

Retraining-free Customizable End-to-end Automatic Speech Recognition

Yui SUDO, Ph.D.¹Shakeel MUHAMMAD, Ph.D.¹Naoaki SUMIDA¹

ABSTRACT

In this paper, the authors propose a method of customizing automatic speech recognition models using editable term lists to increase recognition accuracy for phrases registered at the discretion of the user. The proposed method adds a text encoder and cross-attention layer to the conventional Connectionist Temporal Classification/Attention-based automatic speech recognition model to detect registered terms in speech signals. To increase registered term recognition accuracy, the authors propose an inference algorithm that integrates the probability of detected registered words and Connectionist Temporal Classification. Results of their evaluation using the in-house Japanese-language dataset and the English-language Librispeech corpus showed that using the proposed method lowered the character error rate and the word error rate for registered terms.

1. Introduction

The development of deep learning has helped to advance automatic speech recognition technology. In particular, in recent years, end-to-end automatic speech recognition (E2E-ASR) models, a method that estimates text sequences directly from speech input, have achieved great increases in performance⁽¹⁾. These dramatic increases in performance have led to the incorporation of ASR into many applications, including smart speakers, car navigation systems, meeting minutes transcription systems, and the Honda Communication Assistance System^{(2),(3)}.

E2E-ASR uses a single neural network to convert input speech signals directly into a text sequence, unlike conventional systems that include multiple components such as an acoustic model, lexicon, and language model. Neural networks are optimized using deep learning to mitigate error accumulation between components in conventional systems. A typical ASR model comprises an encoder that encodes the input speech and a decoder that converts the encoded speech features into a text sequence. Previously, various decoders have been proposed, such as Connectionist Temporal Classification (CTC)^{(4),(5)}, Recurrent Neural Network Transducer^{(6),(7)}, and attention mechanism^{(8),(9)}. Hybrid systems combining these methods have also been proposed^{(10),(11)}, among which CTC/Attention-based E2E-ASR is one of widely used methods due to its high performance⁽¹⁰⁾.

The performance of ASR using deep learning, however, tends to be highly dependent on the training data, making

it a challenge to correctly recognize words and phrases that rarely appear in the training data. For example, technical terms and personal names are often incorrectly recognized because they rarely appear in training data. However, such words may be important keywords for certain users, and their misrecognition may greatly reduce user satisfaction. It is possible to enhance recognition performance by collecting training data that includes such words and retraining the neural network. However, frequent retraining is not possible because human resources and high-performance GPU servers are required for collecting training data, annotating, and training neural networks. Therefore, a retraining-free ASR model customization method is needed.

Other research has approached this need by proposing customization methods that can register any word^{(12),(13)}. These methods add a cross-attention layer⁽¹⁴⁾ to the decoder, enabling detection of registered terms with any parts of speech without retraining. Furthermore, to increase registered term detection accuracy, methods to introduce auxiliary branches^{(13),(15)} and auxiliary loss functions in the cross-attention layer have been proposed⁽¹⁶⁾. These methods, however, do not explicitly use auxiliary branches or auxiliary losses during inference, limiting how much their performance with registered terms can be increased.

2. Aims of the Research

This paper proposes a method to customize E2E-ASR models utilizing user-editable term lists without retraining. The proposed method detects registered terms in speech

¹ Honda Research Institute Japan Co., Ltd.

signals by adding a text encoder and cross-attention layer to a CTC/Attention-based E2E-ASR model (Fig. 1)⁽¹⁰⁾, similar to the conventional customization method⁽¹⁶⁾ (Fig. 2). Unlike the conventional customization method⁽¹⁶⁾, however, to further increase registered term recognition accuracy, this paper proposes an inference algorithm that increases the probability of detected registered terms and integrates the output probabilities with a CTC decoder. The aims of this research are as follows.

- Enhance registered term recognition performance by increasing the probability of registered terms detected during inference.
- Enhance overall recognition performance by integrating output probabilities using the CTC decoder during inference.
- Conduct a detailed analysis of performance with a Japanese-language dataset.

This paper extends earlier research⁽¹⁶⁾. Specifically, in respect to the preceding research⁽¹⁷⁾, the researchers integrated the output probabilities using a CTC decoder and conducted a new detailed performance analysis with a Japanese-language dataset.

