



DYNAC: Dynamic Vocabulary-based Non-Autoregressive Contextualization for Speech Recognition

Yui Sudo¹, Yosuke Fukumoto¹, Muhammad Shakeel¹, Yifan Peng², Chyi-Jiunn Lin², Shinji Watanabe²

¹Honda Research Institute Japan, Japan

²Carnegie Mellon University, USA

shakeel.muhammad@jp.honda-ri.com, yifanpen@andrew.cmu.edu, shinjiw@ieee.org

Abstract

Contextual biasing (CB) improves automatic speech recognition for rare and unseen phrases. Recent studies have introduced dynamic vocabulary, which represents context phrases as expandable tokens in autoregressive (AR) models. This method improves CB accuracy but with slow inference speed. While dynamic vocabulary can be applied to non-autoregressive (NAR) models, such as connectionist temporal classification (CTC), the conditional independence assumption fails to capture dependencies between static and dynamic tokens. This paper proposes DYNAC (Dynamic Vocabulary-based NAR Contextualization), a self-conditioned CTC method that integrates dynamic vocabulary into intermediate layers. Conditioning the encoder on dynamic vocabulary, DYNAC effectively captures dependencies between static and dynamic tokens while reducing the real-time factor (RTF). Experimental results show that DYNAC reduces RTF by 81% with a 0.1-point degradation in word error rate on the LibriSpeech 960 test-clean set.

Index Terms: speech recognition, biasing, dynamic vocabulary, non-autoregressive, self-conditioned CTC

1. Introduction

End-to-end automatic speech recognition (E2E-ASR) [1, 2] can be classified into autoregressive (AR) and non-autoregressive (NAR) models. AR models, such as the attention-based encoder-decoder [3–6] and recurrent neural network transducer (RNN-T) [7, 8], predict the next token based on the previous token sequence in an AR manner. In contrast, NAR models, such as connectionist temporal classification (CTC) [9–12], predict all tokens simultaneously under the conditional independence assumption. This approach allows faster inference but often results in lower performance compared to AR models. Several methods have been proposed to relax the conditional independence assumption [13–17]. For example, self-conditioned CTC incorporates intermediate predictions as feedback to refine subsequent encoder layers, allowing the model to capture contextual dependencies [18]. Despite these advances, the performance of E2E-ASR models remains highly dependent on the training data, leading to performance inconsistencies for rare and unseen phrases. Frequent retraining to adapt to these phrases is impractical, highlighting the need for a method to contextualize models without additional retraining.

Contextual biasing (CB) [19–23] provides an effective strategy for adapting E2E-ASR models to specific context phrases through a bias list without requiring additional retraining. Most CB methods are designed for AR models, leveraging a cross-attention layer in the decoder for accurate bias phrase recognition. Although AR-CB methods achieve accurate bias phrase recognition, their slow inference speed makes them unsuitable

for latency-sensitive applications. Several CB methods have been proposed for NAR models or adapted to CTC-based NAR models to improve inference speed [24–26]. However, most existing CB methods represent bias phrases as sequences of subwords from a pre-defined vocabulary (referred to as static vocabulary), resulting in unnatural token patterns with low occurrence probabilities. For example, the personal name “*Raphael*” might be segmented into “*Ra*”, “*pha*”, and “*el*”. If such static token patterns are rare in the training data, their recognition accuracy decreases significantly. Several studies mitigate this issue by employing additional information, such as text-only data [27, 28], synthesized speech [29, 30], phonemes [31, 32], and named entity tags [33]. While these approaches improve CB performance, they introduce additional computational and operational overhead.

To mitigate this issue without requiring such additional information, dynamic vocabulary expansion has been proposed [34]. This method introduces a dynamic vocabulary, where each bias phrase is represented as a dynamically expandable single token instead of being decomposed into a sequence of static tokens. For example, the bias phrase “*Raphael*” is represented as a single dynamic token [*<Raphael>*] rather than the static token sequence [*“Ra”, “pha”, “el”*]. While this method improves CB performance without relying on additional information, its evaluation has been limited to AR models (CTC/attention and CTC/RNN-T) [35, 36], which are computationally expensive. Although dynamic vocabulary expansion can be applied to CTC-based NAR models, the conditional independence assumption limits the ability to capture contextual dependencies between static and dynamic tokens.

