JUSTC

Article

http://justc.ustc.edu.cn

Received: December 19, 2024; Accepted: November 15, 2025

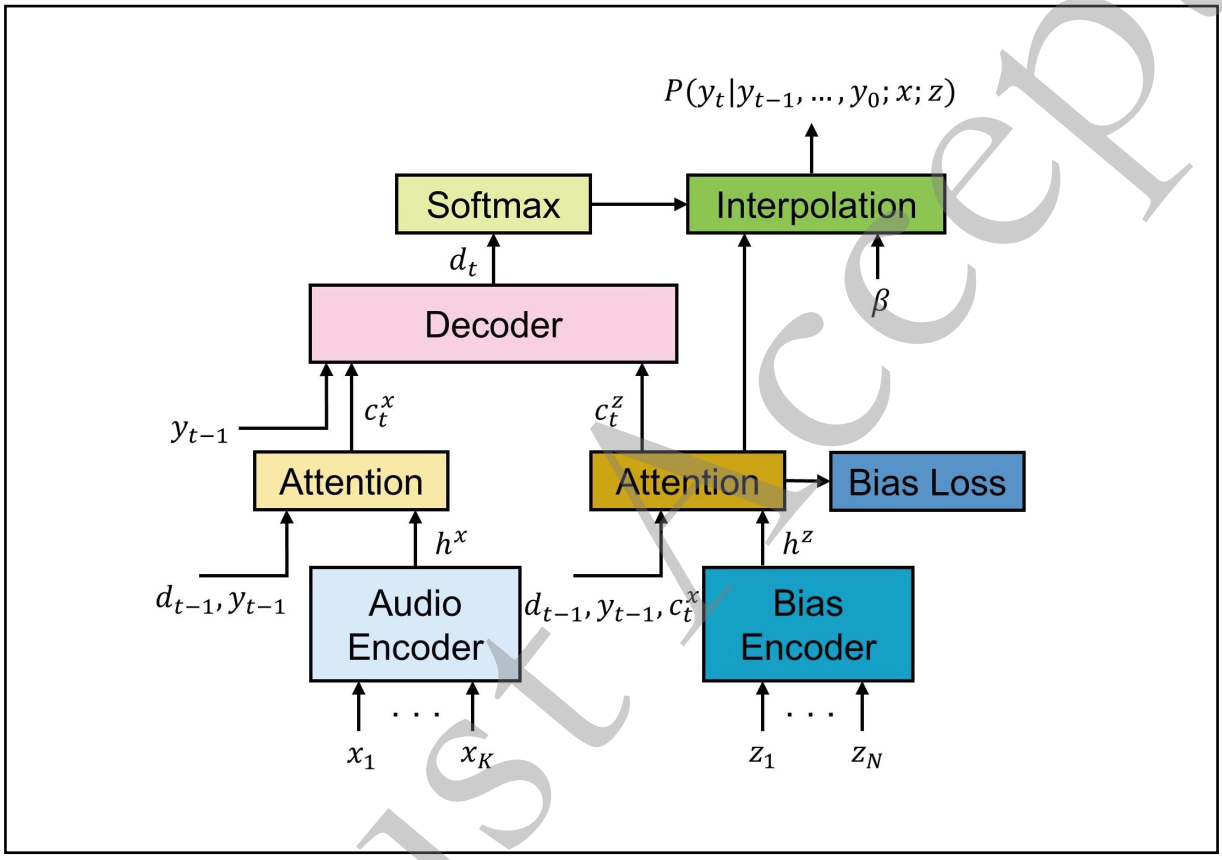
Deep CLAS: deep contextual listen, attend, and spell

Mengzhi Wang[1] ✉, Shifu Xiong[1,2], Genshun Wan[1,2], Hang Chen[2], Jianqing Gao[1], and Lirong Dai[2]

[1]*iFLYTEK Research, iFLYTEK Co. Ltd., Hefei 230088, China;*
[2]*National Engineering Research Center of Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China*

✉Correspondence: Mengzhi Wang, E-mail: mzwang@ustc.edu.cn
© 2026 The Author(s). This is an open access article under the CC BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Graphical abstract



The framework of Deep CLAS model.

Public summary

- Propose Deep CLAS to deeply utilize contextual information for improved ASR of rare words.
- Introduce bias loss and enriched query of bias attention to enhance focus and accuracy on contextual information.
- Replace phrase-level encoding with character-level encoding and use conformer for fine-grained contextual representation.
- Directly utilize bias attention scores to correct output probability distribution and employ prefix tree to avoid irrelevant interference.

Citation: Wang M Z, Xiong S F, Wan G S, et al. Deep CLAS: deep contextual listen, attend, and spell. *JUSTC*, **2026**, 56(0): . CSTR: 32290.14.JUSTC-2024-0158. DOI: 10.52396/JUSTC-2024-0158

Just Accepted

JUSTC

Article

http://justc.ustc.edu.cn

Received: December 19, 2024; Accepted: November 15, 2025

Deep CLAS: deep contextual listen, attend, and spell

Mengzhi Wang[1] ✉, Shifu Xiong[1,2], Genshun Wan[1,2], Hang Chen[2], Jianqing Gao[1], and Lirong Dai[2]

[1]*iFLYTEK Research, iFLYTEK Co. Ltd., Hefei 230088, China;*
[2]*National Engineering Research Center of Speech and Language Information Processing, University of Science and Technology of China, Hefei 230027, China*

✉Correspondence: Mengzhi Wang, E-mail: mzwang@ustc.edu.cn
© 2026 The Author(s). This is an open access article under the CC BY-NC-ND 4.0 license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Cite This: *JUSTC*, **2026**, 56(X): (9pp) Read Online SI Supporting Information

Abstract: Contextual Listen, Attend and Spell (CLAS) has been shown effective in improving Automatic Speech Recognition (ASR) of rare words. It relies on phrase-level contextual modeling and attention-based relevance scoring without explicit contextual constraint which lead to insufficient use of contextual information. In this work, we propose Deep CLAS to deeply utilize contextual information. We introduce bias loss forcing model to focus on contextual information. The query of bias attention is also enriched to improve the accuracy of the bias attention score. To get fine-grained contextual information, we replace phrase-level encoding with character-level encoding and encode contextual information with conformer. Furthermore, the bias attention score is directly utilized to correct the model's output probability distribution. Additionally, a prefix tree is employed to prevent interference from irrelevant information. On the public AISHELL-1 dataset, Deep CLAS achieves a 65.78% relative increase in recall and a 53.49% relative increase in F1-score over the CLAS baselines in named entity recognition.

