

IMPROVING LARGE-SCALE DEEP BIASING WITH PHONEME FEATURES AND TEXT-ONLY DATA IN STREAMING TRANSDUCER

Jin Qiu*, Lu Huang*, Boyu Li, Jun Zhang, Lu Lu, Zejun Ma

ByteDance

{qiujin, huanglu.thu19}@bytedance.com

ABSTRACT

Deep biasing for the Transducer can improve the recognition performance of rare words or contextual entities, which is essential in practical applications, especially for streaming Automatic Speech Recognition (ASR). However, deep biasing with large-scale rare words remains challenging, as the performance drops significantly when more distractors exist and there are words with similar grapheme sequences in the bias list. In this paper, we combine the phoneme and textual information of rare words in Transducers to distinguish words with similar pronunciation or spelling. Moreover, the introduction of training with text-only data containing more rare words benefits large-scale deep biasing. The experiments on the Librispeech corpus demonstrate that the proposed method achieves state-of-the-art performance on rare word error rate for different scales and levels of bias lists.

Index Terms— automatic speech recognition, conformer transducer, deep biasing, rare word, text-only

1. INTRODUCTION

Recently, E2E models have been widely explored in the ASR community and have achieved significant improvements [1, 2]. Compared to hybrid models, E2E ASR directly maps speech features into word sequences by optimizing a single neural network with E2E criteria. The most popular methods, such as CTC [3, 4], Transducer [1, 5, 6], and Attention-based Encoder-Decoder (AED) [7, 8, 9], have become mainstream.

Since the vocabulary comprises sub-word units, it is difficult for E2E models to recognize rare words, as they are frequently decomposed into infrequent sub-word sequences [10]. However, in practical applications, the accurate recognition of rare words is crucial for providing a better user experience, such as in the case of songs, contacts, installed applications. Moreover, rare words and text corpus containing such words are often available in advance. Therefore, finding ways to leverage this information and benefit E2E ASR models has become increasingly important.

One of the most common methods to improve the recognition performance of rare words is language model (LM) fusion. It can be achieved by constructing an FST based on rare words or contextual words [11, 12] and incorporating it during beam search. Besides, an external task-specific LM trained on extra text-only data can be used during inference [13, 14]. Also, the external LM can be incorporated in the E2E model during training [15, 16], like MWER training [17].

An alternative solution is to bias the E2E model with an all-neural framework. CLAS [11] was proposed to incorporate contextual information dynamically into the E2E model. In [18], a deep personalized LM was introduced to influence the model’s predictions earlier, and the performance was further improved by combining shallow fusion (SF), deep biasing, and LM contextualization [19]. CATT [20, 21] was proposed by jointly training a context-biasing network with the original Transducer. In [22], a contextual adapter is added to the pre-trained ASR model, which is conditioned with the outputs of the encoder’s different layers [23]. Besides, some works use an auxiliary loss or decoder to predict the rare words directly [24, 25]. Additionally, the phoneme information of rare words was also considered for deep biasing [26], and different embedding extractors were explored in [27].

In addition, there are some works that explore the usage of extra text corpus through text injection, rather than relying on LM. This is achieved by joint training of speech and text data, like JOIST [28, 29], MAESTRO [30], JEIT [31]. Furthermore, text-only domain adaptation has gained popularity recently, as seen in approaches like TOG [32] and USTR [33].

In this study, we first explore deep biasing modules using different hidden states as condition (query in attention). These modules are trained with a pre-trained Transducer, and the proposed learning rate policy can achieve better accuracy on biased words, and maintain the performance on unbiased words. Specifically, encoder-predictor query is chosen for its better performance and lower computational cost. To enhance the performance of deep biasing with large-scale bias lists, the phoneme information of rare words is also combined with the textual information. To our best knowledge, this is the first time that phoneme information is adopted for biasing Transducer. Additionally, we introduced the previous USTR approach to further improve the training of biasing module, and

* Equal contribution.