3. Conventional Model (CTC/Attention Model)

The CTC/Attention model⁽¹⁰⁾, a conventional ASR model, is shown in Fig. 1. The CTC/Attention model has an audio encoder, an attention decoder, and a CTC decoder. The attention decoder and the CTC decoder share an audio encoder. Given an audio feature sequence $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_L]$ of length L , posterior probability $P_{\text{att}}(y | \mathbf{X})$ and $P_{\text{ctc}}(y | \mathbf{X})$ of text sequence $y = [y_1, \dots, y_S]$ of length S are respectively estimated. The decoders and audio encoder are trained jointly, and the output probabilities of the two decoders are integrated during inference. This approach could stabilize learning and enhance performance during inference. The following section describes each component in detail.

3.1. Audio Encoder

A conformer is used for the audio encoder⁽⁷⁾, and converts the audio feature sequence $\mathbf{X}$ to a hidden state vector $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T] \in \mathbb{R}^{T \times d}$ of length T as follows.

$$\mathbf{H} = \text{AudioEnc}(\mathbf{X}) \quad (1)$$

where d is the dimensions of the hidden state vector.

3.2. Attention Decoder

A transformer decoder is used as the attention decoder⁽¹⁴⁾. When given the hidden state vector $\mathbf{H}$ extracted by the audio encoder shown in Eq. (1) and the text sequence $y_{0:s-1} = [y_0, \dots, y_{s-1}]$ up to one frame previously, the attention decoder estimates the posterior probability $P(y_s | y_{0:s-1}, \mathbf{X})$ of the s th token y_s as follows.

$$P_{\text{att}}(y_s | y_{0:s-1}, \mathbf{X}) = \text{AttDecoder}(y_{0:s-1}, \mathbf{H}) \quad (2)$$

By repeating Eq. (2) for $s = 1, \dots, S$, the posterior probability of text sequence y is formulated as follows.

$$P_{\text{att}}(y | \mathbf{X}) = \prod_{s=1}^S P_{\text{att}}(y_s | y_{0:s-1}, \mathbf{X}) \quad (3)$$

Cross entropy loss is introduced to the above.

$$L_{\text{att}} = -\log P_{\text{att}}(y | \mathbf{X}) \quad (4)$$

3.3. CTC Decoder

The CTC decoder comprises one linear layer^{(4), (5)}. As Fig. 1 shows, the CTC decoder uses the hidden state vector $\mathbf{H}$ output by the audio encoder to estimate the posterior probability $P_{\text{ctc}}(y | \mathbf{X})$ of text sequence y , as shown below.

$$P_{\text{ctc}}(y | \mathbf{X}) = \text{CTC}(\mathbf{H}) \quad (5)$$

The CTC loss shown below is introduced to the above output.

$$L_{\text{ctc}} = -\log P_{\text{ctc}}(y | \mathbf{X}) \quad (6)$$

3.4. Joint Training and Joint Decoding

The CTC/Attention model is optimized to minimize loss by joint training using the weight λ shown below for the loss function defined in Eqs. (4) and (6).

$$L = (1 - \lambda) L_{\text{att}} + \lambda L_{\text{ctc}} \quad (7)$$

Additionally, to enhance performance during inference, the output probabilities of the attention decoder and CTC decoder are integrated using the weight μ shown below.

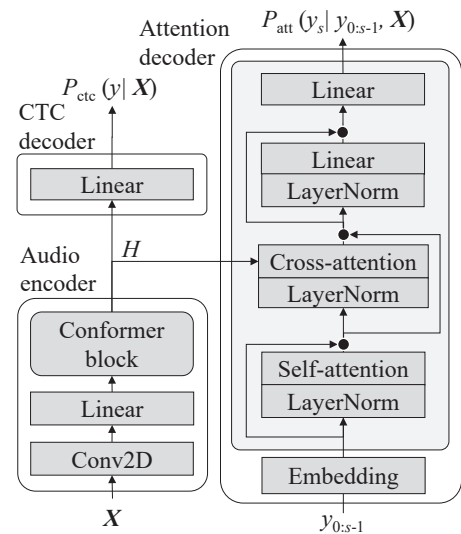


Fig. 1 Overall architecture of CTC/Attention model

$$\alpha = (1 - \mu) \alpha_{\text{att}} + \mu \alpha_{\text{ctc}}, \quad (8)$$

$$\alpha_{\text{att}} = \log P_{\text{att}}(y | \mathbf{X}), \quad (9)$$

$$\alpha_{\text{ctc}} = \log P_{\text{ctc}}(y | \mathbf{X}), \quad (10)$$

where α , α_{att} , and α_{ctc} are the respective output probabilities of the integration probability, attention decoder, and CTC decoder.

