Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.041772

ARTICLE

A Robust Conformer-Based Speech Recognition Model for Mandarin Air Traffic Control

Peiyuan Jiang[1], **Weijun Pan**[1,*], **Jian Zhang**[1], **Teng Wang**[1] **and Junxiang Huang**[2]

[1]College of Air Traffic Management, Civil Aviation Flight University of China, Deyang, 618307, China

[2]East China Air Traffic Management Bureau, Xiamen Air Traffic Management Station, Xiamen, 361015, China

*Corresponding Author: Weijun Pan. Email: wjpan@cafuc.edu.cn

Received: 05 May 2023 Accepted: 10 August 2023 Published: 31 October 2023

ABSTRACT

This study aims to address the deviation in downstream tasks caused by inaccurate recognition results when applying Automatic Speech Recognition (ASR) technology in the Air Traffic Control (ATC) field. This paper presents a novel cascaded model architecture, namely Conformer-CTC/Attention-T5 (CCAT), to build a highly accurate and robust ATC speech recognition model. To tackle the challenges posed by noise and fast speech rate in ATC, the Conformer model is employed to extract robust and discriminative speech representations from raw waveforms. On the decoding side, the Attention mechanism is integrated to facilitate precise alignment between input features and output characters. The Text-To-Text Transfer Transformer (T5) language model is also introduced to handle particular pronunciations and code-mixing issues, providing more accurate and concise textual output for downstream tasks. To enhance the model's robustness, transfer learning and data augmentation techniques are utilized in the training strategy. The model's performance is optimized by performing hyperparameter tunings, such as adjusting the number of attention heads, encoder layers, and the weights of the loss function. The experimental results demonstrate the significant contributions of data augmentation, hyperparameter tuning, and error correction models to the overall model performance. On the Our ATC Corpus dataset, the proposed model achieves a Character Error Rate (CER) of 3.44%, representing a 3.64% improvement compared to the baseline model. Moreover, the effectiveness of the proposed model is validated on two publicly available datasets. On the AISHELL-1 dataset, the CCAT model achieves a CER of 3.42%, showcasing a 1.23% improvement over the baseline model. Similarly, on the LibriSpeech dataset, the CCAT model achieves a Word Error Rate (WER) of 5.27%, demonstrating a performance improvement of 7.67% compared to the baseline model. Additionally, this paper proposes an evaluation criterion for assessing the robustness of ATC speech recognition systems. In robustness evaluation experiments based on this criterion, the proposed model demonstrates a performance improvement of 22% compared to the baseline model.

KEYWORDS

Air traffic control; automatic speech recognition; conformer; robustness evaluation; T5 error correction model

1 Introduction

In the International Civil Aviation Organization (ICAO) document 4444 PAN-ATM, it is stated that the primary purpose of ATC is to prevent collisions between aircraft and between aircraft and obstacles in maneuvering areas and to expedite and maintain an orderly flow of air traffic. To achieve this objective, air traffic control officers (ATCOs) issue ATC instructions to aircraft based on the provisions of the Air Traffic Service Plan and other relevant documents. Therefore, ensuring the effective transmission of ATC instruction information is crucial for air traffic safety. In ATC, control instructions are primarily conveyed through pilot-controller voice communications (PCVCs) [1,2] indicates that digital automation is critical in addressing challenges in air traffic flow management (ATFM), such as safety, capacity, efficiency, and environmental aspects, and future Controller-Pilot Data Link Communication (CPDLC) will replace PCVCs. Currently, the Civil Aviation Administration of China (CAAC) is promoting intelligent air traffic management (ATM), which includes key technologies such as multimodal fusion, ASR, ATC instructions intent recognition, automatic response to ATC instructions, and semantic verification of ATC instructions [3]. Among these technologies, ASR systems aim to convert speech signals into text sequences for text-based communication purposes or device control [4]. In recent years, researchers have applied ASR technology in the ATC domain to address various existing issues, such as operational safety monitoring [5], reducing ATCOs' workload [6], and developing simulation interfaces [7,8] points out that when the ASR system has poor recognition performance, it can lead to deviations in the understanding of control instructions and increase controllers' workload. Hence, this paper proposes a context-aware speech recognition and understanding system for the ATC domain to reduce the error rate of automated systems in executing control instructions. Reference [9] presented an Assistant Based Speech Recognition (ABSR) for ATM. ABSR integrates an assistant system with the ASR system, providing contextual information to ASR and readjusting the generated results to improve the final accuracy. AcListant and its follow-up project AcListant-Strips have demonstrated that ABSR reduces the controller's mouse click time by three times [6] and reduces fuel consumption per flight by 60 liters [10]. The Horizon 2020 SESAR-funded project Machine Learning for Controller-Assisted Speech Recognition Models (MALORCA) [11] also adjusts the recognition results to improve recognition accuracy. However, the projects above utilize a traditional architecture where ASR models incorporate both an acoustic and a language model, which introduces complexity in the training process and hampers global optimization [12]. Due to these limitations and the advancement of deep learning techniques, more researchers have started exploring end-to-end (E2E) ASR. E2E ASR eliminates the need for designing multiple modules to map various intermediate states and employs a globally optimized objective function highly correlated with the final evaluation criteria, thereby seeking global optimization results [13,14]. The current main E2E models are based on Connectionist Temporal Classification (CTC) [15] and attention-based ASR models [16]. CTC is essentially a loss function that eliminates the need for explicit data alignment, allowing deep neural network models to be widely adopted in E2E ASR with improved performance. However, CTC is based on the assumption that labels in the output sequence are independent, limiting its ability for language modeling [17]. Attention mechanisms overcome this limitation by implicitly learning soft alignments between input and output sequences, and the encoding vectors are no longer restricted to fixed-length vectors, resulting in stronger encoding capabilities. Reference [18] indicated that CTC is superior to attention in terms of latency, computational complexity, and training difficulty, while attention has stronger language modeling capabilities than CTC. In addition to the modeling capabilities for input sequences, the performance of speech recognition systems is also influenced by the representation of audio features. Reference [19] used four local feature extraction blocks and one Long Short-Term Memory (LSTM) layer to learn the local and long-term correlations in the log Mel spectrogram of input speech samples, achieving robust speech emotion recognition. Reference [20] employed a hierarchical Convolutional Long Short-Term Memory (CnvLSTM) for further feature representation of input audio, improving the recognition performance for Speech Emotion Recognition (SER). Reference

[21] stated that efficient SER system performance depends on feature learning and proposed a two-stream deep convolutional neural network with an iterative neighborhood component analysis (INCA) for audio feature extraction, effectively enhancing the recognition system's performance. Recently, Transformer architecture based on Multi-Head Self-Attention (MHSA) has been widely applied in sequence modeling due to its ability to capture long-range interactions and high training efficiency [22]. Compared to Deep Convolutional Neural Network (DCNN) and LSTM, Transformer has stronger feature extraction capabilities, enabling effective handling of long sequence dependencies, but it is relatively weaker in extracting fine-grained local feature information.

2 The Work and Chapter Arrangement

Based on the characteristics of ATC speech, this paper considers the importance of global and local interactions in parameterizing the ATC speech recognition system. Therefore, this paper utilizes the Conformer encoder [23] to capture global and local information in ATC speech and achieve advanced feature representation of audio. In terms of modeling the audio feature sequence, this paper combines Attention with CTC, using the Attention mechanism to assist in achieving more accurate input-output sequence alignment, while CTC is utilized to reduce alignment time. To reduce the error rate in executing control instructions by automated systems, this paper employs the T5 language model to correct further and transcribe the recognized text. The transcription process aims to convert the Chinese representation of control instructions, including numbers and letters, into Arabic numerals and English letters, addressing the challenges of code-mixing in ATC speech recognition, enhancing the readability of recognized text, and improving the accuracy of automated system recognition. Finally, this paper proposes a new evaluation metric system and provides corresponding evaluation methods specifically designed to measure the robustness of ATC speech recognition systems. The proposed model architecture and evaluation criteria contribute to accelerating the application deployment of ASR technology in practical scenarios and provide insights for the development of automation in air traffic control. The remaining parts of this paper are arranged as follows: In Section 3, the paper analyzes the difficulties faced in applying speech recognition technology in this field. In Section 4, the article describes the strategies adopted to address the challenges identified in Section 3 and provides a detailed introduction to the proposed model architecture. Additionally, a set of evaluation metrics specifically designed to assess the robustness of ATC speech recognition systems and a comprehensive evaluation method for scoring the systems are presented. In Section 5, the paper conducts an experimental analysis of the hyperparameters that affect model robustness to determine the optimal parameter configuration. Furthermore, ablation experiments are designed to validate the effectiveness of the proposed improvement strategies. In addition, comparative experiments are conducted to demonstrate the proposed model's superiority further. Finally, robustness evaluation experiments of the ATC speech recognition system are performed to compare the robustness of different models. In Section 6, the article summarizes the main findings and provides prospects for future work.

3 Challenges

Due to the unique characteristics of the ATC field, applying ASR technology to this field poses certain difficulties [24]. Currently, implementing advanced ASR technology in the ATC field still faces the following main challenges [25].

(1) Difficulties in data collection and annotation.

Almost all state-of-the-art ASR models are built via a data-driven mechanism, so the quality of training samples significantly impacts the model's performance [26]. However, collecting sufficient training samples to develop a qualified speech recognition system is difficult due to safety and intellectual property concerns in the ATC field. Even groups related to air traffic cannot share ATC speech with other research institutions or companies. Additionally, annotating ASR training samples in the ATC field is a specialized task, and staff must learn a significant amount of necessary ATC knowledge to be competent. Therefore, collecting and annotating training samples is an expensive and laborious task. Table 1 shows some corpora collected from other fields and the ATC field.

Table 1: Summary of corpora related to ATC speech recognition [27]

Corpus	Language	Domain	Length (hour)	Access
LibriSpeech	English	Novels	960	Public
TED-LIUM3	English	TED talks	452	Public
Switchboard	English	Telephone	260	Public
THCHS30	Chinese	Newspapers	30	Public
AISHELL-V1	Chinese	Multi-domain	500	Public
AISHELL-V2	Chinese	Multi-domain	1000	Application
ATCSpeech	Chinese/English	Real ATC	59	Application
ATCOSIM	English	Simulated ATC	11	Public
LDC94S14	English	Airport	70	Paid
Airbus	English	Pilot	40	Unavailable

As seen from Table 1, due to the field's uniqueness, collecting and annotating sufficient samples to develop the required ASR system in the ATC field is extremely difficult, especially for Mandarin ATC speech corpora.

