



Improving Synthetic Data Training for Contextual Biasing Models with a Keyword-Aware Cost Function

Chin Yuen Kwok¹, Jia Qi Yip¹, Eng Siong Chng¹

¹College of Computing and Data Science, Nanyang Technological University, Singapore

kwok0062@e.ntu.edu.sg

Abstract

Rare word recognition can be improved by adapting ASR models to synthetic data that includes these words. Further improvements can be achieved through contextual biasing, which trains and adds a biasing module into the model architecture to prioritize rare words. While training the module on synthetic rare word data is more effective than using non-rare-word data, it can lead to overfitting due to artifacts in the synthetic audio. To address this, we enhance the TCPGen-based contextual biasing approach and propose a keyword-aware loss function that additionally focuses on biased words when training biasing modules. This loss includes a masked cross-entropy term for biased word prediction and a binary classification term for detecting biased word positions. These two terms complementarily support the decoding of biased words during inference. By adapting Whisper to 10 hours of synthetic data, our method reduced the word error rate on the NSC Part 2 test set from 29.71% to 11.81%.

Index Terms: speech recognition, human-computer interaction, computational paralinguistics

1. Introduction

Accurate recognition of rare words in automatic speech recognition (ASR) is crucial, as these words often carry key semantic information. However, ASR models frequently misidentify them due to their limited presence in training data. Contextual biasing addresses this challenge by incorporating additional contextual information, such as contact lists, domain-specific terms, or user-defined vocabularies, to improve recognition. This technique is typically implemented through specialized modules [1–7] integrated into pre-trained ASR models, which adjust model activations based on a predefined list of biasing words, thereby enhancing recognition accuracy.

While training these modules on non-rare-word data improve rare word recognition, they remain less effective than direct training on data containing these rare words, which is often unavailable. This data limitation creates a bottleneck for effective contextual biasing. To address this, synthetic training data generated via text-to-speech (TTS) systems can be used as a substitute [8]. By generating audio samples from sentences containing the target rare words, TTS enables the creation of training datasets tailored to specific contexts. This approach alleviates the need for extensive real-world recordings and allows the ASR model to automatically learn rare words.

However, synthetic audio data introduce artifacts that differ from natural speech, such as unnatural prosody or acoustic distortions, which can lead to overfitting when training the biasing module. Modules trained on such data may perform poorly when exposed to natural audio, limiting the effective-

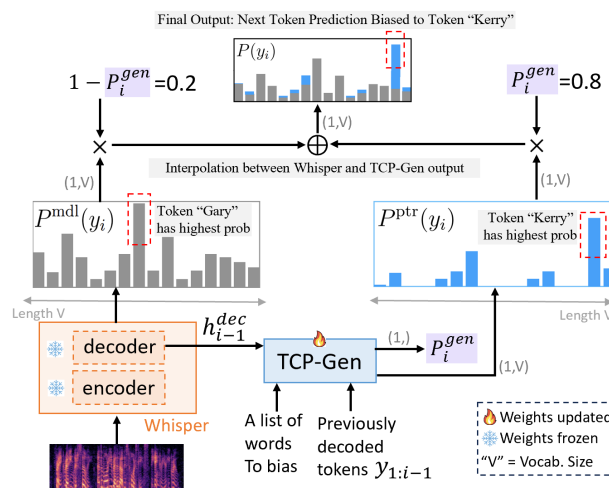


Figure 1: Example of applying TCPGen to Whisper for contextual biasing to predict the i -th token. Tensor shapes are shown in parenthesis. Bottom-left: Whisper takes in an audio to output next token probabilities $P^{mdl}(y_i)$. Bottom-center: TCPGen takes in Whisper decoder’s last hidden state h_{i-1}^{dec} , a list of words to bias and the previously decoded tokens $y_{1:i-1}$ to output biased probabilities $P^{ptr}(y_i)$ and an interpolation weight P_i^{gen} . Top-center: Interpolation between $P^{mdl}(y_i)$ and $P^{ptr}(y_i)$ is performed using the weight P_i^{gen} to output the final probabilities $P(y_i)$.

ness of contextual biasing in real-world scenarios. To mitigate overfitting, prior work has introduced training constraints to the standard ASR loss used for training the biasing modules, addressing two key challenges in contextual biasing:

- When to perform biasing:** Xu et al. [1] introduced an Entity Detector to identify the presence of bias words, enabling selective application of biasing. Similarly, Huang et al. [9] refined biasing by modifying layer outputs only at time frames corresponding to bias phrases.
- What to bias:** Kulshreshtha et al. [10] restricted the CTC encoder to cross-attend only to the correct bias phrase from a given list. Tang et al. [11] extended this by constraining the prediction network.