Conventional CTC/Attention models do not assume external input such as term lists, so these models cannot be customized after training, and it is a challenge to accurately recognize words that do not appear in the training data.

4. Network Structure of the Proposed Model

The overall structure of the proposed model is shown in Fig. 2. In addition to the conventional CTC/Attention model, the proposed method has a text encoder for inputting term lists and adds cross-attention to the attention decoder (green parts in Fig. 2). The text encoder encodes a list of terms that can be edited at the discretion of the user, and the cross-attention layer detects registered terms in the input speech.

When given an audio feature sequence $\mathbf{X}$ and a term list $\mathbf{B} = \{b_0, b_1, \dots, b_N\}$, the extended attention decoder estimates the posterior probability $P_{\text{ext}}(y | \mathbf{X}, \mathbf{B})$ of a text sequence y of length S and the posterior probability $P_{\text{idx}}(n | \mathbf{X}, \mathbf{B})$ of index $n = [n_1, \dots, n_S]$ of the detected registered terms (Fig. 2). The following section discusses each additional component.

4.1. Text Encoder

A transformer encoder is used as the text encoder⁽¹²⁾. The transformer encoder consists of an embedding layer,

a transformer block, a mean pooling layer, and an editable term list $\mathbf{B} = \{b_0, b_1, \dots, b_N\}$. b_N denotes the text sequence of the N th registered term (e.g. [morning, assembly]). b_0 is a dummy word used when there is no registered term in the speech signal. The term list $\mathbf{B}$ is input to the embedding layer and transformer block and converted into a token-by-token feature sequence $\mathbf{G} \in \mathbb{R}^{(N+1) \times l_{\text{max}} \times d}$ as shown below.

$$\mathbf{G} = \text{Transformer}(\text{Embedding}(\mathbf{B})) \quad (11)$$

l_{max} is the maximum length of a word in the term list (e.g. $l_{\text{max}} = 2$ if [morning, assembly] is the maximum length). It is then converted into a feature $\mathbf{V} = [v_0, \dots, v_N] \in \mathbb{R}^{(N+1) \times d}$ representing the features of the whole word using the mean pooling layer.

$$\mathbf{V} = \text{MeanPool}(\mathbf{G}) \quad (12)$$

4.2. Extended Attention Decoder

The extended attention decoder uses the hidden state vectors $\mathbf{H}$ and $\mathbf{V}$ extracted by the audio encoder and text encoder shown in Eqs. (1) and (12) to estimate the posterior probability $P_{\text{ext}}(y | \mathbf{X}, \mathbf{B})$ of the text sequence and the index probability $P_{\text{idx}}(n | \mathbf{X}, \mathbf{B})$ of the detected registered term⁽¹²⁾.

4.2.1. Text sequence probability

When given the hidden state vectors $\mathbf{H}$ and $\mathbf{V}$ extracted by the audio encoder and text encoder shown in Eqs. (1) and (12), and the text sequence $y_{0:s-1}$ up to one frame previously, the decoder estimates the s th token y_s as follows.

$$P_{\text{ext}}(y_s | y_{0:s-1}, \mathbf{X}, \mathbf{B}) = \text{ExtDecoder}(y_{0:s-1}, \mathbf{H}, \mathbf{V}) \quad (13)$$

By repeating Eq. (13) for $s = 1, \dots, S$, the posterior probability of text sequence y is formulated as follows.

$$P_{\text{ext}}(y | \mathbf{X}, \mathbf{B}) = \prod_{s=1}^S P_{\text{ext}}(y_s | y_{0:s-1}, \mathbf{X}, \mathbf{B}) \quad (14)$$

Cross entropy loss is introduced to the above.

$$L_{\text{ext}} = -\log P_{\text{ext}}(y | \mathbf{X}, \mathbf{B}) \quad (15)$$

4.2.2. Registered term index probability

Writing the input and output of the cross-attention layer of Fig. 2 as $\mathbf{U}$ and $\mathbf{U}'$, respectively, the cross-attention layer is defined as follows.

