INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

A Unified Recognition and Correction Model under Noisy and Accent Speech Conditions

Zhao Yang[1,3], *Dianwen Ng*[2,3], *Chong Zhang*[2], *Rui Jiang*[1], *Wei Xi*[1], *Yukun Ma*[2], *Chongjia Ni*[2], *Jizhong Zhao*[1], *Bin Ma*[2], *Eng Siong Chng*[3]

[1]Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, China
[2]Speech Lab of DAMO Academy, Alibaba Group [3]Nanyang Technological University, Singapore

zhaoyang9425@gmail.com, dianwen.ng@alibaba-inc.com, {xiwei,zjz}@xjtu.edu.cn, aseschng@ntu.edu.sg

Abstract

Automatic speech recognition (ASR) and its post-processing, such as recognition error correction, are usually cascaded in a pipeline ignoring their strong interconnection. Inspired by the recent progress of leveraging text data to improve linguistic modeling, we propose a **U**nified **A**SR and error **C**orrection framework (**UAC**), coupling speech recognition and error correction to capture richer semantic information for improving the performance of speech recognition. The proposed framework established interaction between speech and textual representations via explicitly fusing their uni-modal embeddings in a shared encoder. Additionally, the proposed framework is flexible to operate in either synchronous or asynchronous variant and could be equipped with modality and task tags enhancing its adaptation to heterogeneous inputs. Experimental results on accented and noisy speech datasets demonstrate that our method effectively produces improved word error rate when compared against the pipeline baselines.

Index Terms: Speech Recognition, Error Correction, Unified Model, Interactive Training, Noisy and Accented Speech

1. Introduction

End-to-end Automatic Speech Recognition (ASR) [1, 2, 3, 4, 5] that directly transcribes human speech into text sequence has shown remarkable advance. However, ASR systems can make errors, particularly when dealing with accents, noise, or complex language. To address these issues, error correction techniques [6, 7, 8] can be applied to improve the accuracy of ASR output. In previous studies, ASR and ASR error correction were usually modeled as two independent modules connected in a cascade fashion, in which ASR first transcribes speech into a noisy transcription, then the errors will be fixed by the correction module using linguistic or global information.

Recent studies have investigated directly unifying ASR and recognition error correction in a single model, where ASR and error correction are tackled simultaneously. Existing approaches can be categorized into the three: *a) Mask-CTC*. These methods [9, 10, 11] first generate CTC-based ASR hypothesis and low-confidence tokens are masked based on the CTC probabilities, then masked tokens are iteratively refined conditioning on the other unmasked tokens based on the iterative refinement decoding [12]. The models are trained with the joint CTC and mask-predict objectives. *b) Stacked-ASR-LM*. These methods [13, 14] straightforward incorporate pre-trained acoustic encoder and linguistic decoder, *i.e.*, BERT as error correction model in an end-to-end framework. *c) Text-Supervision-ASR*. These methods [15, 16] utilize knowledge distillation based language model integration for ASR. The pre-trained language model transfers knowledge to ASR model.

On the one hand, the encoder-decoder architecture [17, 18] used in a large variety of sequence-to-sequence transformation tasks, including ASR [19] and error correction [20, 21] has proven to be a powerful and flexible approach. On the other hand, recent studies [22, 23, 24, 25] have shown the effectiveness of using text data to improve linguistic modeling ability in speech-to-text tasks. [23, 24] focus on improving speech-to-text via joint-training a speech model with auxiliary text-based tasks. [25] leverages large-scale unlabeled speech and text data to pre-train a common encoder-decoder model, further supporting various spoken language processing tasks across different modalities. Inspired by recent advances jointly training between speech and text and architecture similarity between ASR and error correction tasks, we propose to **U**nify **A**SR and **C**orrection tasks in a single framework by sharing a common conditional language model, namely **UAC**, in this work. The proposed UAC framework contains two modality-specific embeddings, a modality-agnostic shared encoder and a shared decoder. The input speech and text are embedded into modality-specific space with two embeddings and are converted to shared space with a shared encoder, from which the decoder generates text output. Specifically, we design two variants, synchronous and asynchronous models, based on different interactions of speech and text embeddings. The synchronous model focuses on the joint modeling of paired speech-text representations, while the asynchronous model focuses on progressive training for ASR and error correction tasks. We verify the effectiveness of the proposed framework in both restricted and unrestricted settings. Experiments show the proposed system can effectively reduce word error rate (WER) for both ASR and correction tasks. In addition, our pre-training strategies result in additional reductions in WER compared to the same system when trained from scratch.

