Zero-shot Context Biasing with Trie-based Decoding using Synthetic Multi-Pronunciation

Changsong Liu*†, Yizhou Peng*†, and Eng Siong Chng†
* Alibaba-NTU Global e-Sustainability CorpLab, Nanyang Technological University, Singapore
† College of Computing and Data Science, Nanyang Technological University, Singapore
E-mail: changsong.liu@ntu.edu.sg

***Abstract*—Contextual automatic speech recognition (ASR) systems allow for recognizing out-of-vocabulary (OOV) words, such as named entities or rare words. However, it remains challenging due to limited training data and ambiguous or inconsistent pronunciations. In this paper, we propose a synthesis-driven multi-pronunciation contextual biasing method that performs zero-shot contextual ASR on a pretrained Whisper model. Specifically, we leverage text-to-speech (TTS) systems to synthesize diverse speech samples containing each target rare word, and then use the pretrained Whisper model to extract multiple predicted pronunciation variants. These variant token sequences are compiled into a prefix-trie, which assigns rewards to beam hypotheses in a shallow-fusion manner during beam-search decoding. Subsequently, any recognized variant is mapped back to the original rare word in the final transcription. The evaluation results on the LibriSpeech dataset show that our method reduces biased-word error rate (B-WER) by 43% on test-clean and 44% on test-other while maintaining unbiased-WER (U-WER) essentially unchanged.**

Index Terms: ASR, TTS, Contextual biasing, Zero-shot.

I. Introduction

Contextual automatic speech recognition (ASR), also known as "hotword" or "named-entity" customization, enables ASR systems to accurately recognize user-specified terms, such as names, places, or technical jargon, that are rare or even out-of-vocabulary (OOV) in training data [1]. Modern end-to-end (E2E) ASR models achieve an impressive overall word error rate (WER) compared to traditional hybrid systems, thanks to advanced architectures such as RNNs [2], Transformers [3], and Conformers [4]. However, due to data scarcity and confusion in pronunciation for rare words, contextual ASR remains challenging [5].

Traditional contextual biasing techniques augment decoding via shallow-fusion with an external language model [6] or on-the-fly WFST rescoring [7]. While effective, these methods require careful tuning of fusion weights and can degrade WER/U-WER performance when the bias list becomes large, due to an increase in false positive distractor terms with similar spellings or pronunciations. To be more straightforward, the E2E approaches embed context phrases directly into the ASR model, for example, Contextual Listen, Attend and Spell (CLAS) uses an attention-based contextual module to bias decoding [8], and trie-based deep biasing integrates dynamic n-gram constraints into transducer decoding [9]. Recently, large language model (LLM)-based ASR has emerged, leveraging prompt engineering to incorporate hotwords into decoding. For example, a CTC-assisted LLM-based contextual ASR model uses coarse CTC hypotheses to select effective hotwords from a large hotword list, which are then fed into an LLM as a prompt [10]. These yield substantial improvements in biased-WER (B-WER) by enhancing the model's capability for contextual understanding during the inference phase.

Despite these advances, one critical factor has often been overlooked: the variability in pronunciation of hotwords. In practice, many hotwords, especially specialized terms such as technical jargon, foreign names, or branded entities, are unfamiliar to users, leading to inconsistent or incorrect pronunciations. As a result, the same hotword may be spoken in multiple ways across different speakers or even by the same speaker over time. This mismatch between the intended hotword and its spoken realization can significantly degrade the effectiveness of contextual biasing, as models may fail to associate these variant forms with the intended target.

In this paper, we introduce a novel approach to enhancing the recognition accuracy of rare words in a pretrained ASR system without requiring any adaptation. Our method leverages synthetic speech to generate diverse pronunciation variants for each hotword, which are then transcribed by the same ASR model. Technically, we synthesize speech using a variety of TTS systems and voices for each templated hotword sample and apply Whisper Large-v3 model to transcribe them. We integrate a multi-pronunciation prefix-trie-based hotword module into the same Whisper Large-v3 model in a shallow-fusion manner, where the prefix-trie is constructed using extracted multiple pronunciation variants of each hotword.

Our experiments on the LibriSpeech test sets demonstrate that the proposed method reduces WER/B-WER of the baseline Whisper Large-v3 model without affecting the recognition performance of common words. To the best of our knowledge, this is the first work to leverage synthesized multi-pronunciation variants for contextual ASR without requiring model adaptation.