$$\mathbf{U}' = \text{Softmax}\left(\frac{\text{LN}(\mathbf{U})\mathbf{V}^T}{\sqrt{d}}\right)\mathbf{V} + \mathbf{U} \quad (16)$$

The term Softmax in Eq. (16) is the inner product of the input feature sequence $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_s]$ and the term list feature sequence $\mathbf{V} = [v_0, \dots, v_N]$, so it indicates their degree

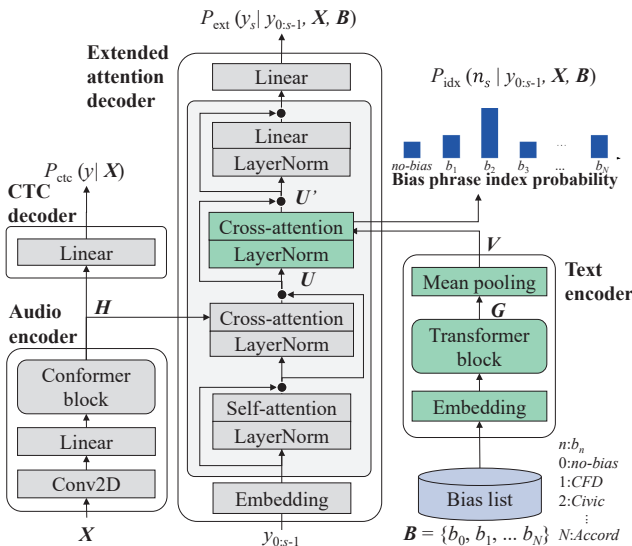


Fig. 2 Overall architecture of proposed method

of similarity; that is, it can be interpreted as the probability of the index sequence $n = [n_1, \dots, n_s]$ of registered terms in the input speech, as shown below.

$$P_{\text{idx}}(n | \mathbf{X}, \mathbf{B}) = \text{Softmax}\left(\frac{\text{LN}(\mathbf{U})\mathbf{V}^T}{\sqrt{d}}\right) \quad (17)$$

The index with the highest probability is a registered term contained in the speech. For example, if two words (1: morning assembly, 2: PR) are registered in the term list and the utterance $y = [\text{do PR at morning assembly}]$ is spoken, the index sequence of registered terms ($n = [0, 2, 0, 1, 1]$, $n = 0$ [no-phrase]) is used for tokens that are not registered terms. Registered term index loss using cross entropy is introduced to the above output.

$$L_{\text{idx}} = -\log P_{\text{idx}}(n | \mathbf{X}, \mathbf{B}) \quad (18)$$

5. Model Training and Inference Methods

5.1. Training Method

The proposed method is optimized to minimize the loss by multi-task learning using the weights λ and γ shown below for the loss function defined in Eqs. (6), (15), and (17).

$$L = (1 - \lambda) L_{\text{ext}} + \lambda L_{\text{ctc}} + \gamma L_{\text{idx}} \quad (19)$$

During training, any word can be registered to the term list during inference, whereas a term list is randomly created from ground truth for each batch during training. This random selection is intended to allow the model to accommodate a wide range of words and phrases.

5.2. Inference Algorithm

To further enhance recognition performance, this paper proposes a new inference algorithm. Specifically, for the posterior probability $P_{\text{ext}}(y | \mathbf{X}, \mathbf{B})$ of text sequence y estimated by the extended attention decoder, (1) the index probabilities $P_{\text{idx}}(n | \mathbf{X}, \mathbf{B})$ of registered terms are integrated (red part of Fig. 3), and (2) the output probabilities by the CTC decoder are integrated (blue part of Fig. 3). The proposed algorithm is shown in Fig. 3.

- (1) First, the extended attention decoder calculates the probability α_{ext} of each s th token based on Eq. (4) (line 5). Next, the s th registered term index n_s is estimated using Eq. (8) and the argmax function (line 6). If a registered term is detected ($n_s \neq 0$), the output probability of the tokens in the registered term is increased based on the weight α_{bonus} (line 9). For example, if the detected registered term is $b_n = [\text{morning, assembly}]$, the output probability of the tokens [morning] and [assembly] is increased.
- (2) Then, the output probability α_{ctc} of the CTC decoder is calculated using CTC prefix scoring⁽¹⁰⁾. However,

since term list $\mathbf{B}$ is not input to the CTC decoder (Fig. 2), the probability of registered terms is not calculated. In other words, only if no registered terms are detected ($n_s = 0$), output probabilities are integrated using the weight μ (line 16).