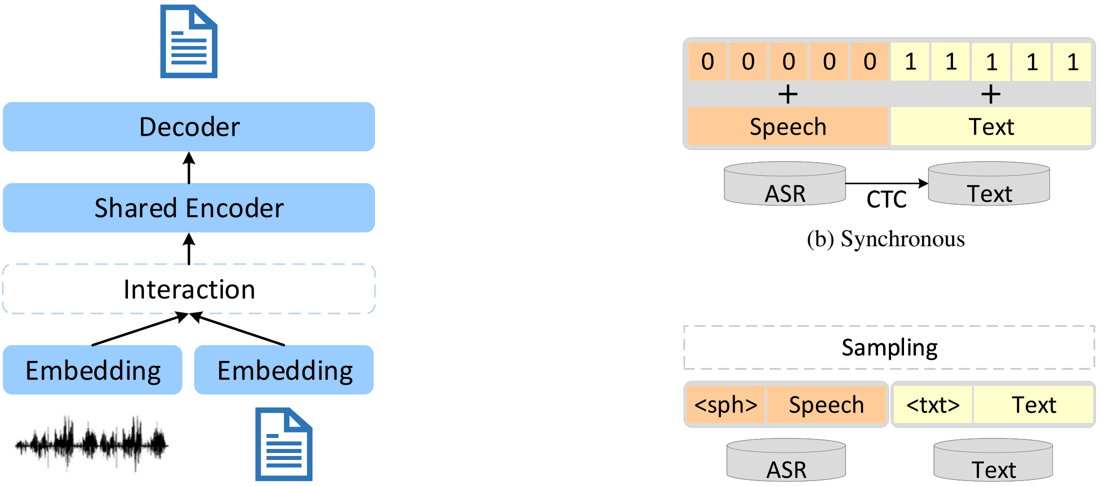
2. Methodology

2.1. Unified ASR and Correction Framework

As shown in Figure 1(a), the framework is fed with speech and text as the input and generates the corresponding text output, which consists of four main components: speech embedding, text embedding, shared encoder and decoder. These components are all transformer-based sub-structures.

The speech embedding takes the 80-channel log Mel-filter bank feature $\mathbf{X}_a^{T\times 80}$ as the audio input, where T is the input audio length and produces context-aware features $\mathbf{E}_a^{T\times D}$, where D is the model dimension. The text embedding converts text input to contextualized features. Specifically, it first transforms a sequence of token indexes $\mathbf{X}_t^{T'}$ into a sequence of embedding vectors via the embedding layer, where T' is the tokens length of input text and then feeds them into stacks of Transformer en-

10.21437/Interspeech.2023-1300



(a) Unified ASR and correction framework

(b) Synchronous

(c) Asynchronous

Figure 1: *The diagram of unified ASR and correction framework. (a) Both variants consist of four common components: speech embedding, text embedding, shared encoder and decoder. (b) Synchronous model. (c) Asynchronous model.*

coder layers to generate context-aware features $\mathbf{E}_t^{T' \times D}$. The shared encoder receives speech and text representations synchronously or asynchronously to produce unified semantic representations. The decoder uses input representations from the shared encoder to generate output tokens in an auto-regressive way.