Although these approaches are effective, they rely on assumptions that may not hold for speech foundation models like Whisper [12]. Specifically, they assume that encoders can produce frame-level predictions [13] or that audio and text modalities are modeled separately through distinct encoder and predictor networks. However, Whisper does not generate frame-level

(A) Biasing list:	Ground-truth:	My	name	Is	Kerry	(B)
1. Kerry	Whisper output:	My	name	Is	Gary	(C)
2. Tom						
$P^{ptr}(y_i)$ decides which token to bias:		Tom	Tom	Tom	Kerry	(D)
P_i^{gen} decides when to bias:		0.012	0.009	0.001	0.81	(E)
TCP-Gen's output	Combined output:	My	name	Is	Kerry	(F)
	Our training target for $P^{ptr}(y_i)$				Kerry	(G)
	Our training target for P_i^{gen}	0	0	0	1	(H)

Figure 2: Overview of our modified TCPGen training objective. As an example, (B) given an audio with ground truth transcript “My name is Kerry”, (C) Whisper wrongly outputs “Gary” instead. To perform contextual biasing, (A) a biasing list is provided to TCPGen. (D) Then it predicts the token to bias at each transcript position and (E) if biasing should be performed. Originally, (F) the combined output of Qwen Audio and TCPGen is trained on the vanilla ASR cross-entropy loss using the ground-truth in (B) as training targets. To better guide the TCPGen to model the pronunciations of rare words, we propose to replace the ASR loss with two loss that explicitly trains TCPGen to output the desirable (G) $P^{ptr}(y_i)$ and (H) P_i^{gen} targets.

predictions, and its decoder processes both encoded acoustic features and text-based inputs via cross- and self-attention, challenging the applicability of these methods to such architectures.

To address this, we integrate the pointer-network-based biasing module TCPGen [6] into Whisper for decoder-side adaptation [14, 15]. We propose to modify the ASR objective function, which is often used to train the biasing modules, to additionally focus on the biasing words to reduce overfitting. Specifically, given that TCPGen is trained to output a bias probability $P^{ptr}(y_i)$ and an interpolation weight P_i^{gen} as shown in Fig. 1, we introduce two novel loss terms: 1) to determine what to bias, we introduce a masked cross-entropy loss for $P^{ptr}(y_i)$ to predict the posterior probabilities of the rare words as shown in Fig. 2G. 2) to determine when to perform biasing, we introduce a binary classification for P_i^{gen} to indicate the presence of biasing words at each token position, as shown in Fig. 2H. These two objectives complementarily support the decoding of rare words during inference.

Our contributions are fourfold. We show that: 1) applying the vanilla TCPGen module to Whisper can cause overfitting, 2) by changing the objective function of TCPGen to our two novel loss terms, a significant improvement of up to 16.6% relative WER reduction is observed. 3) We show that our two novel loss terms, which are specifically designed for contextual biasing, can be used without the standard ASR loss to effectively train the biasing modules, unlike previous work that still includes the ASR loss when training the modules. 4) Our objective function improves the interpretability of the TCPGen output P_i^{gen} and allows explicit analysis of the false acceptance rate for contextual biasing.