Keywords: speech recognition; CLAS; deep context

CLC number: **Document code:** A

1 Introduction

Speech recognition is one of the most popular human-computer interaction methods at present. Thanks to the development of deep learning technology and the accumulation of data in recent years[1–4], the accuracy of machine speech recognition in some scenes has exceeded that of humans[5]. However, the data-driven deep learning model suffers long-tail distribution problems[6, 7]. Rare words, such as entity names, are hard to recognize.

There are two mainstream methods to improve the recognition rate of rare words in the traditional ASR system. One is adding more training text about rare words of the language model[8, 9]. Another way is to construct a prefix tree[10] to increase the corresponding posterior probability during decoding.

Compared with the traditional model, the end-to-end model significantly improves speech recognition accuracy in general scenes[11]. However, using the above two methods in the end-to-end model is challenging because the end-to-end model has no pronunciation dictionary and uses a word or subword modeling[12] directly. As a result, the model may assign a higher probability to homophonic words, making the contextualization very difficult to take effect. Therefore, the end-to-end model usually introduces contextual information into the model and optimizes jointly to improve the contextualization effect[13–26]. Accordingly, the Contextual Listen-Attend-Spell (CLAS)[27–29] model was proposed to encode a set of bias phrases first. A bias phrase refers to any contextual unit, such as single words or multi-word expressions, that is used to condition the ASR model for improved recognition accuracy in specific contexts. Then the state of the decoder is used to extract relevant contextual information through the attention mechanism. Finally, the context is passed to decoder layers to obtain the output probability. A particular ⟨/*bias*⟩ symbol is inserted at the corresponding positions of the reference to make the model pay attention to the contextual information.

We make a deep analysis of the CLAS model and have some interesting findings. First, the correct recognition of ⟨/*bias*⟩ does not mean the correct recognition of bias phrases. The model tends to output ⟨/*bias*⟩ directly when the decoding history matches a bias phrase and ignores contextual information before output ⟨/*bias*⟩. Second, CLAS uses the last decoder state as the query of bias attention, which is missing the last recognized word information. However, the model should pay more attention to the bias phrases that contain the last recognized word. Furthermore, the current acoustic context is vital for bias attention because the acoustic context contains the pronunciation information of the current word, which is not contained in the last decoder state. Third, CLAS uses coarse-grained contextual information encoding. Each bias phrase has only one embedding. The single embedding may not be able to encode the information of all characters if the bias phrase is long. Finally, CLAS feeds contextual information into the decoder, but the model may not use it at all. Therefore, CLAS does not make effective use of contextual information.

To enhance the contextualization performance of the CLAS model, we propose several interrelated improvements in our

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

Deep CLAS model. First, we introduce a bias loss that explicitly encourages the model to focus more on contextual information during training. Building upon this, to further improve the accuracy of bias attention scores, we design a bias attention query that jointly incorporates the last decoder state, the last decoded word, and the current acoustic context. This combination enables the model to more precisely capture relevant contextual cues. Second, recognizing that phrase-level encoding may be too coarse to capture fine-grained contextual details, we replace it with character-level encoding. The finer granularity of character-level representations enriches the contextual information available to the bias attention mechanism, thereby enhancing its effectiveness. Third, we explicitly utilize the bias attention scores to better exploit contextual information within the decoding process. This step ensures that the model can effectively leverage the focused context identified by the attention mechanism. Finally, to mitigate the negative effects of irrelevant or noisy bias phrases—especially in long-context scenarios—we incorporate a prefix tree structure. This allows us to efficiently filter out unlikely bias phrases early in the decoding process, thereby reducing interference and improving recognition accuracy in complex contexts. Together, these improvements form a cohesive pipeline: from encouraging contextual focus via the bias loss, to refining attention query design and representation granularity, to explicitly leveraging attention scores, and finally to constraining irrelevant bias phrases with prefix trees. This integrated approach enables the Deep CLAS model to achieve significantly better contextual awareness and bias phrase recognition performance.

2 Related work

The CLAS model derives from the Listen, Attend and Spell (LAS) model[30] but increases the use of contextual information. This model has been proven to be an effective way for contextual biasing. Fig. 1(a) illustrates the overall structure of the CLAS model.

CLAS model contains two encoders, one decoder and two attention networks. One encoder is an audio encoder and the other is a bias encoder. The audio encoder encodes speech frames $x = (x_1, ..., x_K)$ into high-level acoustic features h^x. And bias encoder encodes bias phrases $z = (z_1, ..., z_N)$ into high-level contextual features h^z. The bias encoder is a multilayer long short-term memory network (LSTM)[31]. The last state of the LSTM is considered as the embedding of the entire bias phrase. Since the bias phrases may not be relevant to the current utterance, CLAS includes an additional learnable vector h^z_{nb}, corresponding to the 'no-bias' option. Similar to LAS, CLAS generates acoustic context vector c^x using the previous decoder state d_{t-1} and the output of the audio encoder h^x through the audio attention network.

$$u^x_{it} = v^x tanh(W^x_h h^x_i + W^x_d d_{t-1} + b^x_a) \tag{1}$$

$$\alpha^x_t = softmax(u^x_t) \tag{2}$$

$$c^x_t = \sum_{i=0}^{K} \alpha^x_{it} h^x_i \tag{3}$$

Then generate bias context vector c^z using the previous decoder state d_{t-1} and the output of bias encoder h^z through bias attention network.

