Context Biasing for Pronunciation-Orthography Mismatch in Automatic Speech Recognition

Christian Huber[1], *Alexander Waibel*[2]

[1] Interactive Systems Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2] Interactive Systems Lab, Carnegie Mellon University, Pittsburgh PA, USA
christian.huber@kit.edu, alexander.waibel@cmu.edu

Abstract

Neural sequence-to-sequence systems deliver state-of-the-art performance for automatic speech recognition. When using appropriate modeling units, e.g., byte-pair encoding, these systems are in principle open vocabulary systems. In practice, however, they often fail to recognize words not seen during training, e.g., named entities, acronyms, or domain-specific special words. To address this problem, many context biasing methods have been proposed; however, these methods may still struggle when they are unable to relate audio and corresponding text, e.g., in case of a pronunciation-orthography mismatch. We propose a method where corrections of substitution errors can be used to improve the recognition accuracy of such challenging words. Users can add corrections on the fly during inference. We show that with this method we get a relative improvement in biased word error rate between 22% and 34% compared to a text-based replacement method, while maintaining the overall performance.

Index Terms: context biasing, pronunciation-orthography mismatch, automatic speech recognition

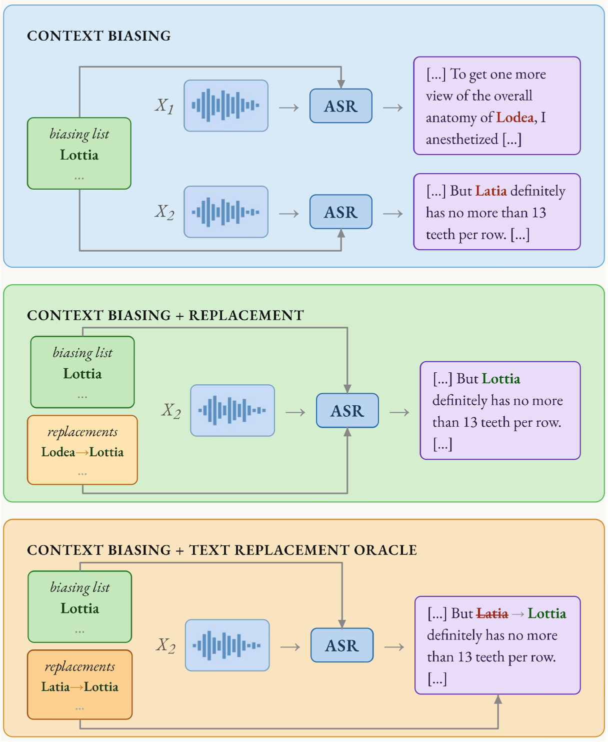
1. Introduction

Up until a few years ago automatic speech recognition (ASR) systems were implemented as Bayes classifiers in order to search for the word sequence $\hat{Y}$, among all possible word sequences Y, with the highest posterior probability given a sequence of feature vectors X which is the result of preprocessing the acoustic signal to be recognized:

$$\begin{aligned}\hat{Y} &= \operatorname*{argmax}_{Y} P(Y|X) \\ &= \operatorname*{argmax}_{Y} P(X|Y)P(Y) \qquad (1)\end{aligned}$$

In the context of ASR $P(X|Y)$ is called the acoustic model, $P(Y)$ the language model. The space of allowed word sequences to search among was usually defined by a list of words, the vocabulary, of which permissible word sequences could be composed. Words that were not in the vocabulary could not be recognized. In turn this means that by adding words to the vocabulary and appropriate probabilities to the language model, previously unknown words could be added to the ASR system.

In contrast, for neural end-to-end trained ASR systems [1–3], this is no longer possible. In principle, end-to-end systems are open-vocabulary systems, when using appropriate modeling units, such as byte-pair encoded (BPE) [4] characters. However, in practice, words not seen during training are often not reliably recognized. This is especially true for named entities. The reasons are that, a) the end-to-end network implicitly learns language model knowledge when being trained on transcribed speech data, and b) especially named entities often have a grapheme-to-phoneme relation that deviates from the general pronunciation rules of the language, as learned implicitly by the networks of the end-to-end system.



