Phoneme-grapheme Dictionary-based Prompting for Robust Proper Noun Recognition in Japanese ASR

Ryuga Sugano*, Hiroaki Sato* Asahi Sakuma*, Tadashi Kumano*, Yoshihiko Kawai*, and Shinji Watanabe**
* NHK Science and Technology Laboratories, Tokyo, Japan
E-mail: {sugano.r-hm, satou.h-fk, sakuma.a-fc, kumano.t-eq, kawai.y-lk}@nhk.or.jp
**Carnegie Mellon University, Pittsburgh, PA, USA
E-mail: shinjiw@ieee.org

***Abstract*—Recent End-to-End (E2E) speech recognition models still struggle with low-frequency words such as proper nouns. Whereas Whisper models utilize prompt-based contextual biasing for improvement, Japanese presents a challenge because of its non-unique grapheme-phoneme correspondence, which limits the biasing effectiveness. Furthermore, rare kanji and special symbols often lead to out-of-vocabulary (OOV) issues, preventing model processing. To address these problems, we propose a novel speech recognition method that leverages a dictionary for proper nouns, which stores their graphemes and phonemes, along with assigned special tokens. Our method biases the model by providing phoneme sequences from the dictionary as prompts. When input speech matches these sequences, the model outputs special tokens, which are then replaced with the corresponding correct graphemes during post-processing by utilizing dictionary entries. Experiments on Japanese CSJ datasets demonstrate that our method drastically reduces the proper noun error rate from 24.5% to 0.6% compared with the conventional grapheme-based prompting method.**

I. Introduction

With the development of deep learning technology, End-to-End (E2E) architectures have become mainstream in the automatic speech recognition (ASR) field [1–6]. However, recognition accuracy often degrades owing to domain mismatch with training data, and this trend is particularly noticeable for low-frequency words, especially proper nouns. Proper nouns are important speech recognition targets in many applications, such as subtitle generation and dialogue systems. Therefore, recognition errors in proper nouns significantly impact user trust, necessitating robust countermeasures.

To address these difficulties, contextual biasing techniques have been actively researched in recent speech recognition studies [7–14]. These techniques aim to improve the recognition of specific words or phrases by providing the model with additional context. Models such as OpenAI's Whisper [15] and OWSM [16] utilize prior context learned by their Transformer decoders [17] during training, which allows for prompt input during inference. This enables prompt-based contextual biasing, thereby improving proper noun recognition accuracy. However, Japanese has a linguistic characteristic where the correspondence between phonemes and graphemes such as kanji is not one-to-one and is complex. Let us take the grapheme "梅雨" (rainy season) as an example. Whereas its individual kanji characters "梅" and "雨" are often read as "*ume*" (plum) and "*ame*" (rain) when appearing separately, they can have multiple pronunciation candidates such as "*tsuyu*" or "*baiu*" when combined. Additionally, the proper noun "*Saitou*" (a common family name in Japan) can have as many as 85 diverse grapheme candidates, including rare notations such as "斉藤", "斎藤", or "西藤". It is highly likely that many of these rare grapheme forms are not included in the training data. Because of these characteristics, if only a grapheme is provided as a prompt and its pronunciation is unknown to the model, the effectiveness of existing contextual biasing methods, which commonly use only graphemes as a biasing list, is limited. Furthermore, Japanese proper nouns often include rare kanji characters or special symbols that frequently become out-of-vocabulary (OOV) terms, fundamentally making it impossible for the model to input or output them. Although the models in [11–14] enhance biasing effectiveness by providing both graphemes and phonemes, their output is only grapheme-based, and thus they cannot handle OOV words.

