PERL: Pinyin Enhanced Rephrasing Language Model for Chinese ASR N-best Error Correction

Junhong Liang
Mohamed Bin Zayed University of Artificial Intelligence
Abu Dhabi, UAE
junhong.liang@mbzuai.ac.ae

Bojun Zhang
Institute of Automation, Chinese Academy of Sciences
Beijing, China
zhangbojun2022@ia.ac.cn

***Abstract*—Chinese ASR correction is challenging because errors are often *phonetic* (many characters share similar Pinyin) while the correction model must also obey a *length constraint* under noisy N-best hypotheses. Existing approaches either exploit Pinyin only at the prompt/feature level without integrating it into model representations or rely on generative decoding that can drift in length. We propose PERL, a constrained rephrasing pipeline for Chinese N-best ASR correction that (i) predicts the target length and enforces it via mask budgeting, and (ii) fuses *semantic* and *phonetic* (Pinyin) representations through token-wise gates conditioned on sentence semantics. Experiments on Aishell-1 and our new domain N-best benchmark DoAD show that PERL consistently reduces CER (29.11% on Aishell-1 and up to ∼70% on DoAD) while maintaining low latency. We also provide analyzes of length generalization and phonetic–semantic interactions, showing when PERL relies on phonetic cues versus semantic constraints.**

***Index Terms*—ASR, Error Correction**

I. Introduction

Automatic Speech Recognition (ASR) has been widely adopted in applications such as voice interaction and information retrieval. Despite advances in end-to-end models [1], ASR output remains vulnerable to accents, background noise, and speaker variability. These errors not only reduce recognition quality, but also propagate to downstream tasks, motivating the need for practical post-correction modules.

In Mandarin Chinese, ASR errors often follow phonetic patterns, as many characters share similar pronunciations. This is closely related to the Chinese phonetic system, *Pinyin*, a Romanized representation of character pronunciations. Previous work on Chinese Spelling Correction (CSC) has demonstrated that incorporating phonological features improves error correction [2]–[5]. Extending this idea to ASR, recent studies have integrated Pinyin features as explicit prompts for LLMs [6]. However, to bridge the gap between text and phonetic modalities, it is essential to incorporate phonological cues directly into model representations rather than relying solely on explicit prompts.

Existing approaches are largely dependent on limited embeddings or generative models, which cannot guaranty fixed-length outputs and are thus less suitable for ASR correction [7]–[9]. Although large language models (LLMs) have achieved strong results in CSC [10], their decoder-only structure tends to produce predictions of variable-length. Unlike grammatical error correction, where flexible generation is acceptable [11], or spelling correction, which typically assumes fixed-length outputs [12], ASR correction requires a constrained setting in which the model must infer the correct sentence length from noisy hypotheses. Existing methods have explored constrained decoding [13], alignment algorithms [14], and LLM-based re-ranking [15], but balancing phonetic and semantic information under strict length constraints remains an open challenge.

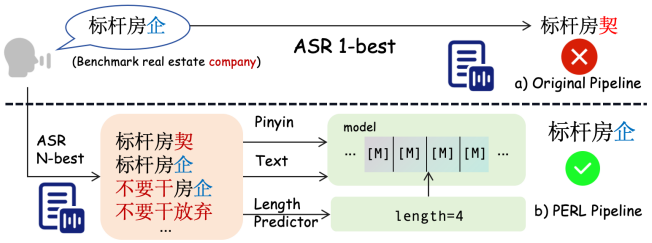

Fig. 1. Comparison between the standard ASR pipeline (a) and our proposed PERL pipeline (b). Our proposed pipeline employs ASR N-best as the input and incoporates length information to determine masked tokens, and jointly employ Pinyin and Semantic information for error correction.