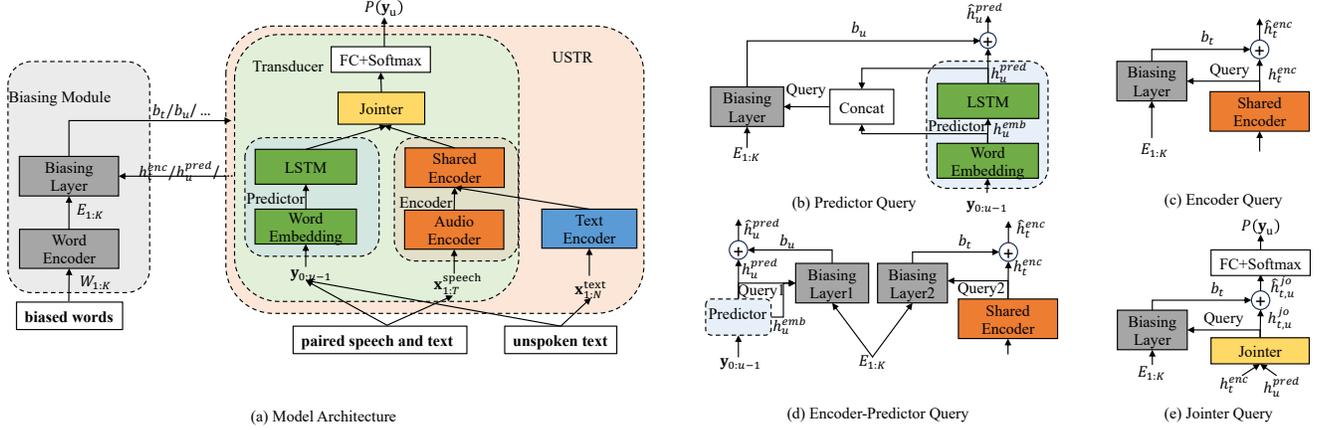


Fig. 1. The model structures of proposed method. (a) is the overall architecture. (b) is the case where the query of biasing layer is from Predictor. (c) is Encoder Query. (d) is the Enc-Pre Query. (e) is Joiner Query.

two types of text-only corpora are explored to demonstrate the feasibility of our methodology. Compared to previous methods, our experiments on Librispeech corpus showed that the proposed framework achieves competitive or even superior performance regarding the recognition accuracy of rare words.

2. RELATED WORK

2.1. Deep biasing

Attention-based deep biasing methods are most relevant to our work, such as CLAS [11] and C-RNNT [34], where an attention mechanism is used to direct the model’s focus towards specific contextual entities. Meanwhile, in C-RNNT [20, 22], deep biasing is employed on both the encoder and predictor in Conformer Transducers (CT). Also, CLAS is further improved with phoneme representations [26, 27], and leveraging phoneme information achieves better discrimination for similar grapheme sequences [10, 27].

However, these works may increase the decoding complexity without biasing module due to the combiner module. Furthermore, phoneme information hasn’t been explored for biasing Transducers before, and we show that phoneme information brings a gain for large scale deep biasing.

Additionally, it is found that optimizing the adapter by fixing the pre-trained model gets even worse performance on general test set [22]. We propose to use a group-based learning rate policy, and achieved better performance on both biased and unbiased words.

2.2. Text-only ASR

Due to the sparsity of audio training data, additional text data is explored to improve the accuracy of rare words. Several studies focus on leveraging external knowledge to enrich the

representations of rare words, such as selecting text data for LM training [14], augmenting the rare word embedding to enhance LM’s performance [35, 36]. Nevertheless, all these methods require an external LM during inference, which incurs computational cost.

Other methods try to increase the accuracy of rare words through joint training with text-only data [29, 30, 31]. However, these works focus on enhancing the general performance of rare words, instead of particular biasing words. In contrast, we propose to employ USTR, which is adopted for text-only domain adaptation [33], to further enhance the capability of deep biasing with more unpaired text data.

3. PROPOSED METHODS

3.1. Model architecture

The overall architecture is illustrated in Figure 1(a), which consists of Transducer, USTR’s TextEncoder and BiasingModule. The BiasingModule includes two parts, named WordEncoder and BiasingLayer.

For the paired speech and text data, let $\mathbf{X}^{\text{speech}} \in \mathbb{R}^{B_1 \times T \times D_1}$ be the audio features like Fbank, and the output of encoder is computed by

$$\mathbf{H}^{\text{speech}} = \text{Encoder}(\mathbf{X}^{\text{speech}}), \quad (1)$$

where $\mathbf{H}^{\text{speech}} \in \mathbb{R}^{B_1 \times T' \times H}$, and $\text{Encoder}(\cdot)$ is the same as $\text{SharedEncoder}(\text{AudioEncoder}(\cdot))$.

For unspoken text data, let $\mathbf{X}^{\text{text}} \in \mathbb{R}^{B_2 \times N \times D_2}$ be the text features, and the output of encoder is computed by

$$\mathbf{H}^{\text{text}} = \text{SharedEncoder}(\text{TextEncoder}(\mathbf{X}^{\text{text}})), \quad (2)$$

where $\mathbf{H}^{\text{text}} \in \mathbb{R}^{B_2 \times N' \times H}$.

Then the encoder output of paired speech-text data $\mathbf{H}^{\text{speech}}$ and unspoken text data \mathbf{H}^{text} are concatenated on the batch

dimension by filling to the same size on length dimension, as well as the output label sequence,

$$\mathbf{H}^{\text{enc}} = \text{BatchConcat}(\mathbf{H}^{\text{speech}}, \mathbf{H}^{\text{text}}), \quad (3)$$

$$\mathbf{Y} = \text{BatchConcat}(\mathbf{Y}^{\text{speech}}, \mathbf{Y}^{\text{text}}), \quad (4)$$

where $\mathbf{H}^{\text{enc}} \in \mathbb{R}^{B \times L \times H}$, $B = B1 + B2$, $L = \max(T', N')$, $\mathbf{Y} \in \mathbb{R}^{B \times U}$.