2. Method

2.1. Tree-constrained pointer generator (TCPGen)

TCPGen is a neural network component that is added to ASR models for contextual biasing, where TCPGen is trained end-to-end and the weights of the pre-trained ASR model is frozen. As shown in Fig. 1, at output step i , TCPGen takes the last hidden state h_{i-1}^{dec} of Whisper’s decoder, a list of words to bias and the previously decoded tokens $y_{1:i-1}$ to output a distribution $P^{ptr}(y_i)$ over all tokens to determine which token to bias. Fur-

thermore, TCPGen predicts a generation probability P_i^{gen} that determines when to perform biasing. The final output $P(y_i)$ is an interpolation between $P^{ptr}(y_i)$ and the original ASR model distribution $P^{mdl}(y_i)$ weighted by P_i^{gen} :

$$P(y_i) = P^{mdl}(y_i)(1 - P_i^{gen}) + P^{ptr}(y_i)P_i^{gen} \quad (1)$$

An example of contextual biasing is shown in Fig. 2, where given an audio with ground truth transcript “My name is Kerry”, Whisper incorrectly outputs “Gary” instead of “Kerry”. To perform contextual biasing, a biasing list consisting of “Kerry” and “Tom” is provided to TCPGen. Then TCPGen outputs $P^{ptr}(y_i)$ at each decoding step i to find the most probable word to bias as shown in Fig. 2D, and uses P_i^{gen} to decide when to perform biasing as shown in Fig. 2E. Finally, the combined output of whisper and TCP-Gen has the output “Gary” biased to “Kerry” as shown in Fig. 2F.

2.2. Training Objectives

Originally, TCPGen is trained to optimize the vanilla ASR cross entropy loss ℓ_{asr} :

$$\ell_{asr} = - \sum_{i=1}^U \sum_{c=1}^V \mathbb{1}(y_i = c) \log(P(y_i = c)) \quad (2)$$

where U and V are the target sequence length and vocabulary size and $\mathbb{1}(y_i = c)$ is an indicator function that outputs 1 if the i -th token in y is token c or outputs 0 otherwise.

2.2.1. Binary Classification Loss

To more effectively train TCPGen and to reduce overfitting, we propose replacing the ASR loss of TCPGen with two losses ℓ_{ptr} and ℓ_{gen} to explicitly guide TCPGen to learn the desired $P^{ptr}(y_i)$ and P_i^{gen} outputs, respectively. Specifically, as shown in Fig. 2H, P_i^{gen} should be explicitly optimized to classify if the i -th decoded token needs biasing:

$$\ell_{gen} = - \sum_{i=1}^U \alpha \cdot \mathbb{1}(i \in K) \log(P_i^{gen}) + (1 - \alpha)(1 - \mathbb{1}(i \in K)) \log(1 - P_i^{gen}) \quad (3)$$

where K is the set of the position indexes of the biased tokens, e.g. the index of “Kerry” is 4 in Fig. 2B, and P_i^{gen} is trained on the negative log-likelihood loss ℓ_{gen} to perform binary classification.

As rare and non-rare word distributions are imbalanced, we include a weight $\alpha > 0.5$ to further encourage the model to classify the minority class.

2.2.2. Masked Cross Entropy Loss

Next, as shown in Fig. 2G, $P^{ptr}(y_i)$ should be optimized to predict the rare word probabilities only for tokens that need biasing:

$$\ell_{ptr} = - \sum_{i=1}^U \sum_{c=1}^V \mathbb{1}(i \in K) \mathbb{1}(y_i = c) \log(P^{ptr}(y_i = c)) \quad (4)$$

This is different from optimizing ℓ_{asr} as $P^{ptr}(y_i)$ will redundantly learn to predict the rare word probabilities at the positions where there are no rare words under the formulation of ℓ_{asr} in Eq. 2, which leads to overfitting. Given the two losses, TCPGen is trained to optimize $\ell_{gen} + \ell_{ptr}$.

2.3. False Acceptance Rate (FAR)

To analyze the effectiveness of our proposed objective functions, the false acceptance rate (FAR) and the true acceptance rate (TAR) are calculated for P_i^{gen} . False acceptance occurs when the i -th decoded token needs biasing, but P_i^{gen} predicts otherwise. True acceptance occurs when the i -th decoded token needs biasing and P_i^{gen} predicts correctly.