According to the data interaction mode fed into the shared encoder, we design two variant models based on the intuition of jointly training of recognition and correction tasks.

Synchronous Model As shown in Figure 1(b), the synchronous model takes speech-text pairs as input, where the text is derived from from CTC predictions of speech embedding when training. The latent features from both speech and text embeddings are concatenated to get fused features $\mathbf{E}_s^{(T+T') \times D}$, which are then fed into the shared encoder. This approach allows for better modeling of the correlations between the two modalities.

Asynchronous Model As shown in Figure 1(c), the asynchronous model is trained by alternately sampling speech and text data with probability, which relaxes the constraints of paired speech-text input. If speech input is selected, the ASR task will be performed. If text input is selected, the error correction task will be performed. The noisy text input is either from ASR predictions or from noisy ground truth.

2.2. Training Strategy

The UAC framework is trained in both restricted and unrestricted settings. In the restricted setting, the framework is trained from scratch with target datasets. In the unrestricted setting, the framework is pre-trained with large-scale labeled datasets and then fine-tuned with downstream target datasets.

Synchronous Model The synchronous model is trained in the same way in restricted and unrestricted settings. The training loss of the synchronous model can be formulated as:

$$\mathcal{L} = \alpha \mathcal{L}_{asr} + \gamma \mathcal{L}_{ctc} \tag{1}$$

where $\mathcal{L}_{asr}$, $\mathcal{L}_{ctc}$ are cross-entropy and CTC loss for ASR task respectively. The weight α, γ are set to 0.7, 0.3 respectively

Asynchronous Model In unrestricted setting, the asynchronous model is first pre-trained by alternately sampling speech and text data with probability, then the model is fine-tuned by sequentially performing ASR and error correction tasks, in which ASR predictions are used as input for error correction task. In restricted setting, the training strategy is same as that of the fine-tuning stage in unrestricted setting.

Due to the difference in granularity and data size between speech and text data, uniform sampling is not the optimal strategy for the asynchronous model. In this paper, we use a sampling method similar to that of SpeechT5[1] and the details of speech text pre-training by probability sampling are shown in Algorithm 1.

The overall loss of the asynchronous model can be formulated as:

$$\mathcal{L} = \alpha \mathcal{L}_{asr} + \beta \mathcal{L}_{correction} + \gamma \mathcal{L}_{ctc} \tag{2}$$

where $\mathcal{L}_{correction}$ is cross-entropy loss for correction task. The weight α, β, γ are set to 0.5, 0.5, 0.3 respectively.

2.3. Modality and Task Tags

Since the feature distribution of various modalities is heterogeneous, we leverage modality and task tags as priori information to demarcate the boundary of fused features and indicate the modality and task type currently being processed for better modeling.

To this end, we incorporate the modality embedding to the feature embedding for the synchronous model through a summation operation, as demonstrated in Figure 1(b). For the asynchronous model, before feeding the feature embedding to the shared encoder, the modality tag *<spc>*/*<txt>* are appended to the starting position of the embedding, as demonstrated in Figure 1(c). Meanwhile, the task tag *<asr>*/*<corr>* are added to the beginning of the decoder input.

2.4. Differences with Existing Methods

While UAC bears some similarities to previous unifying ASR and correction studies [9, 13, 15], it differs in several key aspects. Firstly, UAC focuses on utilizing a unified model architecture that incorporates a shared conditional language model. This unique design allows us to explore the potential mutual

[1] https://github.com/microsoft/SpeechT5/blob/main/SpeechT5/speecht5/data/multitask_dataset.py

Table 1: *Word error rate (WER, %) results on noisy and accented English datasets in both restricted and unrestricted settings. En-De ASR & Correction represents a cascade model, which first uses the encoder-decoder ASR model to recognize text, and then uses the Transformer error correction model to refine the text. FairseqS2T represents the Speech-Transformer model implemented by Fairseq, which is pre-trained on* LIBRISPEECH *dataset and fine-tuned on cross-domain datasets.*