This paper proposes DYNAC (Dynamic Vocabulary-based NAR Contextualization), an NAR-CB method using self-conditioned CTC that incorporates dynamic vocabulary into the intermediate encoder layers. By conditioning the encoder on the dynamic vocabulary, DYNAC relaxes the conditional independence assumption. This approach allows the model to capture dependencies between static and dynamic tokens while reducing the real-time factor (RTF) compared to AR-CB methods [34]. The main contributions of this paper are as follows:

- We propose DYNAC, an NAR-CB method using self-conditioned CTC that integrates dynamic vocabulary, enabling efficient CB while reducing the RTF.
- We analyze the effectiveness of DYNAC by comparing the token-wise scores of the static and dynamic vocabulary to highlight its impact on recognition accuracy.
- We demonstrate the effectiveness of DYNAC on the LibriSpeech 960 corpus and our in-house Japanese dataset, showing robust performance even on unseen phrases.

2. CTC-based CB with dynamic vocabulary

This section describes the CTC-based CB method using dynamic vocabulary, which consists of an audio encoder, a bias encoder, and a scoring layer based on CTC. Although dynamic vocabulary expansion was originally proposed for AR-CB methods, it can be applied to CTC-based NAR models [34].

2.1. Audio encoder

We use the Conformer [8] for the audio encoder, comprising L encoder layers. The l -th encoder layer transforms its T -length d -dimensional hidden state representation input $\mathbf{X}_{(l)}^{\text{in}} \in \mathbb{R}^{T \times d}$ into $\mathbf{X}_{(l)}^{\text{out}} \in \mathbb{R}^{T \times d}$ as follows:

$$\mathbf{X}_{(l)}^{\text{out}} = \text{AudioEnc}_{(l)}(\mathbf{X}_{(l)}^{\text{in}}). \quad (1)$$

Then, the output of the l -th encoder layer $\mathbf{X}_{(l)}^{\text{out}}$ is directly used as the input of the $(l+1)$ -th encoder layer $\mathbf{X}_{(l+1)}^{\text{in}}$ as follows:

$$\mathbf{X}_{(l+1)}^{\text{in}} = \mathbf{X}_{(l)}^{\text{out}}. \quad (2)$$

By iterating this process L times, the final output $\mathbf{X}_{(L)}^{\text{out}} \in \mathbb{R}^{T \times d}$ is obtained.

2.2. Bias encoder

The bias encoder comprises Transformer [37] layers and a mean pooling layer with a bias list $\mathbf{B} = \{b_1, \dots, b_N\}$, where N represents the number of bias phrases. Each bias phrase b_n is represented as a sequence of static tokens (e.g., [*“Ra”*, *“pha”*, *“el”*]) in the pre-defined static vocabulary $\mathcal{V}^{\text{static}}$. The bias list \mathbf{B} is processed by the bias encoder to obtain phrase-level representations $\mathbf{V} = [v_1, \dots, v_N] \in \mathbb{R}^{N \times d}$ as follows:

$$\mathbf{V} = \text{BiasEnc}(\mathbf{B}). \quad (3)$$

The bias phrases in the bias list $\mathbf{B} = \{b_1, \dots, b_N\}$ are incorporated into the dynamic vocabulary $\mathcal{V}^{\text{dynamic}} = \{<b_1>, \dots, <b_N>\}$ of size N , where each entry represents the corresponding bias phrase as a single dynamic token (e.g., [*<Raphael>*]).

2.3. CTC scoring layer with dynamic vocabulary

Similar to conventional CTC, the scoring layer predicts an alignment sequence $A = [a_1, \dots, a_T]$ instead of directly predicting the output token sequence Y . Unlike conventional CTC, the scoring layer expands the static vocabulary $\{\mathcal{Y}^{\text{static}} \cup \{\phi\}\}$ of size K to include the dynamic vocabulary $\mathcal{V}^{\text{dynamic}}$ of size N by incorporating \mathbf{V} as follows:

$$P(Y | \mathbf{X}_{(L)}^{\text{out}}, \mathbf{V}) = \sum_{A \in \mathcal{B}^{-1}(Y)} P(A | \mathbf{X}_{(L)}^{\text{out}}, \mathbf{V}), \quad (4)$$

where ϕ represents the blank token, and $\mathcal{B}^{-1}(Y)$ represents the set of all possible alignment sequences of token sequence Y . $P(A | \mathbf{X}_{(L)}^{\text{out}}, \mathbf{V})$ is computed based on the conditional independence assumption as follows:

$$P(A | \mathbf{X}_{(L)}^{\text{out}}, \mathbf{V}) = \prod_{t=1}^T P(a_t | \mathbf{X}_{(L)}^{\text{out}}, \mathbf{V}). \quad (5)$$