[...] To get one more view of the overall anatomy of Lodea, I anesthetized [...]

[...] But Latia definitely has no more than 13 teeth per row. [...]

[...] But Lottia definitely has no more than 13 teeth per row. [...]

[...] But ~~Latia~~ → Lottia definitely has no more than 13 teeth per row. [...]

Figure 1: *Approaches: Top: Inference of the baseline context biasing ASR model for two utterances containing the same named entity "Lottia" in the reference transcript; the model failed to recognize the named entity. Middle: Approach context biasing + replacement; the context biasing list contains the wrongly recognized word "Lodea" from another utterance mapped to "Lottia" (for details how the model uses this see Section 3). Bottom: Approach context biasing + text replacement oracle for comparison; the replacement "Latia" mapped to "Lottia" from the same utterance is used.*

The recognition of words not seen during the training of an automatic speech recognition system, e.g., named entities, acronyms, or domain-specific special words, has been studied in classical ASR systems [5–9]. Generally, the language model $P(Y)$ in equation 1 was modified. More recently, other works have combined statistical or neural language models with end-to-end ASR models using shallow fusion [10–14]. On the other hand, many recent works have used attention-based deep biasing [15–27]. Some of them use only textual context information and some also include pronunciation information. The problem with the former is that the model might not be able to relate audio and corresponding text of previously unseen words. In that case the model is not able to recognize the previously unseen word and users have no effective way to correct this. The problem with the latter is that such information is difficult to annotate by users.

In order to address this problem we 1) propose a method (see Section 3) that can take advantage of corrections of substitution errors provided during inference, 2) demonstrate that this method achieves a relative improvement between 22% and 34% in biased word error rate (BWER; see Section 4.4) compared to a text-based replacement method, while maintaining the overall word error rate, and 3) show that this method uses one correction of a substitution error more efficiently than the text-based replacement method.

2. Background

An auto-regressive end-to-end ASR model directly estimates the probability distribution

$$P(Y_t|Y_0,\dots,Y_{t-1};X) \tag{2}$$

of the next token Y_t given the already decoded sequence $Y_0,\dots,Y_{t-1}$ and the audio input X. The model is then used to find the word sequence $\hat{Y}$ with the highest probability

$$\hat{Y} = \underset{Y}{\operatorname{argmax}}\, P(Y|X) = \underset{Y}{\operatorname{argmax}} \prod_{t=1}^{T} P(Y_t|Y_0,\dots,Y_{t-1};X),$$

where Y_0 is the start of sequence token.

We work with a transformer-based encoder-decoder ASR model. First, $Y_0,\dots,Y_{t-1}$ is embedded:

$$E = Emb(Y_0,\dots,Y_{t-1}) \in \mathbb{R}^{t\times d}, \tag{3}$$

with $d \in \mathbb{N}$. Then, the decoder output is computed:

$$O = Dec(H_X, E) \in \mathbb{R}^{t\times d},$$

where $H_X = Enc_{Audio}(X)$ is the encoded audio input. Finally, the output and softmax layers are applied:

$$\alpha = Linear(O) \in \mathbb{R}^{t\times n_{vocab}},$$

$$p = Softmax(\alpha) \in \mathbb{R}^{t\times n_{vocab}}, \tag{4}$$

where n_{vocab} is the vocabulary size.

Based on that model, we trained a context biasing model using the training scheme from [18] together with the architecture from [27]. The method works as follows: Equation 2 is replaced by

$$P(Y_t|Y_0,\dots,Y_{t-1};X;Z),$$

where Z is some context provided to the model. In our case,

$$Z = (Z_1,\dots,Z_L), L \in \mathbb{N},$$

is a list denoted context biasing list and each Z_l, $l \in \{1,\dots,L\}$, is a word or short phrase the model is biased towards.