To tackle these challenges, we propose the **Pinyin Enhanced Rephrasing Language (PERL)** pipeline (Figure 1), which improves ASR correction by jointly modeling semantic and phonetic information. In addition, to comprehensively evaluate error correction in domain-specific scenarios, we introduce the **Do**main N-best **A**SR **D**ataset (DoAD). Our key contributions can be summarized as follows:

1) We propose **PERL**, a constrained rephrasing framework for Chinese ASR N-best error correction that explicitly models the target length via a dedicated predictor and enforces it through mask-based generation.
2) We design a **phonetic–semantic fusion mechanism** that integrates Pinyin-based phonological representations with contextual semantic embeddings via sentence-conditioned token-wise gating.
3) We construct **DoAD**, a domain-specific N-best ASR benchmark with a high proportion of unequal-length cases, providing a challenging testbed for length-constrained correction.

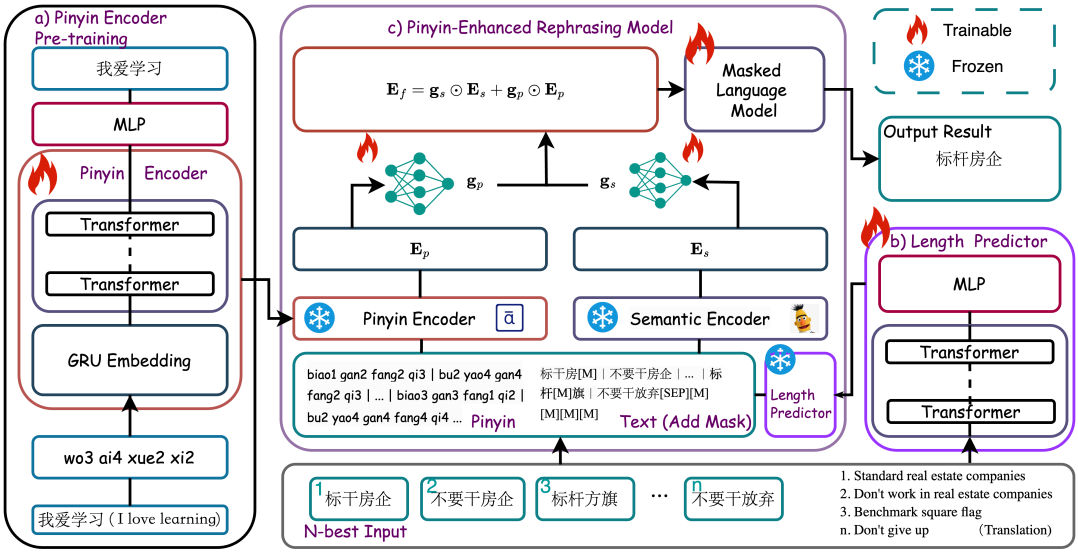

Fig. 2. The overall architecture of the proposed model comprises three main components: (a) a pre-trained Pinyin encoder (left), which generates phonetic embeddings; (b) a length predictor (right), which estimates the target output length from the N-best inputs; and (c) a Pinyin-Enhanced Rephrasing Model (center), which integrates phonetic and semantic embeddings to perform masked token prediction. The Pinyin encoder and the length predictor are trained in the first two stages, after which their parameters are frozen and incorporated into the training of the rephrasing model.

II. Method

Our method consists of three main stages: 1) pretrain a Pinyin encoder to capture phonological information (Section II-A), 2) pretrain a length predictor to handle the variable-length issue in ASR correction (Section II-B), and 3) train a rephrasing model that integrates semantic and phonetic features for the final correction (Section II-C). Figure 2 provides an overview.

A. Pinyin Encoder

The Chinese Pinyin system provides a Latin-based transcription of Chinese characters that carries rich phonological information. To exploit this, we design a Pinyin encoder that maps a character sequence to phonetic embeddings. Given a sentence $x = (x_1, x_2, \ldots, x_n)$ with n characters, the encoder converts each character x_i into a Pinyin sequence $p_{i,1}, \ldots, p_{i,l_i}$, where l_i is the length of the transcription. The encoder then produces the phonetic representation $\mathbf{E}_p = \text{Enc}_p(x)$. It operates in three steps:

a) Pinyin embedding: Each token $p_{i,j}$ is assigned to a vector representation $\mathbf{p}_{i,j}$.