Setting	**Method**	**Noisy**				**Accented**	
		Recognition		**Correction**		**Recognition**	**Correction**
		test	test-other	test	test-other	test	test
Restricted	En-De ASR & Correction	30.00	51.48	25.56	36.88	9.93	9.67
	Synchronous (Ours)	29.86	50.04	18.53	33.27	16.44	8.27
	Asynchronous (Ours)	**18.87**	**33.02**	**17.04**	**27.93**	**8.89**	**7.51**
Unrestricted	FairseqS2T	12.09	22.65	-	-	8.36	-
	Synchronous (Ours)	24.69	35.16	13.37	21.54	14.57	6.07
	Asynchronous (Ours)	**11.32**	**20.54**	**10.42**	**18.73**	**6.47**	**5.42**

improvement of both ASR and text correction. This inspiration sets our approach apart from other methods. Secondly, UAC emphasizes maintaining a high degree of flexibility by considering the independence and association of the model components. After completing the model training, we have the option to either separate the ASR model and the error correction model from the unified model or utilize the unified model directly for recognition-refinement tasks. This trade-off enables us to strike a balance between efficiency and accuracy based on the specific requirements of the application.

3. Experiments

3.1. Datasets

Noisy Speech and Text Generation We built up the noisy speech dataset[2] by uniformly sampling a noise clip from DNS Challenge 2020 noise dataset [26] and adding it to 100 hours of clean subset from LIBRISPEECH. The SNR levels are sampled from a uniform distribution between 0dB, 5dB, 10dB, 15dB, 20dB. Following the text interrupting approach in BART [27, 25], the noisy text is generated[3], where the synthesized noisy text data is used for pre-training.

Accented Speech AESRC2020 [28] is used to evaluate the performance of the proposed approach. AESRC2020 is a 164-hour accented English speech corpus that includes recordings from non-native speakers of English. Since no labeled test set is publicly released, we split a subset of about 10% from the training speech as the test set.

ASR We conduct experiments on 960 hours of LIBRISPEECH during pre-training in unrestricted setting. The noisy and accented ASR data are used to measure the performance of the model in cross-domain scenarios.

Error Correction Text transcription in LIBRISPEECH dataset and co-training text data for language modeling coming with the LIBRISPEECH dataset are used during pre-training. The one-best hypothesis from ASR model is used as input to the error correction task during fine-tuning.

Note that all models evaluated are by default trained in restricted setting unless otherwise stated.

3.2. Experimental Setup

The proposed encoder-decoder architecture follows the Transformer sub-structures, which contains 12-layer speech embedding, 3-layer text embedding with embedding layer, 3-layer shared encoder and 6-layer decoder. Due to the constraints of computing resources, each layer comprises 256 hidden units, 4 attention heads, and 2,048 feed-forward size.

Algorithm 1 Speech-Text Sampling Pre-training Algorithm

```
Input: Speech training data $\mathcal{D}_s$, Text training data $\mathcal{D}_t$
Initialize: Speech embedding $\theta_s$, Text embedding $\theta_t$,
Shared encoder $\theta_{share}$, Decoder $\theta_d$
$M = Sum(\mathcal{D}_s)$                          ▷ sum of speech frames
$N = Sum(\mathcal{D}_t)$                           ▷ sum of text tokens
$Ratio = M/(M+N)$
while Training do
    $SpeechTag, TextTag$ = False, False
    $r \sim \mathcal{U}(0,1)$
    if $r < Ratio$ then
        $SpeechTag$ = True
    else $TextTag$ = True
    end if
    if $SpeechTag$ then                            ▷ ASR task
        $(X_s, Y_s) \leftarrow Next(\mathcal{D}_s)$
        $Z_{asr}, Z_{ctc} = f(X_s; [\theta_s, \theta_{share}, \theta_d])$
        $\mathcal{L}_{asr} = CrossEntropy(Z_{asr}, Y_s)$
        $\mathcal{L}_{ctc} = CTC(Z_{ctc}, Y_s)$
    else if $TextTag$ then                         ▷ Correction task
        $(X_t, Y_t) \leftarrow Next(\mathcal{D}_t)$
        $Z = f(X_s; [\theta_t, \theta_{share}, \theta_d])$
        $\mathcal{L}_{corr} = CrossEntropy(Z, Y_t)$
    end if
    Update model with Eq. 2
end while
return $\theta_s, \theta_t, \theta_{share}, \theta_d$
```