We emphasize that this analysis is only possible with our ℓ_{gen} formulation. This is because in the original ℓ_{asr} formulation, P_i^{gen} is optimized as an interpolation weight in the range of $(0, 1)$ and a higher value means that biasing is needed for the i -th decoded token, but the decision boundary is not obvious. By explicitly optimizing P_i^{gen} to perform binary classification using our ℓ_{gen} , the FAR and TAR can be calculated by setting the decision boundary to 0.5.

2.4. Masked Probabilities

Given that M_i is the set of tokens to bias at the i -th decoding step. TCPGen originally assigns zero probabilities to $P^{ptr}(y_i = c) : c \notin M_i$ such that according to Eq. 1:

$$\forall c \notin M_i : P(y_i = c) = P^{mdl}(y_i = c)(1 - P_i^{gen}) \quad (5)$$

where $P(y_i = c) : c \notin M_i$ is just a scaled version of $P^{mdl}(y_i = c)$. However, we empirically find that the scaling effect may be undesirable when we optimize P_i^{gen} to have values close to 1 using ℓ_{gen} as it completely suppresses $P(y_i = c) : c \notin M_i$ and increases the chance of overbiasing. Therefore, we remove the $(1 - P_i^{gen})$ term in Eq. 5 and set $\forall c \notin M_i : P(y_i = c) = P^{mdl}(y_i = c)$.

3. Experiment Setup

3.1. Dataset and metrics

Experiments were conducted on the National Speech Corpus Part 2 (NSC-Part-2) dataset [16], a subset of a large-scale Singapore English corpus that includes road names and addresses. They are recordings of people asking for directions and consist of 13K unique utterances. This dataset is chosen because synthetic audio has been publicly released for the dataset [17], and about 15% of the words in the synthetic audio are testset rarewords, which is significantly more than Librispeech¹.

We use the 20 hours of real audio train set, 10 hours for synthetic audio train set (VITS-SPKSET1 [17]) and 2 hours of real audio test set for our experiments. Both the synthetic and real audio train sets used contain the same number of sentences. Synthetic audios are generated from the same text as the real train set, using the VITS [19] text-to-speech model from coqui-ai², which is trained on the VCTK corpus [20]. We find that although the synthetic audio has a different English accent compared to the NSC-Part-2 dataset, it is sufficient to adapt Whisper to the NSC-Part-2 text domain using the audio.

¹Although synthetic audio has also been released for Librispeech [18], only about 2% of the words are testset rarewords. As a result, synthetic data training for Librispeech is less effective.

²<https://github.com/coqui-ai/TTS>

Similar to Sun et al. [21], we treat words that do not exist in the 10K common word list³ as rare words. In total, there are 3856 and 1106 rare words in the train and test sets respectively. In this setup all the testset rarewords are considered OOV words as we never train TCPGen with any real audio data containing the words. During inference, a list of biasing words for each utterance was obtained by collecting rarewords that appear in the utterance and adding N distractors, which are randomly sampled from the set of all possible rarewords.

In addition, we also experiment on DSTC2 [22], a human-machine task-oriented dialogue dataset where each of 10 hours of user-side speech input was used for training and evaluation. The biasing list for inference was organised by the error-based biasing list obtained from the training set following [21]. The full biasing list contains 274 words.

To evaluate our method, two additional metrics were used to evaluate system performance besides word error rate (WER): the biased word error rate (B-WER) and the unbiased word error rate (U-WER) [23]. B-WER is calculated as the number of error words from the biasing list divided by the total number of words from the biasing list in the test set, with insertion errors counted if the inserted word is from the bias list. U-WER is similar to B-WER but focuses solely on words not belonging to the biasing list.

3.2. Implementation details

We implement our methods based on the popular SpeechBrain [24] toolkit. We adapt whisper-small and Qwen Audio with vanilla fine-tuning (FT) or AGEM [25] for 2 epochs and set the train batch size to 6. AGEM is a regularization method that constrains the gradients, and we find it to be effective in reducing overfitting on synthetic training data. We simply follow the setup from Kwok et al. [14] to apply AGEM. We set the learning rate to 0.005 and 0.0001 for Whisper and Qwen Audio respectively. We use the AdamW optimizer with a variant⁴ of the ReduceLROnPlateau learning rate (LR) scheduler. The encoder weights are frozen to prevent overfitting [26, 27]. We sweep through the hyper-parameters to optimize WER.