b) Sequential encoding: A uni-directional GRU encodes the Pinyin sequence $\hat{\mathbf{p}}_{i,j} = \text{GRU}(\hat{\mathbf{p}}_{i,j-1}, \mathbf{p}_{i,j})$. The last hidden state $\hat{\mathbf{p}}_{i,l_i}$ is used as the representation of the character x_i, forming the sequence $\mathbf{E}_{p,0} = (\hat{\mathbf{p}}_{1,l_1}, \hat{\mathbf{p}}_{2,l_2}, \ldots, \hat{\mathbf{p}}_{n,l_n})$.

c) Global encoding: To capture long-range dependencies, we apply L_p Transformer layers $\mathbf{E}_{p,l} = \text{Transformer}(\mathbf{E}_{p,l-1}), \quad l \in [1, L_p]$

The final phonetic representation $\mathbf{E}_{p,L_p}$ is projected to predict the corresponding Chinese characters using learnable parameters $\mathbf{W}^p$ and $\mathbf{b}^p$. For position i, the prediction is defined as

$$\hat{y}_i = \text{Softmax}(\mathbf{W}^p \mathbf{E}_{p,L_p,i} + \mathbf{b}^p), \quad \mathbf{E}_{p,L_p,i} \in \mathbf{E}_{p,L_p}.$$

The encoder is pre-trained with a cross-entropy loss, where y_i denotes the ground-truth character at position i:

$$\mathcal{L}_p = -\frac{1}{n} \sum_{i=1}^{n} y_i \log \hat{y}_i.$$

B. Length Predictor

Chinese N-best ASR correction is a *length-constrained* rephrasing problem: the corrected output may be shorter or longer than any single hypothesis, and an incorrect length budget can propagate errors to the downstream mask-filling stage. We therefore introduce a dedicated length predictor to estimate the target character length from the N-best list.

a) Input: Given an N-best list $\{S_k\}_{k=1}^{n}$, we concatenate hypotheses with a delimiter "|":

$$S_{\text{concat}} = S_1 \mid S_2 \mid \ldots \mid S_n.$$

The ground-truth label is the character length l of the reference sentence. We set the maximum length to L (e.g., L=128) and clamp labels into $[1, L]$.

b) Ordinal length modeling: A flat multi-class classifier over lengths can be brittle when the test-time length distribution shifts across domains. To improve generalization, we model length as an *ordinal* variable. Specifically, we encode S_{concat} using a BERT encoder and take the [CLS] embedding $\mathbf{h} \in \mathbb{R}^d$:

$$\mathbf{h} = \text{Enc}_s(S_{\text{concat}}).$$

For each threshold $m \in \{1, \ldots, L\}$, the predictor estimates the cumulative probability

$$p_m = P(l \geq m \mid S_{\text{concat}}) = \sigma(\mathbf{w}_m^\top \mathbf{h} + b_m),$$

where $\sigma(\cdot)$ is the sigmoid function. The supervision signal for the threshold m is $y_m = 1[l \geq m]$. We train the predictor using a sum of binary cross-entropy losses across thresholds:

$$\mathcal{L}_{\text{len}} = -\sum_{m=1}^{L} \Big(y_m \log p_m + (1 - y_m) \log(1 - p_m) \Big).$$

This formulation explicitly exploits the natural ordering of length values and is less sensitive to unseen lengths than a flat softmax classifier.

c) Inference with length uncertainty: Instead of relying on a single point estimate, we decode with a small candidate set of lengths to mitigate prediction uncertainty. We first recover a discrete distribution over lengths by

$$P(l = m) = \begin{cases} 1 - p_2, & m = 1, \\ p_m - p_{m+1}, & 1 < m < L, \\ p_L, & m = L, \end{cases}$$

and take the top-K lengths $\mathcal{C} = \{l^{(1)}, \ldots, l^{(K)}\}$ For each candidate length $l^{(k)}$, we run the downstream rephrasing model (Section II-C) with $l^{(k)}$ appended masks, and select the final output with the lowest loss in masked language modeling (i.e., the highest likelihood in masked positions). This adds only a small overhead, while improving robustness under the length distribution shift.