$$u^z_{it} = v^z tanh(W^z_h h^z_i + W^z_d d_{t-1} + b^z_a) \tag{4}$$

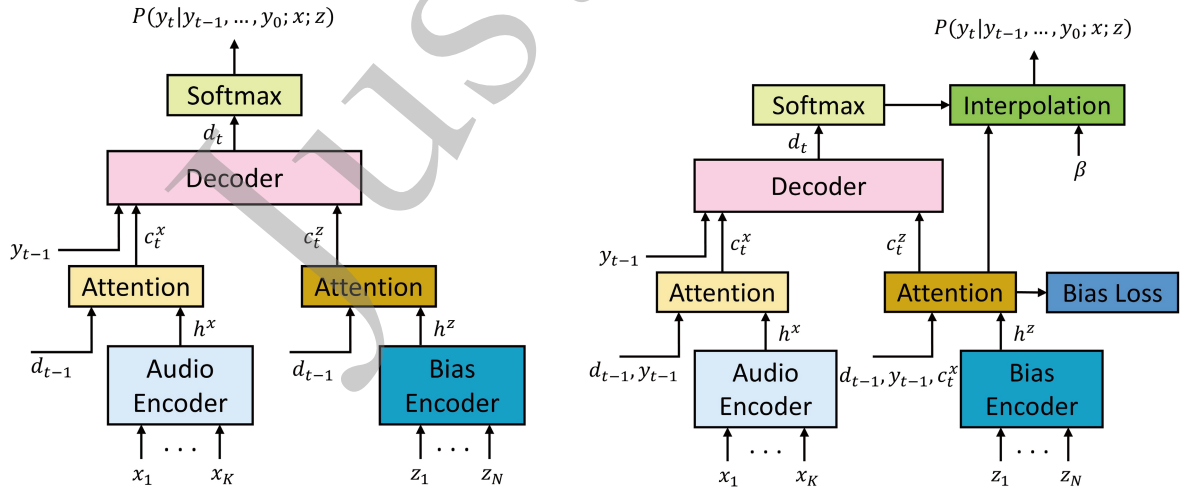
$$\alpha^z_t = softmax(u^z_t) \tag{5}$$

$$c^z_t = \sum_{i=0}^{N} \alpha^z_{it} h^z_i \tag{6}$$

After obtaining c^x and c^z, they are spliced and fed into the decoder to get the decoder state d_t and output probabilities over all modeling units.

$$d_t = RNN(d_{t-1}, y_{t-1}, c^x_t, c^z_t) \tag{7}$$

$$P(y_t|h^x, y < t, z) = softmax(W_s d_t + b_s) \tag{8}$$



(a) CLAS Model (b) Deep CLAS Model

Fig. 1. The framework of CLAS model and Deep CLAS model

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

$$L_{CLAS} = -logP(y|x,z) \tag{9}$$

During training, the bias phrase list is randomly created from the reference transcripts associated with the utterances in the training batch. To exercise the 'no-bias' option, not all reference transcripts extract bias phrases. The proportion of extracting bias phrases is P_{keep}. k word n-grams are randomly selected from each kept reference, where k is picked uniformly from $[1, N_{phrases}]$ and n is picked uniformly from $[1, N_{order}]$. P_{keep}, $N_{phrases}$ and N_{order} are hyperparameters of the training process. A special $\langle/bias\rangle$ symbol is inserted where bias phrases appear in reference transcripts. The purpose of $\langle/bias\rangle$ is to introduce a training error which can be corrected only by considering the correct bias phrase.

3 Materials and methods

In this section, we will introduce the proposed Deep CLAS model. Fig. 1(b) illustrates the overall structure of the Deep CLAS model.

3.1 Bias loss

The CLAS inserts $\langle/bias\rangle$ into the reference transcripts to make the model focuses on contextual information. However, the model can ignore contextual information and output $\langle/bias\rangle$ when the historical output is the same as a bias phrase. Besides, the model does not know when to start paying attention to contextual information. Therefore, we used the bias loss[32], a cross-entropy loss of the bias attention coefficient. The bias loss makes the model focuses on the corresponding bias phrases in the bias list when decoding bias phrases. Otherwise, it focuses on the 'no-bias'. The bias loss is defined as:

$$L_{bias} = -\sum_{t}\sum_{k} y_{tk} log(\alpha_{tk}) \tag{10}$$

Where α_{tk} is the bias attention score of the kth word of the bias list at the decoding step t, and y_t is a one-hot vector. If the t-th word of reference is part of the kth bias phrase, y_{tk} is 1; otherwise, it is 0. If there is no bias phrase match, y_{t0} is 1 which indicates attending 'no-bias'. The bias loss is combined with the standard joint Connectionist Temporal Classification (CTC)/attention loss as the optimization object:

$$L = (1-\lambda)L_{att} + \lambda L_{ctc} + L_{bias} \tag{11}$$

3.2 The query of bias attention

The CLAS uses d_{t-1} as the query of the bias attention. Fig. 1(a) shows d_{t-1} does not contain the last recognized word information y_{t-1}. However, the last recognized word is essential for bias attention. A bias phrase can have a higher attention score if it contains y_{t-1} or a lower attention score if not. Besides, d_{t-1} lacks of c_t^x which is acoustic contextual information at the current time. A bias phrase should have a higher attention score if it matches acoustic information at the current time rather than historical time. Thus, we modify Eq. 4 to:

$$u_{it}^z = v^z tanh(W_h^z h_i^z + W_d^z [d_{t-1} : y_{t-1} : c_t^x] + b_a^z) \tag{12}$$

$[d_{t-1} : y_{t-1} : c_t^x]$ indicates splicing d_{t-1}, y_{t-1} and c_t^x. In this way, the model has more information when scoring bias phrases, so bias phrases are scored more accurately.

3.3 The granularity of bias information

The CLAS model utilizes phrase-level embeddings to represent bias phrases, with each bias phrase being represented by a fixed-size vector. However, there is a potential problem of information loss when the bias phrase is long. Furthermore, phrase-level embeddings do not correspond to the output modeling unit, which can be either sub-words or characters. In this case, the model doesn't know which character in the bias phrase to output when using phrase-level embeddings. To address these issues, we use the output of all time steps of the LSTM instead of just the last state, to obtain fine-grained contextual information. Additionally, we integrate the BLSTM and conformer architecture to replace the LSTM. This change results in better embeddings.