For simplicity, let $\mathbf{y} \in \mathbb{R}^U$ and $\mathbf{h}^{\text{enc}} \in \mathbb{R}^{L \times H}$ be an utterance in the batch of \mathbf{H}^{enc} and \mathbf{Y} , and the predicted probability on vocabulary of CT at frame t and step u is computed by

$$\mathbf{h}_u^{\text{pred}} = \text{Predictor}(\mathbf{y}_{0:u-1}), \quad (5)$$

$$\mathbf{h}_{t,u}^{\text{joint}} = \text{Joiner}(\mathbf{h}_t^{\text{enc}}, \mathbf{h}_u^{\text{pred}}), \quad (6)$$

$$\hat{\mathbf{y}}_{t,u} = \text{Softmax}(\text{FC}(\mathbf{h}_{t,u}^{\text{joint}})). \quad (7)$$

Then with forward-backward algorithm [6], Transducer loss is computed as the training objective function.

For **BiasingModule**, it takes the bias words $\mathbf{W}_{1:K} \in \mathbb{R}^{K \times S}$ as input and converts $\mathbf{W}_{1:K}$ to $\mathbf{E}_{1:K} \in \mathbb{R}^{K \times M}$ by a **WordEncoder**, where K is the number of biasing words (including one empty word for no biasing), S is the max length of rare words, M is the dimension of word embedding, \mathbf{W}_k is a word or word sequence, and \mathbf{E}_k is the corresponding embedding. More details are provided in Section 3.3.

BiasingLayer contains multi-head attention (MHA), which takes the input from Transducer/USTR as query and $\mathbf{E}_{1:K}$ as key/value. The output of MHA is reshaped to the same size as the query by a projection layer, then is added to the original query as a biasing vector. More details about **BiasingLayer** can be found in Section 3.2.

3.2. Biasing with different queries

Predictor-Query (Figure 1(b)). The query is the concatenated value of predictor’s embedding output $\mathbf{h}_u^{\text{emb}}$ and final output $\mathbf{h}_u^{\text{pred}}$ at each step u , and the biasing process is

$$\text{Query} = \text{Concat}(\mathbf{h}_u^{\text{emb}}, \mathbf{h}_u^{\text{pred}}), \quad (8)$$

$$\mathbf{b}_u = \text{MHA}(\text{Query}, \mathbf{E}_{1:K}, \mathbf{E}_{1:K}), \quad (9)$$

$$\hat{\mathbf{h}}_u^{\text{pred}} = \mathbf{b}_u + \mathbf{h}_u^{\text{pred}}, \quad (10)$$

where $\hat{\mathbf{h}}_u^{\text{pred}}$ will replace $\mathbf{h}_u^{\text{pred}}$ for Transducer/USTR training and inference.

Encoder-Query (Figure 1(c)). The query is the encoder output $\mathbf{h}_t^{\text{enc}}$ at each time step t , and the biasing process is

$$\mathbf{b}_t = \text{MHA}(\text{Query} = \mathbf{h}_t^{\text{enc}}, \mathbf{E}_{1:K}, \mathbf{E}_{1:K}) \quad (11)$$

Enc-Pre Query. This is the combination of Encoder-Query and Predictor-Query. As noted in Figure 1(d), there are two **BiasingLayer** modules with separated parameters.

Joiner-Query (Figure 1(e)). In this case, the query is the hidden states $\mathbf{h}_{t,u}^{\text{joint}}$ in joiner, and the biasing process is

$$\mathbf{b}_{t,u} = \text{MHA}(\text{Query} = \mathbf{h}_{t,u}^{\text{joint}}, \mathbf{E}_{1:K}, \mathbf{E}_{1:K}) \quad (12)$$

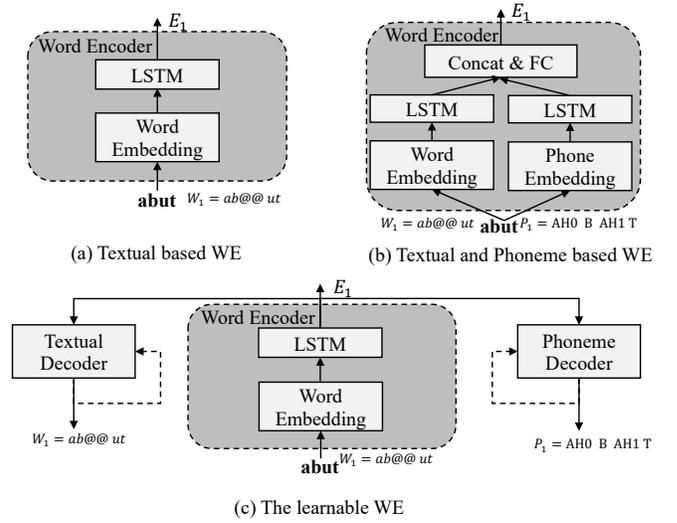


Fig. 2. The model structures of different word encoders (WE). Only one word **abut** is considered here for better understanding, where “ab@@ ut” is the sub-word sequence and “AH0 B AH1 T” is the phoneme sequence.

3.3. Combining textual and phoneme features

Textual-WE, as shown in 2(a). A word is first converted to a sub-word sequence and an embedding layer is used to extract token embedding. Then the sequence of token embedding is fed into an unidirectional long short-term memory (LSTM) layer, and the final state is adopted as the word embedding.