C. Pinyin-Enhanced Rephrasing Model

The final correction model integrates semantic and phonetic information while incorporating the predicted length. It consists of two stages: input processing and feature encoding/prediction.

a) Input processing: The N-best hypotheses are concatenated into S_{concat}. The frozen length predictor is then used to estimate the target length $\hat{l}$ (Section II-B). To align with this prediction, $\hat{l}$ mask tokens are appended to S_{concat}. Following [16], we further apply a dynamic masking strategy, where a subset of tokens in S_{concat} is randomly replaced with the [M] symbol, yielding the masked sequence S_{masked}.

b) Feature encoding: The masked sequence S_{masked} is processed by the semantic encoder, while the concatenated sentence S_{concat} is passed through the Pinyin encoder; the parameters for the Pinyin encoder in Section II-A will be frozen.

$$\mathbf{E}_s = \text{Enc}_s(S_{\text{masked}}), \quad \mathbf{E}_p = \text{Enc}_p(S_{\text{concat}}).$$

The sentence-level semantic embedding is obtained by averaging token embeddings over the maximum sequence length:

$$\bar{\mathbf{e}}_s = \tfrac{1}{l_{\max}} \sum_{i=1}^{l_{\max}} \mathbf{E}_s[i],$$

where $l_{\max}$ denotes the maximum sentence length and $\mathbf{E}_s[i]$ is the semantic embedding of the i-th token.

c) Feature fusion: For each token, semantic and phonetic features are combined through learned gating weights:

$$g_{s,i} = \sigma\big(\text{MLP}_s\big([\mathbf{E}_s[i]; \mathbf{E}_p[i]; \bar{\mathbf{e}}_s]\big)\big),$$
$$g_{p,i} = \sigma\big(\text{MLP}_p\big([\mathbf{E}_s[i]; \mathbf{E}_p[i]; \bar{\mathbf{e}}_s]\big)\big).$$

where $[\cdot;\cdot]$ denotes vector concatenation, MLP_s and MLP_p are two-layer perceptrons with ReLU activation, and $\sigma(\cdot)$ is the sigmoid function.

The final fused embedding is computed as

$$\mathbf{E}_f[i] = g_{s,i}\mathbf{E}_s[i] + g_{p,i}\mathbf{E}_p[i].$$

d) Prediction: The fused representation $\mathbf{E}_f$ is passed to a mask prediction model to predict masked tokens. The model is trained with a cross-entropy objective over masked positions:

$$\mathcal{L}_{\text{mask}} = -\sum_{i \in \mathcal{M}} y_i \log \hat{y}_i,$$

where $\mathcal{M}$ denotes the set of masked indices and y_i is the ground-truth label.

III. Experimental Setup

A. Data

We perform experiments on the ChineseHP/Aishell-1 (CHP/Aishell-1) dataset [6] and construct **the Domain ASR** dataset (DoAD). The construction process is summarized as follows:

a) Data Processing: We start with ECSpell data [17], a text domain error correction dataset. Each gold-standard sentence is segmented by punctuation and normalized to remove Arabic numerals and punctuation marks.

b) Text-to-Speech: Normalized sentences are converted into speech using Microsoft Azure[1]. To simulate noisy conditions, white noise between 10 dB and 20 dB is added randomly.

c) ASR Generation: A Whisper ASR model [1] is used to generate N-best hypotheses. Specifically, we adopt the Belle-distilwhisper-large-v2-zh model [18].

Table I reports dataset statistics. Compared to Aishell-1, DoAD contains a much larger proportion of unequal-length sentences, which poses additional challenges for LLMs to recover the correct sequence length. We attribute this to the noise, which makes ASR correction more difficult than in clean speech. Although DoAD is synthetically constructed using TTS and noise injection, it enables controlled generation of unequal-length ASR errors. However, it may not fully capture real-world phenomena such as disfluencies, accents, or spontaneous speech patterns, which we leave for future work.