II. Methodology

In this section, we describe our end-to-end pipeline for augmenting Whisper's contextual biasing with TTS-generated multi-pronunciation variants. The pipeline comprises three sequential stages: Hotword Speech Synthesis, Hotword Token Extraction, and Prefix-Trie Implementation in Whisper.

979-8-3315-7206-8/25/$31.00

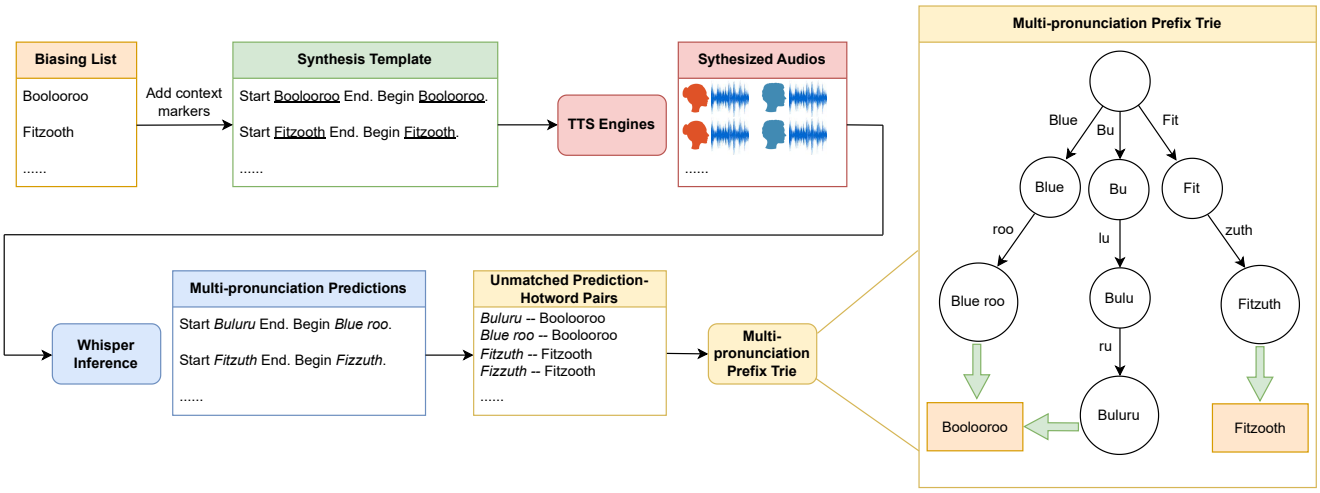

Fig. 1. Workflow of TTS-powered multi-pronunciation prefix-trie construction for Whisper model. Given a list of hotwords, each is inserted into a designated template and synthesized using multiple TTS engines to mimic three distinct speaker voices. The synthetic speech is then transcribed using the vanilla Whisper model, from which diverse pronunciation variants are extracted. Each unique transcription is included as an effective hotword in the prefix-trie. During decoding, any recognized variant shown in the final results is mapped back to the original hotword.

A. Multi-Pronunciation Hotword Speech Synthesis via TTS

We begin by synthesizing multiple pronunciation variants for each target hotword using three complementary TTS engines, including CosyVoice [11], F5-TTS [12], and GPT-SoVITS [13].

- **F5-TTS** is a non-autoregressive (NAR) TTS model that builds on a Diffusion Transformer (DiT) architecture and incorporates ConvNeXt V2 blocks to effectively address text-speech alignment in an in-context learning setting. It is trained on 95K hours of **English-Chinese** bilingual dataset and supports good-quality synthesis in both languages.
- **CosyVoice** is a scalable zero-shot multilingual TTS system that employs a text-to-token-to-speech architecture. It leverages supervised semantic tokens extracted from a quantized multilingual ASR encoder, uses a large language model (LLM) to predict token sequences from text and speaker embeddings, and reconstructs speech via a conditional flow matching model followed by a HiFi-GAN vocoder. It is trained on around 170K hours of multilingual dataset and supports five languages and dialects, including **Chinese**, **English**, **Cantonese**, **Japanese**, and **Korean**.
- **GPT-SoVITS** integrates a speech-to-unit encoder, a speaker encoder, and a GPT-style text-to-unit decoder within the So-VITS framework. It enables prompt-based zero-shot voice cloning by generating discrete unit sequences from text and synthesizing them into a waveform via a neural vocoder. It supports the same five languages as CosyVoice.