To overcome these issues, in this paper, we propose a retraining-free method to improve the recognition accuracy of difficult-to-recognize proper nouns, including OOV words, by providing their phonemes as a prompt. Figure 1 illustrates the proposed method. As with other contextual biasing methods, our approach assumes that a list of proper nouns is available beforehand. First, the list of proper nouns and their phoneme sequences are registered in a dictionary. Next, all phoneme sequences from this dictionary are input into the model as prompts. The model uses an encoder-decoder network to first estimate the phoneme sequence from the speech input, and then estimate the grapheme sequence. When the speech matching the phoneme sequence of a prompted word is input, the model outputs a special token at the corresponding grapheme sequence output location. In post-processing, this special token can be automatically replaced with the correct grapheme using

979-8-3315-7206-8/25/$31.00 ©2025 IEEE

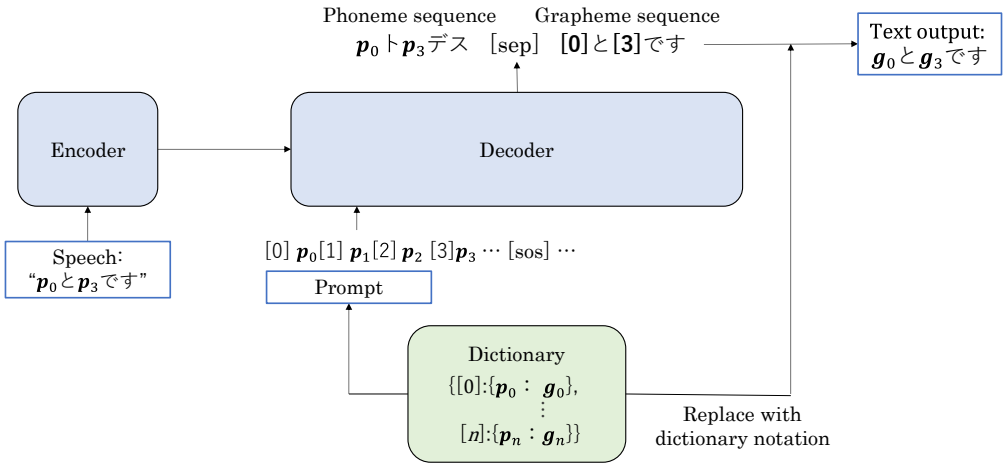

Fig. 1 Overview of our approach for robust proper noun ASR. Key components include phoneme-based prompts with special tokens and a dictionary for grapheme conversion.

the dictionary mapping.

We conducted experiments to evaluate the proper noun error rate in Japanese using the Corpus of Spontaneous Japanese (CSJ) and an in-house dataset. Compared with existing Whisper-based contextual biasing methods, the results show that our method significantly improves proper noun recognition accuracy.

The remainder of this paper is organized as follows. In Section II, we explain related work. In Section III, we describe the attention-based encoder-decoder models and prompt-based contextual biasing techniques. In Section IV, we detail the proposed method. In Section V, we present the experiments and their results. Finally, in Section VI we provide the conclusion and outline our future work.

II. Related work

Whisper [15] is an encoder-decoder model extensively trained on 5,000,000 hours (large-v3) or 680,000 hours (others) of audio-text paired data obtained from the web, achieving low word error rates across multiple languages. During training, Whisper learned to utilize the preceding context by inputting it before the output text in the Transformer decoder, thereby allowing text to be provided as a prompt during inference. According to OpenAI's official documentation [18], this improves accuracy by inputting lists of low-frequency words or prior context, and also allows the user to control the transcription style. Furthermore, [19] has demonstrated Whisper's capabilities for tasks such as audio-visual speech recognition, code-switched speech recognition, and speech translation. [20, 21] focused on improving the accuracy of low-frequency words through prompting and succeeded in further enhancing the output accuracy of such words by fine-tuning Whisper using low-frequency words in its prompt section.

We also focus on the utilization of prompts in speech recognition models. Our goal is to extend their capabilities to handle unknown words and OOV.

III. Attention-based encoder-decoder model

In this section, we introduce attention-based encoder-decoder ASR models, such as Whisper.

Attention-based encoder-decoder ASR models first input the audio feature sequence $\boldsymbol{X}$ to the encoder. The encoder then generates the hidden state $\boldsymbol{H}$:

$$\boldsymbol{H} = \mathrm{Encoder}(\boldsymbol{X}). \quad (1)$$

The decoder estimates the next token y_t using $\boldsymbol{H}$ and the previously estimated token sequence $\boldsymbol{y}_{<t} := (\boldsymbol{y}_1, \dots, \boldsymbol{y}_{t-1})$:

$$y_t = \mathrm{Decoder}(\boldsymbol{y}_{<t}, \boldsymbol{H}). \quad (2)$$

In models such as Whisper, a prompt $\boldsymbol{p} := (p_1, p_2, p_3, \dots)$ can be provided as an additional input to the decoder during inference. This allows contextual information to be incorporated into the estimation process, as shown below:

$$y_t = \mathrm{Decoder}(\boldsymbol{y}_{<t}, \boldsymbol{H}, \boldsymbol{p}). \quad (3)$$

