $f = 5$ frames and 2) all top $k = 3$ hypotheses are considered complete.

[5]We do not express an extra preference for very small τ so it is less likely that multiple non-blank tokens are emitted at a single frame.

[6]The computation for this condition cannot be considered as a search over the frames after τ since the series $\{p(\varnothing | \mathbf{x}_{1:t}, \hat{\mathbf{l}}_{1:u})\}$ can be computed in parallel fashion and requires no loop.

2.4. Streaming Application: Early Emission

A streaming ASR system is expected to emit each token accurately and timely. The accuracy and latency, however, usually form a trade-off: better recognition accuracy requires longer context, which results in higher latency. For streaming ASR, BRT is designed to encourage all tokens to emit at early time stamps, even at the cost of slight performance degradation. By doing so, BRT achieves a better accuracy-latency trade-off than the vanilla transducer, which is further demonstrated in Sec.3.3.

The vanilla transducer only attempts to transcribe the speech correctly but poses no constraint on when tokens would be emitted. By contrast, the proposed BRT can reduce the latency by enforcing the paths that emit each token at a smaller time stamp. Formally, with any non-blank token l_u and the exponentially decayed risk function $r_g(\tau) = e^{-\lambda \cdot (\tau - \tau')/T}$, a BRT objective is derived from Eq.8 with the goal of encouraging l_u to be emitted earlier:

$$\mathbf{J}_{\text{BRT}}(\mathbf{l}, \mathbf{x}, u) = -\log \sum_{\tau} e^{-\lambda \cdot (\tau - \tau')/T} \cdot \mathcal{G}(\tau, u) \tag{10}$$

where τ is the concerned property that specifies the time-stamp when l_u is emitted and is enforced to be smaller; $\tau' = \arg\max_{\tau} \mathcal{G}(\tau, u)$ is a bias term to ensure that the path group with the highest summed posterior $\mathcal{G}(\tau, u)$ would always receive the risk value of $r_g(\tau) = 1$ so that the absolute value of $\mathbf{J}_{\text{BRT}}(\mathbf{l}, \mathbf{x}, u)$ does not vary along with u significantly. Here λ is still an adjustable hyper-parameter varying with datasets. Subsequently, to guide every token l_u to be emitted earlier requires the consideration of all tokens. So we simply attempts to minimize the mean of $\mathbf{J}_{\text{BRT}}(\mathbf{l}, \mathbf{x}, u)$ in Eq.10 over every u:

$$\mathbf{J}(\mathbf{l}, \mathbf{x}) = \frac{1}{U} \cdot \sum_{u=1}^{U} \mathbf{J}_{\text{BRT}}(\mathbf{l}, \mathbf{x}, u) \tag{11}$$

3. Experiments

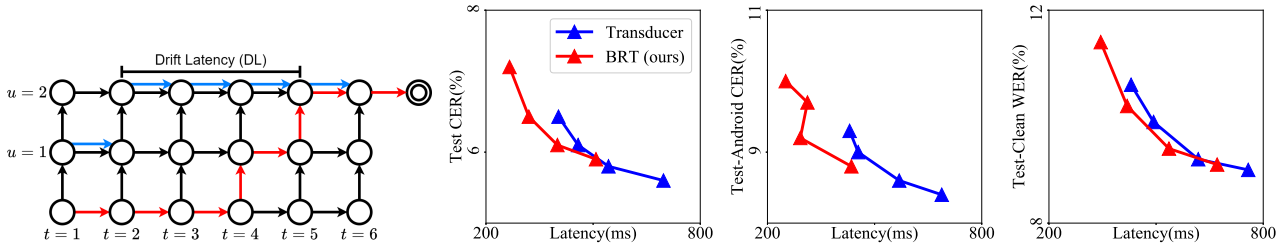
3.1. Experimental Setup

Datasets: Experiments are conducted on Aishell-1 [25], Aishell-2 [26] and Librispeech-100 [27] datasets. The volumes of these datasets range from 100 hours to 1k hours. Librispeech-100 is in English and the others are in Mandarin. All data is augmented by SpecAugment [28] and speed perturbation. For English, tokens are 500 BPE units.

Evaluation Metrics: CER / WER% is adopted to show the recognition accuracy. To compare the decoding efficiency of non-streaming ASR, the average number of decoding frames (DF) before the decoding termination and the real-time factor (RTF) over CPU[7] are reported. For streaming ASR, the overall latency is defined as the sum of data collecting latency (DCL) and drift latency (DL) [21][8]. DCL is the time to wait before the

[7]Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50GHz, single thread.

[8]The latency caused by computation is not considered in this work: it is marginal for a light model and without an external language model.