All three systems support prompt-based speech synthesis by conditioning on a reference audio, its corresponding text, and a new target text, allowing for zero-shot speech generation in the reference speaker's voice and style.

To ensure that the hotword appears in a fixed, easily locatable context, we embed it in two minimal utterances: "Start <Hotword> End" and "Begin <Hotword>". The context can be customized, but we adopt such simple and rigid templates to facilitate reliable extraction from Whisper decoding results and to ensure stable and consistent synthesis from TTS models. Longer or more complex contexts tend to introduce unwanted variability in TTS outputs, which can compromise the reliability of anchor-based hotword localization, as the surrounding context serves as a temporal marker for extracting hotword speech variations. Each engine renders both utterances with three distinct voices and accents from [14], including British female, British male, and American male. This process yields a total of 3 engines × 2 utterances × 3 voices = 18 synthesized utterances per hotword, capturing a rich and comprehensive variety of pronunciation patterns.

B. Multi-Pronunciation Hotword Token Extraction

Next, each synthesized waveform is fed into the vanilla Whisper Large-v3 model to obtain the recognition results, where we require not only the final sentences but also the sequences of byte-pair encoding (BPE) tokens. This is to align with the requirement of prefix-trie as mentioned in Section II-C, which performs hotword biasing through BPE-level rewarding scores in each auto-regressive decoding step of the Whisper decoder that generates the probability distribution of the entire BPE dictionary.

From the token sequences, we locate the tokens corresponding to the known context markers "Start," "End," and "Begin," and we isolate the subsequences that occur between "Start" and "End" as well as those immediately following "Begin". Any extracted sequence whose token IDs exactly match the original hotword is discarded, so that only unmatched pronunciation-

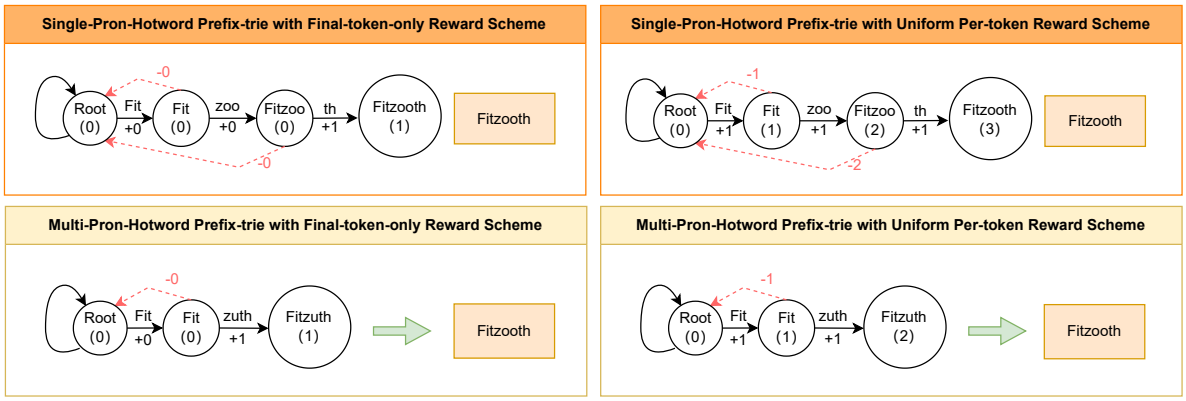

Fig. 2. An illustration of Single-Pron-Hotword and Multi-Pron-Hotword prefix-trie using **Final**-token-only and **Uniform** per-token reward schemes. In Single-Pron-Hotword prefix-trie, each hotword corresponds to only one path, representing the correct spelling. In contrast, in a Multi-Pron-Hotword prefix-trie, each hotword may correspond to multiple paths with different spellings. When using the **Final** reward, only the last state of a valid path will give a reward of 1; whereas when using **Uniform** reward, each active state will credit a reward of 1 when matched. Numbers in the parentheses stand for the rewards gained up to the current state. Red dashed lines are examples of failing to match in the middle state of a prefix-trie path.

prediction variant pairs remain. This extends the search space to include more predictions that the Whisper model can make for each hotword, allowing us to improve hit accuracy. Additionally, we refine the variant set to eliminate faulty options by Syllable Filtering (SF). Specifically, we use the syllapy [15] package to estimate the number of phonemes of the predicted variants. Only those with syllable count matches that of the original hotword and at least three are retained. This strategy significantly improves performance by removing inappropriate pronunciation variants introduced by TTS generation instability or the Whisper model's hallucinations through the automated pipeline.