4. Results and Discussions

The contextual biasing results of whisper-small are shown in Table 1. The first block of the table shows that whisper-small originally performs worse on NSC-Part-2, and the performance improves if the model is vanilla fine-tuned (FT) on the synthetic data. AGEM further improves the performance, as it regularizes the training on synthetic data.

Based on the AGEM model, we further apply contextual biasing baseline methods as shown in the second block of the table. First, the results of contextual adapters and TCPGen show no improvements, and we observe that TCPGen’s $P^{ptr}(y_i)$ is always close to 0. We hypothesize that this occurs because the AGEM model is already overfitted to the NSC-Part-2 synthetic train set and has been fully optimized with respect to the ASR loss. Consequently, further training of the contextual biasing modules, which are essentially error correction modules, can barely optimize the ASR loss as the AGEM model has almost no errors on the train set already. Instead, further training only

³<https://github.com/first20hours/google-10000-english/blob/master/google-10000-english.txt>

⁴https://speechbrain.readthedocs.io/en/latest/_modules/speechbrain/nnet/schedulers.html#NewBobScheduler

Table 1: OOV contextual biasing for whisper-small. Either the real or synthetic (Syn) version of the NSC-Part-2 train/development set is used for training/hyper-parameter tuning. N is the number of distractors added to the biasing words list. The first block of results have repeated results as N is irrelevant. TCPGen-2L means TCPGen trained with our two novel loss terms.

Method	NSC-Part-2		N=10			N=50			N=100		
	Real	Syn	WER	BWER	UWER	WER	BWER	UWER	WER	BWER	UWER
unadapted			29.71	54.03	17.21	29.71	54.03	17.21	29.71	54.03	17.21
Kwok et al. [17]		✓	16.5	-	-	16.5	-	-	16.5	-	-
FT	✓		9.40	16.95	5.52	9.40	16.95	5.52	9.40	16.95	5.52
FT		✓	15.00	28.02	8.31	15.00	28.02	8.31	15.00	28.02	8.31
AGEM [25]		✓	14.16	28.80	6.64	14.16	28.80	6.64	14.16	28.80	6.64
<i>reproduced baselines</i>											
AGEM [25]		✓									
+ Contextual Adapter [28]		✓	15.90	31.11	8.08	15.72	30.71	8.01	15.76	30.68	8.09
+ TCPGen [21]		✓	14.16	28.80	6.64	14.16	28.80	6.64	14.16	28.80	6.64
+ Alter. pron. [29]		✓	14.20	28.29	6.96	14.54	28.29	7.47	15.01	28.45	8.11
+ Alter. pron. [30]	✓		13.08	25.39	6.76	13.75	26.28	7.32	14.56	27.32	8.00
<i>our method</i>											
AGEM		✓									
+ TCPGen-2L		✓	11.81	20.74	7.22	12.81	22.65	7.75	12.86	22.41	7.95

Table 2: Contextual biasing results ($N = 100$) for Qwen-Audio on the DSTC2 dataset. Δ BWER means relative % improvement in BWER. TCPGen-2L means TCPGen trained with our two novel loss terms.

Method	WER	BWER	UWER	Δ BWER
unadapted	26.5	107	25.2	-
Sun et al. [21]	13.9			
FT	7.39	87.9	6.14	0.0
+ TCPGen	7.39	87.9	6.14	0.0
+ TCPGen-2L	7.01	72.9	5.98	17.1

overfits the modules to the artificial patterns of the synthetic audio.

Next, when we apply our two novel losses onto TCPGen (TCPGen-2L), significant improvement is observed over the vanilla TCPGen as we hypothesize that our method can effectively optimize the contextual-biasing-specific objectives without relying on the ASR loss and this reduces overfitting.

4.1. More overfitting scenarios

To validate the effectiveness of our method, we further perform experiments using Qwen Audio on the DSTC2 dataset. Specifically, we first fine-tune (FT) Qwen Audio on DSTC2 real train set, and further train TCPGen on top of the fine-tuned model using the same train set.

