LeVoice ASR Systems for the ISCSLP 2022 Intelligent Cockpit Speech Recognition Challenge

Yan Jia*, Mi Hong, Jingyu Hou, Kailong Ren, Sifan Ma, Jin Wang, Yinglin Ji,
Fangzhen Peng, Lin Yang, Junjie Wang

Lenovo Research, Beijing, China

Abstract

This paper describes LeVoice automatic speech recognition systems to track2 of intelligent cockpit speech recognition challenge 2022. Track2 is a speech recognition task without limits on the scope of model size. Our main points include deep learning based speech enhancement, text-to-speech based speech generation, training data augmentation via various techniques and speech recognition model fusion. We compared and fused the hybrid architecture and two kinds of end-to-end architecture. For end-to-end modeling, we used models based on connectionist temporal classification/attention-based encoder-decoder architecture and recurrent neural network transducer/attention-based encoder-decoder architecture. The performance of these models is evaluated with an additional language model to reduce word error rates. As a result, our system achieved 10.2% character error rate on the challenge test set data and ranked third place among the submitted systems in the challenge.

Index Terms: speech recognition, neural network, multi-model

1. Introduction

With the blooming of driverless technology and intelligent cockpit in recent years, voice interaction central control platforms have become increasingly popular. However, there still remain many challenges for accurate speech recognition in vehicle environments: the complex acoustic environment in and out of the cockpit and insufficient computing power on the cockpit devices. This year ISCSLP 2022 intelligent cockpit speech recognition challenge (ICSRC2022) [1] focuses on automatic speech recognition (ASR) in intelligent cockpit scenes, and we will introduce our ASR system submitted to the competition.

Recently, end-to-end (E2E) ASR [2, 3] has been significantly developed. Compared to traditional hybrid ASR systems [4], which are generally composed of acoustic model, language model and pronunciation dictionary, the E2E ASR system uses a single neural network to transform the acoustic feature sequence into a token sequence. Besides, with the application of self-attention based transformer [5], conformer [6] and other variants based on self-attention in E2E ASR, its performance has gradually improved.

There are three basic mainstream E2E ASR approaches, including connectionist temporal classification (CTC) [7, 8], attention-based encoder decoder (AED) [9], and recurrent neural network transducer (RNN-T) [10]. Among these three approaches, CTC is the earliest and can map the input speech signal to target labels without requiring any auxiliary alignments. However, it doesn’t perform satisfactorily as the conditional frame independence assumption. RNN-T extends CTC modeling by changing the objective function and the model architecture to remove the frame independence assumption. Moreover, AED is initially proposed for machine translation but got dramatic effects when applied to ASR in offline scenarios because of its naturally non-streaming by default. In recent years, two types two-pass joint models, including CTC/AED and RNNT/AED, are applied to E2E ASR and show that they perform well compared with the single basic model. In two-pass decoding, the second pass model is often used to improve the initial outputs from first-pass models by using n-best re-ranking.

Experiments suggest that all these two-pass joint models can achieve better performance with an internal or external language model, which can be a neural network language model (NNLM) or an N-gram model based on a weighted finite-state transducer (WFST). These methods can be used to make E2E ASR systems benefit from prior context information [11].

In this challenge, we use all two-pass architectures to build our system, aiming to utilize various advantages from different architectures by system fusion. Simultaneously, various data augmentation methods are used to train our model, which is proven effective for better performance.

The rest of the paper is organized as follows: Section 2 describes the details of the model structure of the ASR system we submitted, as well as some methods we use to improve the performance of the system. Section 3 describes the experimental details and results of this challenge. We make a summary of our work and propose some future directions for improvement in section 4.

2. Proposed System

2.1. Overview

The data set of ICSRC2022 challenge task2 contains 20 hours of audio data under various noise. The main content of the data is the user’s command, including controlling the air conditioner, playing songs, making phone calls, navigating, chatting, etc. These contents involve a large number of special words, such as contacts, singer names, navigation destinations and other out-of-vocabulary words. In addition, there are various noises from inside and outside the car, such as wipers, wind, engine, wheels, background music and interference speakers. The recording room is a small enclosed space, which causes serious reverberation. These raise great challenges for the performance of ASR system. To solve this problem, we made a rough analysis on this issue.

Firstly, the core problem of track2 is the speech in the background with reverberation and vehicle interior background noise. Therefore, it is necessary to have a front-end that provides high-quality speech. Secondly, there is no limit to the
The architecture of our system is shown in Figure 1. It briefly describes our training and inference process, in which the orange arrow represents the training process of each subsystem, the purple arrow represents the inference process of the subsystem, and the gray arrow represents the information transmission process required for the inference of the subsystem.