When using $\boldsymbol{p}$ for contextual biasing, a text sequence of words separated by commas is typically input as the prompt. For instance, a prompt could be structured as:

$$\boldsymbol{p} = \text{"[sop]}, \boldsymbol{g}_0, \boldsymbol{g}_1, \boldsymbol{g}_2, \dots \text{[eop]"}, \quad (4)$$

where [sop] and [eop] represent the start-of-prompt token and the end-of-prompt token, respectively, and $\boldsymbol{g}_n$ is a word in the bias list.

The parameters of the encoder and decoder are trained to maximize the log-likelihood $\log(\boldsymbol{y}|\boldsymbol{X})$ for the training dataset.

IV. Proposed Method

The proposed method uses the phoneme sequence of nouns along with special tokens as a prompt, allowing the model to output special tokens instead of graphemes. It then utilizes dictionary notation for words that are difficult to recognize. In

the following subsections, we describe the rules for prompts and output sequences, the creation of training data, pretraining with text-only data, main training with speech-text pairs, and inference using the dictionary.

A. Rules for prompts and output sequences

In this subsection, we detail how our proposed model utilizes prompts and predicts sequences. The model utilizes the following prompt sequence $\boldsymbol{p}$:

$$\boldsymbol{p} = ([0], \boldsymbol{p}_0, [1], \boldsymbol{p}_1, [2], \boldsymbol{p}_2, \dots), \tag{5}$$

where $[n]$ (n = 0, 1, 2, …) is a special token, and $\boldsymbol{p}_n$ represents the phoneme sequence of a noun. As shown in Fig. 1, $\boldsymbol{p}$ is the input before the special token [sos], which serves as the initial input to the decoder. The model outputs a phoneme sequence $\boldsymbol{y}^{(\mathrm{p})}$ and a grapheme sequence $\boldsymbol{y}^{(\mathrm{g})}$ for speech input. Therefore, the model autoregressively predicts the sequence $\boldsymbol{z}$ as follows:

$$\boldsymbol{z} = \{y_1^{(\mathrm{p})}, \dots, y_L^{(\mathrm{p})}, [\mathrm{sep}], y_1^{(\mathrm{g})}, \dots, y_M^{(\mathrm{g})}, [\mathrm{eos}]\}, \tag{6}$$

where L and M are the numbers of tokens in $\boldsymbol{y}^{(\mathrm{p})}$ and $\boldsymbol{y}^{(\mathrm{g})}$, respectively, [sep] is a separation token, and [eos] is an end-of-sentence token. In particular, when $\boldsymbol{p}$ is the input and the prompt words $\boldsymbol{p}_0$ and $\boldsymbol{p}_2$ are uttered, the corresponding parts in $\boldsymbol{y}^{(\mathrm{g})}$ are output as special tokens. Therefore, $\boldsymbol{z}$ is:

$$\boldsymbol{z} = \begin{matrix} \{y_1^{(\mathrm{p})}, \dots, \boldsymbol{p}_0, \dots, \boldsymbol{p}_2, \dots, y_L^{(\mathrm{p})}, [\mathrm{sep}], \\ y_1^{(\mathrm{g})}, \dots, [0], \dots, [2], \dots, y_M^{(\mathrm{g})}, [\mathrm{eos}]\}. \end{matrix} \tag{7}$$

B. Creating training text data

Our primary focus is on proper nouns. To robustly output special tokens only at noun positions, we incorporate morphological knowledge by selecting only nouns as prompts and special token replacement locations. This prevents the incorrect insertion of special tokens in cases of accidental phoneme matches, such as within a word or when a word is combined with a particle.