B. Experiment Setup and Evaluation

The experiments are conducted on two NVIDIA 3090 GPUs. The Pinyin encoder is pre-trained on the Wang271k [19] gold-labeled dataset, with 10,000 sentences held out for validation. In principle, any error-free Chinese corpus could

[1]https://learn.microsoft.com/zh-cn/azure/ai-services/speech-service/text-to-speech

TABLE I

DATASET STATISTICS. #SEN DENOTES THE NUMBER OF SENTENCES, #LEN THE AVERAGE SENTENCE LENGTH, AND #EQUAL THE NUMBER OF SENTENCES WHERE THE 1-BEST OUTPUT MATCHES THE REFERENCE LENGTH. BOTH TRAINING AND TEST SETS ARE REPORTED.

Dataset	#Sen (Train/Test)	#Len	#Equal
DoAD-Law	4,040 / 1,010	13.69 / 13.64	2,469 / 263
DoAD-Med	10,978 / 1,820	11.30 / 11.55	5,878 / 1,032
DoAD-Odw	5,137 / 1,480	12.45 / 12.41	2,502 / 696
CHP/Aishell1	120,099 / 7,176	14.41 / 14.60	113,039 / 6,699

be used for this pre-training. The encoder is trained for 10 epochs with an initial learning rate of 5×10^{-4}. The Aishell-1 and DoAD datasets are used to train the length predictor, starting with a learning rate 2×10^{-5}. Finally, PERL training follows the same training parameters shown in [16]. The mask rate is set to 0.2, and the maximum length is 128. We choose the BERT model [20] as the semantic encoder and select the 5-best results as input for each scenario.

For evaluation, we adopt Character Error Rate (CER) and Character Error Rate Reduction (CERR). CER is computed as the edit distance between the predicted and reference sequences normalized by the reference length, while CERR measures the relative reduction of CER compared to the best ASR baseline.

C. Baselines

To thoroughly investigate the effectiveness of our proposed model, we select several representative pretrained language models (PLMs) and large language models (LLMs) as baselines. For PLMs, we employ **MacBERT** [21], a Chinese pre-trained model that improves on BERT by using a masked correction pretraining strategy, where original tokens are replaced with semantically similar words instead of the special mask symbol [22]. We also include **T5** [23], a unified text-to-text Transformer that reformulates every NLP problem as a sequence-to-sequence task, allowing flexible adaptation to classification, summarization, and error correction. In addition, we use **ReLM** [16], the Rephrasing Language Model trained with a mask filling objective, to rephrase entire sentences rather than detecting errors token by token, emphasizing fluency and natural rewording.

For LLMs, we benchmark against **GPT-4o**, a generative model developed by OpenAI and optimized from the GPT-4 architecture with state-of-the-art reasoning ability, **DeepSeek-V2.5**, which demonstrates strong capabilities in coding and mathematical reasoning, and **Qwen 2.5**, a large multilingual model pre-trained on diverse corpora. For generative baselines, we use a zero-shot prompt setup following [6], where the N-best hypotheses are provided as input and the model is instructed to generate a corrected sentence. We do not apply additional constrained decoding or length control for LLMs.

TABLE II

PERFORMANCE OF OUR POST-CORRECTION MODEL UNDER DIFFERENT ASR SYSTEMS. O_{cp} AND O_{nb} DENOTE THE CHARACTER-LEVEL ORACLE AND THE N-BEST SELECTION ORACLE, RESPECTIVELY. THE BEST RESULTS ARE SHOWN IN **BOLD**.

Method	CER%↓ $_{-\text{CERR\%}\downarrow}$			
	CHP/Aishell1	DoAD-Law	DoAD-Med	DoAD-Odw
Baseline	5.84	12.64	16.09	15.94
O_{cp}	$2.93_{-49.83}$	$3.83_{-69.70}$	$6.13_{-61.90}$	$4.41_{-73.59}$
O_{nb}	$3.49_{-32.70}$	$9.44_{-25.31}$	$12.66_{-28.37}$	$11.89_{-28.80}$
MacBERT	$5.06_{-13.36}$	$13.16_{+4.11}$	$17.28_{+7.40}$	$16.76_{+5.14}$
T5	$5.49_{-6.00}$	$14.34_{+13.45}$	$18.52_{+15.10}$	$18.49_{+16.00}$
ReLM	$5.20_{-10.96}$	$5.21_{-58.78}$	$6.91_{-57.05}$	$6.17_{-61.23}$
GPT-4o	$4.41_{-24.49}$	$9.83_{-22.23}$	$13.27_{-17.53}$	$12.40_{-22.20}$
DeepSeek	$4.25_{-27.22}$	$9.12_{-27.85}$	$12.61_{-21.63}$	$11.68_{-26.72}$
Qwen2.5	$4.22_{-27.74}$	$8.28_{-34.49}$	$12.16_{-24.43}$	$11.15_{-30.05}$
PERL	$\mathbf{4.10}_{-29.11}$	$\mathbf{3.41}_{-73.02}$	$\mathbf{3.91}_{-75.70}$	$\mathbf{4.87}_{-69.44}$