We do a lot of data simulation and augmentation work during the training stage. The training data consists of two parts. The training set of AIShell-1 [12] and the development set of track2 are used to train and fine-tune a TTS model respectively. It synthesizes the development set of track2 without noise and the training subset with vehicle noise. On the other hand, the development set of track2 and the training set with noise are handled by a neural network based deep noise suppression model. Finally, the data of training ASR model is composed of simulation data and original training data. Speed perturb, room impulse response and specaugment [13] are used when training the ASR model.

2.2. Data Processing

2.2.1. Speech Enhancement

In order to suppress the noise, the DCCRN [14] model is adopted. We first use the clean data from the AIShell-1 to mix with the noise from MUSAN [15] at a certain SNR to simulate noisy speech to train the base model. Then the background noise in the ICSRC2022 development dataset is used to generate a new training data set to finetune the base model. Finally, the noisy training data is denoised by the best model to obtain enhanced speech, which is used to finetune the ASR model.

2.2.2. TTS-based Speech Generation

We propose to use text-to-speech (TTS) to expand training data. We use VITS [16] model with a reference encoder, which can extract embedding, including speaker and channel information from the raw waveform. So when input reference audio with vehicle noise, we believe the reference encoder can extract the information of vehicle channel. Specifically, we first pre-train the model with AIShell-1 as a multi-speaker base model and then fine-tune the model with ICSRC’s development data to generate noisy speech. Figure 2 depicts the overall TTS pipeline. TTS data augmentation we used consists of two parts: multi-speaker clean speech and noisy speech. Clean speech is generated with the content of development data and voices from AIShell-1. Voices from development data and content of AIShell-1 are collected and used for synthesizing noisy speech. Given the limited amount of speech from new energy vehicles, we believe TTS is a good way to synthesize more data with similar content and similar noise in this particular task.

2.3. ASR Framework

We mainly construct three structures: Hybrid system, RNN-T/AED based two-pass joint model and CTC/AED based two-pass E2E model. We further fuse these models for improving robustness by using ROVER toolkit [17].

2.3.1. Hybrid system

Our hybrid model is based on the Kaldi [18] toolkit, which consists of an acoustic model (AM) and a language model (LM). In order to get the alignments, we trained a gaussian mixture model (GMM) based ASR model. An n-gram LM is built using the SRILM [19] toolkit.

2.3.2. RNN-Transducer/AED based system

Our proposed RNNT/AED based two-pass architecture is shown in Figure 3. It consists of five parts. 1) The shared encoder consists of multiple Squeezeformer [20] layers. 2) CTC decoder is a linear layer that converts the output of the shared encoder into CTC activation. 3) The RNNT decoder consists of a two-layer LSTM and a linear layer, which generates hypotheses at each time step. 4) The additional encoder is a 2-layer Conformer, which aims to process the output of the shared encoder further. 5) The AED decoder consists of a left-to-right attention decoder and a right-to-left attention decoder. A left-to-right attention decoder (L2R) models an ordered token sequence from left to right to represent past context information. A right-to-left attention decoder (R2L) models a right-to-left tag sequence to represent future context information.

In the first pass, each acoustic frame $X$ is passed through the shared encoder to obtain a shared output $E_{shared}$. Then $E_{shared}$ pass to the RNN-T decoder for generating hypotheses. In the second pass, $E_{shared}$ is passed to the an additional encoder to get $E_{ad-shared}$. Finally, $E_{ad-shared}$ is passed to the AED decoder. The AED decoder computes output according to $E_{ad-shared}$ during training. A single CTC decoder is used to compute auxiliary loss. The three losses generated by the three decoders can be simultaneously trained and jointly optimize the...
Figure 3: RNNT/AED based Model

Figure 4: CTC/AED based Model.

The total loss function $L_{\text{total}}$ defined as:

$$L_{\text{total}} = \lambda_{\text{ctc}} L_{\text{ctc}} + \lambda_{\text{rnnt}} L_{\text{rnnt}} + \lambda_{\text{aed}} L_{\text{aed}}$$  \hspace{1cm} (1)

where $L_{\text{ctc}}$ is the principal transducer loss, $L_{\text{rnnt}}$ is the AED loss based on cross-entropy loss with label smoothing, and $L_{\text{ctc}}$ is the CTC loss. The AED loss consists of a L2R AED loss and a R2L AED loss. $\lambda$ defines their respective contribution to the overall loss.

During decoding, LM shallow fusion [21] and ILME [22] decoding methods are adopted in this experiment. Then AED decoder can re-rank the hypotheses generated by RNNT decoder.