To comprehensively create training data that can execute the rules mentioned in the previous subsection for all use cases, first, extract all nouns from each sentence of text by morphological analysis. Let i be the number of nouns obtained. Next, we randomly select the k ($0 \le k \le i$) nouns from the i obtained nouns and the total number j ($0 \le k \le j \le J$) of nouns to be included in the prompt, including dummy nouns, where J is the maximum value of j. Then, we obtain $j - k$ nouns not included in the text as dummy prompts from a cached nouns list, which consists of nouns randomly extracted from the training dataset and dynamically updated by random insertion and deletion. After that, the nouns including k nouns in the text and $j - k$ dummy nouns not in the text are shuffled. Subsequently, we create a prompt sequence by pairing their phoneme sequences with special tokens, as in Equation (5). At this time, special tokens are also chosen randomly. Finally, to create the ground truth data for $\boldsymbol{y}^{(\mathrm{g})}$, replace the k selected noun locations in the text with their paired special tokens.

C. Pretraining using text data

In our model, the decoder can be pretrained on a substantial amount of text-only data $\boldsymbol{z}$ by predicting the next token, excluding the cross-attention layers. Through this process, the decoder develops the capability to predict $\boldsymbol{y}^{(\mathrm{p})}$, convert $\boldsymbol{y}^{(\mathrm{p})}$ into $\boldsymbol{y}^{(\mathrm{g})}$, and effectively use prompts. This pretraining approach is more effective than relying solely on limited paired speech and text data. The prompt $\boldsymbol{p}$ is also utilized during this pretraining.

D. Training with paired speech and text data

We then train the entire model, including the cross-attention layer in the decoder, using paired speech $\boldsymbol{X}$ and text $\boldsymbol{z}$ data, enabling the pretrained decoder to process the speech input from the encoder. During training with paired data, the decoder uses prompts $\boldsymbol{p}$, and a gated cross-attention mechanism [22] is implemented to preserve the knowledge gained during pretraining. Gated cross-attention employs a tanh gating mechanism with a trainable scalar parameter θ, which was initialized to zero. To ensure stable training with speech features from the encoder and the pretrained decoder, θ is applied at the cross-attention as follows:

$$\boldsymbol{O} = \tanh(\theta) \times \mathrm{MHA}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) + \boldsymbol{Q}, \tag{8}$$

where $\boldsymbol{O}$ is the output of this gated cross-attention layer, MHA refers to multi-head attention, $\mathbf{Q}$ represents query derived from previous decoder layers, and $\boldsymbol{K}$ and $\boldsymbol{V}$ are the key and value obtained from the encoder output, respectively.

The estimation of $\boldsymbol{y}^{(\mathrm{p})}$ should reference speech features from the encoder, but the inference of $\boldsymbol{y}^{(\mathrm{g})}$ also has access to $\boldsymbol{y}^{(\mathrm{p})}$ through the self-attention layer in the decoder. Therefore, we establish two sets of parameters $\{\theta^{(\mathrm{p})}, \theta^{(\mathrm{g})}\}$, one for $\boldsymbol{y}^{(\mathrm{p})}$ and the other for $\boldsymbol{y}^{(\mathrm{g})}$, which are switched depending on the sequence being processed.

E. Inference method using dictionary

The proposed model, when given special tokens and noun phoneme sequences as prompt inputs, outputs those special tokens when the corresponding speech is uttered. At this time, to mechanically replace the special tokens with the correct graphemes, nouns are registered in the dictionary $\boldsymbol{D}$ by associating special tokens $[n]$, phoneme sequences $\boldsymbol{p}_n$, and grapheme sequences $\boldsymbol{g}_n$ as follows:

$$\boldsymbol{D} = \{[0]: \{\boldsymbol{p}_0 : \boldsymbol{g}_0\}, \dots, [n]: \{\boldsymbol{p}_n : \boldsymbol{g}_n\}\}. \tag{9}$$

From $\boldsymbol{D}$, the prompt $\boldsymbol{p}$ is created as in Equation (5) and input to the model as a prompt. Next, the model estimates $\boldsymbol{z}$ from the speech $\boldsymbol{X}$. Finally, if $\boldsymbol{y}^{(\mathrm{g})}$ contains special tokens, the corresponding grapheme sequence is retrieved from $\boldsymbol{D}$ in Equation (9) and automatically replaced with the correct notation to obtain the final recognition result.