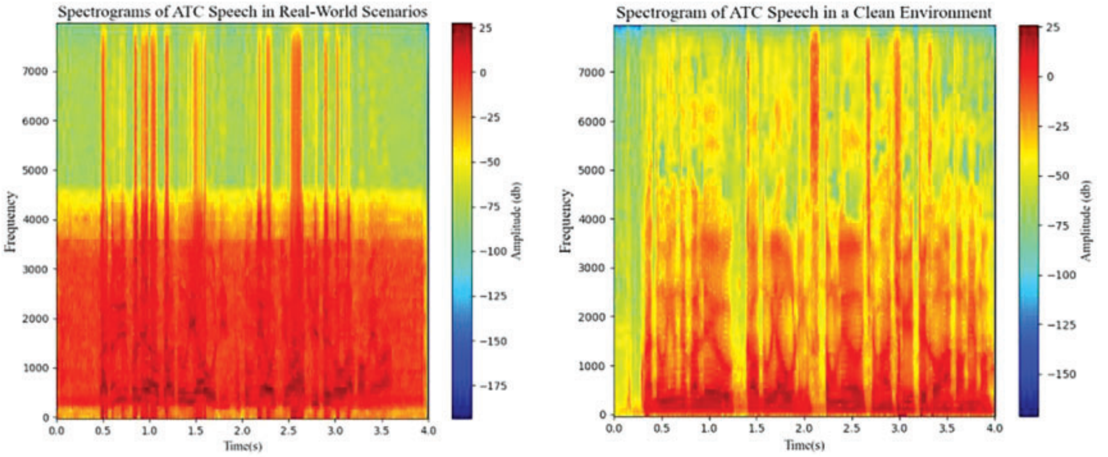
(2) Poor audio quality.

As ATCOs and pilots rely on radio communication to exchange information, ATCOs typically communicate with multiple pilots at the same frequency. During conversations, channel occupancy, changes in the conditions of electronic equipment, and disturbances in the environment surrounding the input device can all lead to the generation of current background noise. Fig. 1 shows the spectrograms of ATC speech in real-world and clean scenarios.

From Fig. 1a, it can be observed that the ATC speech in the real-world scenario is filled with a significant amount of electrical noise in the range of 100–4000 Hz. This results in a spectrogram with high energy distribution and no noticeable random frequency variation. In contrast, Fig. 1b displays the spectrogram of ATC speech in a clean environment without noise interference. The energy distribution is smoother than in Fig. 1a, with distinct spectral features with clear shapes.

(3) Speaking too fast.

The speed of ATC communication is generally faster than that of daily life due to the job's special nature. For example, when facing heavy traffic or during peak hours, ATCOs tend to speak faster unconsciously. Table 2 shows the speaking speed of ATCOs, where "Our ATC Corpus" refers to the dataset used in this paper.



(a)

(b)

Figure 1: The spectrograms of ATC speech in real-world and clean scenarios. (a) Spectrogram of control speech in a real-world scenario; (b) spectrogram of clean control speech

Table 2: Comparison of speaking speeds in different corpora

Language	Corpus	Mean (w/s)	Standard deviation (w/s)
Chinese	Our ATC corpus	4.5	1.09
	THCHS-30 [28]	3.48	0.47
English	ATCSpeech [29]	3.28	0.75
	LibriSpeech [30]	2.73	0.47

According to Table 2, it can be seen that the speaking speed of Chinese is higher than that of English. In addition, the speaking speed of ATC is more unstable, with a higher standard deviation than that of ordinary corpora.

(4) Code-Switching.

Code-switching refers to the phenomenon of mixing native language and other languages during communication between pilots and ATCOs [31]. To avoid communication ambiguity between ATCOs and crew members, the Civil Aviation Administration of China has developed a set of guidelines called “Radio Telephone Communications for Air Traffic Services” based on the guidelines of the ICAO to regulate radio communication in China. For example, the number 1 (yī) is pronounced as “yāo” and the number 7 (qī) is pronounced as “guǎi”. In addition, it is difficult to find a single Chinese character to correspond to the pronunciation of English letters. For instance, the letter A (alpha) is pronounced as “ai ou fe” in Chinese, requiring three Chinese characters to correspond, making the recognition results cumbersome.

(5) Lack of targeted evaluation criteria.

Currently, general metrics [32], such as WER and CER, are commonly used to evaluate the performance of ATC speech recognition systems. However, for Chinese ATC instructions, since the instructions have strict structural features, the information of ATC instructions is mainly contained in the keywords. Therefore, keywords are the core of ATC instruction content. However, the commonly used evaluation metrics for ATC speech recognition systems do not consider keyword information, and the results cannot reflect the model performance in detail.

(6) Difficult to deploy.

The quality of ATC speech recognition system is crucial to constructing intelligent ATC system, as errors in the recognition system will propagate to downstream tasks such as language understanding (LU) and directly affect these tasks. Although the current ASR systems perform well overall, they still face challenges when deployed in practice [33]. Therefore, further improving the accuracy of the ATC speech recognition system is a critical factor in promoting the deployment and application of ASR in the ATC field.

4 Methodology and Methods

4.1 Strategies to Address Challenges

To address the problem of difficult data collection and annotation, this paper adopts transfer learning to alleviate the problem of poor model performance due to small sample sizes. In addition, this paper addressed the challenges of poor audio quality and fast speech affecting model performance by using data augmentation techniques, which included noise suppression and speech rate adjustment to increase data diversity and improve model robustness. To tackle the problem of code-switching, this paper uses the pre-trained language model T5 to correct further and transcribe output results, which improved recognition accuracy and simplified the text's complexity by mapping Chinese to Arabic numerals and English letters, to some extent alleviating deployment difficulties. Finally, to address the lack of targeted evaluation criteria, this paper proposed a set of evaluation indicators and scoring methods based on keyword metrics for the ATC speech recognition system, which can comprehensively measure the system's performance.

4.2 Strategy Introduction

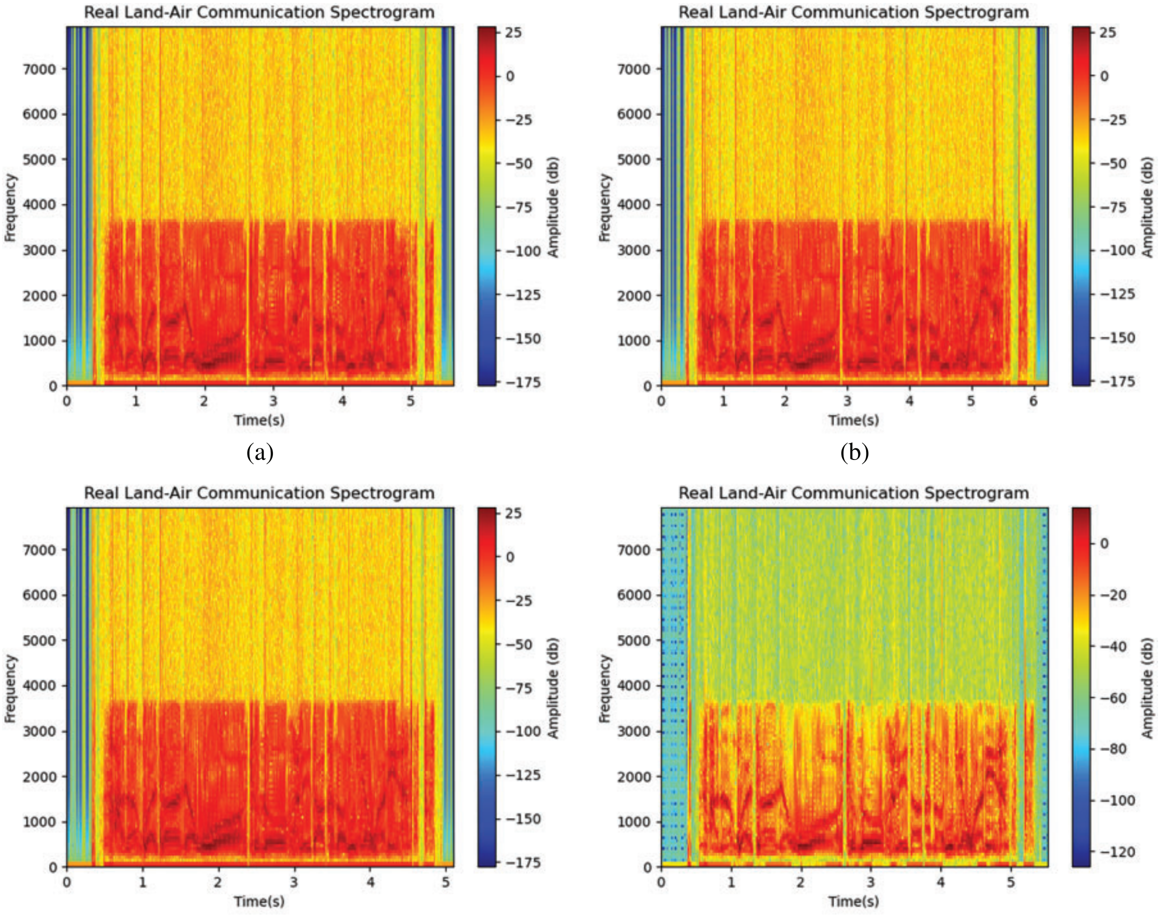
4.2.1 Data Augmentation

Speech data augmentation refers to generating new speech data using various transformation methods without changing the original content. Data augmentation can increase the diversity and quantity of training data, thereby improving the model's generalization ability and robustness and reducing overfitting. A study [34] found that using data augmentation can effectively enhance the performance of speech recognition systems for low-resource languages. Another study [35] investigated an audio-level speech enhancement method that directly processes the original signal, generating three versions of the original signal by changing the audio signal's speed, with speed factors of 0.9, 1.0, and 1.1. This low-cost and easy-to-use technique can effectively improve the robustness of ASR systems due to insufficient training data samples.