6. Evaluation Experiment

6.1. Experimental Conditions

The input audio feature $\mathbf{X}$ is an 80-dimensional mel filter bank with a window length of 512 samples. A mel filter bank represents a time-frequency characteristic using a mel scale that is modeled on human auditory characteristics. The audio encoder consists of two convolutional layers, a 256-dimensional linear projection layer, and 12 conformer blocks. The text encoder includes an embedding layer and six transformer blocks and has a mean pooling layer in the final layer. The extended attention decoder comprises an embedding layer, six transformer blocks, and a linear layer. The CTC decoder consists of one linear layer. The weights λ and γ during training as shown in Eq. (19) were respectively set to 0.3 and 1.0. The parameters α_{bonus} and μ during inference, as described in Section 5.2, were respectively set to 1.0 and 0.3.

An in-house dataset (Japanese) and the Librispeech-100 corpus (English)⁽¹⁸⁾ were used for evaluation. The in-house dataset consists of 93 hours of in-house speech data including meeting and morning assembly scenarios,

Algorithm 1 Decoding algorithm

```

1:  $\text{hyps} = \{<\text{sos}>; 1.0\}$ ;  $\text{end\_hyps} = \{\}$ ;
2: for  $s = 1$  to  $\text{maxlen}$  do
3:    $\text{ext\_hyps} = \{\}$ 
4:   for  $l \in \text{hyps}$  do
5:      $\alpha_{\text{ext}} = \log P_{\text{ext}}(y_s | y_{1:s-1}, \mathbf{X}, \mathbf{B})$ 
6:      $n_s \leftarrow \text{Argmax}(P(n_s | y_{1:s-1}, \mathbf{X}, \mathbf{B}))$ 
7:     if  $n_s \neq 0$  then
8:       for token  $\in b_{n_s}$  do
9:          $\alpha_{\text{ext}}[\text{token}] \leftarrow \alpha_{\text{ext}}[\text{token}] + \alpha_{\text{bonus}}$ 
10:      end for
11:    end if
12:    for  $y_s \in \text{Top-k}_{\text{beam}}(\alpha_{\text{ext}})$  do
13:      Add  $\tilde{l} = l \oplus y_s$  and the score  $\alpha_{\text{ext}}$  to  $\text{ext\_hyps}$ 
14:       $\alpha_{\text{ctc}} = \text{CTCPrefixScore}(\tilde{l}, \mathbf{H})$ 
15:      if  $\hat{n}_s = 0$  then ("no-bias")
16:         $\text{ext\_hyps}[\tilde{l}] = \mu \alpha_{\text{ctc}} + (1 - \mu) \alpha_{\text{ext}}$ 
17:      else
18:         $\text{ext\_hyps}[\tilde{l}] = \alpha_{\text{ext}}$ 
19:      end if
20:    if  $y_s = <\text{eos}>$  or  $\text{len}(\tilde{l}) = \text{maxlen}$  then
21:      Add  $\tilde{l}$  to  $\text{end\_hyps}$ 
22:    end if
23:  end for
24: end for
25:  $\text{hyps} = \text{Top-k}(\text{ext\_hyps})$ 
26: end for
27: return  $\text{Top-1}(\text{end\_hyps})$ 

```

Fig. 3 Joint decoding algorithm

the Corpus of Spontaneous Japanese⁽¹⁹⁾, and a database developed by the Advanced Telecommunications Research Institute International⁽²⁰⁾. The proposed method was evaluated based on character error rate (CER), CER for registered terms (B-CER), and CER for other (U-CER). The objective of the proposed method is to increase B-CER while minimizing the increase in U-CER and overall CER. With Librispeech-100, word error rate (WER), U-WER, and B-WER were used.