C. Prefix-Trie Implementation in Whisper Decoder

We implement a simple prefix-trie module into the Whisper beam-search decoding to enable contextual biasing in Whisper without requiring model adaptation or other learnable modules, such as Tree-constrained Pointer Generator (TCPGen) [16]. In our prefix-trie architecture, each arc, corresponding to a token, is annotated with a Uniform per-token reward of $+1$, so that every token in every pronunciation variant contributes equally to decoding. Since each token in a hotword contributes a fixed reward during decoding, longer hotwords (i.e., those with more tokens) inherently receive more cumulative bias, giving them a greater advantage when matched correctly. During beam-search decoding, we maintain, for each beam hypothesis, a corresponding "active" trie state that records how far along any hotword prefix that hypothesis has traversed. Given the audio X, at each time step t, when considering an output token y_t, we first compute the **W**hisper score $S_{\mathbf{W}}$ as in Eq. 1.

$$S_{\mathbf{W}}(y_t) = -\log P_{\mathbf{W}}(y_t \mid y_{t-1}, ..., y_0, X) \tag{1}$$

Here we define the cumulative score of each beam **H**ypothesis as in Eq. 2.

$$S_{\mathbf{H}}(y_t) = S_{\mathbf{H}}(y_{t-1}) + S_{\mathbf{W}}(y_t) \tag{2}$$

We then attempt to advance the hypothesis's current trie state with y_t: if y_t continues a valid hotword prefix conditioned on y_{t-1}, the trie returns a reward $\rho_{y_{t-1}\to y_t} = 1$; otherwise we remove the accumulated reward of partial unsuccessful match from the hypothesis's cumulative score. We define the reward as $-\sum_{i=1}^{n} \rho_{y_{t-i}\to y_{t-i+1}}$ where n is the depth from root to current state of prefix-trie as shown in Figure 2. We form the final score S augmented with contextual rewards, as in Eq. 3, which is then used in place of $S_{\mathbf{W}}$ when ranking and pruning beam candidates.

$$S(y_t) = S_{\mathbf{W}}(y_t) + \rho_{y_{t-1}\to y_t} \tag{3}$$

In this way, any hypothesis that follows one of our multi-pronunciation hotword paths receives a consistent positive boost, biasing the search toward recognizing those variants. After the beam-search decoding, we select the highest-scoring beam, and then traverse its recorded trie states to map to the correct hotword spelling. Since we only add these Uniform per-token rewards to the beam scores without altering the underlying acoustic or language model logits, our trie-based biasing improves recognition of the target hotwords while preserving Whisper's overall decoding behavior and general ASR accuracy.

Through this three-stage methodology, as shown in Figure 1, we incorporate a prefix-trie-based multiple pronunciation hotword mechanism with Whisper for contextual biasing, leading to significant reductions in B-WER without degrading overall ASR accuracy.

III. Experiments

In this section, we introduce the dataset and evaluation metrics used, as well as the experimental setup.

A. Dataset and Evaluation Metrics

1) Data and Hotword Selection: We follow previous research [9] on their settings regarding contextual biasing

and benchmark our method on the widely used LibriSpeech dataset [17], including the *test-clean* and *test-other* evaluation sets. Specifically, we adopt the artificial biasing word list setup[1] where the 5,000 most frequent words from the LibriSpeech training data are labeled as *common*, while all remaining words are treated as *rare*, including over 209.2K words. For each test utterance, the biasing list is composed of two components: rare words that appear in the reference transcription, and distractor words randomly selected from the *rare* words list. We use a fixed $N = 1000$ distractors to align with prior works for a fair comparison. The statistics of both test sets are as demonstrated in Table I. Rare words take up 10% of the word count in both test sets.