2.3.3. CTC/AED based system

We use a model structure similar to [23]. Our CTC/AED model architecture is shown in Figure 4, which is composed of a shared encoder, a CTC decoder and an asymmetry Left-Right attention decoder. Each part is consistent with RNN-T as mentioned above. When training, there is a strategy for the loss computation:

$$L_{\text{total}} = \lambda_{\text{ctc}} L_{\text{ctc}} + \lambda_{\text{rnnt}} L_{\text{rnnt}} + \lambda_{\text{aed}} L_{\text{aed}}$$  \hspace{1cm} (2)

where $L_{\text{rnnt}}$ can be calculated in Eq.(2). While Decoding, CTC decoder joint with LM could output n-best hypothesis in the first pass. Then, rescoring will be done on the n-best candidates by the AED decoder part based on the corresponding encoder output.

3. Experiments and Results

3.1. Data

The training data mainly consist of the base training datasets and the ICSRC2022 development dataset in this challenge. The base training set is part of the data in the open source corpus of OpenSLR¹, about 1300 hours. The statistics of the data we used for training are shown in Table 2, where the term “Openslr” denotes natural speech from the open-source data set. For synthetic data, 173k utterances from “Openslr” samples are used as the synthetic “TTS-openslr_subset” train set. After adding noise, we selected 1.17M utterances from “Openslr” as the “DNS-Openslr_subset” simulated through the DNS model randomly. Two different training setups in our experiments:

1. **Train-base**: use “Openslr” set, which are shown in Table 2.
2. **Train-all**: use all sets shown in Table 2.

In the data augmentation stage, we use the room impulse response from the RIR [24] to convolve training data to simulate far-field data. Speed perturbation is used to generate 0.9 and 1.1 speed ratio data. At last, we mix noises provided by MUSAN with a random SNR from -5 to 0 dB.

3.2. Training Setup

**Feature Extraction**: 80-dimensional fbank features are extracted from audios, and all the features are normalized with global CMVN. SpecAugment is used for data coverage and model robustness.

**Hybrid Model**: We use 13-dimensional MFCC acoustic features with delta to build speaker-adapted GMM based models. We also build hybrid ASR systems using TDNNs comprising 37-layer TDNN-F [25] blocks with dimension 1536.

**RNNT/AED based model**: The encoder is a 12-layer Squeezeformer, each with 2048 hidden units and 512-dimensional attention vector. Each layer contains eight attention heads. The prediction network has a 2-layer LSTM of 2048 hidden units, a 512-dimensional projection per layer, and an embedding layer of 512 units. The outputs of encoder and prediction network are fed to a joint network that has 512 hidden units. The L2R model is a 6-layer transformer decoder. The R2L model is a 3-layer transformer decoder. Each layer contains eight attention heads which are concatenated to form a 512-dimensional attention vector. Models are trained for a maximum of 100 epochs with the Adam optimizer with a learning rate of 0.001 and 25000 warmup steps. The top 20 models with the best validation accuracy are averaged and this averaged checkpoint is used for decoding. We set the weight $\lambda_{\text{ctc}}$ to 0.1, $\lambda_{\text{rnnt}}$ to 0.75 and $\lambda_{\text{aed}}$ to 0.15.

**CTC/AED based model**: CTC/AED based model uses a CTC weight of 0.3 and an attention weight of 0.7. A 12-layer

¹http://openslr.org/resources.php
Table 1: Overall results on ICSRC2022 test dataset with various ASR models and fusions.

| Framework | Data          | Decoding Method                        | WER   |
|-----------|---------------|----------------------------------------|-------|
| CTC/AED   | Train-base    | CTC prefix beam search + attention rescoring | 14.95 |
|           | Train-all     |                                        | 12.38 |
|           | Train-base    | CTC NNLM shallow fusion + attention rescoring | 14.67 |
|           | Train-all     |                                        | 12.11 |
|           | Train-base    | CTC TLG shallow fusion + attention rescoring | 16.86 |
|           | Train-all     |                                        | 11.72 |
| RNNT/AED  | Train-base    | RNNT prefix beam search + attention rescoring | 15.06 |
|           | Train-all     |                                        | 11.99 |
|           | Train-base    | RNNT NNLM shallow fusion + attention rescoring | 15.03 |
|           | Train-all     |                                        | 11.92 |
|           | Train-base    | RNNT ILME + attention rescoring        | 15.19 |
|           | Train-all     |                                        | 12.06 |
| K2-RNNT   | Train-base    | RNNT beam search + attention rescoring  | 14.32 |
|           | Train-all     |                                        | 12.73 |
| Hybrid    | Train-base    | beam search                            | 33.22 |
|           | Train-all     |                                        | 26.19 |
| ROVER fusion | -            |                                        | 10.20 |

Table 2: Dataset statistics

| Samples       | utterances |
|---------------|------------|
| Openslr       | 1196458    |
| ICSRC2022_dev | 7024       |
| TTS-ICSRC2022 | 7024       |
| TTS-openlr_subset | 173404  |
| DNS-openlr    | 1177393    |
| DNS-ICSRC2022 | 7024       |

Figure 5: Example of enhanced spectra after DNS module.

squeezeformer, a 6-layer L2R decoder and a 3-layer R2L decoder are used, each with 2048 units, with a 0.1 dropout rate. Each layer contains eight 64-dimensional attention heads concatenated to form a 512-dimensional attention vector. The training process is the same as RNNT/AED based model. Decoding is performed with a beam size of 10 and a CTC weight of 0.5.