6.2. Verification with Japanese Dataset

Table 1 shows the evaluation results when $N = 203$ words or phrases are registered in the term list. The baseline was a conventional non-customizable CTC/Attention model⁽¹⁰⁾ (A1). Results were also compared with conventional customization methods^{(15), (21)} (B1, 2). Unlike the proposed method, these methods do not explicitly use the probability of registered terms detected during inference. For the proposed method, the researchers also tested the impact of excluding the inference algorithms (registered term index probability integration and probability integration by CTC decoder) described in Section 5.2. (C2, 3).

First, the proposed method was compared with the baseline and conventional customization methods (A1, B1, and B2 vs. C1). C1 decreased B-CER (CER for registered terms) by 11.71 points compared to baseline, although U-CER (CER for other than registered terms) increased by 0.06 points. As a result, overall CER decreased by 1.40 points. The proposed method also showed a decrease in both overall CER and B-CER over conventional customization methods^{(15), (21)}.

Next, this paper discusses the impact of excluding registered term index probability integration and probability integration by CTC decoder. As shown in C2 and C3, excluding registered term index probability integration increased the B-CER by 1.96 points, and excluding probability integration by CTC decoder increased the U-CER by 0.97 points. This confirmed that their exclusion is effective for enhancing recognition performance for respective registered terms and other words.

6.2.1. Performance analysis

Registered terms were then classified into words written in katakana, the alphabet (such as acronyms written with the alphabet), kanji, and English, and the recognition accuracy was analyzed. Words written with multiple character types were classified according to the type of character they used the most. For example, “CFD 計算” was categorized as written with the alphabet, because it uses mostly alphabetical characters. Words in the alphabet category are those pronounced as readings of the alphabetic characters, such as “PR,” whereas words in the English category are those pronounced otherwise, such as “sketch.”

Table 2 and Fig. 4 show the recognition accuracy for each category of registered terms, as well as examples of

Table 1 Experimental results on in-house dataset

ID	Model	CER [%]	U-CER [%]	B-CER [%]
A1	Baseline	9.85	8.17	21.86
B1	NEA-ASR [21]	9.75	8.11	21.90
B2	Intermediate DB [15]	9.28	8.23	16.93
C1	Proposed method	8.45	8.23	10.15
C2	C1 w/o index probability	8.65	8.19	12.11
C3	C2 w/o CTC integration	9.78	9.16	14.54

Table 2 Performance analysis on in-house dataset

Class	Accuracy [%]	Examples
Katakana	86	ジャイロ, リファイン
Alphabet	84	CFD, PR
Kanji	64	接種, 風洞
English	29	SKETCH, CASCADE

typical inference results. In inference results examples 1 to 3, the registered terms were correctly recognized by the proposed method, while in example 4, the registered term was not correctly recognized. Table 2 shows that for words where each character and its pronunciation are uniquely determined, such as in the katakana and alphabet categories, the recognition accuracy was over 80%, whereas the recognition accuracy for the English category was clearly low, just 29%. In particular, the recognition accuracy of English words like “sketch” and “cascade,” which are commonly written in katakana, was far lower.

This can be attributed to the fact that the words registered during training were selected at random. The cross-attention layer added to the text encoder and the extended attention decoder in the proposed method (C1) is trained to detect the pronunciation of registered words from the input speech. This may have resulted in insufficient training for words with uncommon combinations of spelling and pronunciation.

- Reference: 一番のネックは風洞だって事だ
Baseline: 一番のネッ**カー**フー**ド**だって事だ
Proposed (C1): 一番の**ネッ**クは風洞だって事だ
- Reference: それとは別の**職域接種**
Baseline: それとは別の**職域**撰**取**
Proposed (C1): それとは別の**職域**接**種**
- Reference: **被引例情報**のときに**第一引例**って
Baseline: **非陰例情報**のときに**第一因例**って
Proposed (C1): **非引例情報**のときに**第一引例**って
- Reference: **SKETCH** モードを開いて
Baseline: **スケッチ**モードを開いて
Proposed (C1): **スケッチ**モードを開いて

Boldface: Registered phrase
Red: Incorrectly recognized characters
Blue: Correctly recognized characters

Fig. 4 Typical inference results

6.2.2. Scalability verification

In actual use cases, the number of words registered to a term list tends to increase unless users explicitly delete registered terms. Figure 5 shows the effect of term list size on B-CER (CER for registered words). In this experiment, we compared B-CERs when no words were registered to the term list ($N = 0$), when words appropriate to the use case were registered ($N = 203$), and when unnecessary words were added ($N = 500$ to 2000).