From the rare word vocabulary, we select a subset of rare words that exhibit a non-zero initial WER when decoded with the official Whisper Large-v3 model without contextual biasing. We then perform speech synthesis on this subset. In contrast, rare words that are correctly transcribed are explicitly excluded from further processing and do not go through the synthesis pipeline. Both sections of rare word vocabulary above are taken into account when calculating B-WER.

TABLE I
STATISTICS OF THE LIBRISPEECH DATASET. APPEARING RARE WORDS REFERS TO THE RARE WORD COUNTS THAT ARE PRESENT IN THE REFERENCE TRANSCRIPTS.

Metric	test-clean	test-other
Total words	52,576	52,343
Common words	46,815	46,993
Appearing rare words	5,761	5,350
Utterances	2,620	2,930
Avg. rare words per utterance	2.20	1.83
Rare word rate	10.96%	10.22%

2) Evaluation Metrics: We utilize commonly used evaluation metrics for contextual ASR tasks: overall WER to measure general recognition quality, biased-WER (B-WER) to specifically evaluate the accuracy on targeted hotwords, and unbiased-WER (U-WER) to measure the accuracy on common words. Comparing contextual ASR with standard ASR systems, the combination of these three WERs shows the effectiveness (true positives) and distractions (false positives) of contextual biasing strategies embedded in the ASR systems. Thus, an optimal contextual biasing ASR should achieve improved B-WER while maintaining U-WER unchanged. For transcription normalization and metrics calculation, we utilize the scripts provided by MetaAI [9].

B. Experimental Setup

All of our experiments are based on the Whisper Large-v3 model, and we use beam-search decoding with a beam size of 10. We use the vanilla Whisper Large-v3 model as a baseline, comparing against the Whisper Large-v3 with prefix-trie hotword module under various configurations as illustrated in Figure 2:

- **Single-Pron-Hotword**: Each hotword is bound with only one way of spelling, which is the original correct one, and corresponds to only one path in the prefix-trie.
- **Multi-Pron-Hotword**: As proposed in Section II, each hotword is bound with several selected pronunciations generated through a TTS-Whisper pipeline, and may correspond to several paths in the prefix-trie.

Furthermore, we explore two reward strategies in the prefix-trie: a **Final**-token-only reward scheme that gives a reward of 1 only when the decoding beam reaches the terminal node of the hotword in the trie, and a **Uniform** per-token reward scheme where each hit token of the hotword returns a reward of 1. The **Final** scheme only rewards those beams where Whisper predicts the entire correct sequence of hotwords on its own. In contrast, the **Uniform** scheme continuously rewards a beam each time a partial token of a hotword is predicted; however, if at any point the next expected token does not match, the awarded reward is immediately removed.

IV. RESULTS

Table II summarizes our evaluation results on LibriSpeech test-clean and test-other subsets. Firstly, we test the **Final** reward scheme to evaluate Whisper's ability to recognize rare words, as we only reward hypotheses that contain correctly recognized hotwords and prioritize these decoding paths. Then, SF is applied to improve the reliability of the multi-pronunciation hotword list. Finally, we utilize the **Uniform** reward scheme to guide Whisper's decoding process toward hotword recognition by providing rewards for partial matches, thereby introducing a bias in favor of contextual biasing.

In Table II, with the reward scheme set to **Final**, comparing **S1** and the Baseline system, we observe a significant B-WER reduction (19% for *test-clean* and 22% for *test-other*), along with slightly better U-WER results. This suggests that a prefix-trie hotword implementation already provides substantial improvements in hotword recognition for Whisper. When we implement the multi-pronunciation hotword strategy without using SF (as in **S2** and **S3**), the U-WER performance degrades dramatically. At the same time, B-WER does not improve at all, showing that our pipeline for generating the multi-pronunciation hotwords brings us a large amount of noisy variations, leading to an incorrect rewarding paradigm. However, when SF is applied to remove those faulty variations, comparing **S5** with S1, we can conclude that the strategy using multi-pronunciation hotwords with their original forms introduces further B-WER improvement by 10% and 7% for *test-clean* and *test-other*, respectively, while the U-WER remains essentially unchanged. Since the rare words only occupy around 10% of the entire test sets, **S5** only achieves 2-3% improvement on the overall WER.