2.1. Context encoding

The model incorporates the context biasing list by first tokenizing and embedding each item. Then, an encoder is applied independently for each item, followed by a mean pooling over the sequence dimension. This results in one vector per list entry:

$$Z^s = (Avg(Enc_{Context}(Emb(Tokenize(Z_l)))))_{l=1}^{L}.$$

2.2. Context decoding

Then Z^s is used to extend the vocabulary of the decoder. This is done by extending the output layer, which maps the output of the final decoder layer to the vocabulary, and extending the embedding layer.

In particular,

$$\alpha_{Context} = \frac{Linear_2(O)\cdot Linear_3(Z^s)^T}{\sqrt{d}} \in \mathbb{R}^{t\times L}. \tag{5}$$

is calculated and α in equation 4 is replaced by

$$Concat(\alpha, \alpha_{Context}) \in \mathbb{R}^{t\times(n_{vocab}+L)}.$$

Furthermore, $Y_0,\dots,Y_{t-1}$ is replaced by $Y'_0,\dots,Y'_{t-1}$, where $Y'_0,\dots,Y'_{t-1}$ is calculated by replacing all subsequences of $Y_0,\dots,Y_{t-1}$ which correspond to a context biasing list entry Z_l with a dynamic token v_l. Finally, E in equation 3 is replaced by

$$E' = Emb(Y'_0,\dots,Y'_{t-1}),$$

where dynamic tokens v_l are embedded by $Linear_4(Z^s_l)$ and the rest of the tokens is embedded using Emb.

2.3. Training

During model training, in each step, the context bias list Z is sampled from the labels of the corresponding batch. Specifically, we used a batch size of 16 and sampled on average three context biasing list entries per utterance of the batch. Then the context biasing list is filled up to a length of 200 with distractors sampled from other batches.

3. Method

The context biasing model (see Section 2) learns during training to relate words in the context biasing list and the corresponding audio. This works well and during inference the model can generalize to new words not seen during training [18,27]. However, if this fails and the model is not able to relate audio and corresponding text, e.g. when there is a mismatch between pronunciation and orthography compared to what was learned during training, the model is not be able to recognize such words.

An example of the Yodas test set (see Section 4.1) in which this is the case can be seen in Figure 1. Both the audio features X_1 and X_2 contain the named entity ”Lottia” in the corresponding reference transcript. However, the context biasing model is not able to recognize that (see Figure 1, top). In the first utterance the word ”Lodea” is recognized, in the second utterance the word ”Latia” is recognized. Therefore, a text-based replacement of e.g. Lodea→Lottia would not work for the first utterance. The same is shown quantitatively in Section 5.

To deal with that problem, we noticed that for the words we are interested in (named entities, acronyms, and domain-specific special words) most of the time a substitution error occurs (the results in Section 5 show that more than 84% of errors can be resolved by a substitution). Let Z_1 be the word that should had been recognized and $\tilde{Z}_1$ the wrongly recognized one. When we add $\tilde{Z}_1$ to the context biasing list and run the model again, we observe that the model mostly predicts the token of $\tilde{Z}_1$. Therefore, the idea is to use the summary vector of $\tilde{Z}_1$ (instead of Z_1) in equation 5 but keep using Z_1 for E'. We denoted this approach as context biasing + replacement and use a context biasing list entry $\tilde{Z}_1 \rightarrow Z_1$ for illustration purposes (see Figure 1).

In practice, our method would be applied as follows: Before running the model, a context biasing list can be supplied that contains words (named entities, acronyms, or domain-specific special words) that are likely to occur in the speech. While running the model to recognize speech, users can correct substitution errors of important words and add them to the context biasing list. Our method then improves using these corrections (as shown in Section 5).