Input speech is represented as 80-D log Mel-filter bank coefficients computed every 10ms with a 25ms window. For reducing the computational cost, the input speech features are processed by two convolutional layers, which have a stride of 2×2 and downsample the sequence by a factor of 2. The SpecAugment [29] data augmentation with the LB policy is applied. We tokenize training text data using subword units with a vocabulary size of 10k, learned from SentencePiece.

All experiments are implemented based on the Fairseq[4] toolkit. We use the Adam optimizer with β_1 = 0.9, β_2 = 0.997 and adopt inverse square root learning schedule. We apply a dropout rate of 0.1 and label smoothing of 0.1 for regularization. In a restricted setting, we train for atotal of 80k steps. In an unrestricted setting, we train 400k steps for pre-training and

[2] https://github.com/microsoft/DNS-Challenge

[3] https://github.com/microsoft/SpeechT5/tree/main/SpeechT5/speecht5/data

[4] https://github.com/facebookresearch/fairseq

80k steps for fine-tuning.

3.3. Experimental Results

We compare our UAC framework with Transformer baseline systems in both restricted and unrestricted settings. The main results on noisy and accented English datasets are summarized in Table 1. The results show that both proposed variant models give an improvement over the baseline and compared models in terms of WER, which proves the effectiveness of combining recognition and correction tasks in a unified framework under noisy and accented speech conditions.

In addition, we find some other interesting points by comparing the results. First, the recognition result of the asynchronous model outperforms the encoder-decoder based ASR model, which indicates that the auxiliary error correction task benefits the primary ASR task within the multitask learning framework. Second, the final result of the synchronous model outperforms the encoder-decoder based ASR model. This proves that paired speech-text representation helps to preserve more information from different input modalities and thus boost the performance. Third, the asynchronous model surpasses the synchronous model. One possible reason is that the asynchronous model reuses conditional language components in sequential ASR recognition and error correction tasks while the synchronous model suffers from terrible CTC prediction results in noisy and accented speech conditions without language model, which implies the importance of language model plays in noisy and accented speech recognition tasks. Finally, proposed UAC framework outperforms the cascaded model with the same amount of data, which demonstrates the effectiveness of the proposed model to a certain extent.

3.4. Effects of the Number of Shared Encoder Layers

We explore the effect of the number of shared encoder layers on the synthesized noisy dataset. For a fair comparison, the total number of layers of the shared encoder and text embedding is unchanged to eliminate the interference of model parameters on model performance.

Table 2: *WER(%) results on the noisy test set with different numbers of shared layers.*

Model	**Shared Layers**	**Recognition**	**Correction**
Synchronous	0	30.03	18.90
	3	**29.86**	**18.53**
	6	29.87	18.55
Asynchronous	0	**18.75**	17.33
	3	18.87	17.04
	6	18.78	**16.91**

Table 2 shows the both synchrinous and asynchronous models benefit from using a shared encoder. However, increasing the number of shared encoder layers (e.g., from 3 to 6) might not necessarily introduce further improvement.