This paper employs two data augmentation methods: noise suppression and speed perturbation. Since there is significant noise in real-world ATC speech, this research did not use noise addition to expand the training samples. In the noise suppression method, the sound intensity of the target speech is correspondingly weakened, which can increase the robustness of the model under low sound intensity conditions. The noise suppression is implemented based on the trained U-shaped Network (U-Net) speech denoising model that will be discussed in Section 4.2.2, and the speed perturbation is implemented based on the ffmpeg toolkit in Python. Table 3 describes the dataset after data augmentation, and Fig. 2 displays four audio spectrograms, including the original audio spectrogram, the denoised audio spectrogram, and the spectrogram after speed perturbation.

Table 3: Description of the augmented dataset after data augmentation

Dataset	Number of data samples (item)	Total audio duration (hour)	Average rate (characters/s)
Original ATC dataset	13368 items	23 h	4.5
Noise suppression [36]	11000 items	16.14 h	4.5
Speed perturbation	22000 items	32.28 h	4.5



(a)

(b)

(c)

(d)

Figure 2: Four audio spectrograms. (a) Original audio spectrogram; (b) spectrogram of audio at 0.9× speed; (c) 1.1× speed audio spectrogram; (d) spectrogram of denoised audio

4.2.2 U-Net Noise Suppression

The mixed audio is transformed into a frequency domain spectrogram through Short-Time Fourier Transform (STFT). The ‘librosa.magphase’ function in Python is used to obtain the magnitude and the phase spectrogram. The U-Net model learns the mapping between the magnitude spectrogram of the noisy audio and the magnitude spectrogram of the clean audio through supervised learning. The restored magnitude spectrogram and the original phase spectrogram are multiplied element-wise to obtain the restored frequency domain spectrogram. Finally, the Inverse Short-Time Fourier Transform (ISTFT) is applied to convert it back to the time domain waveform, resulting in denoised audio.

1) The mixed audio is transformed into the frequency domain spectrum through the STFT, and the ‘librosa.magphase’ function is used to obtain the magnitude spectrum and phase spectrum. The phase spectrum is preserved, and the magnitude spectrum is fed into the U-Net encoder for feature extraction. The conversion formula between the spectrum and the time-domain samples of the mixed audio is as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n] \times e^{-\frac{j2\pi kn}{N}}, k = 0, 1 \ldots K - 1 \tag{1}$$

where N is the total number of time-domain samples, K is the total number of frequency components, $X[k]$ represents the Fourier spectrum value of the kth point, and $x[n]$ denotes the value of the nth time-domain sample.

2) The U-Net encoder consists of five CNN blocks, where the input spectrogram undergoes a series of convolutional and pooling operations to be encoded into advanced feature representations. This process can be expressed by the following formula:

$$Z = ConvolutionalLayers\,(X) \tag{2}$$

where *ConvolutionalLayers* represents the convolutional layer operations in the encoder, X represents the input vector, and Z represents the vector of encoded feature representations.

3) The U-Net decoder remaps the encoded feature representation vector back to an output with the same size as the input spectrogram through a series of upsampling and convolution operations. The specific formula can be represented as:

$$Z' = UpsamplingLayers\,(Z) \tag{3}$$

$$X' = ConvolutionalLayers\,(Z') \tag{4}$$

where *UpsamplingLayers*() represents the upsampling operations in the decoder, used to restore the size of the feature maps to a larger scale. *ConvolutionalLayers*() represents the convolutional layer operations in the decoder, used to process the upsampled feature maps further, increase the number of channels, and capture more detailed information.

4) The output of the decoder is the denoised magnitude spectrogram, which needs to be combined with the phase spectrogram and transformed into the audio form using the ISTFT. The specific formula is as follows:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \times e^{\frac{j2\pi kn}{N}} \tag{5}$$

where $x[n]$ represents the value of the nth sample point in the time-domain signal, $X[k]$ represents the value of the kth spectral coefficient in the frequency-domain signal obtained by element-wise multiplication of X' and the phase spectrum values.

5) The output time-domain signal is overlapped and added until all frames are processed, resulting in the restored denoised audio signal.

4.2.3 Error Correction Model

One way to improve the accuracy of speech recognition output is to use a language or neural network model to reorder different output hypotheses and select the one with the highest score for downstream tasks. NLC [37,38] aims to use a neural network architecture to map an output sentence $X = (x_1, \ldots, x_T)$ containing errors to a corrected sentence $Y = (y_1, \ldots, y_T)$ to eliminate the mistakes introduced by ASR systems. Based on this strategy, this paper used the pre-trained language model T5 [39] to correct and transcribe the output of ASR, improve the accuracy of ASR output, simplify output text, and promote the deployment of ASR models in the ATC domain.

T5 is a neural network architecture based on the Transformer, developed by Google for various natural language processing (NLP) tasks such as text generation, machine translation, text summarization, and question answering. Its training utilizes large-scale unsupervised pre-training methods, allowing the model to learn general language representations and only requiring fine-tuning for specific tasks to perform well.

How to collect error correction data to train the model is a key issue for a correction model. Many researchers have proposed different strategies to address this issue. Some work focuses on obtaining output from ASR systems to construct training data with specific sources such as N-best lists [40], Word Confusion Networks (WCNs) [41], and lattice [42]. Since the T5 model is pre-trained on a large-scale text corpus through unsupervised learning, the model can quickly adapt to plain text input, such as N-best lists, rather than more complex representations, such as lattices or WCNs. Therefore, this paper chooses the N-best list output by the ASR system as the training data for the T5 correction model. For the choice of N, when N equals 1, the model's input is the best hypothesis generated by the ASR model. The reduced error information in the best hypothesis significantly affects the training effectiveness of the correction model. Therefore, data augmentation techniques are needed to enable the correction model to generalize to speech data that the ASR model cannot see. Reference [43] conducted experiments on the value of N, and the results showed that the correction model obtains richer information from diversified inputs with larger N, thereby achieving more accurate error detection and correction. This paper finds that when N is greater than 3, the model output contains more invalid information. Therefore, this paper chooses N = 3 and uses speed perturbation and noise suppression training samples to generate training data for the correction model. This paper uses regularization methods to simplify the output labels so that the correction model can transcribe the text further while correcting it. In addition, to expand the generality of language model transcription correction, this paper randomly generates and annotates control instruction text based on control instruction rules. Finally, this paper obtains a total of 54,907 training samples for the T5 correction model, and the total training time of the model was 16 h.

4.2.4 Evaluation Metrics and Methods

Currently, the evaluation metrics commonly used for Chinese ASR speech recognition systems are CER and Real-Time Factor (RTF). The specific calculation formulas are as follows:

$$CER = \frac{I + D + S}{N} \times 100\% \tag{6}$$

where N represents the total length of characters in the reference transcripts, and the symbols I, D, and S, respectively represent the number of insertion deletion, and substitution operations.

$$RTF = \frac{T_{\mathrm{d}}}{T_s} \tag{7}$$

where T_s represents the decoding time and T_d represents the duration of the audio.

Reference [44] pointed out that WER is an objective method for evaluating the quality of Automatic Speech Recognition (ASR). However, WER has certain limitations regarding its applicability across different domains. Therefore, a Semantic WER (SWER) is proposed to address the limitations of WER in terms of usability. The ICAO standard mentioned in [45] states that ATCO instructions must start with aircraft identification (ACID) to identify the communicating aircraft, and pilot instructions must end with their ACID to distinguish them from ATCO instructions, indicating that control instructions have significant structural features. A correct control instruction contains several essential elements, such as the call sign, action instruction, and action parameters. The instruction will be considered invalid if any of these elements need to be corrected. Based on these elements' importance, this paper refers to them as keyword information and proposes a keyword-based evaluation metric system to measure the performance of control speech recognition systems from an ATC perspective. This paper divided the information contained in control instructions into three parts: call sign, action instruction, and action parameters. This paper defines that recognition errors outside these parts will not cause control instructions to fail. For example, "Shunfeng 6954, contact the tower on 123.5, goodbye" was recognized as "Shunfeng 6954, contacting the tower on 123.5". The word "goodbye" here is not an action instruction, so omitting it will not affect the instructions. The call sign consists of the airline abbreviation and flight number; the action instruction is the action contained in the ATC instruction, such as climb, descend, maintain, etc.; and the action parameters refer to the key supplementary information of the instruction action, including speed, altitude, heading, and waypoints. The specific calculation formula is as follows:

$$CSA = \frac{1}{N}\sum_{i}^{N} g\left(i\right) \tag{8}$$

$$AIA = \frac{1}{\mathrm{N}}\sum_{i}^{N} q\left(i\right) \tag{9}$$

$$APA = \frac{1}{N}\sum_{i}^{N} h\left(i\right) \tag{10}$$

The above equation shows that call sign accuracy (CSA) represents the accuracy of call signs, action instruction accuracy (AIA) represents the accuracy of action instructions, and action parameter accuracy (APA) represents the accuracy of action parameters. N represents the number of samples to be tested. In addition, $g(i)$, $q(i)$, and $h(i)$ respectively represent the feature functions of call signs, action instructions, and action parameters with the following formulas:

$$g\left(i\right), q\left(i\right), h\left(i\right) \begin{cases} 1 & pred_i = truth_i \\ 0 & \quad\text{else} \end{cases} \tag{11}$$

Finally, these three keywords work together to affect sentence accuracy (SA), which means that any error in any of these keywords will result in an error in the entire sentence. Therefore, the definition of SA is as follows:

$$SA = \frac{1}{N}\sum_{i}^{N} T\left(i\right) \tag{12}$$

In which N represents the number of samples to be tested, and $T(i)$ is the feature function for sentence accuracy. The specific calculation method is as follows:

$$T(i) = \begin{cases} 1 & g(i) = q(i) = h(i) = 1 \\ 0 & else \end{cases} \tag{13}$$

Finally, based on sentence accuracy, this paper has defined a three-level evaluation system to measure the performance of an ATC recognition system. The evaluation index system is shown in Fig. 3, where gender test data is a first-level evaluation indicator, test data with different rates and combinations of signal-to-noise ratios are second-level evaluation indicators, and sentence accuracy is a third-level evaluation indicator.