Tex-Pho-WE, as illustrated in 2(b). Different from Textual-WE, there is another branch, which converts the word to a phoneme sequence, and the sequence of phoneme embedding is fed to the LSTM layer to get the final state as a phoneme representation of the word. The textual and phoneme representations are concatenated and reshaped to the same size by a fully connected (FC) layer.

Learnable-WE, as illustrated in 2(c). Different from Textual-WE, there are two additional transformer decoders, which take the word embedding as input and predict textual and phoneme sequences respectively. The Learnable-WE is similar to that in [27]. It should be noticed that the decoders can be removed during inference.

4. EXPERIMENTAL SETUP

4.1. Data sets

The experiments are conducted on LibriSpeech [37] corpus, where the 960-hour audio data is adopted as paired speech-text corpus, and the normalized text data with size of 1.5G¹ is used as unspoken text for training USTR. Also, in this work, we proposed to use the 209.2k rare words (defined as words

¹<https://www.openslr.org/11/>

Table 1. The WER(U-WER/B-WER)(%) results of deep biasing with different queries. The size of bias list is 100 here.

model	test-clean	test-other
CT Baseline	3.28 (2.15/12.43)	7.88 (5.71/26.97)
+ deep biasing (Predictor-Query)	2.93 (2.16/9.22)	7.11 (5.61/20.36)
+ deep biasing (Encoder-Query)	2.78 (2.11/8.18)	6.63 (5.38/17.66)
+ deep biasing (Enc-Pre Query)	2.67 (2.06/7.64)	6.54 (5.48/15.81)
+ Freezing CT	2.92 (2.14/9.27)	7.01 (5.61/19.31)
+ deep biasing (Jointer-Query)	2.67 (2.07/7.50)	6.67 (5.51/16.88)

not in the 5,000 most common words in the paired audio training set, i.e., **Rare5k**) as unspoken text to improve the biasing performance when training with USTR.

For the paired audio data, 3-fold speed perturbation [38] with factors of 0.95, 1.0, and 1.05 is used for data augmentation. Besides, the 80-dim filter-bank (Fbank) is extracted and Spec-Augment [39] is applied on Fbank features before feeding into `AudioEncoder`.

When training USTR with unspoken text data, the phonemes are adopted as text features, which is similar to that in [33], and the text features are masked with a probability of 0.15 before repeating and feeding into `TextEncoder`.

4.2. Model

The model’s structure is described as that in Section 3.1, where `AudioEncoder` consists of 2-layer 2D convolution with channel=128, kernel=3, stride=2 and ReLU activation [40, 41], resulting in downsampling of 40ms. `TextEncoder` contains an embedding layer and a Transformer layer. And `SharedEncoder` consists of 12 streaming Conformer [42] layers, where the attentions of first 7 layers have a look ahead of 1 frame (i.e., 40ms) and there is no look ahead for all convolutions and attentions of last 5 layers. The total look ahead of the `Encoder` is 310ms (280ms for Conformer layers and 30ms for `AudioEncoder`). `Predictor` has an embedding layer and 2 LSTM layers and `Jointer` has a linear layer. The output of RNN-T is 4,048 subword units [43]. All models are implemented and trained with PyTorch [44].

4.3. Training

During training, the bias list of current batch is extracted from all the batch references, including the rare words in **Rare2k** (defined as words that fall outside the 2k most common words in the paired audio training set).

The textual feature of a bias word is the subword units. Besides, the phoneme features are generated by a Grapheme-to-Phoneme system, i.e., $g2pE^2$.

When training USTR with unspoken text data, single-step is adopted, as the training process is simpler and more efficient, and better performance can be obtained [33]. Besides, the USTR is trained from scratch with `BiasingModule`. Also, during the training of USTR, paired speech-text data is fed into the `TextEncoder` by using text features instead of audio features with a probability 0.15 to force the `TextEncoder` and `AudioEncoder` to learning an unified representation for audio and text features.

However, when training `BiasingModule` with paired speech-text data, a pre-trained Conformer Transducer (CT) is used for initialization. In this case, to maintain the model’s performance when there is no biasing, group-based learning rate (lr) policy is proposed, where `BiasingModule` and CT are trained with $lr=1e-5$ and $lr=1e-7$ jointly.

In addition to the Transducer loss, CTC and an extra AED decoder are adopted for multi-task learning. Also, internal LM estimation (ILMT) loss is chosen as an auxiliary loss. The overall training loss is

$$\mathcal{L} = \mathcal{L}_{\text{Transducer}} + \mathcal{L}_{\text{CTC}} + \mathcal{L}_{\text{AED}} + \lambda \mathcal{L}_{\text{ILMT}}, \quad (13)$$

where λ is set to 0.2 in all experiments. When Learnable-WE is used, there still exists two AED losses with scale of 0.1. Besides, subword regularization, i.e., BPE dropout [45], is applied with a probability of 0.1 during training.

4.4. Inference

During inference, for each utterance, the biasing list is constructed by extracting the rare words (in **Rare5k**) in the reference and adding a certain number of distractors. The biased words with different size (100/500/1000/2000) are explored to check the performance of proposed method with different scale of bias list size, which is the same as that in [19].