To estimate the alignment probability in Eq. (5), the scoring layer is formulated as follows:

$$\mathbf{S}_{(L)}^{\text{static}} = \text{Linear}(\mathbf{X}_{(L)}^{\text{out}}), \quad (6)$$

$$\mathbf{S}_{(L)}^{\text{dynamic}} = \frac{\text{Linear}(\mathbf{X}_{(L)}^{\text{out}})\text{Linear}(\mathbf{V}^T)}{\sqrt{d}}, \quad (7)$$

$$\mathbf{Z}_{(L)} = \text{Softmax}(\text{Concat}(\mathbf{S}_{(L)}^{\text{static}}, \mathbf{S}_{(L)}^{\text{dynamic}})), \quad (8)$$

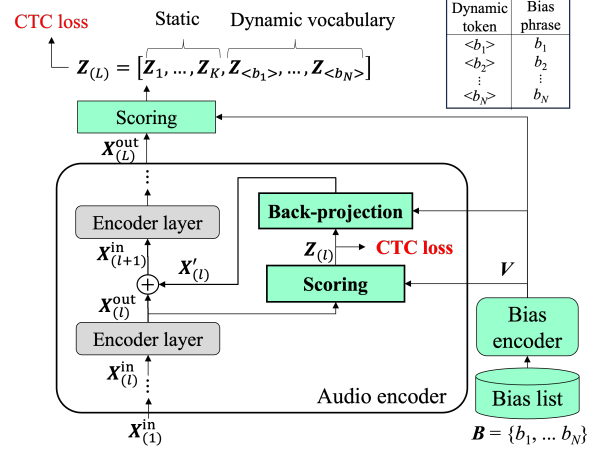


Figure 1: Overall architecture of DYNAC. Bolded blocks highlight the expansions beyond the conventional CTC-based CB method with dynamic vocabulary.

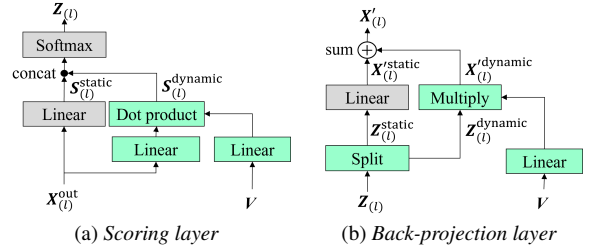


Figure 2: Components of DYNAC.

where $\mathbf{S}_{(L)}^{\text{static}} \in \mathbb{R}^{T \times K}$ and $\mathbf{S}_{(L)}^{\text{dynamic}} \in \mathbb{R}^{T \times N}$ represent the alignment scores for static and dynamic vocabulary, respectively. $\mathbf{Z}_{(L)} \in \mathbb{R}^{T \times (K+N)}$ denotes the alignment probability distribution combining static and dynamic vocabulary. $\mathbf{Z}_{(L)}$ is used to predict the alignment probability $P(A | \mathbf{X}_{(L)}^{\text{out}}, \mathbf{V})$ in Eq. (5). The model parameters are optimized by minimizing the negative log-likelihood as follows:

$$L_{\text{ctc}} = -\log P(Y | \mathbf{X}_{(L)}^{\text{out}}, \mathbf{V}) = -\log P(Y | \mathbf{Z}_{(L)}). \quad (9)$$

While CTC’s conditional independence assumption in Eq. (5) allows fast inference, it often leads to suboptimal performance. In particular, since the dynamic vocabulary is incorporated only in the final layer in Eqs. (7) and (8), the model cannot capture dependencies between dynamic tokens and surrounding static tokens, which results in poor recognition accuracy.