3.5. Effects of Extra Text Training

We investigate the effect of extra text in an unrestricted setting. For the synchronous model, we first train the correction modules with additional text data, then continue to train the joint model with large-scale paired speech-text data, and finally finetune the pre-trained model. In the experiment, we initialize the correction modules with pre-trained BART model parameters to simulate the extra text data training. For the synchronous model, we use LIBRISPEECH language dataset as extra text data.

Table 3: *Effects of extra text training on noisy and accented test sets in terms of WER(%).*

Model	**Extra Text**	**Recognition**	**Correction**
Noisy Speech			
Synchronous	Yes	**29.44**	**18.40**
	No	29.86	18.53
Asynchronous	Yes	**18.04**	**16.12**
	No	18.87	17.04
Accented Speech			
Synchronous	Yes	**9.05**	**5.87**
	No	9.57	6.07
Asynchronous	Yes	**6.16**	**5.08**
	No	6.47	5.42

Table 4: *Effects of modality and task tags on the accented test set in terms of WER(%).*

Model	**Use Tags**	**Recognition**	**Correction**
Synchronous	Yes	**16.44**	**8.27**
	No	16.46	8.31
Asynchronous	Yes	**8.89**	**7.51**
	No	8.96	7.59

Table 3 shows the results on noisy and accented test sets. After the extra single text data is used, the final performance is further improved, which is intuitive. Compared with the asynchronous model, the improvement of the synchronous model is limited. An important reason is that the three-stage training weakens the gain brought by the extra text.

3.6. Effects of Modality and Task Tags

We conduct experiments on accented test set to verify the effectiveness of modality and task tags and the results are shown in Table 4. The results show that the addition of tags further improves the accuracy of the recognition and error correction task, which verifies the effectiveness of the tags for the unified multi-task framework.

4. Conclusion and Future work

In this paper, we present a unified framework by combining ASR and error correction tasks, whose design follows structure similarity and task relatedness. Based on design intuition, we design two variant models: the synchronous model and the asynchronous model, where the synchronous model fuses speech-text pairs for synchronous training, and the asynchronous model is interactively trained between ASR and error correction tasks. Experiment results demonstrate the effectiveness of the proposed framework in cross-domain scenarios. In future work, we will utilize pre-trained acoustic models and pre-trained language models to achieve better performance via continuing pre-training.

5. Acknowledgements

This work was supported in part by National Key R&D Program of China 2020YFB2104000, NSFC Grant No. 61832008, 62002284, Alibaba Group through Alibaba Innovative Research (AIR) Program and China Scholarship Council.

6. References

[1] A. Graves, “Sequence transduction with recurrent neural networks,” in *International Conference of Machine Learning (ICML) 2012 Workshop on Representation Learning*, 2012.

[2] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, “State-of-the-art speech recognition with sequence-to-sequence models,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[3] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, “End-to-end attention-based large vocabulary speech recognition,” in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4945–4949.

[4] S. Karita, N. E. Y. Soplin, S. Watanabe, M. Delcroix, A. Ogawa, and T. Nakatani, “Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration,” *Proc. Interspeech 2019*, pp. 1408–1412, 2019.

[5] S. Kim, T. Hori, and S. Watanabe, “Joint ctc-attention based end-to-end speech recognition using multi-task learning,” in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[6] A. Mani, S. Palaskar, N. V. Meripo, S. Konam, and F. Metze, “Asr error correction and domain adaptation using machine translation,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6344–6348.

[7] L. Zhu, W. Liu, L. Liu, and E. Lin, “Improving asr error correction using n-best hypotheses,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 83–89.

[8] Y. Leng, X. Tan, R. Wang, L. Zhu, J. Xu, W. Liu, L. Liu, X.-Y. Li, T. Qin, E. Lin *et al.*, “Fastcorrect 2: Fast error correction on multiple candidates for automatic speech recognition,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 4328–4337.

[9] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, “Mask ctc: Non-autoregressive end-to-end asr with ctc and mask predict,” *Proc. Interspeech 2020*, pp. 3655–3659, 2020.