WER is evaluated on Librispeech `test-clean` and `test-other` sets. To indicate the performance of (biased) rare words, B-WER (biased WER) is measured on words in the biasing list as that in [19, 24], while U-WER (unbiased WER) is also measured to prevent degrading the performance of words not in the biasing list.

Moreover, similar to that in [46], rare WER (R-WER) is used to evaluate the performance of proposed methods for utterance-level, chapter-level and book-level biasing.

5. EXPERIMENTAL RESULTS

5.1. Biasing with different queries

Deep biasing with different queries in CT has been initially explored, and the results are illustrated in Table 1. Compared

²<https://github.com/Kyubyong/g2p>

Table 2. The WER(U-WER/B-WER)(%) results of deep biasing when using difference embedding modules. The size of bias list is 100, and Enc-Pre Query is used for all experiments.

model	test-clean	test-other
CT Baseline	3.28 (2.15/12.43)	7.88 (5.71/26.97)
deep biasing (Textual-WE)	2.67 (2.06/7.64)	6.54 (5.48/15.81)
deep biasing (Tex-Pho-WE)	2.56 (2.03/6.84)	6.33 (5.38/14.69)
deep biasing (Learnable-WE)	2.68 (2.09/7.50)	6.44 (5.40/15.55)

to the CT baseline, all models with deep biasing achieve significant reductions on B-WER, and the U-WER performances remain almost the same or even better. It indicates that the proposed method enhances the performance of rare words without sacrificing the performance of common words.

Joint Query achieves the best B-WER on test-clean, while Enc-Pre Query obtains better WERs on test-other. When applying deep biasing with Enc-Pre Query, we observe relative improvements of 38.5% and 41.4% on the B-WER. However, when we tried to freeze the parameters of CT rather than using different learning rates, the performance becomes much worse on both biased and unbiased words.

Besides, for an utterance in which lengths of encoder’s output and predictor’s output are L and U , the computational complexity of Enc-Pre Query is $O(L + U)$, in contrast the computational complexity of Jointer Query is $O(L \times U)$, which is larger when $U \geq 2, L > U$. Therefore, Enc-Pre Query has a lower computational complexity in most cases and is chosen as the default in the following experiments.

5.2. Combining textual and phoneme features

We evaluate the effectiveness of combining textual and phoneme information with different word encoders. As shown in Figure 2, compared with Textual-WE, Tex-Pho-WE obtained the best results, with relative reductions of 10.47% (7.64% \rightarrow 6.84%) and 7.08% (15.81% \rightarrow 14.69%) on the B-WERs of two test sets, respectively. Learnable-WE, explored in [27], performs worse than Tex-Pho-WE, as Tex-Pho-WE uses the phoneme information without errors. The improvement indicates the benefits of phoneme information for deep biasing in CT, and Tex-Pho-WE is chosen as the default configuration in following experiments.

5.3. Combined with USTR

Then, we investigate the impact of introducing unpaired text data containing rare words by combining deep biasing and USTR. As illustrated in Table 3, compared to CT baseline, not only the USTR with LM corpus obtains better performance

Table 3. The WER(U-WER/B-WER)(%) results when combining deep biasing with USTR. The size of bias list is 100 here. USTR-CT(C/L) denotes the USTR model trained using Librispeech LM Corpus and rare word List respectively.

model	test-clean	test-other
CT Baseline	3.28 (2.15/12.43)	7.88 (5.71/26.97)
+ deep biasing	2.56 (2.03/6.84)	6.33 (5.38/14.69)
USTR-CT(C)	3.05 (2.09/10.83)	7.49 (5.55/24.58)
+ deep biasing	2.39 (2.27/3.38)	6.30 (6.23/6.99)
USTR-CT(L)	3.13 (2.06/11.84)	7.58 (5.57/25.31)
+ deep biasing	2.19 (1.99/3.82)	5.61 (5.38/7.57)
USTR-CT(C+L)	2.98 (1.97/11.14)	7.45 (5.54/24.24)
+ deep biasing	2.15 (2.00/3.33)	5.56 (5.46/6.45)

on all WERs, but also the USTR with rare word list obtains slight improvements. When trained using both LM corpus and rare word list, the B-WERs on test-clean and test-other are reduced from 12.43%/26.97% to 11.14%/24.24%.

When combining USTR with deep biasing, significant improvements are observed. When trained with unpaired text data C/L, the B-WER of deep biasing model on test-other is improved from 14.69% to 6.99%/7.57%, much better than CT or USTR baselines. The improvements are mainly attributed to utilization of more rare words for training and the capacity of biasing module capacity is enhanced.

It should be noted that the rare words list contains only 209.2k rare words, which is significantly smaller than the LM corpus and demonstrates the feasibility of our method. Consequently, we combined the LM corpus and rare word list, and achieved the best WER/B-WER on both test sets. Compared with the deep biasing baseline, USTR(C+L) with deep biasing provides relative improvements of 51.32%/56.09% on B-WER (6.84%/14.69% \rightarrow 3.33%/6.45%).