Table 1. Overall character error rates (Overall), proper noun error rates (P.N.), and Non-proper noun character error rates (Non-P.N.) of proposed method and Whisper-style method. Results are presented for models without prompts (w/o) and with prompts (w/).

	CSJ eval3			In-house dataset		
	Overall	P.N.	Non-P.N.	Overall	P.N.	Non-P.N.
Whisper-style w/o	4.3	43.6	3.9	0.9	15.8	0.3
Whisper-style w/	4.1	24.5	4.0	0.5	7.6	0.3
Ours w/o	4.4	52.8	3.9	0.9	19.9	0.4
Ours w/	**4.0**	**0.6**	4.1	**0.4**	**0**	0.4

V. EXPERIMENTS

To verify the effectiveness of the proposed phoneme-sequence-based prompting, we compare our approach with the standard prompt-based biasing method that provides words as graphemes directly, as seen in Whisper's approach in Section III.

A. Model settings

We implemented the following two models for comparison using the ESPnet toolkit [23]:

Ours: The proposed model trained from scratch described in Section IV.

Whisper-style: A model trained from scratch following the same process as ours for a fair comparison, differing only in prompt design and output rules, as described in Section III.

Each encoder was composed of a 12-layer Conformer [24] and each decoder was a 12-layer Transformer. The number of attention heads, the attention layer dimensions, and FFN dimensions were set to 12, 768, and 3072, respectively. Both models were trained as hybrid CTC/Attention models [6], with the CTC loss weighted at 0.3. Specifically, for our proposed model, the CTC target was the phoneme sequence $\boldsymbol{y}^{(\mathrm{p})}$, whereas for the Whisper-style model, it was the grapheme sequence $\boldsymbol{y}^{(\mathrm{g})}$. The pretrained decoder employed a gated cross-attention mechanism with gate parameters. We used 80-dimensional Mel-scale filter-bank features as the speech input, with a window size of 512, a hop length of 128, and a sampling frequency of 16 kHz. SpecAugment [25] was then applied for data augmentation.

B. Data preparation

For pretraining, we utilized the Japanese portion of the CC-100 web crawl dataset [26], which, after cleaning, contained 390 million sentences. For training with paired speech and text, we used the Corpus of Spontaneous Japanese (CSJ) [27], comprising approximately 581 hours of audio and 400,000 lines of transcriptions. Training, validation, and evaluation datasets were prepared following the CSJ recipe in ESPnet.

To train our proposed model, the phonemic transcription for each utterance in the training data should precede its graphemic transcription. For pretraining with text-only data, we inferred the phoneme sequences from the text and applied the same approach to speech-text pairs. Specifically, in our Japanese experiments, we automatically generated phoneme sequences in katakana using the MeCab Japanese morphological analyzer [28] with the mecab-ipadic-NEologd named-entity-enhanced analyzing dictionary [29] (ver. 2020-08-20). The character error rate (CER) for inferring katakana transcription from graphemic transcription with this analyzer was 2.7%, as determined by our manual evaluation of 100 sentences.

Next, we created prompts for both pretraining and paired data training as described in Section IV-B. In this paper, the maximum number of nouns to include in the prompt was set to $J = 100$, and special tokens $[n]$ ($n = 0, 1, \ldots, 99$) were used. For comparison, the prompts used for training the Whisper-style method also set the maximum number of nouns to $J = 100$. As described in Section IV-B, nouns in the text and dummy prompt nouns were similarly selected randomly, and then a text with noun graphemes separated by commas, as in Equation (4), was used as the prompt.

C. Training settings

Both models were trained using the same process, except for the training data. For pretraining, the models were trained for 500,000 steps using the Adam optimizer at a learning rate of 0.0003 with 100,000 warm-up steps. Paired data training was conducted over 100 epochs, using the same optimizer at a learning rate of 0.0015 with 15,000 warm-up steps. The label smoothing weight was set to 0.1, and the vocabulary consisted of 5,000 subwords, excluding special tokens.

D. Evaluation settings

For evaluation, we used the average of the ten models with the highest validation accuracy during training. During inference, the beam size was set to 5, and CTC joint decoding was not used for either model. The evaluation dataset was eval3

Table 2. Examples of recognition results. The boldface characters "**和泉**" indicate the target proper noun in the prompt. Correctly recognized proper noun is shown in blue, whereas incorrectly recognized characters are shown in red.