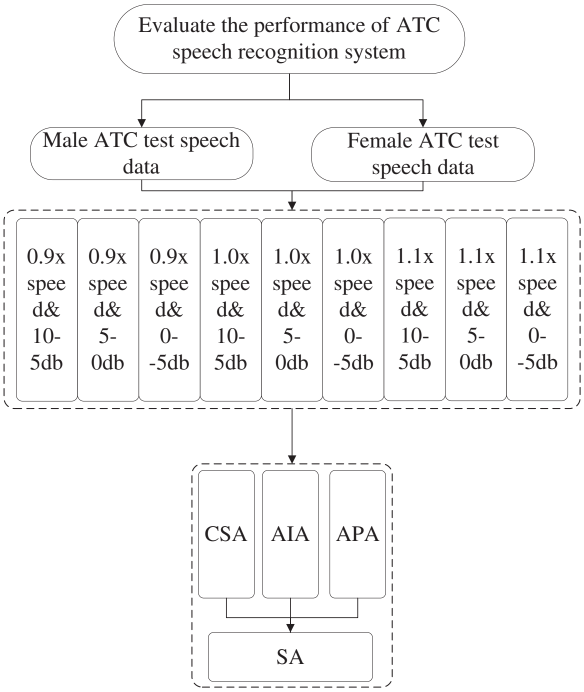


Figure 3: Evaluation index system for ATC speech recognition system

Based on the established evaluation indicator system, this paper provides an ATC speech recognition system evaluation method called Critic Weighting Method—VlseKriterijumska Optimizacija I Kompromisno Resenje (Critic-VIKOR). This paper first uses the Critic method to determine the weight of each indicator. Then it uses the VIKOR comprehensive evaluation method to score and rank the ATC speech recognition systems. The Critic method and the VIKOR method are described as follows:

(1) The Critic comprehensive weighting method is an objective weighting method. Its objective basis for weighting comes from the comparison strength between evaluation indicator data and the conflict between indicators. The comparison strength refers to the size of the difference in values of each scheme for the same indicator, which reflects the information contained in the indicator, and is often measured by the standard deviation. Conflict is based on the correlation between indicators. If two indicators have a strong positive correlation, it means that the conflict between the two indicators is low. The specific calculation process is as follows: Assuming there are n evaluation objects and m

evaluation indicators, the first step in the calculation is to standardize the indicators. For positive indicators, the standardization formula is shown as formula (14).

$$X_{ij} = \frac{X_{ij} - X_{\min}}{X_{\max} - X_{\min}} \tag{14}$$

For negative indicators, the standardization formula is shown as formula (15).

$$X_{ij} = \frac{X_{\max} - X_{ij}}{X_{\max} - X_{\min}} \tag{15}$$

Here, X_{ij} represents the value of the jth indicator for the ith evaluation object, $X_{min} = arg\ min\{X_{ij}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, m\}$, $X_{max} = arg\ max\{X_{ij}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, m\}$.

The second step in the calculation is to compute the comparative strength of the indicators, which is measured by the standard deviation. The specific calculation formula is shown as formula (16).

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} \left(X_{ij} - \overline{X}_j\right)^2}{n - 1}} \tag{16}$$

Here, σ_j represents the comparative strength of the jth indicator, and $\bar{X}_j$ represents the mean value of the jth evaluation indicator for all evaluation objects.

The third step in the calculation is to compute the conflict or correlation between indicators using the specific formula shown in formula (17).

$$S_j = \sum_{k=1}^{m} \left(1 - r_{jk}\right) \tag{17}$$

Here, S_j represents the degree of contradiction of the jth indicator, and r_{jk} represents the Pearson correlation coefficient between the jth and kth evaluation indicators.

The fourth step in the calculation is to compute the information content of the indicators, which is calculated using the specific formula shown in formula (18).

$$C_j = \sigma_j S_j \tag{18}$$

Here, C_j represents the information content of the jth indicator. The larger the C_j, the greater the contribution of the jth evaluation indicator in the overall evaluation indicator system.

The final objective weight calculation formula is shown as formula (19).

$$W_j = \frac{C_j}{\sum_{j=1}^{m} C_j} \tag{19}$$

Here, W_j represents the objective weight of the jth evaluation indicator.

(2) VIKOR was proposed by Opricovic and Tzeng. This evaluation method can simultaneously consider maximizing group utility, minimizing individual regret, and incorporating decision makers' subjective preferences, thus having higher ranking stability and credibility. The specific calculation process is shown below.

The first step of the calculation is data normalization, with the specific formulas shown in formulas (14) and (15).

The second step is to determine the positive and negative ideal solutions with the specific formulas shown in formulas (20) and (21).

$$r^{+} = \left\{ \arg \max_{i=1,2,\ldots,n} (X_{1i}), \arg \max_{i=1,2,\ldots,n} (X_{2i}), \ldots, \arg \max_{i=1,2,\ldots,n} (X_{mi}) \right\} \tag{20}$$

$$r^{-} = \left\{ \arg \min_{i=1,2,\ldots,n} (X_{1i}), \arg \min_{i=1,2,\ldots,n} (X_{2i}), \ldots, \arg \min_{i=1,2,\ldots,n} (X_{mi}) \right\} \tag{21}$$

where r^{+} represents the positive ideal solution, which is a set consisting of the maximum values of each indicator, and r^{-} represents the negative ideal solution, which is a set consisting of the minimum values of each indicator.

The third step of the calculation is to compute the group utility value U_i and individual regret value R_i, with the specific formulas shown in formulas (22) and (23).

$$U_i = \sum_{i=1}^{n} w_j \left(\frac{r_j^{+} - X_{ij}}{r_j^{+} - r_j^{-}} \right) \tag{22}$$

$$R_i = \arg \max_{j=1,\ldots,m} \left\{ w_j \frac{r_j^{+} - X_{ij}}{r_j^{+} - r_j^{-}} \right\} \tag{23}$$

where w_j is the weight of the jth evaluation criterion, $r_j{}^{+}$ is the positive ideal solution for the jth evaluation criterion, $r_j{}^{-}$ is the negative ideal solution for the jth evaluation criterion, and X_{ij} represents the value of the jth indicator of the ith evaluated object after normalization.

The fourth step of the calculation is to compute the compromise solution value Q_i, with the specific formulas shown in formula (24) through (28).

$$U^{+} = \arg \max_{i=1,2\ldots,n} (U_i) \tag{24}$$

$$U^{-} = \arg \min_{i=1,2\ldots,n} (U_i) \tag{25}$$

$$R^{+} = \arg \max_{i=1,2\ldots,n} (R_i) \tag{26}$$

$$R^{-} = \arg \min_{i=1,2\ldots,n} (R_i) \tag{27}$$

$$Q_i = \beta \frac{U_i - U^{-}}{U^{+} - U^{-}} + (1 - \beta) \frac{R_i - R^{-}}{R_{+} - R^{-}} \tag{28}$$

Here, U^{+}, U^{-}, R^{+}, and R^{-}, respectively represent the maximum utility value, minimum utility value, maximum individual regret value, and minimum individual regret value. The parameter β represents the decision-making mechanism coefficient, which ranges from 0 to 1 and is set according to the decision maker's preference. The default value is 0.5. When this value is greater than 0.5, it indicates a preference for risk-taking, while when it is less than 0.5, it indicates a more conservative choice. In this paper, β is chosen to be 0.5.

4.3 Model Architecture Introduction

This paper uses the Conformer model as the encoder of the speech recognition error correction system, whose architecture is shown in Fig. 4.

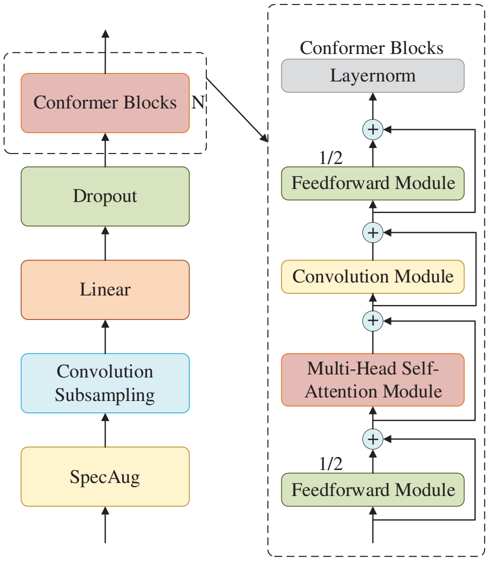


Figure 4: Conformer encoder architecture

In this architecture, the Conformer Block comprises three modules: the Feedforward Module, the Multi-Head Self-Attention Module, and the Convolution Module. There is a Feedforward layer before and after the Conformer Block, and the Multi-Head Self-Attention Module and Convolution Module are sandwiched in the middle. The Feedforward layer adopts a half-step residual connection and performs layer normalization at the end of the module. Each module uses a residual unit to concatenate the convolution and attention for enhanced effect. In the Conformer Block, the two Feedforward layers contribute half of the value each, known as a half-step Feedforward Neural Network (FFN). The input x_i and output h_i of the ith Conformer Block can be described using the following mathematical formula:

$$\begin{cases} \widetilde{x}_i = x_i + \dfrac{1}{2} FFN\left(x_i\right) \\ x'_i = \widetilde{x}_i + MHSA\left(\widetilde{x}_i\right) \\ x''_i = x'_i + Conv\left(x'_i\right) \\ h_i = Layernorm\left(x_i + \dfrac{1}{2} FFN\left(x_i\right)\right) \end{cases} \tag{29}$$

In the above equation, *FFN* refers to the Feedforward module, *MHSA* refers to the multi-head self-attention module, *Conv* refers to the convolutional module, *Layernorm* represents layer normalization, and residual connections are used between each module.

For the decoding part of the model, this paper adopted a CTC/Attention joint decoding strategy during the model training process. The main reason for using this strategy is that although attention-based ASR systems can achieve relatively high recognition accuracy, they do not perform well in small sample situations. In addition, the alignment of the attention mechanism has no specific order, which brings certain difficulties to the model training. In contrast, the from-left-to-right algorithm in CTC can align the input and the output sequence in chronological order. Considering both advantages, this paper combined them for decoding during the training process. In this paper, the main role of the attention decoding mechanism is to assist the CTC decoder so that the CTC decoder can learn a more accurate alignment relationship between audio features and output characters without slowing down the decoding speed.

The ASR model training architecture is shown in Fig. 5, where λ represents the weight size.