3. DYNAC

To address the limitation described in Section 2.3, we introduce scoring and back-projection layers into the intermediate encoder layers. This architecture extends the conventional self-conditioned CTC [18], enabling the integration of dynamic vocabulary into intermediate representations of the encoder, as shown in Figure 1.

3.1. Self-conditioned CTC with dynamic vocabulary

The scoring layer estimates the token-wise probability $\mathbf{Z}_{(l)} \in \mathbb{R}^{T \times (K+N)}$ similar to Eqs. (6), (7), and (8), but using the output of the l -th intermediate layer $\mathbf{X}_{(l)}^{\text{out}}$ and \mathbf{V} (Figure 2a).

Then, the back-projection layer (Figure 2b) converts the token-wise probability $\mathbf{Z}_{(l)}$ back into the d -dimensional hidden representation $\mathbf{X}'_{(l)} \in \mathbb{R}^{T \times d}$ while maintaining the dynamic vocabulary information. Since $\mathbf{Z}_{(l)}$ contains both static and dynamic vocabulary, which have fixed and dynamically changing sizes, we first split $\mathbf{Z}_{(l)}$ into the probabilities for the static and dynamic vocabulary components ($\mathbf{Z}_{(l)}^{\text{static}} \in \mathbb{R}^{T \times K}$ and $\mathbf{Z}_{(l)}^{\text{dynamic}} \in \mathbb{R}^{T \times N}$) as follows:

$$\mathbf{Z}_{(l)}^{\text{static}}, \mathbf{Z}_{(l)}^{\text{dynamic}} = \text{Split}(\mathbf{Z}_{(l)}). \quad (10)$$

Subsequently, $\mathbf{Z}_{(l)}^{\text{static}}$ and $\mathbf{Z}_{(l)}^{\text{dynamic}}$ are transformed to the d -dimensional hidden representation $\mathbf{X}'_{(l)}$ as follows:

$$\mathbf{X}'_{(l)} = \text{Linear}(\mathbf{Z}_{(l)}^{\text{static}}) + \mathbf{Z}_{(l)}^{\text{dynamic}} \mathbf{V}. \quad (11)$$

Here, the T -length d -dimensional hidden representation is obtained by multiplying $\mathbf{Z}_{(l)}^{\text{dynamic}} \in \mathbb{R}^{T \times N}$ with the bias phrase representation $\mathbf{V} \in \mathbb{R}^{N \times d}$ in Eq. (3). This transformation does not contain trainable parameters, allowing for efficient handling of dynamically changing vocabulary. Finally, the hidden representation $\mathbf{X}'_{(l)}$ is added to $\mathbf{X}_{(l)}^{\text{out}}$, unlike Eq. (2), and used as the input of the $(l+1)$ -th layer as follows:

$$\mathbf{X}_{(l+1)}^{\text{in}} = \mathbf{X}_{(l)}^{\text{out}} + \mathbf{X}'_{(l)}. \quad (12)$$

This feedback mechanism (Figure 1) enables the encoder to refine feature representations by incorporating contextual information through self-attention layers in the encoder, thereby capturing dependencies between static and dynamic tokens.

3.2. Intermediate CTC loss

To train DYNAC effectively, we follow [18] and introduce an auxiliary CTC loss at intermediate layers:

$$L_{\text{inter}} = -\frac{1}{|\mathcal{S}|} \sum_{l \in \mathcal{S}} \log P(Y | \mathbf{Z}_{(l)}). \quad (13)$$

\mathcal{S} represents the set of intermediate layers where the auxiliary loss is applied. We also introduce an auxiliary attention loss during training following [34]. The model parameters are optimized by combining Eqs. (9) and (13) using a tunable hyperparameter λ as follows:

$$L_{\text{total}} = \lambda L_{\text{ctc}} + \lambda L_{\text{inter}} + (1 - 2\lambda) L_{\text{att}}. \quad (14)$$

3.3. Training and inference

We follow the same training and inference strategy as [34]. For each batch, a bias list \mathbf{B} containing N bias phrases is randomly generated from the reference transcriptions. Once the bias list \mathbf{B} is defined, the corresponding static tokens are replaced with the dynamic tokens. For example, if ["Ra", "pha", "el"] is extracted from a complete utterance ["Hi", "Ra", "pha", "el"], the reference transcription is modified to ["Hi", "<Raphael>"]. During inference, we apply the bias weight to adjust the token-wise probability $\mathbf{Z}_{(L)}$ in Eq. (8) to avoid over/under-biasing. Specifically, we apply the tunable weight μ to the dynamic token probability ($\mu \mathbf{Z}_{(L)}^{\text{dynamic}}$). If $\mu < 1.0$, the dynamic tokens are underweighted compared to the static tokens; otherwise, the dynamic tokens are overweighted compared to the static tokens.