Model	Text
Ground Truth	…**和泉**校舎の方でえー…
Whisper-style w/o	…**泉**校舎の方でえー…
Whisper-style w/	…**泉**校舎の方でえー…
Ours w/o	…**泉**校舎の方でえー…
Ours w/	…**和泉**校舎の方でえー…

from CSJ, which contains 94 proper nouns and includes 1,375 utterances from ten different lectures by various speakers.

E. Experiments on in-house dataset

In addition to CSJ, an in-house dataset was also utilized for further experiments. The primary motivation for using this in-house dataset was to confirm its applicability in ideal acoustic environments, such as news broadcasts (characterized by clear audio and precise articulation), where errors in speech recognition are predominantly attributed to proper nouns. Addressing these errors can thus prevent most recognition mistakes in this scenario. This dataset comprises approximately 4,500 hours of audio and 2 million lines of transcriptions, collected from Japanese broadcast audio. The preparation of this dataset, including phonemic transcription generation and prompt creation, followed the same procedures as those for the CSJ dataset.

With the in-house dataset, training was conducted for 30 epochs to optimize performance for this dataset. Other hyperparameters, such as learning rate, warm-up steps, optimizer (Adam), label smoothing weight, and vocabulary size, remained consistent with the CSJ training setup. In the evaluation of the in-house dataset, we utilized 5-minute news programs from 10 days, totaling 314 utterances, containing 69 proper nouns.

F. Results

As described in Section IV-E, inference results were obtained by registering the phonemes and graphemes of all proper nouns present in the evaluation data into the dictionary and constructing the prompt using all of the phonemes. Table 1 shows the evaluation results of the proposed method and the baseline Whisper-style method under conditions without prompts (w/o) and with prompts(w/).

On the CSJ eval3 dataset, the proper noun error rate was 43.6% without prompts for the Whisper-style method, whereas the proposed method (Ours w/o) showed a slightly higher value of 52.8%. However, when prompts were used, the proper noun error rate of the Whisper-style method improved to 24.5%, whereas the proposed method (Ours w/) markedly improved to 0.6%. Regarding the overall CER, whereas that of the Whisper-style method improved from 4.3% to 4.1%, that of our proposed method improved from 4.4% to 4.0%, indicating that the improvement in proper noun recognition accuracy also contributed to the overall recognition accuracy, although there were slight degradations in error rates for non-proper nouns.

A similar trend was observed in the in-house dataset. Under the no-prompt condition, the proper noun error rate of the Whisper-style method was 15.8% and that of our proposed method was 19.9%. With prompts, the proper noun error rate of the Whisper-style method improved to 7.6%, whereas our proposed method achieved a 0% proper noun error rate. This suggests that our method achieves nearly perfect performance for proper nouns in clean acoustic environments. Regarding the overall CER, that of the Whisper-style method improved from 0.9% to 0.5% and that of our method improved from 0.9% to 0.4%, demonstrating a large improvement for our method.

Table 2 presents examples of recognition results. In this example, the proper noun "和泉" appears, which has the pronunciation "*izumi*". Typically, "泉" alone is often read as "*izumi*". However, in rare cases for proper nouns, the two characters together "和泉" can also be read as "*izumi*". Without prompting, both models produced incorrect notations. Furthermore, the Whisper-style model, even when the grapheme "和泉" was provided as a prompt, failed to output the correct notation, suggesting that the pronunciation of "和泉" was unknown to the model. In contrast, our proposed method successfully output the correct notation by providing the pronunciation as a prompt, generating a special token, and utilizing a dictionary.

These results confirm that our proposed method, which combines a dictionary associating phonemes and graphemes with prompts, is more effective for Japanese proper noun recognition compared with existing Whisper-style methods that do not handle pronunciation. In particular, the significant reduction in proper noun error rates validates the effectiveness of the proposed method.

VI. Conclusion

In this paper, we proposed a method to improve the accuracy of proper nouns by using a dictionary that lists phonemes and graphemes, as well as phoneme prompting. It was found that by efficiently avoiding the estimation of diverse grapheme output candidates in Japanese and relying on the dictionary, errors in critical proper nouns were dramatically reduced. Future work will focus on countermeasures for proper nouns not anticipated to appear in advance.