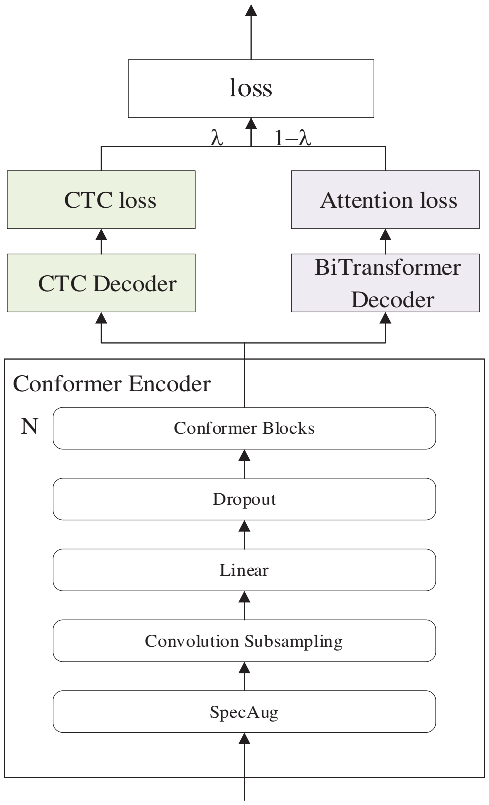


Figure 5: The training architecture of conformer CTC/attention

In Fig. 5, the loss function of CTC is the negative log probability sum of all labels, and the CTC network can be trained through backpropagation. CTC decoding includes two sub-processes: path probability calculation and path aggregation. In addition, CTC introduces a new blank label during the decoding process, indicating no output for this frame. The specific calculation process is as follows: assuming that the input feature sequence is $X = \{x_1, x_2, \ldots, x_T\}$ and the target sequence is $Y = \{y_1, y_2, \ldots, y_T\}$ after the input feature sequence is output through the Softmax layer, the network output is $P(q|X)$, where $q = \{q_1, q_2, \ldots, q_T\}$ and q_T is the output at time T. The probability of the label sequence Y is the sum of all path probabilities, and the specific calculation is shown in formulas (30)–(32).

$$p\left(q|X\right) = \prod_{t=1}^{T} p\left(q_t|X\right) \tag{30}$$

$$P\left(Y|X\right) = \sum_{Y \in \Gamma(q)} p\left(q|X\right) \tag{31}$$

$$CTC_{loss} = -\ln P\left(Y|X\right) \tag{32}$$

In the above formulas, q is the target sequence with the blank label, $p(q_t|X)$ is the probability of outputting q_t given the input sequence X, and $p(q|X)$ is the probability of outputting the target sequence q with blank labels given the input sequence X. Since the same label sequence may correspond to multiple paths with blank labels, $\Gamma(q)$ is mainly used to map the output target sequence with blank labels to the target sequence Y, and it is a many-to-one mapping function.

In the model prediction, this paper uses CTC beam search as the decoding strategy and adds the error-correcting language model T5 to the output layer to correct and further transcribe the decoding results. The final prediction model architecture is shown in Fig. 6.

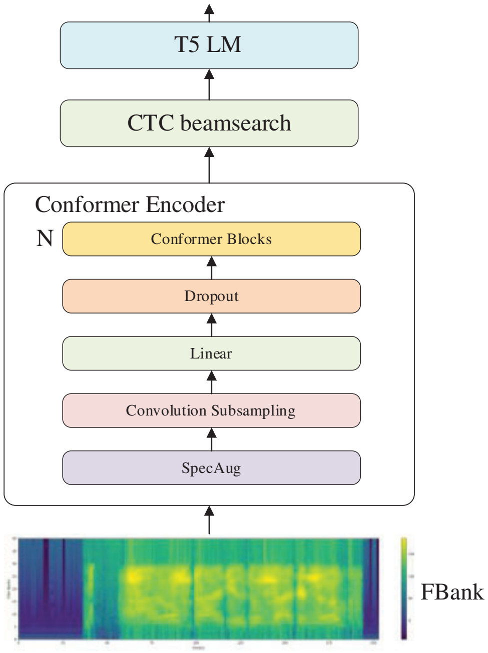


Figure 6: CCAT model architecture

As shown in Fig. 6, the Conformer encoder receives Fbank features as input. A higher dimensionality of Fbank features provides the encoder with more frequency information. Therefore, this paper adopts 80-dimensional Fbank features. The input features undergo a series of compression and transformation operations in the Conformer encoder to obtain advanced representations of audio features, as illustrated in Fig. 4. The CTC beam search decoder takes the audio feature representations outputted by the Conformer encoder and maps them to text sequences while considering repeated and blank labels in the text sequence. The beam search algorithm searches for the most probable text sequence by balancing high-probability labels and merging repeated labels to generate recognition results. The T5 model receives the recognition results from the CTC beam search decoder and generates the final recognition text through encoding, inference, and post-processing of the input. After these processing steps, the obtained recognition text exhibits concise representation and high recognition accuracy. Regarding model initialization, this paper utilizes the XavierUniform initialization method from the Paddle framework for the Conformer encoder. For the initialization of CTC beam search, the parameters are set through configuration files, including a language model (LM) coefficient of 2.2, a word count (WC) coefficient of 4.3, a beam search width of 300, a pruning probability of 0.99, and a maximum pruning value of 40. For the initialization of the T5 language model, the pre-trained model's weights are used as the initialization parameters, accelerating the model training. The specific configuration of model parameters will be introduced in the next section.

5 Experiment

5.1 Introduction to Experimental Data

The dataset used for model pre-training consists of over 10,000 h of labeled speech from the WenetSpeech corpus [46], which is currently the most extensive open-source Mandarin speech corpus constructed by multiple units such as Northwestern Polytechnical University. The transfer learning dataset comprises 7,831 control recordings and 5,537 control simulation training recordings from China's Central-South Air Traffic Management Bureau and the Civil Aviation Flight University of China. There are 13,368 recordings with a combined duration of 23 h. The pre-training dataset includes 5,534 Chinese characters and four special characters <blank>, <unk>, <space> and <eos>. In the transfer learning dataset, there are a total of 543 characters, including three unique characters: <blank>, <unk> and <eos>. To improve the model's generalization ability, this paper uses data augmentation techniques to augment the transfer learning dataset, resulting in 46,368 extended speech recordings totaling 71.42 h. Among them, 44,000 recordings are used for training, 868 for testing, 496 for validation during the training process, and 1,004 for evaluating the robustness of the ATC speech recognition system.

5.2 Experimental Platform and Model Configuration

The experiments were conducted on a Windows operating system. The computer configuration is as follows: Intel Core i5-8400 processor, 56 GB of RAM, NVIDIA RTX 4090 24 GB graphics card, 250 GB SSD, and a 3.6 TB HDD. The PaddlePaddle framework was used to build the neural network models. The specific model hyperparameters configuration is shown in Tables 4 and 5.

Table 4: Conformer CTC/attention model training hyperparameters configuration

Name	Setting
Input feature	Fbank (80 dimensions)
Number of attention heads	8
Output feature dimensions of conformer	512
Conformer blocks	12
BiTransformer blocks	3
CTC weight	0.5
CTC linear output size	543
Optimizer	Adam
Scheduler	WarmupLR
Batch size	16
Number of training epochs	100

Table 5: T5 model training hyperparameters setting

Name	Setting
Input feature	Text vector (Text sequence length, 768)
Number of attention heads	12

(Continued)

Table 5 (continued)

Name	Setting
Encoding dimension	768
Maximum padding length	128
Transformer blocks	12
Top_k	5
Optimizer	Adam
Scheduler	WarmupLR
Number of training epochs	100

5.3 Experimental Validation of the Proposed ASR Model

5.3.1 Model Hyperparameters Experiment

This paper experiments with the CCAT model hyperparameters using 11,000 ATC training data. The experiments were conducted without transfer learning and data augmentation techniques. The purpose of the experiments was to explore through sensitivity analysis which hyperparameters are vital factors influencing model robustness and to determine the optimal configuration of the hyperparameters. The experimental results are shown in Tables 6–8. In Table 6, λ represents the proportion of CTC loss in the training loss, and C_Size means the size of the CTC classifier.

Table 6: Model performance table under different values of λ and C_Size

Experimental ID	Parameter settings	CER	RTF
0	$\lambda = 1.0$, C_Size = 5537	5.98%	0.042
1	$\lambda = 1.0$, C_Size = 543	5.60%	0.023
2	$\lambda = 0.5$, C_Size = 543	5.45%	0.026
3	$\lambda = 0.3$, C_Size = 543	7.33%	0.032
4	$\lambda = 0.0$, C_Size = 543	8.72%	0.410

Table 7: Table of model performance with different attention heads

Experimental ID	Attention heads	CER	RTF
0	1	8.78%	0.041
1	4	7.31%	0.039
2	8	5.45%	0.026
3	16	6.34%	0.031

Table 8: Table of model performance with different conformer blocks

Experimental ID	Conformer blocks	CER	RTF
0	4	6.41%	0.031
1	8	7.06%	0.033
2	12	5.45%	0.026

In Table 6, this paper uses CER and RTF to measure the impact of different values of λ and C_Size on model performance. Through the comparison between Experiment 0 and Experiment 1, it can be concluded that reducing the size of the CTC classifier can improve model performance when the weight λ is fixed. This is because a smaller word list can reduce homophone errors in Chinese, and the traversal time for a smaller word list is shorter, leading to less decoding time for the model. Through Experiment 0–4, it can be seen that the optimal model performance is achieved when λ is 0.5. This is because when λ is 1.0, the model training only considers the CTC loss and not the attention loss, which prevents the model from using the alignment learned by the attention mechanism to assist the CTC in achieving more accurate alignment. When λ is 0.0, the model training only considers the attention loss and not the CTC loss, which can result in long decoding times for the model. In addition, due to the small sample training, the performance of the attention-based ASR model cannot be fully realized, resulting in poorer performance.

Table 7 shows the impact of the number of attention heads on model performance, with λ set to 0.5, C_Size set to 543, and the number of Conformer blocks set to 12. This indicates that an appropriate number of attention heads helps enhance the model's expressive power and ability to model complex speech and language structures. Additionally, reducing the number of attention heads and having excessive attention heads increase the RTF. This is mainly because an inadequate number of attention heads leads to the generation of more erroneous information by the model, thereby increasing the processing time for audio.

Table 8 shows the impact of the number of Conformer blocks on model performance when the attention heads are set to 8, λ is 0.5, and C_Size is 543. The results indicate that the model achieves optimal performance when the number of Conformer blocks is 12. Additionally, the model performs better with 4 Conformer blocks compared to 8 Conformer blocks. This suggests that having an encoder with an appropriate depth facilitates the learning of high-level audio representations.

Lastly, from the experimental results of the three tables, it can be observed that changing the attention heads has a significantly more pronounced impact on model robustness. Increasing the attention heads appropriately for the given dataset can help the model better learn the dependencies between the data.