The results in Table 2 show that originally the unadapted Qwen-Audio has a large BWER as the biasing list is error-based – the words are only included in the list if they are originally wrongly recognized by Qwen-Audio. Then, vanilla TCPGen show no improvements, and we observe that TCPGen’s $P^{ptr}(y_i)$ is always close to 0. This is expected, as we hypothesize that the FT model is already fine-tuned to the DSTC2 real train set and the ASR loss is already optimized. As a result, further training the contextual biasing modules, which are essentially error correction modules, on the same train set can barely optimize the ASR loss as the FT model has almost no errors on the train set already. This makes training TCPGen inefficient

Table 3: Effect of changing α on the WER of the NSC-Part-2 development set, and on the true acceptance rate (TAR) and false acceptance rate (FAR) of P_i^{gen} . The number of unbiased word errors caused by biased words (#U-WE-B) and B-WER is also reported to study their relationship with FAR and TAR respectively.

Metrics	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$
WER	30.3	28.3	27.2	25.3
FAR	0.08	1.43	4.28	9.59
#U-WE-B	61	62	74	73
TAR	16.9	41.9	61.6	67.9
B-WER	60.0	48.7	43.4	37.9

and causes overfitting. However, training TCPGen using our two novel losses effectively avoids the issue caused by the ASR loss and makes the training more effective.

4.2. Ablation Study

We emphasize that with our novel ℓ_{gen} objective function, it is possible to analyze the FAR and TAR of P_i^{gen} as discussed in Section 2.3. Table 3 shows that as we increase α , TAR and FAR gradually increase as a larger α encourages P_i^{gen} to classify the i -th decoded token as a token that needs biasing. The results also show that a higher TAR causes a lower B-WER as more words that need biasing can be correctly biased, and a higher FAR causes more unbiased word error as more of the unbiased words are wrongly biased.

5. Conclusion

This paper proposes two novel training objectives for TCPGen to replace the traditional ASR loss. These objectives work together to reduce overfitting and enhance performance, achieving up to a 16.6% relative WER reduction for Whisper and Qwen Audio.

6. Acknowledgements

This research is supported by the National Research Foundation, Singapore, under the AI Singapore Programme (AISG Award No.: AISG2-100E-2022-102). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (<https://www.nsc.sg>).