When no words were registered to the term list ($N = 0$), there was no decrease in B-CER, whereas when appropriate words were registered ($N = 203$), B-CER decreased. As the number of words unnecessary for the use case in question increased, the B-CER gradually increased ($N = 500$ to 2000). It was confirmed, however, that the proposed method is robust to increases in term list size, since it maintained a sufficient decrease effect in B-CER compared to the case where no words are registered to the term list ($N = 0$).

6.3. Verification with English Dataset

Next, the researchers evaluated a case of $N = 100$ words registered in a term list as an English dataset (Librispeech-100), since English has different characteristics from Japanese. Since Japanese is an ideographic language and English is a phonetic language, it is important to evaluate languages like these that have different characteristics. Table 3 shows the effectiveness of the proposed method, registered term index probability integration, and probability integration by CTC. A similar trend was observed for the English evaluation results for Japanese. The proposed method decreased the B-CER and thus decreased the overall WER compared to the baseline (A1 vs. C1). The B-WER and U-WER decreased when registered term index probability integration and probability integration by CTC decoder were excluded, thus confirming the effectiveness of these methods in enhancing the recognition performance in each case (C1 vs. C2, 3).

Additionally, Fig. 6 shows the registered term index probabilities estimated by the cross-attention layer based on Eq. (8). The horizontal axis shows the input speech utterances, and the vertical axis shows the registered words.

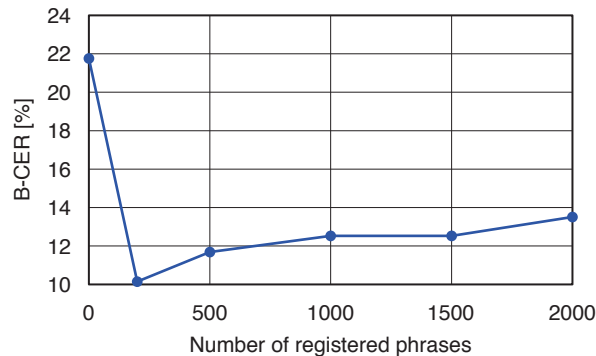


Fig. 5 Robustness to term list size

Table 3 Experimental results on Librispeech-100 h

ID	Model	WER [%]	U-WER [%]	B-WER [%]
A1	Baseline	8.59	5.87	30.71
C1	Proposed method	5.76	4.10	19.30
C2	C1 w/o index probability	6.49	4.83	19.98
C3	C2 w/o CTC integration	7.53	5.27	25.92

These utterances included the registered words “legality” and “legislature.” Figure 6 confirms that the proposed method can accurately detect registered words from the input speech.

The above confirmed that the proposed method is effective for languages with different characteristics (e.g. Japanese and English).



Fig. 6 Registered phrase detection

7. Conclusion

In this paper, the authors propose a method of customizing E2E-ASR models using editable term lists to increase recognition accuracy for phrases registered at the discretion of the user. In the proposed method, a text encoder and cross-attention layer are added to a conventional CTC/Attention-based E2E-ASR model to detect registered terms in speech signals. Additionally, to further increase registered term recognition accuracy, this paper proposes an inference algorithm that increases the probability of detected registered terms and uses probability integration with a CTC decoder. Results of an evaluation using an in-house dataset (Japanese) and the Librispeech corpus (English) demonstrated that the proposed method lowered the character error rate and the word error rate for registered terms.

References

- (1) Prabhavalkar, R., Hori, T., Sainath, T. N., Schlüter, R., Watanabe, S.: End-to-End Speech Recognition: A Survey, IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 32, p. 325-351, (2023), doi: 10.1109/TASLP.2023.3328283
- (2) <https://global.honda/jp/stories/003.html>, (accessed 2024/9/23) (in Japanese)