5.4. Different bias list size

We further evaluate the robustness of the proposed method on large-scale bias lists, in which most words are irrelevant to the audio. As illustrated in Table 4, as the size of bias list increases, B-WER increases gradually when using Enc-Pre Query. The absolute gaps of B-WER on the two test sets between $N = 100/2000$ are 3.2%/7.7% respectively. Tex-Pho-WE alleviates these gaps to 2.7%/5.9% because it provides additional information to discriminate

Table 4. The WER(U-WER/B-WER)(%) results on LibriSpeech test sets with different bias list size (100/500/1000/2000).

Method	$N = 100$		$N = 500$		$N = 1000$		$N = 2000$	
	test-clean	test-other	test-clean	test-other	test-clean	test-other	test-clean	test-other
CT Baseline	3.28 (2.2/12.4)	7.88 (5.7/27.0)	3.28 (2.2/12.4)	7.88 (5.7/27.0)	3.28 (2.2/12.4)	7.88 (5.7/27.0)	3.28 (2.2/12.4)	7.88 (5.7/27.0)
Enc-Pre Query	2.67 (2.1/7.6)	6.54 (5.5/15.8)	2.87 (2.1/9.1)	7.01 (5.5/20.3)	2.97 (2.1/10.0)	7.30 (5.6/22.3)	3.09 (2.2/10.8)	7.44 (5.6/23.5)
+ Tex-Pho-WE	2.56 (2.0/6.8)	6.33 (5.4/14.7)	2.74 (2.1/8.1)	6.70 (5.5/17.5)	2.81 (2.1/8.7)	6.93 (5.5/19.1)	2.91 (2.1/9.5)	7.09 (5.6/20.6)
+ USTR(C+L)	2.15 (2.0/3.3)	5.56 (5.5/6.5)	2.23 (2.1/3.7)	5.83 (5.6/8.2)	2.28 (2.1/3.8)	6.01 (5.7/9.1)	2.30 (2.1/4.4)	6.14 (5.6/11.0)
+ FST	2.06 (2.1/2.0)	5.38 (5.5/4.4)	2.09 (2.1/2.2)	5.62 (5.6/5.6)	2.16 (2.1/2.5)	5.75 (5.7/6.3)	2.17 (2.1/3.0)	5.84 (5.6/7.6)
DB-RNN-T + FST + DB-NNLM[19]	1.98 (1.5/5.7)	5.86 (4.9/14.1)	2.09 (1.6/6.2)	6.09 (5.1/15.1)	2.14 (1.6/6.7)	6.35 (5.1/17.2)	2.27 (1.6/7.3)	6.58 (5.2/18.9)
CT + deep biasing[24]	3.66 (2.8/11.2)	7.63 (6.0/22.1)	3.78 (2.9/11.5)	7.99 (6.2/23.4)	3.88 (2.9/11.9)	8.28 (6.4/24.5)	N/A	N/A

* N/A means that the results are not available.

Table 5. The WER/R-WER(%) results of various systems on LibriSpeech test sets when using deep bias with utterance-level, chapter-level and book-level rare words. The bias list size is 1000 for all methods.

Model	test-clean			test-other		
	Utterance-level	Chapter-level	Book-level	Utterance-level	Chapter-level	Book-level
RNN-T + TCPGen[46] + deep biasing + SF	4.9(13.9) 3.8(11.3)	5.1(13.6) 4.0(11.0)	5.4(28.2) 4.2(24.0)	14.0(35.0) 11.5(29.0)	14.1(32.4) 12.0(29.3)	14.8(52.1) 12.2(50.8)
DB-RNN-T[19]* + FST + DB-NNLM*	3.3(11.9) 2.1(6.8)	N/A N/A	N/A N/A	9.1(31.4) 6.4(21.3)	N/A N/A	N/A N/A
Proposed deep biasing + USTR + FST	2.8(9.8) 2.2(5.0)	3.0(10.9) 2.5(7.1)	3.2(12.9) 2.8(9.8)	6.9(22.9) 5.8(11.7)	7.2(25.4) 6.4(18.0)	7.6(29.3) 7.1(24.2)

* The results of R-WER is generated by using the hypothesis files in https://github.com/facebookresearch/fbai-speech/tree/main/is21_deep_bias with a bias list size of 1000.

similar grapheme sequences. By combining USTR(C+L), the gap shrinks to 1.1%/4.5%. Compared to CT baseline, our best system, which combines deep biasing, USTR, and FST, achieves relative B-WER reductions of 75.81% and 71.85% on the two test sets respectively when $N = 2000$.

5.5. Comparison with other methods

Results of some prior works are also listed in the bottom rows in Table 4. Compared to the best system in [19], the proposed method achieves the best B-WER on all test sets and all sizes of bias list, while no external LM is used. With an external LM, we believe further improvements can be achieved.

R-WERs of our system and other approaches are listed in Table 5 with utterance-level, chapter-level, and book-level rare words as those in [46]. The proposed strategy achieves the best R-WERs on two test sets with all levels of the bias list, indicating the proposed method’s superiority.