5.3.2 Model Optimization Strategy Validation

This paper conducted separate validations for each strategy to verify the effectiveness of the model performance improvement strategies. Strategies 0–4 listed the improvement strategies used by the model, and the experimental results are shown in Table 9. In Table 9, CCA stands for the Conformer CTC/Attention ASR model, and CCAT is the abbreviation for the Conformer CTC/Attention T5 ASR model. CCA0 represents the CCA model based on strategy 0, and CCAT is the model with the T5 error correction model added on top of CCA3.

Table 9: Model performance under different improvement strategies

Experimental ID	Name	CER	RTF
0	CCA0	5.45%	0.026
1	CCA1	4.51%	0.024
2	CCA2	4.47%	0.023
3	CCA3	3.51%	0.025
4	CCAT	3.44%	0.59

Strategy 0: This paper does not use transfer learning, data augmentation, and error correction strategies to train the CCA model.

Strategy 1: This paper only uses the data augmentation strategy and does not use transfer learning and error correction strategies to train the CCA model.

Strategy 2: This paper only uses the transfer learning strategy and does not use data augmentation and error correction strategies to train the CCA model.

Strategy 3: This paper uses transfer learning and data augmentation strategies but does not use the error correction strategy to train the CCA model.

Strategy 4: Based on Strategy 3, this paper uses the T5 model to correct the recognition results of the CCA model.

From Table 9, all the improvement strategies, including pre-training, data augmentation, and error correction, can significantly improve the performance of ASR. Generally, pre-training can obtain specific speech representations for a specific dataset, thus speeding up the model-tuning process. At the same time, data augmentation can increase the size and diversity of the data in a supervised optimization process. Both of these strategies can improve the modeling accuracy of supervised ASR training. From Experiment 1 and Experiment 2, both pre-training and data augmentation are beneficial for enhancing ASR performance. Compared with Experiment 0, Experiment 1 reduced CER by 0.94% through data augmentation, while Experiment 2 reduced CER by 0.98% through pre-training. It can be seen that pre-training is more helpful for improving model performance than data augmentation. In addition, Experiment 3 shows that the combination of data augmentation and transfer learning strategy can further improve the model performance, with CER reduced by 1.94% compared to Experiment 0. Finally, Experiment 4 results show that adding an error correction model can further reduce the model's CER, but the performance improvement comes at the cost of lowering the transcription speed of the model. Nevertheless, the simplification of transcription results by the CCAT model reduces the reading time from the result end.

5.3.3 Model Comparison

For the comparative experiments in this section, this paper selects three baseline models, Deepspeech2, DCNN, and Squeezeformer, to compare and verify with the CCAT model. The evaluation metrics for the models are still CER and RTF, and each model's information and test results are shown in Table 10. The training process loss diagram is shown in Fig. 7.

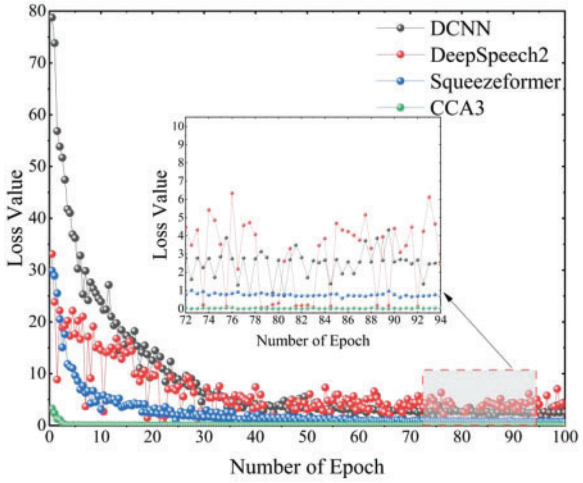


Figure 7: Comparison of training loss among different models

Table 10: Comparison results of different models

Experimental ID	Name	CER	RTF	Model parameters (Million)
0	DCNN	7.50%	0.075	1.47 M
1	DeepSpeech2	7.13%	0.025	40.77 M
2	Squeezeformer	7.08%	0.032	40.68 M
3	CCAT	3.44%	0.59	329 M

According to Table 10, it can be seen that the CCAT model has the lowest CER of 3.44%, which is a 4.06% reduction compared to DCNN. Among the baseline models, DCNN performed the worst, followed by DeepSpeech2 and Squeezeformer. Regarding model parameters, the CCAT model has the highest number of parameters, which means it has a more robust representation ability and better generalization.

In Fig. 7, CCA3 represents the CCA model using transfer learning and data augmentation strategies. From Fig. 7, it can be seen that the CCA3 model has a slight initial loss, and excellent results can be achieved by only fine-tuning the parameters in subsequent training. In addition, it can be observed from Fig. 7 that the loss values of the DCNN model and the DeepSpeech2 model still fluctuate significantly in the 80th to 100th epochs. This may be because these models were not pre-trained and have smaller model parameters, resulting in poor robustness. The relatively stable loss value of the CCA3 model reflects, to some extent, the impact of model parameter size and training strategy on model robustness.

In addition, two open-source corpora, AISHELL-1 and LibriSpeech, were used to validate the generalization of the proposed architecture. AISHELL-1 is a Mandarin Chinese corpus with a total duration of 178 h, comprising approximately 150 h of training data, 10 h of testing data, and 18 h of validation data. LibriSpeech, on the other hand, is an English corpus consisting of various training subsets such as train-clean-360, train-clean-100, and train-other-500, along with test subsets including test-clean and test-other, and validation subsets dev-clean and dev-other. The evaluation metrics used are CER and WER, and the evaluation results are presented in Table 11.

Table 11: Performance comparison of models on open source datasets

Experimental ID	Name	(Aishell-1) CER	(LibriSpeech) WER
0	DCNN	9.26%	18.40%
1	Deepspeech2	8.95%	15.17%
2	Squeezeformer	4.65%	12.94%
3	CCAT	3.42%	5.27%

Table 11 shows that the proposed model architecture consistently outperforms the baseline models on the Chinese and English datasets. This further confirms the robustness of the model, demonstrating its superior performance.

5.4 Robustness Verification of Models

This paper establishes a new evaluation index system to rank the robustness of various models, and the specific evaluation index system is shown in Fig. 4. This paper uses the VIKOR method to score the robustness of each ATC speech recognition system. The third-level indicators in the model evaluation index system include CSA, AIA, and APA. Since any error in these three indicators will lead to a wrong sentence, this paper uses SA to represent the third-level indicators of the evaluation. The second-level indicators include three categories of test speech data: 0.9x, 1.0x, and 1.1x, and each contains three different quality test speech data: 10 to 5 dB, 5 to 0 dB, and 0 to −5 dB. Finally, there are nine categories of second-level indicators, and their weights are determined using the Critic objective weighting method. The first-level indicators are male and female ATC test speech data, and this paper considers that the importance of males and females is the same.

The robustness testing data consists of 1008 samples, with 504 recordings of test speech from male control trainees in a noise-free environment and 504 recordings from female control trainees in a noise-free environment. The clean test speech this paper selected has a signal-to-noise ratio of 10 to 5 dB, and this paper used this as a baseline to construct test speech with different signal-to-noise ratios by extracting various noise audio from frontline control speech. The final ranking of the model's robustness is obtained by linearly weighting the scores obtained from male and female test results. This paper selected three speech rates as indicators: 1.1x, 1.0x, and 0.9x. Additionally, inspired by [47], the test speech used in this paper's experiments has signal-to-noise ratios ranging from 10 to −5 dB, 5 to 0 dB, and 0 to −5 dB. Among them, the cumulative bar charts of recognition results of different models under different indicators in male test data are shown in Figs. 8–10. For female test data, the recognition results of different models under different indicators are shown in radar charts in Figs. 11–13. In Figs. 8–10, the three bar charts corresponding to each model represent the results with signal-to-noise ratios of 10 to −5 dB, 5 to 0 dB, and 0 to −5 dB, respectively.