3.4 Probability fusion

The CLAS model feeds contextual information into the decoder, making the model choose whether to use the contextual information, which is an implicit way to use contextual information and may lead to performance degradation. To use contextual information explicitly, we modify the probability distribution of the model output according to the bias attention score directly. This is done by scaling the bias attention score and the probability distribution of the model output and then combining them to obtain a final probability distribution:

$$P_b^{1\cdots z} = \beta P_b^{1\cdots z} \tag{13}$$

$$P_m = (1-\beta(1-P_b^0))P_m \tag{14}$$

In the formula, P_b denotes the probability distribution of bias attention and P_m denotes the probability distribution of model output. P_b^0 is the score of 'no-bias' and $P_b^{1\cdots z}$ is the score of bias phrases. β is the bias coefficient to control the bias degree. The larger the β is, the easier it is for the model to recognize bias phrases, but more false triggers may occur. We first multiply $P_b^{1\cdots z}$ by β, so the sum of $P_b^{1\cdots z}$ is $\beta(1-P_b^0)$. The remaining score is $1-\beta(1-P_b^0)$, so we multiply P_m by $1-\beta(1-P_b^0)$. Then P_m plus $P_b^{1\cdots z}$ equals one. There are two ways to combine P_m and $P_b^{1\cdots z}$. One way is to concatenate $P_b^{1\cdots z}$ behind P_m to form a larger vocabulary just like pointer generator[33]. In fine-grained contextual encoding, since a character may appear in multiple bias phrases, we can merge the bias scores of the same character from different bias phrases to make the bias scores more concentrated. Another way is to merge the scores of the same characters in P_m and $P_b^{1\cdots z}$, which we call interpolation. In this case, the vocabulary size does not change.

3.5 Prefix tree

The prefix tree[34–36] is widely used in the traditional bias method. Building a prefix tree using bias phrases, which is traversed during decoding, can prevent the excitation of bias phrases whose prefix does not match the decoding history. Because only active tree nodes in the prefix tree will be excited. Fig. 2 illustrates a prefix tree when the decoding his-

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

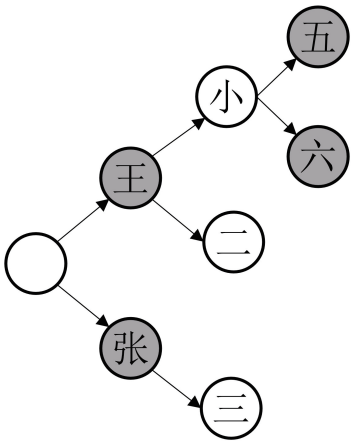


Fig. 2. Prefix tree when a list of bias phrases consists of ("张三","王二", "王小五", "王小六") in character level.

tory is "王 小 ". The active tree nodes are represented by circles with gray background. The CLAS uses the last decoder state to score bias phrases. Theoretically, it can achieve the same effect as the prefix tree. However, neural networks tend to overfitting and the effect drops sharply in mismatched scenes. The longest bias phrase during training is only four characters, which may reduce the effect of bias phrases with more than four characters. Therefore, we conducted a long bias phrase experiment and used the prefix tree to solve the mismatch problem.

4 EXPERIMENTS

4.1 Datasets

We primarily conduct ablation experiments for the Deep CLAS model on AISHELL-1[37], which is a benchmark dataset for mandarin ASR. The AISHELL-1 dataset contains 178 hours of data collected from 400 people. The test set of AISHELL-1 does not contain bias phrases. To make the results reproducible, we use the named entities of the AISHELL-NER[38] as bias phrases. There are 3393 named entities in the test set. We retain named entities containing two to four characters and de-duplicate them to get 891 named entities as the bias phrases list. For comparison with other models, we also used the AISHELL-1 hotword test set[19], which is open-sourced by Alibaba.

AISHELL-1 is a dataset that is relatively formal in language and recorded in a single-speaker manner; hence, it lacks natural speech characteristics and interaction between speakers. To verify the adaptability of the Deep CLAS model in different situations, we also used the open-source MagicData-RAMC dataset[39]. Unlike AISHELL-1, which is a reading-style speech dataset, MagicData-RAMC is a dialog speech dataset, making recognition more challenging. In addition, to verify the performance of the Deep CLAS model on other languages, we also used the LibriSpeech train-clean-100 dataset[40], which contains 100 hours of English speech data.

4.2 Experimental Setup

Our model is based on the joint CTC-attention structure. CTC weight is set to 0.3. The speech features are 80-dimensional log-mel filterbank (FBank) computed on 32ms window with 10ms shift. We apply SpecAugment[41] with mask parameters $(F = 10, T = 50, m_F = m_T = 2)$ and speed perturbation[42] with factors of 0.9, 1.0 and 1.1 to augment training data. The encoder is composed of a subsampling module and 12-layer conformer blocks. The subsampling module contains two Conv2d with kernel size 3*3 and stride 2*2, resulting in 40ms framerate. For conformer blocks, the attention dimension is 256 with 4 attention heads, the dimension in the feed-forward network sublayer is 2048, and the kernel size of the conformer convolution module is 15. The decoder is a layer of LSTMP with a hidden size of 1024 and a projection size of 512. The bias encoder is a layer of (Bi-directional) LSTMP with a hidden size of 512 and a projection size of 256 or two layers of conformer with same configuration as encoder. The encoder attention is computed over 128-dimensional location sensitive attention[43]. The bias attention is computed over 512-dimensional additive attention. Word embedding size is 512. We set attention dropout to 0.1 and label smooth[44] to 0.1. We apply the Adam[45] optimizer with a gradient clipping value of 5.0 and warmup[46] learning rate scheduler with 25,000 warmup steps on the Chinese dataset and 15,000 warmup steps on the English dataset, with a peak learning rate of 0.0016. We train the models for 180 epochs on the Chinese dataset and 300 epochs on the English dataset, and average the last 30 checkpoints to obtain the final model. During training, we set $P_{keep} = 0.5, N_{phrases} = 1, N_{order} = 4$ for consistency with the original CLAS.