4. Experiments

4.1. Data

To evaluate our method, we created a test set from the English data of the Yodas data set [28]. Our goal was to identify cases where a standard context biasing model consistently fails on the same rare words — a necessary condition for meaningfully assessing whether our method can improve. Following [19], we define rare words as words appearing in the references that occur infrequently overall yet are specific to a single YouTube video. Concretely, we retained words occurring at least four times but exclusively within one YouTube video. This yielded 6363 utterances (47.8 hours of audio) containing 8510 rare word occurrences across 1360 unique rare words. For each utterance, we ran our context biasing model using exactly the rare words of that utterance as the biasing list, reflecting an oracle biasing scenario. To focus our evaluation on the cases most relevant to our method, we filtered this set to retain only utterances in which at least one rare word was misrecognized, and only rare words that were misrecognized in at least two distinct utterances. The resulting test set comprises 300 utterances (2.24 hours of audio) with 379 rare word occurrences spanning 94 unique rare words on which the baseline context biasing model reliably struggles.

We applied the same procedure to Earnings-21 [29], LibriSpeech [30], Fleurs [31] and Voxpopuli [32]. The resulting test sets to evaluate misrecognized rare words were too small to yield a statistically meaningful evaluation. For example, in the Earnings-21 test set, only 5 rare words were misrecognized at least twice.

4.2. Models

We use Whisper [3] (whisper-large-v2) as our speech foundation model. The context biasing list is tokenized / embedded using the Whisper tokenizer / Whisper embedding, and the context is encoded using a transformer encoder (the encoder of mBART-50 [33]).

We trained the context biasing model on Common voice [34]. We only trained the context encoder and the added linear layers. In contrast to [27], we did not train the embedding and output layer. This has the advantage to prevent catastrophic forgetting [35] of the representations learned by the baseline model. Since we do not have access to the training data of the baseline model, this approach yielded substantially better overall performance.

During decoding of the test set, the context biasing list contains the rare words belonging to the utterance which is currently decoded. Optionally, we add all other rare words from the test set to the context biasing list as distractors.

4.3. Approaches

To generate the changed context biasing list for the approach context biasing + replacement, we manually annotated the substitution errors $\tilde{Z}_1 \rightarrow Z_1$ of the approach context biasing for all rare words. This resulted in 228 and 226 replacements for the hypotheses without and with distractors, respectively. Finally, we add to the rare words of an utterance the replacements $\tilde{Z}_1 \rightarrow Z_1$ of *other* utterances which contain the same rare word (see Figure 1, middle). Examples for the replacements are: Lodea→Lottia, Latia→Lottia, Röding→Rekin, Röging→Rekin, Lindstra→Lenstra, Lunster→Lenstra, PPAL→PIPOW, PayPal→PIPOW.

To investigate the effect the number of replacements has, we restrict the number of added replacements per rare word between 1 and 4 and randomly sample this number of replacements. We have no rare word with more than 4 different replacements. When not using distractors, we obtain for a maximum of 1, 2, 3 and 4 replacements per rare word in total 244, 344, 380 and 382 replacements, respectively. When using distractors, we obtain for a maximum of 1, 2, 3 and 4 replacements per rare word in total 242, 340, 371 and 376 replacements, respectively. The number of replacements is higher than 228/226 because one replacement can be used for multiple occurrences of the same rare word.

For comparison, we compare with two approaches:
1) Context biasing + text replacement: For this approach, we do not run the context biasing model with a context biasing list containing replacements, instead we take the hypotheses of the approach context biasing and apply the respective replacements that are used in the approach context biasing + replacement on the hypotheses.
2) Context biasing + text replacement oracle: This approach is similar to the previous approach; however, the replacements $\tilde{Z}_1 \rightarrow Z_1$ for the context biasing list are not taken from other utterances containing the same rare word but only from the *same* utterance (see Figure 1, bottom).