(a) Demonstration of drift latency (b) Aishell-1 (c) Aishell-2 (d) Librispeech-100

Figure 3: *Results of streaming ASR. (a) A demonstration of drift latency (DL). Blue arrows stand for the reference duration of each token. The alignment of the hypothesis is represented by the red path. Token l_2 starts at 1st frame but is predicted at 4th frame, which is a 3-frame drift latency. (b) & (c) & (d): the accuracy-latency trade-off achieved with varying data collecting latency (DCL)*

input speech forms a chunk (a.k.a., the latency caused by chunk size and look-ahead length). DL is exemplified in Fig.3.a. and its reference[9] is obtained by standard GMM-HMM systems[10].

Features & Models: 80-dim Fbank features with the window size of 10ms are down-sampled by 4x using CNN before being fed into the encoder. The acoustic encoder is Conformer [29] for non-streaming ASR and Emformer [30] for streaming ASR. For streaming ASR, DCL is set to {160, 320, 480, 640}ms. The prediction network is a standard LSTM and the joint network is linear. For English tasks, an auxiliary CTC criterion is adopted on the top of the encoder to stabilize the training. Model sizes for non-streaming and streaming experiments are 95M and 57M respectively.

Training & Decoding: For {Aishell-1, Aishell-2, Librispeech-100}, models are trained for {100, 100, 300} epochs with λ of {5, 5, 20} and {10, 10, 50} for non-streaming and streaming ASR respectively. The original decoding algorithm proposed in [1] is adopted with a beam size of 10. No language model is adopted in decoding.

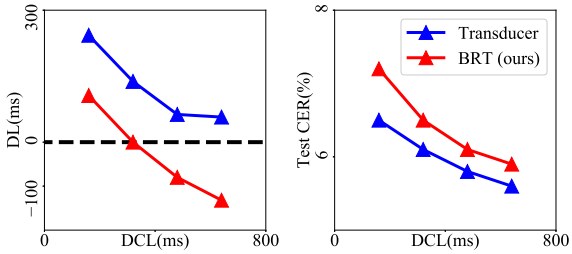
3.2. Results on non-streaming ASR

This part evaluates the effectiveness of the proposed BRT method on non-streaming ASR. Our results are shown in Fig.2. Firstly, as shown in Fig.2.a, with datasets in varying scales and languages, the recognition accuracy achieved by the proposed BRT method and the vanilla transducer are comparable. Secondly, Fig.2.b demonstrates the effectiveness of the proposed BRT in reducing the decoding frames. E.g., by introducing the BRT criterion, the DF for Aishell-2 dataset is reduced from 71 to 16, which is a 77% reduction. Finally, for models trained by BRT, the overall inference cost (a.k.a., RTF) is reduced since the proposed early-stop mechanism allows the model not to explore all the frames. By adopting the BRT criterion and the early-stop mechanism, the RTF of Aishell-1 is reduced from 0.25 to 0.14, which is a 46% relative reduction. The reduction in Fig.2.c is not as considerable as that in Fig.2.b since the encoder inference accounts for a large part of the computation cost.

3.3. Results on streaming ASR

This part evaluates the effectiveness of the proposed BRT method on streaming ASR. Our results are shown in Fig.3. As discussed in Sec.2.4, the streaming ASR has a trade-off between the recognition accuracy and the overall system latency. As shown in Fig.3.{b,c,d}, on the three datasets, the curve of the proposed BRT (the red one) consistently lies in the lower-left direction of its baseline (vanilla transducer, the blue one), which suggested that the proposed BRT criterion achieves better accuracy-latency trade-off than vanilla transducer. In addition, BRT can build systems with extremely low latency that cannot be achieved by the vanilla transducer, even at the cost of recognition performance degradation. E.g., on the Aishell-2 dataset, the lowest overall latency achieved by the vanilla transducer and BRT is 430ms and 251ms respectively, which is a 41% relative reduction in latency, even with accuracy degradation.

Further ablation study is conducted on Aishell-1 dataset. As shown in Fig.4.a, the transducer system with extremely low latency cannot be built by simply reducing the chunk size (a.k.a., small DCL) since the model is allowed to wait for very long context before emitting (a.k.a., larger DL). In addition, the adoption of BRT can effectively reduce the DL, which is aligned with our motivation in Sec.2.4[11]. The BRT model has this strength of early emission since the paths that emit non-blank prediction earlier are enforced during training. Next, Fig.4.b shows that the vanilla transducer outperforms the proposed BRT in accuracy with all DCL settings, which is reasonable since the accessible right context is reduced if the non-blank tokens are emitted earlier. Combining Fig.4.{a,b} will reach Fig.3.b, which demonstrates that: BRT provides an alternative solution for streaming transducer, i.e., increasing DCL with a larger chunk size and reducing DL by using BRT to meet the latency budget so that a better overall accuracy-latency trade-off is achieved.