K2 based model: We also used the "pruned_transducer_stateless5" recipes in the K2 [26] toolkit to build an RNNT/AED based model.

3.3. Experimental Results

3.3.1. Data Processing Results

Here we check the performance of the DNS module. Figure 5 shows the changes in spectrum features after passing to the DNS module. We calculated that the source-to-noise ratio (SNR) of the ICSRC development set is 0.412dB, while the SNR of data through the DNS module is 35.582dB.

3.3.2. ASR Model Comparison

The results of our framework are reported in Table 1. We report the character error rate (CER) for each ASR model. We also fuse the output of each system to improve the performance further. These conclusions can be drawn from our experiments. First, end-to-end models outperform the hybrid system under complex conditions. Second, for CTC/AED based model, decoding with a language model trained with the training data similar to the target domain helps improve performance on the test set. But it is not helpful for the RNNT/AED model. Third, The best result we submitted is a fusion of 16-best results, which achieved 10.2% on the challenge test set, while the official baseline only got 47.98% on the same data set.

4. Conclusions

This paper proposes our submission to the task2 of the ICSRC 2022 challenge. Our work includes the investigation of various data augmentation methods and the comparison of ASR model back-ends. Our proposed system improves against the baseline with an absolute reduction of 78.7% on the test dataset and ranks 3rd out of 20 participating systems in the challenge.

5. References

[1] K. H. L. X. L. W. E. S. C. H. B. B. Z. W. C. X. X. Ao Zhang, Fan Yu, “The iscslp 2022 intelligent cockpit speech recognition challenge (icsrc): Dataset, tracks, baseline and results,” in ISC-SLP. IEEE, 2022.
[2] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, “A comparison of sequence-to-sequence models for speech recognition,” in Interspeech, 2017, pp. 939–943.
[3] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang et al., “A comparative study on transformer vs rnn in speech applications,” in 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019, pp. 449–456.
[4] Y. Miao, M. Gowayyed, and F. Metze, “Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding,” in
A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu et al., “Conformer: Convolution-augmented transformer for speech recognition,” arXiv preprint arXiv:2005.08100, 2020.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in Proceedings of the 23rd international conference on Machine learning, 2006, pp. 369–376.

J. Libovický and J. Helcl, “End-to-end non-autoregressive neural machine translation with connectionist temporal classification,” arXiv preprint arXiv:1811.04719, 2018.

J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” Advances in neural information processing systems, vol. 28, 2015.

K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” in 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2017, pp. 193–199.

C. Peyer, S. Mavandadi, T. N. Sainath, J. Aqfél, R. Pang, and S. Kumar, “Improving tail performance of a deliberation e2e asr model using a large text corpus,” arXiv preprint arXiv:2008.10491, 2020.

H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline,” in 2017 20th conference of the oriental chapter of the international coordinating committee on speech databases and speech I/O systems and assessment (O-COCOSDA). IEEE, 2017, pp. 1–5.

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,” arXiv preprint arXiv:1904.08779, 2019.

Y. Hu, Y. Liu, S. Lv, M. Xing, S. Zhang, Y. Fu, J. Wu, B. Zhang, and L. Xie, “Decrn: Deep complex convolution recurrent network for phase-aware speech enhancement,” 2020.

D. Snyder, G. Chen, and D. Povey, “Musan: A music, speech, and noise corpus,” arXiv preprint arXiv:1510.08484, 2015.

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.

J. G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover),” in 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings. IEEE, 1997, pp. 347–354.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, F. Schwarz et al., “The kaldi speech recognition toolkit,” in IEEE 2011 workshop on automatic speech recognition and understanding, no. CONF. IEEE Signal Processing Society, 2011.

A. Stolcke, “Srilm-an extensible language modeling toolkit,” in Seventh international conference on spoken language processing, 2002.

A. Andrusenko, R. Nasredtinov, and A. Romanenko, “Uconv-conformer: High reduction of input sequence length for end-to-end speech recognition,” arXiv preprint arXiv:2208.07657, 2022.

R. Cabrera, X. Liu, M. Ghodsi, Z. Matteson, E. Weinstein, and A. Kannan, “Language model fusion for streaming end to end speech recognition,” arXiv preprint arXiv:2104.04487, 2021.