4.4. Metrics

The performance of an ASR system is typically measured using the word error rate (WER). To measure how well a context biasing method is working, [19] extended this metric to UWER (unbiased WER measured on words not in the biasing list) and BWER (biased WER measured on words in the biasing list), given a test set together with a corresponding context biasing list. We evaluate these metrics along with WER to compare different approaches.

5. Results

The results for the Yodas test set can be seen in Table 1.

By construction of the Yodas test set (see Section 4.1) the BWER of the Context biasing approach (without distractors) is very high (82.8%). Note, that the BWER is not 100% because some utterances contain a rare word multiple times and not every instance of the rare word is misrecognized. On the other hand, the BWER of the approach Context biasing + text replacement oracle is only 13.2%. Therefore, over 84% of errors can be resolved by a substitution.

When comparing the approach Context biasing + text replacement (without distractors), we see that it is between 44%

Table 1: *Results on the Yodas test set: BWER/UWER/WER in % for the different approaches depending on the maximum number of added replacements per rare word with and without adding distractors.*

Approach	Number repl. / Distractors	1	2	3	4
Context biasing	✗	82.8/6.4/7.8			
Context biasing + text replacement	✗	46.2/6.0/6.8	36.9/6.0/6.6	34.6/5.9/6.5	34.6/5.9/6.5
Context biasing + replacement	✗	30.6/6.0/6.5	27.2/6.0/6.4	26.9/6.0/6.4	26.9/6.0/6.4
Context biasing + text replacement + replacement	✗	24.5/5.9/6.3	21.6/6.0/6.3	21.6/6.0/6.3	21.6/6.0/6.3
Context biasing + text replacement oracle	✗	13.2/5.7/5.9			
Context biasing	✓	83.6/6.6/8.1			
Context biasing + text replacement	✓	47.0/6.3/7.1	38.3/6.2/6.8	35.9/6.2/6.8	35.6/6.2/6.8
Context biasing + replacement	✓	34.3/6.2/6.8	28.0/6.2/6.6	27.7/6.1/6.6	27.7/6.1/6.6
Context biasing + text replacement + replacement	✓	26.4/6.2/6.6	21.4/6.1/6.4	21.4/6.1/6.4	21.4/6.1/6.4
Context biasing + text replacement oracle	✓	14.5/6.0/6.2			

and 58% (relative) better than Context biasing. Furthermore, a higher number of replacements reduces BWER by 25%.

Next, we see that the BWER of Context biasing + replacement is between 22% and 34% better than Context biasing + text replacement. A higher number of replacements also helps this approach in terms of BWER, but only 12% relative. Together with the better performance for number replacements 1, this suggests that one given replacement per rare words is used more efficiently than in Context biasing + text replacement. We checked statistical significance using Bootstrap Resampling [36] with one million samples and found that the difference (46.2% vs. 30.6%) is statistically significant with p-value $2.0e^{-6}$. The same holds for number replacements 4, where we have BWER 34.6% vs. 26.9% and a p-value of $3.9e^{-3}$.

The computational overhead when adding replacements to the context biasing list is negligible. The context encoder output can be reused when decoding multiple utterances and extending the embedding / output layer is insignificant compared to the vocabulary size (which is around 250k).

The approach Context biasing + replacement outperforms Context biasing + text replacement in the cases where the model predicts for different utterances containing the same rare word different words instead of the rare word. In this case, the text replacement does not match. Examples include: Lottia (genus of sea snails), Rekin (name of a festival), Qama (name), Finotex (company), BANI (concept), Chariklo (planet), Lenstra (name), Parasuram (person name), PIPOW (framework name), Kirima (name).

Combining the methods replacement and text replacement delivers even better results: BWER is between 38% and 47% better than text replacement alone. Furthermore, up to 88% of the errors (which can be corrected by the oracle approach) can be corrected by the combination of replacement and text replacement. The oracle approach with BWER 13.2% is still better than that, however, it has access to oracle information.