4. Experiment

We conduct several experiments to verify the effectiveness of the proposed method.

Table 1: Comparison between AR and NAR models on the LibriSpeech 960 test-clean set. **Bold** values represent the best results among the same category.

ID	Model	WER	U-WER	B-WER	RTF
<i>Autoregressive</i>					
A1	CTC/attention	3.0	1.9	12.3	0.213
A2	BPB beam search [23]	3.5	3.0	7.7	N/A
A3	Dynamic vocabulary [34]	2.0	1.9	3.3	0.165
<i>Non-autoregressive</i>					
B1	Self-conditioned CTC [18]	3.1	1.8	14.1	0.027
B2	CTC-based CB (Section 2)	40.8	45.2	6.9	0.024
B3	DYNAC (ours)	2.1	1.9	3.2	0.031

4.1. Experimental setup

The input features are 80-dimensional Mel filterbanks with a window size of 512 samples and a hop length of 160 samples. Then, SpecAugment [38] is applied. In DYNAC, the audio encoder comprises two convolutional layers with a stride of two and a 256-dimensional linear projection layer followed by 12 Conformer layers with 1024 linear units. The proposed self-conditioned CTC architecture is applied at the $\mathcal{S} = \{3, 6, 9\}$ intermediate layers (Section 3.2). The bias encoder has six Transformer blocks with 1024 linear units. DYNAC has 41.92 M parameters, including the bias encoder. During training, a bias list \mathbf{B} is created randomly for each batch, resulting in a total of $N = 50 \sim 200$ bias phrases (Section 3.3). The proposed models are trained with a static vocabulary size K of 5,000 for 70 epochs at a learning rate of 0.002 with 15,000 warmup steps using the Adam optimizer. The training weight λ in Eq. (14) and the bias weight μ (Section 3.3) are set to 0.15 and 0.1, respectively.

The proposed method is evaluated on the LibriSpeech-960 corpus [39] using the ESPnet toolkit [40], in terms of overall word error rate (WER), biased phrase WER (B-WER), and unbiased phrase WER (U-WER) [27]. In addition, we measure RTF using a CPU (Intel(R) Xeon(R) Platinum 8480C @2.00GHz) to assess inference speed. Note that the bias encoder is excluded from RTF measurement because it is only executed once when the bias list is updated (e.g., once before a session), and the resulting \mathbf{V} in Eq. (3) is reused for subsequent use. The goal of this study is to improve B-WER and RTF while minimizing degradation in U-WER.

4.2. Main results

Table 1 presents the results of WER and RTF on the LibriSpeech 960 test-clean set with a bias list size of $N = 1000$. To evaluate the impact of inference speed reduction compared to AR models, we include results from the CTC/attention model and the AR-CB method with dynamic vocabulary [34] with a beam size of 5. For NAR models, we report results from the conventional self-conditioned CTC [18] and the CTC-based CB method with dynamic vocabulary (Section 2).

DYNAC significantly reduces RTF while maintaining a WER comparable to the AR-CB method (A3 vs. B3). Compared to the NAR baseline, DYNAC slightly increases the RTF due to the introduction of the proposed self-conditioned CTC architecture, but substantially improves B-WER (B1 vs. B3). Notably, while applying the dynamic vocabulary-based CB method to a CTC-based NAR model (Section 2) improves B-WER, it severely degrades U-WER (B2). This issue is further analyzed in Section 4.3.

4.3. Analysis on token-wise scores

Figure 3 compares the token-wise scores ($\mathcal{S}_{(L)}^{\text{static}}$ and $\mathcal{S}_{(L)}^{\text{dynamic}}$ in Eqs. (6) and (7)) between the conventional CTC-based CB

Table 2: WER results on the LibriSpeech-960 (U-WER/B-WER). **Bold** values represent the best result among the same bias list size N .