- (3) <https://www.jp.honda-ri.com/en/activity/honda-ca-system/>, (accessed 2024/9/23)
- (4) Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks, ICML '06: Proceedings of the 23rd international conference on Machine learning, p. 369-376, (2006), doi: 10.1145/1143844.1143891
- (5) Peng, Y., Sudo, Y., Shakeel, M., Watanabe, S.: OWSM-CTC: An Open Encoder-Only Speech Foundation Model for Speech Recognition, Translation, and Language Identification, Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Vol. 1, p. 10192–10209, (2024), doi: 10.18653/v1/2024.acl-long.549
- (6) Graves, A.: Sequence Transduction with Recurrent Neural Networks, International Conference of Machine Learning (ICML) 2012 Workshop on Representation Learning, arXiv preprint, (2012), doi: 10.48550/arXiv.1211.3711
- (7) Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., Pang, R.: Conformer: Convolution-augmented Transformer for Speech Recognition, Proc. Interspeech 2020, p. 5036-5040, (2020), doi: 10.21437/Interspeech.2020-3015
- (8) Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-Based Models for Speech Recognition, Advances in Neural Information Processing Systems, Vol. 28, (2015)
- (9) Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust Speech Recognition via Large-Scale Weak Supervision, Proceedings of the 40th International Conference on Machine Learning, p. 28492-28518, (2023)
- (10) Watanabe, S., Hori, T., Kim, S., Hershey, J. R., Hayashi, T.: Hybrid CTC/Attention Architecture for End-to-End Speech Recognition, IEEE Journal of Selected Topics in Signal Processing, Vol. 11, Issue 8, p. 1240-1253, (2017), doi: 10.1109/JSTSP.2017.2763455
- (11) Sudo, Y., Shakeel, M., Yan, B., Shi, J., Watanabe, S.: 4D ASR: Joint modeling of CTC, Attention, Transducer, and Mask-Predict decoders, Proc. INTERSPEECH 2023, p. 3312-3316, (2023), doi: 10.21437/Interspeech.2023-1303
- (12) Pundak, G., Sainath, T. N., Prabhavalkar, R., Kannan, A., Zhao, D.: Deep Context: End-to-end Contextual Speech Recognition, 2018 IEEE Spoken Language Technology Workshop (SLT), (2018), doi: 10.1109/SLT.2018.8639034
- (13) Huang, K., Zhang, A., Yang, Z., Guo, P., Mu, B., Xu, T., Xie, L.: Contextualized End-to-End Speech Recognition with Contextual Phrase Prediction Network, Proc. INTERSPEECH 2023, p. 4933-4937, (2023), doi: 10.21437/Interspeech.2023-767
- (14) Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I.: Attention is all you need, NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, p. 6000-6010, (2017)
- (15) Shakeel, M., Sudo, Y., Peng, Y., Watanabe, S.: Contextualized End-to-end Automatic Speech Recognition with Intermediate Biasing Loss, Proc. Interspeech 2024, p. 3909-3913, (2024), doi: 10.21437/Interspeech.2024-1257
- (16) Huber, C., Hussain, J., Stüker, S., Waibel, A.: Instant One-Shot Word-Learning for Context-Specific Neural Sequence-to-Sequence Speech Recognition, 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), (2021), doi: 10.1109/ASRU51503.2021.9687898
- (17) Sudo, Y., Shakeel, M., Fukumoto, Y., Peng, Y., Watanabe, S.: Contextualized Automatic Speech Recognition With Attention-Based Bias Phrase Boosted Beam Search, ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 10896-10900, (2024), doi: 10.1109/ICASSP48485.2024.10447782
- (18) Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: An ASR corpus based on public domain audio books, 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), p. 5206-5210, (2015), doi: 10.1109/ICASSP.2015.7178964
- (19) Maekawa, K.: Corpus of Spontaneous Japanese: Its design and evaluation, Proc. ISCA/IEEE Workshop on Spontaneous Speech Processing and Recognition, (2003), https://www.isca-archive.org/sspr_2003/maekawa03_sspr.pdf, (accessed 2024/9/23)
- (20) Kurematsu, A., Takeda, K., Sagisaka, Y., Katagiri, S., Kuwabara, H., Shikano, K.: ATR Japanese speech database as a tool of speech recognition and synthesis, Speech Communication, Vol. 9, Issue 4, p. 357-363, (1990), doi: 10.1016/0167-6393(90)90011-W
- (21) Sudo, Y., Hata, K., Nakadai, K.: Retraining-free Customized ASR for Enharmonic Words Based on a Named-Entity-Aware Model and Phoneme Similarity Estimation, Proc. INTERSPEECH 2023, p. 491-495, (2023), doi: 10.21437/Interspeech.2023-1320

■ Author ■



Yui SUDO



Shakeel MUHAMMAD



Naoaki SUMIDA