[10] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, “Non-autoregressive Error Correction for CTC-based ASR with Phone-conditioned Masked LM,” in *Proc. Interspeech 2022*, 2022, pp. 3889–3893.

[11] Y. Higuchi, B. Yan, S. Arora, T. Ogawa, T. Kobayashi, and S. Watanabe, “BERT meets CTC: New formulation of end-to-end speech recognition with pre-trained masked language model,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5486–5503.

[12] J. Lee, E. Mansimov, and K. Cho, “Deterministic non-autoregressive neural sequence modeling by iterative refinement,” in *2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*. Association for Computational Linguistics, 2020, pp. 1173–1182.

[13] C. Yi, S. Zhou, and B. Xu, “Efficiently fusing pretrained acoustic and linguistic encoders for low-resource speech recognition,” *IEEE Signal Processing Letters*, vol. 28, pp. 788–792, 2021.

[14] G. Zheng, Y. Xiao, K. Gong, P. Zhou, X. Liang, and L. Lin, “Wav-bert: Cooperative acoustic and linguistic representation learning for low-resource speech recognition,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 2765–2777.

[15] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, “Distilling the knowledge of bert for sequence-to-sequence asr,” *Proc. Interspeech 2020*, pp. 3635–3639, 2020.

[16] Y. Bai, J. Yi, J. Tao, Z. Tian, Z. Wen, and S. Zhang, “Fast end-to-end speech recognition via non-autoregressive models and cross-modal knowledge transferring from bert,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1897–1911, 2021.

[17] L. Dong, S. Xu, and B. Xu, “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

[18] Q. Zhang, H. Lu, H. Sak, A. Tripathi, E. McDermott, S. Koo, and S. Kumar, “Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7829–7833.

[19] P. Guo, F. Boyer, X. Chang, T. Hayashi, Y. Higuchi, H. Inaguma, N. Kamo, C. Li, D. Garcia-Romero, J. Shi *et al.*, “Recent developments on espnet toolkit boosted by conformer,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5874–5878.

[20] Y. Weng, S. S. Miryala, C. Khatri, R. Wang, H. Zheng, P. Molino, M. Namazifar, A. Papangelis, H. Williams, F. Bell *et al.*, “Joint contextual modeling for asr correction and language understanding,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6349–6353.

[21] S. Katsumata and M. Komachi, “Stronger baselines for grammatical error correction using a pretrained encoder-decoder model,” in *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020, pp. 827–832.

[22] Y. Huang, H.-K. Kuo, S. Thomas, Z. Kons, K. Audhkhasi, B. Kingsbury, R. Hoory, and M. Picheny, “Leveraging unpaired text data for training end-to-end speech-to-intent systems,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7984–7988.

[23] R. Ye, M. Wang, and L. Li, “End-to-End Speech Translation via Cross-Modal Progressive Training,” in *Proc. Interspeech 2021*, 2021, pp. 2267–2271.

[24] Y. Tang, J. Pino, C. Wang, X. Ma, and D. Genzel, “A general multi-task learning framework to leverage text data for speech to text tasks,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6209–6213.

[25] J. Ao, R. Wang, L. Zhou, C. Wang, S. Ren, Y. Wu, S. Liu, T. Ko, Q. Li, Y. Zhang *et al.*, “Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5723–5738.

[26] C. K. Reddy, H. Dubey, V. Gopal, R. Cutler, S. Braun, H. Gamper, R. Aichner, and S. Srinivasan, “Icassp 2021 deep noise suppression challenge,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6623–6627.

[27] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, “Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7871–7880.

[28] X. Shi, F. Yu, Y. Lu, Y. Liang, Q. Feng, D. Wang, Y. Qian, and L. Xie, “The accented english speech recognition challenge 2020: open datasets, tracks, baselines, results and methods,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6918–6922.

[29] 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,” *Proc. Interspeech 2019*, pp. 2613–2617, 2019.