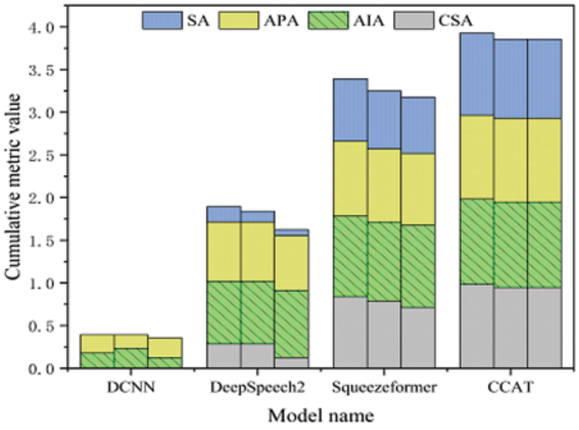


Figure 8: The cumulative bar chart under 0.9× speed

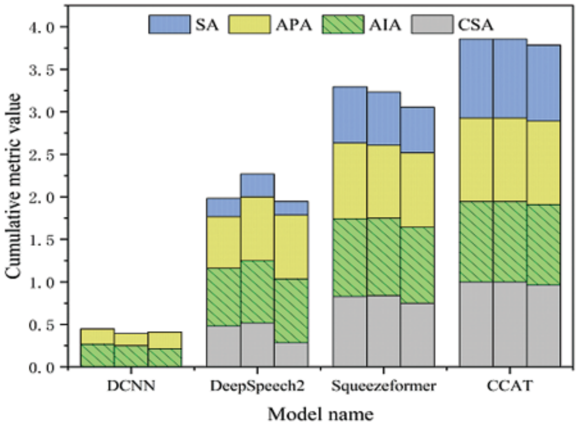


Figure 9: The cumulative bar chart under 1.0× speed

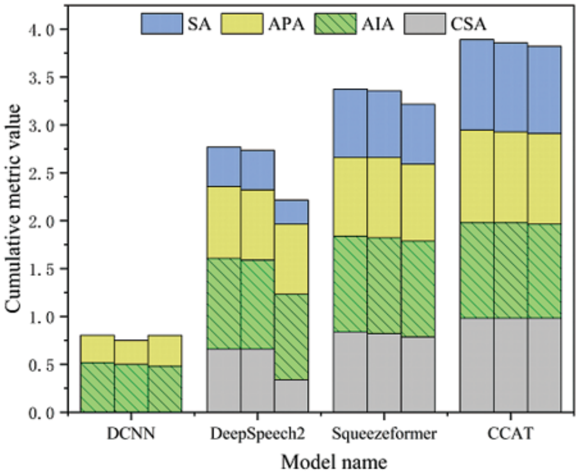


Figure 10: The cumulative bar chart under 1.1× speed

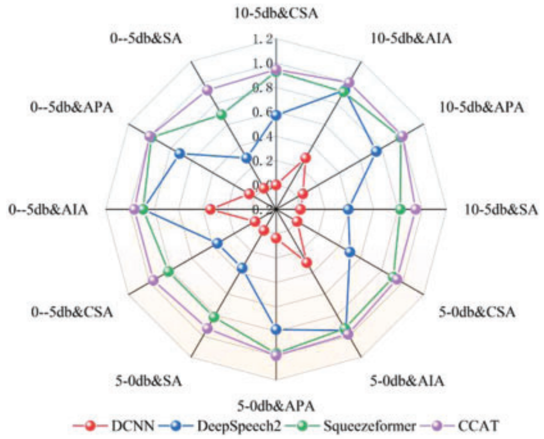


Figure 11: Radar chart of the test results at 0.9× speed

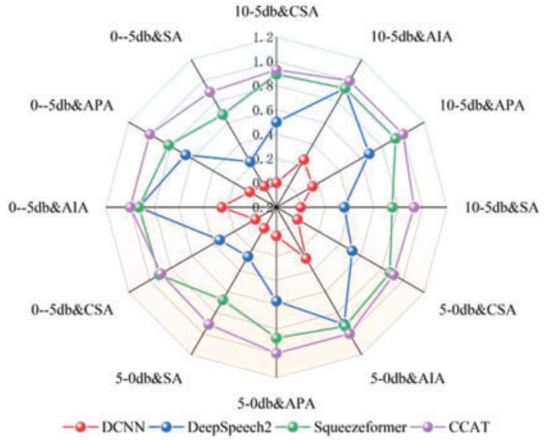


Figure 12: Radar chart of the test results at 1.0× speed

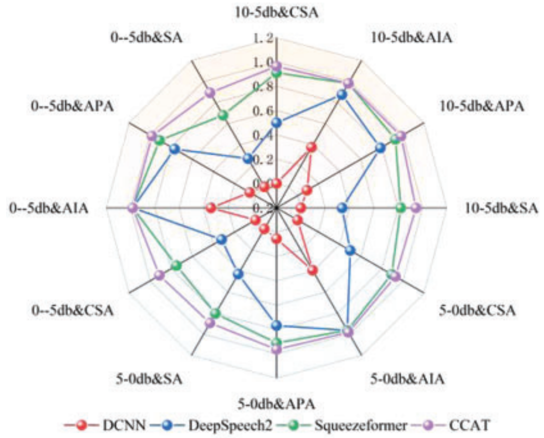


Figure 13: Radar chart of the test results at 1.1× speed

The following conclusions can be drawn from the results from Figs. 8 to 13: (1) By comparing the results from Figs. 8–13 and Table 10, it can be seen that the CER metric has apparent limitations when measuring the performance of an ATC speech recognition system from the perspective of the effectiveness of control instructions. (2) The DCNN model has the worst generalization ability. In Table 10, the CER of DCNN is 7.5% in the test dataset with the same distribution as the training data. However, after the data distribution is changed, the sentence accuracy of DCNN is 0 in 1004 robustness test data, which means that there are character errors in each keyword index of the instructions. (3) The proposed CCAT model has relatively stable robustness and maintains a high sentence accuracy in the test data of different indicators. This is partly due to the large number of model parameters and strong generalization ability. On the other hand, it also reflects the effectiveness of transfer learning, data augmentation, and error correction strategies used. (4) The purpose of this test is to test the robustness of each model. Since the test text under different speeds is not the same, the comparison results in the figure cannot reflect the influence of different speeds on the performance of the ATC speech recognition model. In addition, only the data in the figure cannot indicate the impact of ATC speech test data of different genders on the performance of the ATC speech recognition system. (5) It can be concluded from the results of Figs. 8–13 that ATC speech test data of different qualities will affect the performance of the ATC speech recognition system.

Based on the SA values corresponding to each indicator under male ATC speech test data in Figs. 8–10, this paper calculates the weights using the Critic method. The SA values corresponding to each indicator and the weight calculation results for each indicator are shown in Table 12. Similarly, Table 13 shows the SA values corresponding to each indicator under female ATC speech test data and the weight calculation results for each indicator.

Table 12: The SA values and the weight results for each indicator under male test data

	0.9× speed & 10–5 db	0.9× speed & 5–0 db	0.9× speed & 0–5 db	1.0× speed & 10–5 db	1.0× speed & 5–0 db	1.0× speed & 0–5db	1.1× speed & 10–5 db	1.1× speed & 5–0 db	1.1× speed & 0–5 db
DCNN	0	0	0	0	0	0	0	0	0
DeepSpeech2	0.179	0.125	0.071	0.214	0.268	0.161	0.411	0.411	0.25
Squeezeformer	0.732	0.679	0.66	0.661	0.625	0.536	0.714	0.696	0.625
CCAT	0.964	0.928	0.928	0.928	0.928	0.893	0.946	0.928	0.91
Weight results	0.089	0.114	0.166	0.058	0.065	0.104	0.167	0.178	0.059

Table 13: The SA values and the weight results for each indicator under female test data

	0.9× speed & 10–5 db	0.9× speed & 5–0 db	0.9× speed & 0–5 db	1.0× speed & 10–5 db	1.0× speed & 5–0 db	1.0× speed & 0–5 db	1.1× speed & 10–5 db	1.1× speed & 5–0 db	1.1× speed & 0–5 db
DCNN	0	0	0	0	0	0	0	0	0
DeepSpeech2	0.357	0.268	0.232	0.393	0.357	0.286	0.339	0.429	0.268

(Continued)

Table 13 (continued)

	0.9× speed & 10–5 db	0.9× speed & 5–0 db	0.9× speed & 0–5 db	1.0× speed & 10–5 db	1.0× speed & 5–0 db	1.0× speed & 0–5 db	1.1× speed & 10–5 db	1.1× speed & 5–0 db	1.1× speed & 0–5 db
Squeezeformer	0.75	0.679	0.679	0.821	0.821	0.696	0.821	0.804	0.679
CCAT	0.929	0.911	0.893	0.946	0.929	0.929	0.946	0.893	0.893
Weight results	0.065	0.111	0.140	0.1	0.095	0.098	0.079	0.218	0.094

The weight calculation results in Tables 12 and 13 show that the indicator “1.1× speed & 5–0 dB” has relatively high weights for both male and female ATC speech test data, indicating that the performance of various models under this indicator differs significantly and that it contains much information. In addition, the weights between different indicators are not significantly different for male and female control speech test data, indicating that the information contained in the ATC speech test data selected under different indicators is less affected by gender.

Based on Tables 12 and 13 data, this paper uses the VIKOR method to calculate the system evaluation. The calculation results are shown in Table 14, where a smaller VIKOR score indicates better model performance.

Table 14: The ranking table of each model score

Experimental ID	Name	Model scores for male test data	Model Scores for female test data	Weighted scores	Overall ranking
0	DCNN	1	1	1	4
1	DeepSpeech2	0.796	0.578	0.687	3
2	Squeezeformer	0.275	0.166	0.220	2
3	CCAT	0	0	0	1

According to Table 14, it can be seen that the models this paper constructed have achieved the best scores, whether under male or female ATC speech test data. The Squeezeformer and DeepSpeech2 models both achieved relatively better scores under female ATC speech test data than under male ATC speech test data. The DCNN model has the poorest robustness, while the CCAT model has the best robustness.

6 Conclusion

This study proposes a Conformer-CTC/Attention-T5 model with error correction and transcription capabilities to address the downstream task deviation caused by inaccurate recognition results in applying ASR technology in this field. In the proposed model architecture, Conformer extracts the spectro-temporal features of audio signals, capturing global interactions between sequences. At the same time, the Attention mechanism assists CTC in achieving precise alignment between input features and output characters. The T5 model is employed for error correction of recognition results and further simplifying the corrected text, improving transcription accuracy and alleviating the problem of mixed Chinese and English in ATC speech recognition. A series of parameter-tuning experiments are designed to investigate the impact of hyperparameters on model performance. The influence of CTC classifier size on model performance is examined, showing that using the vocabulary size of a small lexicon as the number of classes for CTC output during fine-tuning effectively improves recognition accuracy and decoding speed. The model can learn the most helpful information when the weights of CTC loss and Attention loss are equal. The number of attention heads in the encoder and decoder and the number of Conformer blocks are analyzed, indicating that an appropriate number of attention heads and Conformer blocks facilitates the model in handling complex audio structural features and enhancing the modeling capability of input information. This study employs transfer learning, data augmentation, and error correction models to address the challenges in Chinese speech recognition in ATC. Ablation experiments demonstrate that compared to data augmentation, pre-training strategies contribute more to improving model performance. The model significantly improves performance when data augmentation and transfer learning strategies are employed simultaneously. Additionally, including an error correction model further enhances the model's performance. The constructed CCAT model achieves a CER of 3.44% on the Our ATC Corpus, exhibiting a performance improvement of 3.64% compared to the baseline model. The effectiveness of the proposed model is also validated on two open-source corpora. On the AISHELL-1 dataset, the CCAT model achieves a CER of 3.42%, resulting in a performance improvement of 1.23% compared to the baseline model. On the LibriSpeech dataset, the CCAT model achieves a WER of 5.27%, exhibiting a performance improvement of 7.67% compared to the baseline model. In robustness evaluation experiments, the proposed model outperforms the baseline model by 22%. Compared to existing models, the proposed model in this paper has the following advantages: In terms of recognition performance, the CCAT model exhibits strong robustness, high accuracy, and the ability to handle mixed Chinese text, effectively mitigating the problem of downstream task deviations caused by inaccurate recognition in the ATC domain when applying ASR technology. Regarding scalability, the CCAT model demonstrates adaptability to different fields and possesses a certain level of generalization. It can be applied in the ATC domain and various human-machine interaction tasks, enhancing interactive capabilities. However, it should be noted that the algorithm's complexity is increased due to the adoption of large models for both the recognition and error correction modules, which pose specific requirements for deployment scenarios. Knowledge distillation techniques will be employed in future work to reduce model complexity and accelerate model inference and deployment.