6. CONCLUSIONS

In this paper, we proposed several significant improvements in large-scale deep biasing for Transducer based streaming ASR. Our approach extends CT by incorporating textual and phoneme information of rare words, resulting in notable relative improvements of 45.16% and 45.56% on B-WER over the baseline. Furthermore, by incorporating the previously established USTR method for text injection during training and incorporating FST during inference, the proposed approach yields remarkable improvements, leading to relative reductions of 83% ~ 84% on B-WER. Moreover, our method demonstrates robustness in large-scale deep biasing scenarios, effectively closing the gap between bias list sizes from 100 to 2000. Notably, compared to other publicly available results, our approach attains state-of-the-art performance on the accuracy of rare words.

7. REFERENCES

- [1] Tara N Sainath, Yanzhang He, Bo Li, Arun Narayanan, Ruoming Pang, Antoine Bruguier, Shuo-yiin Chang, Wei Li, Raziell Alvarez, Zhifeng Chen, et al., “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency,” in *ICASSP*, 2020.
- [2] Jinyu Li et al., “Recent advances in end-to-end automatic speech recognition,” *APSIPA Transactions on Signal and Information Processing*, 2022.
- [3] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *ICML*, 2006.
- [4] Jinyu Li, Guoli Ye, Amit Das, Rui Zhao, and Yifan Gong, “Advancing acoustic-to-word ctc model,” in *ICASSP*, 2018.
- [5] Jinyu Li, Rui Zhao, Zhong Meng, Yanqing Liu, Wenning Wei, Sarangarajan Parthasarathy, Vadim Mazalov, Zhenghao Wang, Lei He, Sheng Zhao, et al., “Developing rnn-t models surpassing high-performance hybrid models with customization capability,” in *Interspeech*, 2020.
- [6] Alex Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.
- [7] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, “Attention-based models for speech recognition,” in *NeurIPS*, 2015.
- [8] Shigeeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyang Jiang, Masao Someki, Nelson Enrique Yalta Soplín, Ryuichi Yamamoto, Xiaofei Wang, et al., “A comparative study on transformer vs rnn in speech applications,” in *ASRU*, 2019.
- [9] Jinyu Li, Yu Wu, Yashesh Gaur, Chengyi Wang, Rui Zhao, and Shujie Liu, “On the comparison of popular end-to-end models for large scale speech recognition,” in *Interspeech*, 2020.
- [10] Xuandi Fu, Kanthashree Mysore Sathyendra, Ankur Gandhe, Jing Liu, Grant P Strimel, Ross McGowan, and Athanasios Mouchtaris, “Robust acoustic and semantic contextual biasing in neural transducers for speech recognition,” in *ICASSP*, 2023.
- [11] Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjali Kannan, and Ding Zhao, “Deep context: end-to-end contextual speech recognition,” in *SLT*, 2018.
- [12] Ding Zhao, Tara N Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang, “Shallow-fusion end-to-end contextual biasing,” in *Interspeech*, 2019.
- [13] Cyril Allauzen, Ehsan Variani, Michael Riley, David Rybach, and Hao Zhang, “A hybrid seq-2-seq asr design for on-device and server applications,” in *Interspeech*, 2021.
- [14] W Ronny Huang, Cal Peyser, Tara N Sainath, Ruoming Pang, Trevor Strohman, and Shankar Kumar, “Sentence-select: Large-scale language model data selection for rare-word speech recognition,” *arXiv preprint arXiv:2203.05008*, 2022.
- [15] Cal Peyser, Sepand Mavandadi, Tara N Sainath, James Apfel, Ruoming Pang, and Shankar Kumar, “Improving tail performance of a deliberation e2e asr model using a large text corpus,” *Interspeech*, 2020.
- [16] Wang Weiran, Tongzhou Chen, Tara Sainath, Ehsan Variani, Rohit Prabhavalkar, W. Ronny Huang, Bhuvana Ramabhadran, Neeraj Gaur, Sepand Mavandadi, Cal Peyser, Trevor Strohman, Yanzhang He, and David Rybach, “Improving rare word recognition with lm-aware mwer training,” in *Interspeech*, 2022.
- [17] Rohit Prabhavalkar, Tara N Sainath, Yonghui Wu, Patrick Nguyen, Zhifeng Chen, Chung-Cheng Chiu, and Anjali Kannan, “Minimum word error rate training for attention-based sequence-to-sequence models,” in *ICASSP*, 2018.
- [18] Duc Le, Gil Keren, Julian Chan, Jay Mahadeokar, Christian Fuegen, and Michael L Seltzer, “Deep shallow fusion for rnn-t personalization,” in *SLT*, 2021.
- [19] Duc Le, Mahaveer Jain, Gil Keren, Suyoun Kim, Yangyang Shi, Jay Mahadeokar, Julian Chan, Yuan Shanguan, Christian Fuegen, Ozlem Kalinli, et al., “Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion,” in *Interspeech*, 2021.
- [20] Feng-Ju Chang, Jing Liu, Martin Radfar, Athanasios Mouchtaris, Maurizio Omologo, Ariya Rastrow, and Siegfried Kunzmann, “Context-aware transformer transducer for speech recognition,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 503–510.
- [21] Junfeng Hou, Jinkun Chen, Wanyu Li, Yufeng Tang, Jun Zhang, and Zejun Ma, “Bring dialogue-context into rnn-t for streaming asr,” in *INTERSPEECH*, 2022.
- [22] Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Feng-Ju Chang, Jing Liu, Jinru Su, Grant P Strimel,