To enable the model to obtain the pronunciation information of bias phrases, we add a phoneme encoder to encode phonemes of Chinese characters or English words in bias phrases[29]. Phoneme embeddings and Chinese character or English word embeddings are concatenated and reduced in dimension with a linear layer to obtain final embeddings. We use a python package called g2p_zh_en to get phonemes. Decoding was performed using the beam search decoding algorithm with beam 10 without using any external language model.

We utilize recall, precision and F1-score as our measurement:

$$R = \frac{n}{N_t} \tag{15}$$

$$P = \frac{n}{N_r} \tag{16}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{17}$$

Where R is the recall, P is the precision, and n is the number of bias phrases correctly identified in the recognition result. N_r is the number of all bias phrases in the recognition result and N_t is the number of all bias phrases in the reference transcript. F1-score is the harmonic mean of recall and precision. The higher the F1-score, the better. We also calculate character error rate (CER) to measure the overall recognition performance:

$$CER = \frac{S + D + I}{N} \tag{18}$$

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

JUST Accepted

Where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the total number of characters in the reference text. The lower the CER, the better.

5 Results and Discussion

In this section, we give the overall results in Table 1, and then analyze the methods proposed in this paper.

5.1 Bias loss

First, we reproduce the LAS and CLAS baseline model. When no bias phrases are added, the recognition rate of CLAS only slightly decreases compared to LAS. However, the recall of the CLAS baseline decreases after adding bias phrases. We guess it was because too many bias phrases added during testing (891) compared with training (about 42). So we try to use only the named entities corresponding to each sentence (less than 5) as the bias phrases list instead of using all named entities. The results are shown in Table 2. F1-score does not change after adding bias phrases either, indicating that the number of bias phrases has nothing to do with bias failure. We then plot the bias attention map of one audio in Fig. 3(a). The vertical axis represents bias phrases, and the horizontal axis represents the decoding result. The numbers on the graph represent the attention score. The label is "在许茹芸看来". So the model should pay attention to the bias phrase when decoding "许茹芸". However, it can be seen from the figure that the model only pays attention to the bias phrase when decoding "许" and ignores it when decoding other words, making "茹芸" recognized as "如云". Therefore, the CLAS baseline sometimes cannot pay attention to the bias information correctly and it is important to add bias loss to constrain bias attention. The bias loss version in Table 1 (E1) indicates that recall increased from 78.46% to 79.59% when adding bias phrases. It proves the effectiveness of the bias loss. We also plot the bias attention map of bias loss version in Fig. 3(b). Although the recognition result is still wrong, the model can basically focus on the correct bias

Table 1. Results of using all named entities as the bias list. The Y/N in the bias column represents whether bias phrases are present. And the number in the bias column represents the bias coefficient.

Methods	bias	CER (%)	Recall (%)	Precision (%)	F1 (%)
B0: LAS	N	4.81	79.07	98.17	87.59
E0: CLAS	N	4.91	79.35	97.88	87.65
	Y	4.92	79.25	97.80	87.55
E1: E0+bias loss	N	5.11	78.46	98.20	87.22
	Y	5.06	79.59	98.10	87.88
E2: E1+query y_{t-1}	N	5.04	79.04	98.04	87.52
	Y	4.99	79.86	97.94	87.98
E3: E2+query c_t^x	N	4.95	79.25	98.30	87.75
	Y	4.83	82.16	97.87	89.33
E4: E3+fine-grained	N	4.89	79.59	97.60	87.68
	Y	4.80	82.85	98.01	89.79
E5: E4+BLSTM	N	4.91	79.90	97.94	88.0
	Y	4.65	85.11	97.72	90.98
E6: E4+conformer	N	4.88	79.49	98.05	87.80
	Y	4.60	86.86	97.80	92.01
E7: E6+pointer generator	0.1	4.49	91.66	96.01	93.79
	0.2	4.52	94.31	93.60	93.95
	0.3	4.64	95.95	90.46	93.12
E8: E7+merge probability	0.1	4.49	91.70	95.98	93.79
	0.2	4.52	94.37	93.48	93.92
	0.3	4.65	96.09	90.27	93.09
E9: E6+merge probability	0.1	4.47	92.90	95.55	94.21
	0.2	4.55	94.72	92.84	93.77

Table 2. Results of using corresponding named entities as the bias list.

Methods	bias	CER (%)	Recall (%)	Precision (%)	F1 (%)
CLAS	N	4.91	78.50	100	87.97
	Y	4.90	78.50	100	87.97