(a) DCL vs. DL (b) DCL vs. CER

Figure 4: *Ablation study for streaming ASR (on Aishell-1)*

4. Conclusion

To achieve controllable alignment prediction in the transducer, this work proposes an extension of the transducer called Bayes Risk Transducer (BRT), which adopts a Bayes risk function to enforce specific paths with the desired properties. By designing different Bayes risk functions, the predicted alignment is enriched with task-specific properties, which provides practical benefits besides recognizing the speech accurately: efficient decoding for non-streaming ASR and early emission for streaming ASR. The claimed two applications are experimentally validated on multiple datasets and in multiple languages.

[9]For an English word that consists of multiple BPE tokens, we only count the last BPE unit of that word.

[10]Models are trained by Kaldi: https://github.com/kaldi-asr/kaldi

[11]The DL can be negative due to the look-ahead of the model

5. References

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

[2] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schlüter, and S. Watanabe, "End-to-end speech recognition: A survey," *arXiv preprint arXiv:2303.03329*, 2023.

[3] D. Wang, X. Wang, and S. Lv, "An overview of end-to-end automatic speech recognition," *Symmetry*, vol. 11, no. 8, 2019.

[4] J. Li, "Recent advances in end-to-end automatic speech recognition," *arXiv preprint arXiv:2111.01690*, 2021.

[5] H. Sak, M. Shannon, K. Rao, and F. Beaufays, "Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping," *Proc. Interspeech 2017*, pp. 1298–1302, 2017.

[6] T. N. Sainath, R. Pang, D. Rybach, Y. He, R. Prabhavalkar, W. Li, M. Visontai, Q. Liang, T. Strohman, Y. Wu, I. McGraw, and C.-C. Chiu, "Two-Pass End-to-End Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2773–2777.

[7] E. Variani, D. Rybach, C. Allauzen, and M. Riley, "Hybrid autoregressive transducer (hat)," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6139–6143.

[8] N. Moritz, T. Hori, S. Watanabe, and J. L. Roux, "Sequence transduction with graph-based supervision," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7212–7216.

[9] H. Xu, F. Jia, S. Majumdar, S. Watanabe, and B. Ginsburg, "Multi-blank transducers for speech recognition," *arXiv preprint arXiv:2211.03541*, 2022.

[10] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "Rnn-transducer with stateless prediction network," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7049–7053.

[11] A. Tripathi, H. Lu, H. Sak, and H. Soltau, "Monotonic recurrent neural network transducer and decoding strategies," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 944–948.

[12] F. Kuang, L. Guo, W. Kang, L. Lin, M. Luo, Z. Yao, and D. Povey, "Pruned RNN-T for fast, memory-efficient ASR training," in *Proc. Interspeech 2022*, 2022, pp. 2068–2072.

[13] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving rnn transducer modeling for end-to-end speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 114–121.

[14] A. Zeyer, A. Merboldt, R. Schlüter, and H. Ney, "A New Training Pipeline for an Improved Neural Transducer," in *Proc. Interspeech 2020*, 2020, pp. 2812–2816.

[15] 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 *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6004–6008.

[16] W. Kang, Z. Yao, F. Kuang, L. Guo, X. Yang, P. Żelasko, D. Povey *et al.*, "Delay-penalized transducer for low-latency streaming asr," *arXiv preprint arXiv:2211.00490*, 2022.

[17] 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 *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 52–59.

[18] Y. Shinohara and S. Watanabe, "Minimum latency training of sequence transducers for streaming end-to-end speech recognition," in *Proc. Interspeech 2022*, 2022, pp. 2098–2102.

[19] J. Kim, H. Lu, A. Tripathi, Q. Zhang, and H. Sak, "Reducing Streaming ASR Model Delay with Self Alignment," in *Proc. Interspeech 2021*, 2021, pp. 3440–3444.

[20] T. N. Sainath, R. Pang, D. Rybach, B. García, and T. Strohman, "Emitting Word Timings with End-to-End Models," in *Proc. Interspeech 2020*, 2020, pp. 3615–3619.

[21] J. Tian, B. Yan, J. Yu, C. WENG, D. Yu, and S. Watanabe, "BAYES RISK CTC: CONTROLLABLE CTC ALIGNMENT IN SEQUENCE-TO-SEQUENCE TASKS," in *The Eleventh International Conference on Learning Representations*, 2023.

[22] 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.

[23] L. Rabiner and B. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[24] G. Saon, Z. Tüske, and K. Audhkhasi, "Alignment-length synchronous decoding for rnn transducer," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7804–7808.

[25] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, 2017, pp. 1–5.

[26] J. Du, X. Na, X. Liu, and H. Bu, "Aishell-2: Transforming mandarin asr research into industrial scale," *arXiv preprint arXiv:1808.10583*, 2018.

[27] 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.

[28] 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.

[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*, 2020, pp. 5036–5040.

[30] Y. Shi, Y. Wang, C. Wu, C.-F. Yeh, J. Chan, F. Zhang, D. Le, and M. Seltzer, "Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 6783–6787.