Using **Uniform** reward scheme, comparing **S6** with S1, U-WER and B-WER have identical improvement for both test sets, with around 3% performance improvement on U-WER, and 23% B-WER reduction. Also, we observe the same trends when comparing **S7** against S5. This indicates that guiding

[1]The rare and common word lists are from https://github.com/facebookresearch/fbai-speech/tree/main/is21_deep_bias.

TABLE II

PERFORMANCE OF THE PROPOSED MULTI-PRONUNCIATION HOTWORD METHOD ON LIBRISPEECH DATASET USING BIASING LIST WITH $N = 1000$. **SF** INDICATES WHETHER WE APPLY SYLLABLE FILTERING TO THE MULTI-PRONUNCIATION VARIANTS. **REWARD** INDICATES THE REWARD PATTERNS (FINAL-TOKEN-ONLY VS. UNIFORM PER-TOKEN) USED. FINAL-TOKEN-ONLY INDICATES ONLY THE FINAL PREFIX-TRIE STATE OF A VALID PATH HAS A REWARD OF 1 WHILE THE PREVIOUS STATES HAVE A REWARD OF 0. UNIFORM PER-TOKEN INDICATES EVERY PREFIX-TRIE STATE OF A VALID PATH HAS A REWARD OF 1. WER REFERS TO THE OVERALL WORD ERROR RATE. B-WER/U-WER REFERS TO BIASED/UNBIASED-WER, INDICATING THE MODEL'S ABILITY TO RECOGNIZE RARE/COMMON WORDS.

System	Condition	SF	Reward	test-clean			test-other		
				WER	U-WER	B-WER	WER	U-WER	B-WER
Baseline	Vanilla Whisper Large-v3	–	-	2.81	1.94	9.86	4.84	3.39	17.55
S1	Single-Pron-Hotword	–	Final	2.57	1.91	7.95	4.27	3.21	13.61
S2	Multi-Pron-Hotword	✗	Final	19.97	21.19	9.98	26.45	27.25	19.42
S3	+ Single-Pron	✗	Final	19.79	21.17	8.54	26.15	27.21	16.84
S4	Multi-Pron-Hotword	✓	Final	2.70	1.96	8.73	4.52	3.25	15.68
S5	**+ Single-Pron**	✓	Final	2.50	1.92	7.19	4.18	3.22	12.60
S6	Single-Pron-Hotword	-	Uniform	2.33	**1.87**	6.11	3.87	**3.11**	10.52
S7	**+ Multi-Pron**	✓	Uniform	**2.31**	1.90	**5.66**	**3.83**	3.14	**9.90**

TABLE III

COMPARISON WITH PRIOR CONTEXTUAL ASR MODELS ON LIBRISPEECH DATASET USING BIASING LIST [9] WITH LIST SIZE $N = 1000$.

Model	test-clean			test-other		
	WER	U-WER	B-WER	WER	U-WER	B-WER
CPPNet [18]	3.81	2.90	11.40	8.75	6.90	25.30
Deep Biasing+BPB [19]	3.47	3.00	7.70	7.34	6.40	15.80
Whisper+TCPGen+GPT-2 [20]	3.40	–	8.20	6.30	–	16.30
TCPGen+GNN enc. [21]	3.10	–	6.70	7.90	–	17.80
GA-CTC [22]	2.40	2.00	6.30	6.20	5.20	15.20
TCPGen+p+phn-aware Q [23]	2.20	–	4.60	6.00	–	12.30
DB-NNLM [9]	2.14	1.60	6.70	6.35	5.10	17.20
CTC+LLM [10]	1.33	1.00	4.16	2.99	2.31	9.33
S7	2.31	1.90	5.66	3.83	3.14	9.90

Whisper's decoding process by rewarding step by step to favor rare words achieves even better performance. Our final system, **S7**, integrates a prefix-trie contextual biasing module built on multi-pronunciation and the original form of rare words to the Baseline system, achieving 18% and 21% overall WER reduction, as well as 43% and 44% B-WER performance improvement, for *test-clean* and *test-other*, respectively.

We also compare our method against many prior contextual ASR approaches evaluated on the LibriSpeech dataset with biasing list from [9] in Table III. Despite utilizing a frozen Whisper backbone without fine-tuning or LLM assistance, our proposed method significantly outperforms several fine-tuned models in terms of B-WER under the same configuration of biasing list and distractor size ($N = 1000$).