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

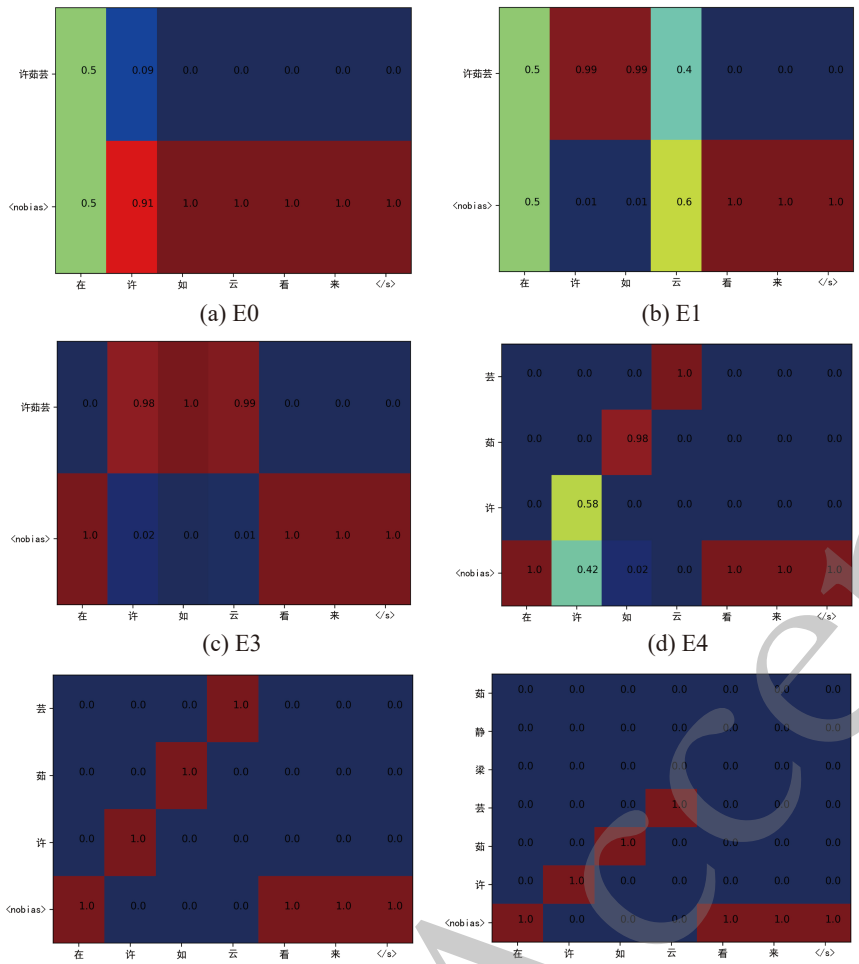


Fig. 3. Bias attention map of different models, only one or two bias phrases were added for simplification.

phrase at the right time. This is the foundation for subsequent optimization.

5.2 The query of bias attention

Then, we verify the effect of adding more information to the query of bias attention. Compared with E1 using d_{t-1} only as the query of bias attention, E2 which use $d_{t-1}+y_{t-1}$ can increase recall from 79.59% to 79.86%. Moreover, the recall was further raised to 82.16% when c_t^x is used as the query of bias attention, indicating the audio contextual information is important for bias attention. We also plot the bias attention map of E3 in Fig. 3c. Although the recognition result is still wrong, the bias attention score is much stronger than E2 in Fig. 3b. It shows that the model has more confidence in bias attention, and the bias attention score is more accurate.

5.3 The granularity of bias information

In this section, we conduct fine-grained modeling of bias information and compare the effects of different model structure encoding bias phrases. Fine-grained model E4 in Table 1 reaches 82.86% recall outperforming coarse-grained model E3 with 82.16% recall. We plot the bias attention map of E4 in Fig. 3d. It can be seen from the figure that it can pay attention to a character in the bias phrase rather than the whole word, reducing confusion when biasing. Further, after switching the bias encoder from LSTM to BLSTM or conformer, the recall can be improved consistently, achieving 85.11% and 86.86% respectively. We plot the bias attention map of E6 in Fig. 3e. The attention score is much stronger than E4, indicating that better bias encoding can also improve the confidence of bias attention and thus improve the bias effect. In addition, the CER of E6 without bias phrases is 4.88, which is better than CLAS and slightly worse than LAS, indicating that this approach has almost no negative impact on the general performance.

5.4 Probability fusion

Although the model's attention to bias phrases has dramatically increased after the above improvements, the recognition result in Fig. 3 is still wrong. It indicates that even if the model focuses on bias phrases, it does not mean that the model will use them for recognition since the model relies more on acoustic and linguistic information than bias information in recognition. Therefore, feeding the bias context vector into the decoder cannot utilize the bias information effectively. Considering that the bias attention score in Fig. 3e has reached one, we can directly use the bias score to correct the output of the model. We compared different correction methods, including pointer generator and interpolation. Table 1 shows that the optimal bias coefficient of the pointer generat-

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

or model (E7) is 0.2 with F1-score 93.95%, which is significantly higher than E6. Merging probabilities of the same characters in different bias phrases (E8) has little effect. Probably because even if there are multiple identical characters in the bias list, only the character with a specific context has a high bias score. To verify our conjecture, we add another bias phrase "梁静茹" which has the same character "茹" as "许茹芸" and plot the bias attention map in Fig. 3f. We can see that only "茹" in "许茹芸" has a high attention score, while the model does not focus on "茹" in "梁静茹" at all. It shows that although the bias score is calculated for each character separately, the adjacent acoustic context is also taken into account, resulting more accurate bias score. And if we merge probabilities of the same characters in the model output and the bias list (E9), F1-score can be further improved to 94.21%, resulting 53.49% relative improvement from the CLAS baseline.

5.5 Prefix tree

The longest bias phrase in our previous experiment has only four characters. However, the number of characters in bias phrases may be vast in the production environment. To verify the effect of the model on long bias phrases, we retain named entities in AISHELL-NER containing two to sixteen characters and de-duplicate them to get 1186 named entities as the bias phrases list. The results are shown in Table 3. It can be seen from the table that the recall of long bias phrases (5-16) is only 91.75% in E9, which is much lower than that of short bias phrases. But after using the prefix tree, the recall is increased to 94.58%. It shows that the prefix tree is important for the performance of the CLAS model on the long bias phrases list.

5.6 Comparison with other models

Since previous works on hotword customization have reported performance on internal test sets, we compare our method with Paraformer-CLAS and SeACo-Paraformer[26], as they also conducted tests on AISHELL-1 and have open-sourced their hotword lists. The hotword list contains a total of 400 words. The results are shown in Table 4. The F1-score of Deep CLAS is 95.25%, which is only slightly lower than the 96% of SeACo-Paraformer+ASF, higher than the 94% of SeACo-Paraformer, and the 91% of Paraformer-CLAS. It is worth noting that the training data for SeACo-Paraformer and Paraformer-CLAS is around 50,000 hours, while the training data for Deep CLAS is only 150 hours. Therefore, the CER of Deep CLAS is higher, but it has achieved a similar F1-score with much less training data, demonstrating the superiority of the method. SeACo-Paraformer did not use strategies such as bias loss, a conformer-based bias encoder, or fine-grained bias encoding, which is why our performance is better.

5.7 Results on other dataset

Since AISHELL-1 is a reading-style speech dataset, to evaluate the adaptability of the Deep CLAS model in different scenarios, we also conducted experiments on the conversational-style MagicData-RAMC dataset. We use LTP[47] to extract 100 named entities from the test set as the list of bias phrases. The results are shown in Table 5. Deep CLAS achieves an F1 score of 81.08%, which is still significantly higher than the 65.50% of CLAS, demonstrating the adaptability of our method.