Model	test-clean				test-other			
	$N=0$ (no-bias)	$N=100$	$N=500$	$N=1000$	$N=0$ (no-bias)	$N=100$	$N=500$	$N=1000$
Baseline (CTC)	3.5 (1.9/16.2)	3.5 (1.9/16.2)	3.5 (1.9/16.2)	3.5 (1.9/16.2)	8.2 (5.4/33.7)	8.2 (5.4/33.7)	8.2 (5.4/33.7)	8.2 (5.4/33.7)
Self-cond. CTC [18]	3.1 (1.8/14.1)	3.1 (1.8/14.1)	3.1 (1.8/14.1)	3.1 (1.8/14.1)	7.3 (4.7/30.2)	7.3 (4.7/30.2)	7.3 (4.7/30.2)	7.3 (4.7/30.2)
CTC-based CPPNet [24]	4.0 (2.3/17.9)	3.8 (2.2/17.0)	3.8 (2.1/17.5)	3.9 (2.4/18.3)	9.0 (5.9/35.7)	8.6 (5.6/34.4)	8.7 (5.6/35.73)	8.9 (5.7/37.3)
Intermediate CB [25]	3.0 (1.7/13.0)	2.9 (1.6/12.8)	3.1 (1.8/13.4)	3.3 (1.9/14.4)	7.4 (5.0/28.6)	7.2 (4.8/28.1)	7.7 (5.1/30.7)	8.2 (5.5/32.2)
DYNAC (ours)	3.5 (2.0/16.2)	2.0 (1.8/4.2)	1.9 (1.8/3.0)	2.1 (1.9/3.2)	8.5 (5.4/35.4)	5.4 (4.9/10.0)	5.4 (5.0/8.3)	5.8 (5.4/9.0)

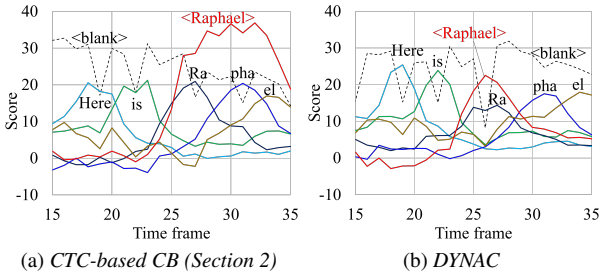


Figure 3: Comparison in token-wise score.

method (Section 2) and DYNAC. The red line represents the score of the dynamically expanded token $\langle \text{Raphael} \rangle$, while the dashed and other lines show the blank and static token scores, respectively.

In the CTC-based CB method (Figure 3a), dynamic vocabulary is incorporated only in the final layer, preventing dependencies between static and dynamic tokens from being captured. As a result, the dynamic token $\langle \text{Raphael} \rangle$ receives an excessively high score, while static tokens (“Here”, “is”) have lower scores than blank tokens, leading to a substantial degradation in U-WER (Table 1). In contrast, DYNAC (Figure 3b) addresses this issue by introducing self-conditioned CTC, allowing intermediate layers to integrate dynamic vocabulary. This approach ensures that static token scores remain higher than blank scores while preventing excessive emphasis on dynamic tokens, thereby mitigating U-WER degradation.

4.4. Impact of bias list size

Table 2 presents the impact of the bias list size N and compares DYNAC with existing NAR-CB methods. While the existing CB methods [24, 25] were primarily designed for AR models, we apply them to CTC-based NAR models for a fair comparison. DYNAC consistently improves B-WER significantly even as the bias list size increases, resulting in a better overall WER than the baseline. Moreover, DYNAC outperforms existing CB methods by a large margin.

4.5. Effectiveness on rare and unseen phrases

Figure 4 illustrates the relationship between the phrase occurrence in the training data and the B-WER. The red and blue lines represent the baseline self-conditioned CTC [18] and DYNAC with a bias list size of $N = 1000$, respectively. Phrases that do not occur in the training data (zero occurrences) correspond to unseen words. In the baseline model, B-WER increases as phrase occurrence decreases, reaching 76% for unseen words. In contrast, DYNAC consistently improves B-WER across all

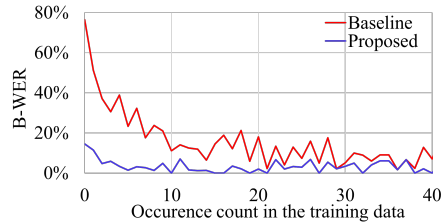


Figure 4: Performance evaluation on rare and unseen phrases.