V. CONCLUSION

We introduced a novel approach to enhancing zero-shot contextual ASR performance by leveraging synthetic speech for Whisper. Our method involves synthesizing diverse hotword speech via TTS engines and integrating these multi-pronunciation variants extracted from the Whisper model into a prefix-trie for shallow-fusion in beam-search decoding. Without any model fine-tuning or reliance on large language models, our system consistently improves hotword recognition performance on the LibriSpeech test sets with a hotword biasing list, achieving competitive B-WER while maintaining and even improving general ASR quality. Our results surpass many fine-tuned contextual ASR systems, making the approach practical, low-cost, and easy to deploy, as it does not require any fine-tuning of the underlying ASR model. Future work will expand the TTS augmentation pipeline by incorporating additional voice styles and more diverse TTS engines, including multilingual and expressive speech synthesis. This would enable coverage of broader pronunciation variability and better simulate real-world user scenarios across regions and accents.

ACKNOWLEDGMENT

This research is supported by the RIE2025 Industry Alignment Fund - Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL).

References

[1] J. Li *et al.*, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[2] L. R. Medsker, L. Jain, *et al.*, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[3] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[4] A. Gulati, J. Qin, C.-C. Chiu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[5] X. Shi, Y. Yang, Z. Li, Y. Chen, Z. Gao, and S. Zhang, "Seaco-paraformer: A non-autoregressive asr system with flexible and effective hotword customization ability," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, pp. 10 346–10 350.

[6] D. Zhao, T. N. Sainath, D. Rybach, *et al.*, "Shallow-fusion end-to-end contextual biasing.," in *Interspeech*, 2019, pp. 1418–1422.

[7] K. B. Hall, E. Cho, C. Allauzen, *et al.*, "Composition-based on-the-fly rescoring for salient n-gram biasing.," in *Interspeech*, 2015, pp. 1418–1422.

[8] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *2018 IEEE spoken language technology workshop (SLT)*, IEEE, 2018, pp. 418–425.

[9] D. Le, M. Jain, G. Keren, *et al.*, "Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion," *arXiv preprint arXiv:2104.02194*, 2021.

[10] G. Yang, Z. Ma, Z. Gao, S. Zhang, and X. Chen, "Ctc-assisted llm-based contextual asr," in *2024 IEEE Spoken Language Technology Workshop (SLT)*, IEEE, 2024, pp. 126–131.

[11] Z. Du, Q. Chen, S. Zhang, *et al.*, "Cosyvoice: A scalable multilingual zero-shot text-to-speech synthesizer based on supervised semantic tokens," *arXiv preprint arXiv:2407.05407*, 2024.

[12] Y. Chen, Z. Niu, Z. Ma, *et al.*, "F5-tts: A fairytaler that fakes fluent and faithful speech with flow matching," *arXiv preprint arXiv:2410.06885*, 2024.

[13] *GPT-SoVITS*, https://github.com/RVC-Boss/GPT-SoVITS, 2024.

[14] *Voiceover Samples*, https://www.voiceover-samples.com/languages/, 2025.

[15] M. Holtzscher, *Syllapy*, https://pypi.org/project/syllapy/, 2025.

[16] G. Sun, C. Zhang, and P. C. Woodland, "Tree-constrained pointer generator for end-to-end contextual speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2021, pp. 780–787.

[17] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, IEEE, 2015, pp. 5206–5210.

[18] K. Huang, A. Zhang, Z. Yang, *et al.*, "Contextualized end-to-end speech recognition with contextual phrase prediction network," *arXiv preprint arXiv:2305.12493*, 2023.

[19] Y. Sudo, M. Shakeel, Y. Fukumoto, Y. Peng, and S. Watanabe, "Contextualized automatic speech recognition with attention-based bias phrase boosted beam search," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, pp. 10 896–10 900.

[20] G. Sun, X. Zheng, C. Zhang, and P. C. Woodland, "Can contextual biasing remain effective with whisper and gpt-2?" *arXiv preprint arXiv:2306.01942*, 2023.

[21] G. Sun, C. Zhang, and P. C. Woodland, "Tree-constrained pointer generator with graph neural network encodings for contextual speech recognition," *arXiv preprint arXiv:2207.00857*, 2022.

[22] 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.

[23] 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. 10 641–10 645.