In addition, we conducted experiments on the English dataset LibriSpeech train-clean-100. We use the Python library spaCy to extract the 100 most frequent person names from the test-clean set as the list of bias phrases. The results are shown in Table 6. Regardless of whether bias is applied or not, the performance of CLAS is worse than that of the baseline LAS. In contrast, Deep CLAS achieves an F1 score of 78.06%

Table 3. Results of long bias phrases list.

Methods	CER (%)	Length	Recall (%)	Precision (%)	F1 (%)
E9	4.40	2-16	94.01	94.22	94.11
		2-4	94.37	94.15	94.26
		5-16	91.75	99.49	95.46
E10: E9+prefix tree	4.30	2-16	94.94	94.66	94.80
		2-4	95.03	94.54	94.78
		5-16	94.58	99.50	96.98

Table 4. Comparison with other models.

Methods	bias	CER (%)	Recall (%)	Precision (%)	F1 (%)
Paraformer-Base[26]	N	5.55	56	99	72
Paraformer-CLAS[26]	Y	3.34	84	99	91
SeACo-Paraformer[26]	Y	2.48	90	99	94
SeACo-Paraformer+ASF[26]	Y	2.27	94	98	96
CLAS	N	9.95	50.61	98.91	66.96
	Y	9.98	50.28	98.68	66.62
Deep CLAS	N	9.95	50.39	97.83	66.52
	0.1	5.78	95.20	95.30	95.25

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

Table 5. Results on MagicData-RAMC.

Methods	bias	CER (%)	Recall (%)	Precision (%)	F1 (%)
LAS	N	18.87	47.30	94.93	63.14
CLAS	N	18.81	47.47	94.29	63.15
	Y	18.83	50.44	93.39	65.50
Deep CLAS	N	18.60	47.82	95.15	63.65
	0.1	18.35	86.67	72.68	79.06
	0.05	18.35	82.14	80.05	81.08

Table 6. Results on LibriSpeech train-clean-100.

Methods	bias	WER (%)	Recall (%)	Precision (%)	F1 (%)
LAS	N	7.78	64.38	94.40	76.55
CLAS	N	8.22	62.60	93.54	75.0
	Y	8.23	62.85	93.56	75.19
Deep CLAS	N	7.59	65.65	96.27	78.06
	0.3	7.46	82.19	80.95	81.57

without bias, which is already higher than the baseline score of 76.55%. With bias, the F1 score reaches 81.57%, representing a relative improvement of 21.4%.

6 Conclusions

In this work, we introduced Deep CLAS, a model that can exploit bias information more effectively than CLAS. We analyzed the shortages of CLAS and made a series of improvements. First, we introduce the bias loss forcing model to focus on contextual information. Second, we use richer information as the query of bias attention. Third, we replace phrase-level encoding with character-level encoding to obtain fine-grained contextual information. Fourth, we explicitly use the bias attention score to utilize contextual information more effectively. Finally, we use the prefix tree to reduce the interference of irrelevant bias phrases. Experiments on the public ASR data demonstrate a significant improvement of Deep CLAS over CLAS.

Conflict of Interest

The authors declare that they have no conflict of interest.

Biography

Mengzhi Wang is currently working at iFLYTEK Research in Hefei. He received his master's degree in Software Engineering from the University of Science and Technology of China in 2019. His research interests focus on automatic speech recognition.

References

[1] Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented transformer for speech recognition. In: Interspeech 2020. ISCA, **2020**: 5036–5040.

[2] Han W, Zhang Z, Zhang Y, et al. ContextNet: Improving convolutional neural networks for automatic speech recognition with global context. In: Interspeech 2020. ISCA, **2020**: 3610–3614.

[3] Radford A, Kim J W, Xu T, et al. Robust speech recognition via large-scale weak supervision. In: Proceedings of the 40th International Conference on Machine Learning. JMLR. org: Honolulu, Hawaii, USA, **2023:** 28492–28518.

[4] Peng Y, Tian J, Chen W, et al. OWSM v3.1: Better and faster open whisper-style speech models based on E-branchformer. In: Interspeech 2024. ISCA, **2024**: 352–356.

[5] Xiong W, Droppo J, Huang X, et al. Toward human parity in conversational speech recognition. *ACM Transactions on Audio, Speech, and Language Processing*, **2017**, *25*: 2410–2423.

[6] Czarnowska P, Ruder S, Grave E, et al. Don't forget the long tail! a comprehensive analysis of morphological generalization in bilingual lexicon induction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Stroudsburg, PA, USA: ACL, **2019**: 974–983.

[7] Deng K, Cheng G, Yang R, et al. Alleviating ASR long-tailed problem by decoupling the learning of representation and classification. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, **2022**, *30*: 340–354.

[8] McGraw I, Prabhavalkar R, Alvarez R, et al. Personalized speech recognition on mobile devices. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Shanghai, China: IEEE, **2016**: 5955–5959.

[9] Shan C, Weng C, Wang G, et al. Component fusion: Learning replaceable language model component for end-to-end speech recognition system. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Brighton, UK: IEEE, **2019**: 5361–5635.

[10] Jung N, Kim G, Chung J S. Spell my name: Keyword boosted speech recognition. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Singapore, Singapore: IEEE, **2022**: 6642–6646.

[11] Vielzeuf V, Antipov G. Are E2E ASR models ready for an industrial usage? arXiv: 2112.12572, **2021.**

[12] Zeineldeen M, Zeyer A, Zhou W, et al. A systematic comparison of grapheme-based *vs*. phoneme-based label units for encoder-decoder-attention models. arXiv: 2005.09336, **2020**.

[13] Fu X, Sathyendra K M, Gandhe A, et al. Robust acoustic and semantic contextual biasing in neural transducers for speech recognition. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Rhodes Island, Greece: IEEE, **2023**: 1–5.

[14] Meng Z, Wu Z, Prabhavalkar R, et al. Text injection for neural contextual biasing. In: Interspeech 2024. ISCA, **2024**: 2985–2989.