Looking at the performance with added distractors, we see the same behavior compared to not adding distractors but mostly with slightly lower performance in all metrics, which is expected. Comparing the approaches Context biasing + replacement and Context biasing + text replacement, we have for number replacement 1 (47.0% vs. 34.3% BWER) and 4 (35.6% vs. 27.7% BWER) p-values of $1.6e^{-4}$ and $7.3e^{-3}$, respectively.

Finally, the UWER performance changes less than 2% relative (excluding the Context biasing and oracle approaches) while the WER performance improves up to 7% between text replacement, replacement and the combination of both because the BWER performance improves.

5.1. Limitations

Our proposed approach context biasing + replacement can only be applied if there is a substitution error, not if there is a deletion error. Furthermore, if the substitution error produced a word with a very high number of occurrences, our approach likely will produce false positives. For such cases, it might be best to keep the replacement in the context biasing list only for the relevant session and then transfer the knowledge through continuous learning [37].

We also tried to generate the replacements automatically from the utterances where the context biasing model was able to correctly recognize a rare word compared to the baseline model which failed to do so. However, this did not lead to improvements, suggesting that manual corrections are necessary.

6. Conclusion

In this work, we addressed the challenge of recognizing words where existing context biasing methods are not able to relate audio and corresponding text, e.g. in case of a pronunciation-orthography mismatch.

We proposed a method, context biasing + replacement, that allows corrections of substitution errors provided during inference to enhance recognition accuracy. By incorporating these corrections into the context biasing list, our method significantly improves the recognition of such problematic words. Our experiments demonstrated a relative improvement in BWER between 22% and 34% compared to a text-based replacement, while maintaining the overall word error rate. Our method can use one correction more efficiently compared to the text-based replacement.

7. Acknowledgment

This research was supported in part by a grant from Zoom Video Communications, Inc. Furthermore, the projects on which this research is based were funded by the Federal Ministry of Education and Research (BMBF) of Germany under the number 01EF1803B (RELATER), the Horizon research and innovation program of the European Union under grant agreement No 101135798 (Meetween), and the KIT Campus Transfer GmbH (KCT) staff in accordance with the collaboration with Carnegie – AI. The authors gratefully acknowledge the support.

8. Use of generative AI tools

Generative AI tools were used in a limited capacity during the preparation of this work. Specifically, AI-assisted code completion was employed to support software development tasks. Language model suggestions were used to refine the clarity and style of the written text. Additionally, generative AI tools assisted in the enhancement of figures. All substantive intellectual contributions, including the research design, methodology, analysis, and conclusions, are entirely the authors' own.

9. References

[1] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[2] N.-Q. Pham, T.-S. Nguyen, J. Niehues, M. Müller, S. Stüker, and A. Waibel, "Very deep self-attention networks for end-to-end speech recognition," *arXiv preprint arXiv:1904.13377*, 2019.

[3] 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, 2023.

[4] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[5] B. Suhm, M. Woszczyna, and A. Waibel, "Detection and transcription of new words." in *Eurospeech*. Citeseer, 1993.

[6] B. Suhm, "Towards better language models for spontaneous speech," in *Proc. ICSLP'94*, vol. 2, 1994.

[7] P. Maergner, A. Waibel, and I. Lane, "Unsupervised vocabulary selection for real-time speech recognition of lectures," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012.

[8] A. Waibel and I. R. Lane, "System and methods for maintaining speech-to-speech translation in the field," Jun. 19 2012, uS Patent 8,204,739.

[9] ——, "Enhanced speech-to-speech translation system and methods for adding a new word," Mar. 3 2015, uS Patent 8,972,268.

[10] A. Sriram, H. Jun, S. Satheesh, and A. Coates, "Cold fusion: Training seq2seq models together with language models," *Interspeech 2018*, 2018.