Acknowledgement: We would like to extend our heartfelt gratitude to all individuals who have contributed to this paper, excluding ourselves. Their unwavering support and invaluable assistance have significantly enriched our research work.

Funding Statement: This study was co-supported by the National Key R&D Program of China (No. 2021YFF0603904), National Natural Science Foundation of China (U1733203), and Safety Capacity Building Project of Civil Aviation Administration of China (TM2019-16-1/3).

Author Contributions: Study conception and design: Peiyuan Jiang, Weijun Pan; data collection: Weijun Pan, Junxiang Huang; analysis and interpretation of results: Peiyuan Jiang, Weijun Pan; draft manuscript preparation: Peiyuan Jiang, Jian Zhang, Teng Wang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Some or all data, models, or codes that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Zhang, J. Kong, C. Chen, Y. Li and H. Liang, “Speech GAU: A single head attention for mandarin speech recognition for air traffic control,” *Aerospace*, vol. 9, no. 8, pp. 395, 2022.

[2] H. Helmke, M. Kleinert, O. Ohneiser, H. Her and S. Shetty, “Machine learning of air traffic controller command extraction models for speech recognition applications,” in *39th Digital Avionics Systems Conf.*, San Antonio, TX, USA, pp. 1–9, 2020.

[3] W. Pan, P. Jiang, Z. Wang, Y. Li and Z. Liao, “Ernie-gram BiGRU attention: An improved multi-intention recognition model for air traffic control,” *Aerospace*, vol. 10, no. 4, pp. 349, 2023.

[4] E. Rahhal, A. El Hannani and H. Ouahmane, “Automatic speech recognition errors detection and correction: A review,” *Procedia Computer Science*, vol. 128, pp. 32–37, 2018.

[5] Y. Lin, L. Deng, Z. Chen, X. Wu, J. Zhang *et al.,* “A real-time ATC safety monitoring framework using a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4572–4581, 2019.

[6] H. Helmke, O. Ohneiser, T. Mühlhausen and M. Wies, “Reducing controller workload with automatic speech recognition,” in *35th Digital Avionics Systems Conf.*, Sacramento, CA, USA, pp. 1–10, 2016.

[7] H. Liu, V. H. Cheng, D. Ballinger, A. Fong, J. Nguyen *et al.,* “A speech-enabled simulation interface agent for airspace system assessments,” in *Modeling and Simulation Technologies Conf.*, Kissimmee, Florida, USA, pp. 0148, 2015.

[8] Y. Oualil, D. Klakow, G. Szaszák, A. Srinivasamurthy, H. Helmke *et al.,* “A context-aware speech recognition and understanding system for air traffic control domain,” in *IEEE Automatic Speech Recognition and Understanding Workshop*, Okinawa, Japan, pp. 404–408, 2017.

[9] H. Helmke, J. Rataj, T. Mühlhausen, O. Ohneiser, H. Ehr *et al.,* “Assistant-based speech recognition for ATM applications,” in *11th USA/Europe Air Traffic Management Research and Development Seminar*, Lisbon, Portugal, pp. 1–10, 2015.

[10] H. Helmke, O. Ohneiser, J. Buxbaum and C. Kern, “Increasing atm efficiency with assistant based speech recognition,” in *Proc. of the 13th USA/Europe Air Traffic Management Research and Development Seminar*, Seattle, USA, pp. 1–10, 2017.

[11] M. Kleinert, H. Helmke, G. Siol, H. Ehr, M. Finke *et al.,* “Machine learning of controller command prediction models from recorded radar data and controller speech utterances,” in *7th SESAR Innovation Days*, Belgrade, The Republic of Serbia, pp. 1–8, 2017.

[12] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. Bengio *et al.,* “Towards end-to-end speech recognition with deep convolutional neural networks,” *ArXiv Preprint*, arXiv: 1701.02720, 2017.

[13] T. Hori, S. Watanabe, Y. Zhang and C. William, “Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM,” *ArXiv Preprint*, arXiv: 1706.02737, 2017.

[14] S. Kim, T. Hori and S. Watanabe, “Joint CTC-attention based end-to-end speech recognition using multitask learning,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, New Orleans, LA, USA, pp. 4835–4839, 2017.

[15] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev *et al.,* “Quartznet: Deep automatic speech recognition with 1D time-channel separable convolutions,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp. 6124–6128, 2020.

[16] L. Dong, S. Xu and B. Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Calgary, AB, Canada, pp. 5884–5888, 2018.

[17] J. Nozaki and T. Komatsu, “Relaxing the conditional independence assumption of CTC-based ASR by conditioning on intermediate predictions,” *ArXiv Preprint*, arXiv: 2104.02724, 2021.

[18] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson *et al.,* "A comparison of sequence-to-sequence models for speech recognition," in *Interspeech*, Stockholm, Sweden, pp. 939–943, 2017.

[19] A. A. Abdelhamid, E. S. M. El-Kenawy, B. Alotaibi, G. M. Amer, M. Y. Abdelkader *et al.,* "Robust speech emotion recognition using CNN+LSTM based on stochastic fractal search optimization algorithm," *IEEE Access*, vol. 10, pp. 49265–49284, 2022.

[20] Mustaqeem and S. Kwon, "CLSTM: Deep feature-based speech emotion recognition using the hierarchical ConvLSTM network," *Mathematics*, vol. 8, no. 12, pp. 2133, 2020.

[21] Mustaqeem and S. Kwon, "Optimal feature selection based speech emotion recognition using two-stream deep convolutional neural network," *International Journal of Intelligent Systems*, vol. 36, no. 9, pp. 5116–5135, 2021.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, vol. 30, pp. 1–11, 2017.

[23] A. Gulati, J. Qin, C. C. Chiu, N. Parmar, Y. Zhang *et al.,* "Conformer: Convolution-augmented transformer for speech recognition," *ArXiv Preprint*, arXiv: 2005.08100, 2020.

[24] H. Holone, "Possibilities, challenges and the state of the art of automatic speech recognition in air traffic control," *International Journal of Computer and Information Engineering*, vol. 9, no. 8, pp. 1933–1942, 2015.

[25] Y. Lin, "Spoken instruction understanding in air traffic control: Challenge, technique, and application," *Aerospace*, vol. 8, no. 3, pp. 65, 2021.

[26] Y. Lin, L. Li, H. Jing, B. Ran and D. Sun, "Automated traffic incident detection with a smaller dataset based on generative adversarial networks," *Accident Analysis & Prevention*, vol. 144, pp. 105628, 2020.

[27] Y. Lin, Q. Li, B. Yang, Z. Yan, H. Tan *et al.,* "Improving speech recognition models with small samples for air traffic control systems," *Neurocomputing*, vol. 445, pp. 287–297, 2021.

[28] D. Wang and X. Zhang, "Thchs-30: A free Chinese speech corpus," *ArXiv Preprint*, arXiv: 1512.01882, 2015.

[29] B. Yang, X. Tan, Z. Chen, B. Wang, D. Li *et al.,* "ATCSpeech: A multilingual pilot-controller speech corpus from real air traffic control environment," *ArXiv Preprint*, arXiv: 1911.11365, 2019.

[30] V. Panayotov, G. Chen, D. Povey and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, South Brisbane, QLD, Australia, pp. 5206–5210, 2015.

[31] T. Tarnavska, L. Baranovska, N. Glushanytsia and S. Yahodzinskyi, "The impact of psychological factor on the aircraft operation safety," *E3S Web of Conf.*, EDP Sciences, vol. 258, pp. 02029, 2021.

[32] S. Badrinath and H. Balakrishnan, "Automatic speech recognition for air traffic control communications," *Transportation Research Record*, vol. 2676, no. 1, pp. 798–810, 2022.

[33] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey *et al.,* "Robust speech recognition via large-scale weak supervision," *ArXiv Preprint*, arXiv: 2212.04356, 2022.

[34] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov *et al.,* "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," in *Int. Cong. on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Chengdu, China, IEEE, pp. 439–444, 2020.

[35] T. Ko, V. Peddinti, D. Povey and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth Annual Conf. of the Int. Speech Communication Association*, Dresden, Germany, pp. 1–11, 2015.

[36] B. Tolooshams, R. Giri, A. H. Song, U. Isik and A. Krishnaswamy, "Channel-attention dense u-net for multichannel speech enhancement," in *Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, pp. 836–840, 2020.

[37] Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky and A. Y. Ng, "Neural language correction with character-based attention," *ArXiv Preprint*, arXiv: 1603.09727, 2016.

[38] T. Tanaka, R. Masumura, H. Masataki and Y. Aono, "Neural error corrective language models for automatic speech recognition," in *Interspeech*, Hyderabad, India, pp. 401–405, 2018.

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang *et al.,* "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[40] L. Zhu, W. Liu, L. Liu and E. Lin, "Improving ASR error correction using N-best hypotheses," in *Automatic Speech Recognition and Understanding Workshop (ASRU)*, Cartagena, Colombia, pp. 83–89, 2021.

[41] Y. Weng, S. S. Miryala, C. Khatri, R. Wang, H. Zheng *et al.,* "Joint contextual modeling for asr correction and language understanding," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp. 6349–6353, 2020.

[42] R. Ma, H. Li, Q. Liu, L. Chen and K. Yu, "Neural lattice search for speech recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Barcelona, Spain, pp. 7794–7798, 2020.

[43] R. Ma, M. J. Gales, K. Knill and M. Qian, "N-best T5: Robust ASR error correction using multiple input hypotheses and constrained decoding space," *ArXiv Preprint*, arXiv: 2303.00456, 2023.

[44] S. Roy, "Semantic-WER: A unified metric for the evaluation of ASR transcript for end usability," *ArXiv Preprint*, arXiv: 2106.02016, 2021.

[45] J. Drayton and A. Coxhead, "The development, evaluation and application of an aviation radiotelephony specialised technical vocabulary list," *English for Specific Purposes*, vol. 69, pp. 51–66, 2023.

[46] B. Zhang, H. Lv, P. Guo, Q. Shao, C. Yang *et al.,* "Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Singapore, pp. 6182–6186, 2022.

[47] J. M. Martin-Donas, A. M. Gomez, J. A. Gonzalez and A. M. Peinado, "A deep learning loss function based on the perceptual evaluation of the speech quality," *IEEE Signal Processing Letters*, vol. 25, no. 11, pp. 1680–1684, 2018.