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):

[15] Huang K, Zhang A, Yang Z, et al. Contextualized end-to-end speech recognition with contextual phrase prediction network. In: Interspeech 2023. ISCA, **2023**: 4933–4937.
[16] Xu Y, Liu B, Huang Q, et al. CB-conformer: Contextual biasing conformer for biased word recognition. In: ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Rhodes Island, Greece: IEEE, **2023**: 1–5.
[17] Yang Z, Sun S, Wang X, et al. Two stage contextual word filtering for context bias in unified streaming and non-streaming transducer. In: Interspeech 2023. ISCA, **2023**: 3257–3261.
[18] Xu T, Yang Z, Huang K, et al. Adaptive contextual biasing for transducer based streaming speech recognition. In: Interspeech 2023. ISCA, **2023**: 1668–1672.
[19] Shi X, Yang Y, Li Z, et al. SeACo-paraformer: A non-autoregressive ASR system with flexible and effective hotword customization ability. In: ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Seoul, Korea: IEEE, **2024**: 10346–10350.
[20] Yang C, Zheng L, Tian S, et al. Contextual biasing with confidence-based homophone detector for mandarin end-to-end speech recognition. In: Interspeech 2024. ISCA, **2024**: 747–751.
[21] Li Y, Li Y, Zhang M, et al. CB-whisper: Contextual biasing whisper using open-vocabulary keyword-spotting. In: Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). Torino, Italia: Association for Computational Linguistics, **2024**: 2941–2946.
[22] Sudo Y, Shakeel M, Fukumoto Y, et al. Contextualized automatic speech recognition with attention-based bias phrase boosted beam search. In: *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seoul, Korea: IEEE, **2024**: 10896–10900.
[23] Sudo Y, Fukumoto Y, Shakeel M, et al. Contextualized automatic speech recognition with dynamic vocabulary. In: 2024 IEEE Spoken Language Technology Workshop (SLT). Macao, China: IEEE, **2025**: 78–85.
[24] Fang M, Wei T, Guo K, et al. Improving contextual ASR with enhanced phrase-level representation based on MCTC loss. In: ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Hyderabad, India: IEEE, **2025**: 1–5.
[25] Fang M, Guo K, Wei T, et al. Token-level contextual network with ladder-shaped attention for end-to-end ASR. In: ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Hyderabad, India: IEEE, **2025**: 1-5.
[26] Huang K, Zhang A, Zhang B, et al. Spike-triggered contextual biasing for end-to-end mandarin speech recognition. In: 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). Taipei, China: IEEE, **2024**: 1–8.
[27] Pundak G, Sainath T N, Prabhavalkar R, et al. Deep context: End-to-end contextual speech recognition. In: 2018 IEEE Spoken Language Technology Workshop (SLT). Athens, Greece: IEEE, **2018**: 418–425.
[28] Alon U, Pundak G, Sainath T N. Contextual speech recognition with difficult negative training examples. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Brighton, UK: IEEE, **2019**: 6440–6444.
[29] Chen Z, Jain M, Wang Y, et al. Joint grapheme and phoneme embeddings for contextual end-to-end ASR. In: Interspeech 2019. ISCA, **2019**: 3490–3494.
[30] Chan W, Jaitly N, Le Q V, et al. Listen, attend and spell. arXiv: 1508.01211, **2015**.
[31] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, **1997**, *9*: 1735–1780.
[32] Zhang Z, Zhou P. End-to-end contextual asr based on posterior distribution adaptation for hybrid ctc/attention system. arXiv: 2202.09003, **2022**.
[33] See A, Liu P J, Manning C D. Get to the point: Summarization with pointer-generator networks. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, Canada: Association for Computational Linguistics, **2017**: 1073–1083.
[34] Sun G, Zhang C, Woodland P C. Tree-constrained pointer generator for end-to-end contextual speech recognition. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). Cartagena, Colombia: IEEE, **2022**: 780–787.
[35] Le D, Keren G, Chan J, et al. Deep shallow fusion for RNN-T personalization. In: 2021 IEEE Spoken Language Technology Workshop (SLT). Shenzhen, China: IEEE, **2021**: 251–257.
[36] Le D, Jain M, Keren G, et al. Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion. In: Interspeech 2021. ISCA, **2021**: 1772–1776.
[37] Bu H, Du J, Na X, et al. 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). Seoul, Korea: IEEE, **2018**: 1–5.
[38] Chen B, Xu G, Wang X, et al. AISHELL-NER: Named entity recognition from Chinese speech. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Singapore, Singapore: IEEE, **2022**: 8352–8356.
[39] Yang Z, Chen Y, Luo L, et al. Open source MagicData-RAMC: A rich annotated mandarin conversational(RAMC) speech dataset. In: Interspeech 2022. ISCA, **2022**: 1736–1740.
[40] Panayotov V, Chen G, Povey D, et al. Librispeech: An ASR corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). South Brisbane, QLD, Australia: IEEE, **2015**: 5206–5210.
[41] Park D S, Chan W, Zhang Y, et al. SpecAugment: A simple data augmentation method for automatic speech recognition. In: Interspeech 2019. ISCA, **2019**: 2613-2617.
[42] Ko T, Peddinti V, Povey D, et al. Audio augmentation for speech recognition. In: Interspeech 2015. ISCA, **2015**: 3586-3589.
[43] Chorowski J, Bahdanau D, Serdyuk D, et al. Attention-based models for speech recognition. In: Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1. Cambridge, MA, USA: MIT Press, **2015:** 577–585.
[44] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, **2016**: 2818–2826.
[45] Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv: 1412.6980, **2014**.
[46] Vaswani A, Shazeer N, Parmar N, et al. Attention is All you Need. In: Advances in Neural Information Processing Systems, volume 30. Long Beach, CA, USA: Curran Associates, Inc., **2017**.
[47] Che W, Feng Y, Qin L, et al. N-LTP: An open-source neural language technology platform for Chinese. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Online and Punta Cana, Dominican: Association for Computational Linguistics, **2021**: 42–49.

Just Accepted

CSTR: 32290.14.JUSTC-2024-0158
DOI: 10.52396/JUSTC-2024-0158
JUSTC, **2026**, 56(X):