7. References

- [1] T. Xu, Z. Yang, K. Huang, P. Guo, A. Zhang, B. Li, C. Chen, C. Li, and L. Xie, "Adaptive contextual biasing for transducer based streaming speech recognition," *arXiv preprint arXiv:2306.00804*, 2023.
- [2] P. Harding, S. Tong, and S. Wiesler, "Selective Biasing with Trie-based Contextual Adapters for Personalised Speech Recognition using Neural Transducers," in *Proc. INTERSPEECH 2023*, 2023, pp. 256–260.
- [3] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual adapters for personalized speech recognition in neural transducers," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8537–8541.
- [4] B. Naowarat, P. Harding, P. D'Alterio, S. Tong, and B. Awwad Shiekh Hasan, "Effective Training of Attention-based Contextual Biasing Adapters with Synthetic Audio for Personalised ASR," in *Proc. INTERSPEECH 2023*, 2023, pp. 1264–1268.
- [5] A. Shamsian, A. Navon, N. Glazer, G. Hetz, and J. Keshet, "Keyword-guided adaptation of automatic speech recognition," in *Proc. INTERSPEECH 2024*, 2024.
- [6] G. Sun, X. Zheng, C. Zhang, and P. C. Woodland, "Can Contextual Biasing Remain Effective with Whisper and GPT-2?" in *Proc. INTERSPEECH 2023*, 2023, pp. 1289–1293.
- [7] H. Futami, E. Tsunoo, Y. Kashiwagi, H. Ogawa, S. Arora, and S. Watanabe, "Phoneme-aware encoding for prefix-tree-based contextual asr," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 10641–10645.
- [8] X. Zheng, Y. Liu, D. Gunceler, and D. Willett, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5674–5678, 2020.
- [9] K. Huang, A. Zhang, B. Zhang, T. Xu, X. Song, and L. Xie, "Spike-triggered contextual biasing for end-to-end mandarin speech recognition," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.
- [10] D. Kulshreshtha, S. Dingliwal, B. Houston, and S. Bodapati, "Multilingual contextual adapters to improve custom word recognition in low-resource languages," *arXiv preprint arXiv:2307.00759*, 2023.
- [11] J. Tang, K. Kim, S. Shon, F. Wu, and P. Sridhar, "Improving asr contextual biasing with guided attention," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 096–12 100.
- [12] 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*, 2022. [Online]. Available: <https://api.semanticscholar.org/CorpusID:252923993>
- [13] C. Y. Kwok, J. Q. Yip, and E. S. Chng, "Improved alignment for score combination of rnn-t and etc decoder for online decoding," in *International Conference on Text, Speech, and Dialogue*. Springer, 2024, pp. 70–80.
- [14] —, "Continual learning optimizations for auto-regressive decoder of multilingual asr systems," *arXiv preprint arXiv:2407.03645*, 2024.
- [15] —, "Continual learning with embedding layer surgery and task-wise beam search using whisper," in *2024 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2024, pp. 140–146.
- [16] J. X. Koh, A. Mislan, K. Khoo, B. Ang, W. Ang, C. Ng, and Y. Tan, "Building the singapore english national speech corpus," *Malay*, vol. 20, no. 25.0, pp. 19–3, 2019.
- [17] C. Y. Kwok, H. Y. Li, and E. S. Chng, "Asr model adaptation for rare words using synthetic data generated by multiple text-to-speech systems," in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2023, pp. 1771–1778.
- [18] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu, "Libritts: A corpus derived from librispeech for text-to-speech," *arXiv preprint arXiv:1904.02882*, 2019.
- [19] J. Kim, J. Kong, and J. Son, "Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech," in *International Conference on Machine Learning*. PMLR, 2021, pp. 5530–5540.
- [20] J. Yamagishi, C. Veaux, and K. MacDonald, "Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92)," 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:213060286>
- [21] G. Sun, X. Zheng, C. Zhang, and P. C. Woodland, "Can contextual biasing remain effective with whisper and gpt-2?" 2023. [Online]. Available: <https://arxiv.org/abs/2306.01942>
- [22] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, K. Georgila, M. Stone, H. Hastie, and A. Nenkova, Eds. Philadelphia, PA, U.S.A.: Association for Computational Linguistics, Jun. 2014, pp. 263–272. [Online]. Available: <https://aclanthology.org/W14-4337/>
- [23] D. Le, M. Jain, G. Keren, S. Kim, Y. Shi, J. Mahadeokar, J. Chan, Y. Shangguan, C. Fuegen, O. Kalinli *et al.*, "Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion," *arXiv preprint arXiv:2104.02194*, 2021.
- [24] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.
- [25] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in neural information processing systems*, vol. 30, 2017.
- [26] C. Y. Kwok, J. Q. Yip, and E. S. Chng, "Low resource language adaptation using two-stage regularization for multilingual asr," in *2024 International Conference on Asian Language Processing (IALP)*. IEEE, 2024, pp. 332–337.
- [27] C. Y. Kwok, S. Li, J. Q. Yip, C. Chu, T. Kawahara, and E. S. Chng, "Extending whisper for emotion prediction using word-level pseudo labels," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.
- [28] X. Gong, A. Lv, Z. Wang, and Y. Qian, "Contextual biasing speech recognition in speech-enhanced large language model," *Proc. Interspeech. ISCA*, pp. 257–261, 2024.
- [29] D. Le, T. Koehler, C. Fuegen, and M. L. Seltzer, "G2g: Tts-driven pronunciation learning for graphemic hybrid asr," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6869–6873.
- [30] D. Le, G. Keren, J. Chan, J. Mahadeokar, C. Fuegen, and M. L. Seltzer, "Deep shallow fusion for rnn-t personalization," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 251–257.