Table 3: Experimental results obtained on Japanese dataset. **Bold** values represent better results in the same category.

Model	CER	U-CER	B-CER	RTF
<i>Autoregressive</i>				
CTC/attention	9.9	8.2	21.8	0.171
Dynamic vocab [34]	9.0	8.9	9.7	0.164
<i>Non-autoregressive</i>				
Self-cond. CTC [18]	11.8	9.9	25.9	0.023
DYNAC (ours)	10.6	10.6	10.6	0.023

occurrence count ranges, with particularly notable gains for words occurring less than 20 times. Moreover, DYNAC remains effective even for unseen phrases, achieving a B-WER of 14.6%.

4.6. Validation on Japanese dataset

We further evaluate DYNAC on a Japanese dataset containing the Corpus of Spontaneous Japanese (581 hours) [41], 181 hours of speech from a database developed by the Advanced Telecommunications Research Institute International [42], and 93 hours of our in-house recordings, with a static vocabulary size K of 3,613. Table 3 shows the results when $N = 203$ phrases are registered in the bias list. Similar to the experiments on the LibriSpeech corpus (Table 1), DYNAC significantly reduces RTF compared to AR-CB methods [34] while substantially improving B-WER. This demonstrates the effectiveness of DYNAC across languages with entirely different vocabulary structures.

5. Conclusion

This paper proposes DYNAC (Dynamic Vocabulary-based NAR Contextualization), an NAR-CB method that integrates dynamic vocabulary into intermediate encoder layers using self-conditioned CTC, enabling efficient inference with low RTF. Experimental results demonstrate that DYNAC significantly reduces RTF while maintaining WER comparable to AR-CB methods on both the LibriSpeech 960 corpus and our in-house Japanese dataset.