- Athanasios Mouchtaris, and Siegfried Kunzmann, “Contextual adapters for personalized speech recognition in neural transducers,” in *ICASSP*, 2022.
- [23] Saket Dingliwal, Monica Sunkara, Srikanth Ronanki, Jeff Farris, Katrin Kirchhoff, and Sravan Bodapati, “Personalization of ctc speech recognition models,” in *SLT*, 2023.
- [24] Kaixun Huang, Ao Zhang, Zhanheng Yang, Pengcheng Guo, Bingshen Mu, Tianyi Xu, and Lei Xie, “Contextualized end-to-end speech recognition with contextual phrase prediction network,” in *Interspeech*, 2023.
- [25] Minglun Han, Linhao Dong, Shiyu Zhou, and Bo Xu, “Cif-based collaborative decoding for end-to-end contextual speech recognition,” in *ICASSP*, 2021.
- [26] Antoine Bruguier, Rohit Prabhavalkar, Golan Pundak, and Tara N Sainath, “Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition,” in *ICASSP*, 2019.
- [27] Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L Seltzer, and Christian Fuegen, “Joint grapheme and phoneme embeddings for contextual end-to-end asr,” in *Interspeech*, 2019.
- [28] Tara N Sainath, Rohit Prabhavalkar, Ankur Bapna, Yu Zhang, Zhouyuan Huo, Zhehuai Chen, Bo Li, Weiran Wang, and Trevor Strohman, “Joist: A joint speech and text streaming model for asr,” in *SLT*, 2023.
- [29] Tara N Sainath, Rohit Prabhavalkar, Diamantino Casero, Pat Rondon, and Cyril Allauzen, “Improving contextual biasing with text injection,” in *ICASSP*, 2023.
- [30] Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Pedro J. Moreno, Ankur Bapna, and Heiga Zen, “MAESTRO: Matched Speech Text Representations through Modality Matching,” in *Interspeech*, 2022.
- [31] Zhong Meng, Weiran Wang, Rohit Prabhavalkar, Tara N Sainath, Tongzhou Chen, Ehsan Variani, Yu Zhang, Bo Li, Andrew Rosenberg, and Bhuvana Ramabhadran, “Jeit: Joint end-to-end model and internal language model training for speech recognition,” in *ICASSP*, 2023.
- [32] Samuel Thomas, Brian Kingsbury, George Saon, and Hong-Kwang J Kuo, “Integrating text inputs for training and adapting rnn transducer asr models,” in *ICASSP*, 2022.
- [33] Lu Huang, Boyu Li, Jun Zhang, Lu Lu, and Zejun Ma, “Text-only domain adaptation using unified speech-text representation in transducer,” in *Interspeech*, 2023.
- [34] Mahaveer Jain, Gil Keren, Jay Mahadeokar, Geoffrey Zweig, Florian Metze, and Yatharth Saraf, “Contextual RNN-T for Open Domain ASR,” in *Interspeech*, 2020.
- [35] Yerbolat Khassanov, Zhiping Zeng, Van Tung Pham, Haihua Xu, and Eng Siong Chng, “Enriching rare word representations in neural language models by embedding matrix augmentation,” in *Interspeech*, 2019.
- [36] W. Ronny Huang, Tara N. Sainath, Cal Peysers, Shankar Kumar, David Rybach, and Trevor Strohman, “Lookupable recurrent language models for long tail speech recognition,” in *Interspeech*, 2021.
- [37] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *ICASSP*, 2015.
- [38] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, “Audio augmentation for speech recognition,” in *Interspeech*, 2015.
- [39] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, “SpecAugment: A simple data augmentation method for automatic speech recognition,” in *Interspeech*, 2019.
- [40] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, “Deep sparse rectifier neural networks,” in *AISTATS*, 2011.
- [41] George E Dahl, Tara N Sainath, and Geoffrey E Hinton, “Improving deep neural networks for lvcsr using rectified linear units and dropout,” in *ICASSP*, 2013.
- [42] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., “Conformer: Convolution-augmented transformer for speech recognition,” in *Interspeech*, 2020.
- [43] Rico Sennrich, Barry Haddow, and Alexandra Birch, “Neural machine translation of rare words with subword units,” in *ACL*, 2016.
- [44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., “Pytorch: An imperative style, high-performance deep learning library,” in *NeurIPS*, 2019.
- [45] Ivan Provilkov, Dmitrii Emelianenko, and Elena Voita, “Bpe-dropout: Simple and effective subword regularization,” in *ACL*, 2020.
- [46] Guangzhi Sun, Chao Zhang, and Philip C Woodland, “Tree-constrained pointer generator for end-to-end contextual speech recognition,” in *ASRU*, 2021.