[11] I. Williams, A. Kannan, P. S. Aleksic, D. Rybach, and T. N. Sainath, "Contextual speech recognition in end-to-end neural network systems using beam search." in *Interspeech*, 2018, pp. 2227–2231.

[12] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1–5828.

[13] R. Huang, O. Abdel-hamid, X. Li, and G. Evermann, "Class lm and word mapping for contextual biasing in end-to-end asr," *Interspeech 2020*, 2020.

[14] A. Kojima, "A study of biasing technical terms in medical speech recognition using weighted finite-state transducer," *Acoustical Science and Technology*, vol. 43, no. 1, pp. 66–68, 2022.

[15] 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.

[16] A. Bruguier, R. Prabhavalkar, G. Pundak, and T. N. Sainath, "Phoebe: Pronunciation-aware contextualization for end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019.

[17] M. Jain, G. Keren, J. Mahadeokar, G. Zweig, F. Metze, and Y. Saraf, "Contextual rnn-t for open domain asr," 2020.

[18] C. Huber, J. Hussain, S. Stüker, and A. Waibel, "Instant one-shot word-learning for context-specific neural sequence-to-sequence speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021.

[19] 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.

[20] M. Han, L. Dong, Z. Liang, M. Cai, S. Zhou, Z. Ma, and B. Xu, "Improving end-to-end contextual speech recognition with fine-grained contextual knowledge selection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8532–8536.

[21] S. Dingliwal, M. Sunkara, S. Ronanki, J. Farris, K. Kirchhoff, and S. Bodapati, "Personalization of ctc speech recognition models," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 302–309.

[22] K. Huang, A. Zhang, Z. Yang, P. Guo, B. Mu, T. Xu, and L. Xie, "Contextualized end-to-end speech recognition with contextual phrase prediction network," in *Proc. Interspeech 2023*, 2023, pp. 4933–4937.

[23] X. Yang, W. Kang, Z. Yao, Y. Yang, L. Guo, F. Kuang, L. Lin, and D. Povey, "Promptasr for contextualized asr with controllable style," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 10 536–10 540.

[24] 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.

[25] F. Yu, H. Wang, X. Shi, and S. Zhang, "Lcb-net: Long-context biasing for audio-visual speech recognition," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.

[26] C. Xiao, Z. Hou, D. Garcia-Romero, and K. J. Han, "Contextual asr with retrieval augmented large language model," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[27] Y. Sudo, Y. Fujita, A. Kojima, T. Mizumoto, and L. Liu, "Owsm-biasing: Contextualizing open whisper-style speech models for automatic speech recognition with dynamic vocabulary," *arXiv preprint arXiv:2506.09448*, 2025.

[28] X. Li, S. Takamichi, T. Saeki, W. Chen, S. Shiota, and S. Watanabe, "Yodas: Youtube-oriented dataset for audio and speech," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[29] M. Del Rio, N. Delworth, R. Westerman, M. Huang, N. Bhandari, J. Palakapilly, Q. McNamara, J. Dong, P. Zelasko, and M. Jetté, "Earnings-21: A practical benchmark for asr in the wild," *arXiv preprint arXiv:2104.11348*, 2021.

[30] 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.

[31] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna, "Fleurs: Few-shot learning evaluation of universal representations of speech," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 798–805.

[32] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," *arXiv preprint arXiv:2101.00390*, 2021.

[33] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan, "Multilingual translation with extensible multilingual pretraining and finetuning," *arXiv preprint arXiv:2008.00401*, 2020.

[34] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[35] R. M. French, "Catastrophic forgetting in connectionist networks," *Trends in cognitive sciences*, vol. 3, no. 4, pp. 128–135, 1999.

[36] T. Berg-Kirkpatrick, D. Burkett, and D. Klein, "An empirical investigation of statistical significance in nlp," in *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 2012, pp. 995–1005.

[37] C. Huber and A. Waibel, "Continuously learning new words in automatic speech recognition," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.