REFERENCES

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in Proc. ICML, pp. 369–376, 2006.

[2] A. Graves, "Sequence transduction with recurrent neural networks," in Proc. ICML Workshop on Representation Learning, 2012.

[3] A. Graves, and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in Proc. ICML, pp. 1764–1772, 2014.

[4] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in Proc. NIPS, vol. 28, pp. 577–585, 2015.

[5] 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, pp. 4960–4964, 2016.

[6] 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.

[7] 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.

[8] M. Jain, G. Keren, J. Mahadeokar, and Y. Saraf, "Contextual rnn-t for open domain asr," in Proc. Interspeech, pp. 11–15, 2020.

[9] S. Zhou, Z. Li, Y. Hong, M. Zhang, Z. Wang, and B. Huai, "Copyne: Better contextual asr by copying named entities," in Proc. ACL, pp.2675−2685, 2024.

[10] Y. Sudo, Y. Fukumoto, M. Shakeel, Y. Peng, and S. Watanabe, "Contextualized automatic speech recognition with dynamic vocabulary," in Proc. SLT, pp. 78−85, 2024.

[11] A. Bruguier, R. Prabhavalkar, G. Pundak, and T. N. Sainath, "Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition," in Proc. ICASSP, pp. 6171– 6175, 2019.

[12] Z. Chen, M. Jain, Y. Wang, M. L. Seltzer, and C. Fuegen, "Joint grapheme and phoneme embeddings for contextual End-to-End ASR," in Proc. Interspeech, pp. 3490−3494, 2019.

[13] R. Pandey, R. Ren, Q. Luo, J. Liu, A. Rastrow, A. Gandhe, D. Filimonov, G. Strimel, A. Stolcke, and I. Bulyko, "PROCTER: Pronunciation-aware Contextual adapter for personalized speech recognition in neural transducers," in Proc. ICASSP, pp. 1–5, 2023.

[14] 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, pp. 10641−10645, 2024.

[15] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision", in *International conference on machine learning*. PMLR, pp. 28492–28518, 2023.

[16] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang, J-W. Jung, and S. Watanabe, "OWSM v3. 1: Better and Faster Open Whisper-Style Speech Models based on E-Branchformer," in Proc. Interspeech, pp. 352−356, 2024.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. NIPS, vol. 30, 2017.

[18] https://platform.openai.com/docs/guides/speech-to-text/prompting/

[19] P. Peng, B. Yan, S. Watanabe, and D. Harwath, "Prompting the hidden talent of web-scale speech models for zero-shot task generalization," in Proc. Interspeech, pp. 396–400, 2023.

[20] X. Wei, and S. McGregor "Prompt Tuning for Speech Recognition on Unknown Spoken Name Entities," in Proc. Interspeech, pp. 762−766, 2024.

[21] Y. Jogi, V. Aggarwal, S. S. Nair, Y. Verma, and A. Kubba, "Improving Rare-Word Recognition of Whisper in Zero-Shot Settings," in Proc. SLT, pp. 216−223, 2024.

[22] J-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mecsch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan "Flamingo: A visual language model for few-shot learning," Advances in Neural Information Processing Systems, 35:23716–23736, 2022.

[23] S.Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in Proc. Interspeech, pp. 2207–2211, 2018.

[24] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for speech recognition," in Proc. Interspeech, pp. 5036–5040, 2020.

[25] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in Proc. Interspeech, pp. 2613–2617, 2019.

[26] G. Wenzek, M.-A. Lachaux, A. Conneau, V. Chaudhary, F. Guzmán, A. Joulin, and E. Grave, "CCNet: Extracting high quality monolingual datasets from Web crawl data," in Proc. LREC, pp. 4003−4012, 2020.

[27] K. Maekawa, H. Koiso, S. Furui, and H. Isahara, "Spontaneous speech corpus of Japanese," in Proc. LREC, 2000.

[28] T. Kudo, MeCab: Yet another part-of-speech and morphological analyzer. [Online]. Available: https://taku910.github.io/mecab/

[29] T. Sato, Neologism dictionary based on the language resources on the Web for MeCab [Online]. Available: https://github.com/neologd/mecab-ipadic-neologd