6. References

- [1] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schluter, and S. Watanabe, "End-to-end speech recognition: A survey," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 325–351, 2023.
- [2] J. Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.
- [3] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in neural information processing systems*, vol. 28, 2015.
- [4] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.
- [5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023, pp. 28 492–28 518.
- [6] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan *et al.*, "OWSM v3.1: Better and faster open Whisper-style speech models based on e-branchformer," in *Proc. Interspeech*, 2024, pp. 352–356.
- [7] A. Graves, "Sequence transduction with recurrent neural networks," in *Proc. ICML*, 2012.
- [8] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020, pp. 5036–5040.
- [9] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.
- [10] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. ICML*, 2014, pp. 1764–1772.
- [11] Y. Peng, Y. Sudo, M. Shakeel, and S. Watanabe, "OWSM-CTC: An open encoder-only speech foundation model for speech recognition, translation, and language identification," in *Proc. ACL*, 2024, pp. 10 192–10 209.
- [12] Y. Higuchi, N. Chen, Y. Fujita, H. Inaguma, T. Komatsu *et al.*, "A comparative study on non-autoregressive modelings for speech-to-text generation," in *Proc. ASRU*, 2021, pp. 47–54.
- [13] J. Lee and S. Watanabe, "Intermediate loss regularization for CTC-based speech recognition," in *Proc. ICASSP*, 2021, pp. 6224–6228.
- [14] W. Chan, C. Saharia, G. E. Hinton, M. Norouzi, and N. Jaitly, "Imputer: Sequence modelling via imputation and dynamic programming," in *Proc. ICML*, 2020, pp. 1403–1413.
- [15] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, "Mask CTC: Non-autoregressive end-to-end asr with CTC and mask predict," in *Proc. Interspeech*, 2020, pp. 3655–3659.
- [16] E. A. Chi, J. Salazar, and K. Kirchhoff, "Align-refine: Non-autoregressive speech recognition via iterative realignment," in *Proc. NAACL HLT*, 2021, pp. 1920–1927.
- [17] A. Tjandra, C. Liu, F. Zhang, X. Zhang, Y. Wang *et al.*, "Deja-vu: Double feature presentation and iterated loss in deep transformer networks," in *Proc. ICASSP*, 2020, pp. 6899–6903.
- [18] J. Nozaki and T. Komatsu, "Relaxing the conditional independence assumption of CTC-based asr by conditioning on intermediate predictions," in *Proc. Interspeech*, 2021, pp. 3735–3739.
- [19] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *Proc. SLT*, 2018, pp. 418–425.
- [20] M. Jain, G. Keren, J. Mahadeokar, and Y. Saraf, "Contextual RNN-T for open domain ASR," in *Proc. Interspeech*, 2020, pp. 11–15.
- [21] C. Huber, J. Hussain, S. Stüker, and A. Waibel, "Instant one-shot word-learning for context-specific neural sequence-to-sequence speech recognition," in *Proc. ASRU*, 2021, pp. 1–7.
- [22] S. Zhou, Z. Li, Y. Hong, M. Zhang, Z. Wang, and B. Huai, "Copyne: Better contextual ASR by copying named entities," *arXiv preprint arXiv:2305.12839*, 2023.
- [23] Y. Sudo, M. Shakeel, Y. Fukumoto, Y. Peng, and S. Watanabe, "Contextualized automatic speech recognition with attention-based bias phrase boosted beam search," in *Proc. ICASSP*, 2024, pp. 10 896–10 900.
- [24] K. Huang, A. Zhang, Z. Yang, P. Guo, B. Mu *et al.*, "Contextualized end-to-end speech recognition with contextual phrase prediction network," in *Proc. Interspeech*, 2023, pp. 4933–4937.
- [25] M. Shakeel, Y. Sudo, Y. Peng, and S. Watanabe, "Contextualized end-to-end automatic speech recognition with intermediate biasing loss," in *Proc. Interspeech*, 2024, pp. 3909–3913.
- [26] Y. Nakagome and M. Hentschel, "Interbiasing: Boost unseen word recognition through biasing intermediate predictions," in *Proc. Interspeech*, 2024, pp. 207–211.
- [27] D. Le, M. Jain, G. Keren, S. Kim *et al.*, "Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion," in *Proc. Interspeech*, 2021, pp. 1772–1776.
- [28] J. Qiu, L. Huang, B. Li, J. Zhang, L. Lu, and Z. Ma, "Improving large-scale deep biasing with phoneme features and text-only data in streaming transducer," in *Proc. ASRU*, 2023, pp. 1–8.
- [29] X. Wang, Y. Liu, J. Li, V. Miljanic, S. Zhao, and H. Khalil, "Towards contextual spelling correction for customization of end-to-end speech recognition systems," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 3089–3097, 2022.
- [30] X. Wang, Y. Liu, J. Li, and S. Zhao, "Improving contextual spelling correction by external acoustics attention and semantic aware data augmentation," in *Proc. ICASSP*, 2023, pp. 1–5.
- [31] A. Bruguier, R. Prabhavalkar, G. Pundak, and T. N. Sainath, "Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition," in *Proc. ICASSP*, 2019, pp. 6171–6175.
- [32] H. Futami, E. Tsunoo, Y. Kashiwagi, H. Ogawa, S. Arora, and S. Watanabe, "Phoneme-aware encoding for prefix-tree-based contextual asr," in *Proc. ICASSP*, 2024.
- [33] Y. Sudo, K. Hata, and K. Nakadai, "Retraining-free customized ASR for enharmonic words based on a named-entity-aware model and phoneme similarity estimation," in *Proc. Interspeech*, 2023, pp. 3312–3316.
- [34] Y. Sudo, Y. Fukumoto, M. Shakeel, Y. Peng, and S. Watanabe, "Contextualized automatic speech recognition with dynamic vocabulary," in *Proc. SLT*, 2024, pp. 78–85.
- [35] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [36] Y. Sudo, M. Shakeel, Y. Fukumoto, B. Yan, J. Shi, Y. Peng, and S. Watanabe, "Joint beam search integrating ctc, attention, and transducer decoders," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 33, pp. 598–612, 2025.
- [37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 5998–6008.
- [38] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu *et al.*, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.
- [39] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.
- [40] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.
- [41] K. Maekawa, "Corpus of spontaneous Japanese: Its design and evaluation," in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.
- [42] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," *Speech Communication*, vol. 9, no. 4, pp. 357–363, 1990.