IV. RESULTS

A. N-best ASR Correction

We utilize the ChineseHP [6] dataset for Aishell-1 and perform our own ASR on distil-whisper [18]. Table II shows the N-best ASR correction result, where CER represents the average character error rate, and CERR represents the percentage decrease in the average character error rate. o_{cp} represents the best character combination oracle and o_{nb} represents the best selection oracle.

Under our experimental framework, the baseline model exhibited relatively high CER in the law, medical, and official document domains of the DoAD dataset compared to Aishell-1, highlighting the challenge posed to our model. Empirical results show that PERL substantially reduces CER across all datasets, achieving a 29.1% reduction on Aishell-1 and around 70% across different domains on DoAD. Notably, the improvement on DoAD can outperform a character-level oracle that is restricted to selecting characters appearing in the N-best list, suggesting PERL benefits from contextual rephrasing beyond per-position selection. This gain can be attributed to the ability of PERL to explore a broader semantic space and take advantage of contextual dependencies, which is particularly effective when the DoAD beam search produces low-quality hypotheses, demonstrating the robustness of our proposed framework.

In contrast, LLMs perform worse than PERL in DoAD. We attribute this to the inherent limitations of generative models when handling tasks with length constraints, where noisy inputs often mislead them into producing erroneous output.

B. Result Analysis

The choice of n. For each erroneous character in the 1-best hypothesis, we calculate the proportion of n-best candidates that contain its correct counterpart. We define this metric as *Wrong Character Coverage (WCC)*, which measures how often the correct character appears within the n-best list. Figure 3 reports the WCC performance of our proposed PERL model under different n settings on the ChineseHP/Aishell-1 dataset. We observe that the Character Error Rate (CER) reaches its minimum when $n = 5$. Interestingly, although

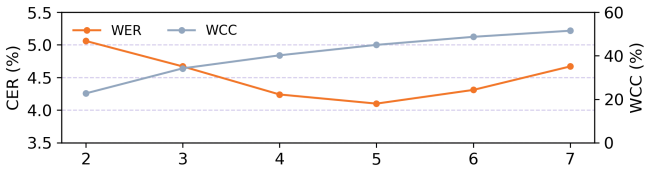

Fig. 3. Character Error Rate (CER) and Wrong Character Coverage (WCC) across different n-best settings. The left y-axis shows CER, while the right y-axis shows WCC.

TABLE III
CER OF WHISPER MODELS ACROSS DIFFERENT DOMAINS OF DOAD.

Model	Domain	Baseline	PERL	o_{nb}	o_{cp}
whisper-large-v3	law	15.81	5.65	9.24	7.67
	med	15.86	7.20	10.37	9.15
	odw	14.75	7.61	9.09	8.04
whisper-small	law	26.76	10.77	18.01	14.72
	med	29.69	13.22	20.82	17.81
	odw	30.02	12.82	19.93	16.51

TABLE IV
LENGTH CORRECTION ON DIFFERENT WHISPER MODELS.

Model	Method	#Equal		
		Law	Med	Odw
whisper-large-v3	1-best	910	1641	1393
	length predict	954	1685	1410
whisper-small	1-best	853	1551	1289
	length predict	913	1650	1310

WCC continues to increase to 6, the CER worsens. This suggests that simply expanding n does not guaranty a better correction and that effectiveness depends on the ability of the model to leverage additional hypotheses and processing long inputs.

ASR Model Selection. To assess the robustness of our model across different ASR systems, we evaluate it on Belle-distilwhisper-large-v2-zh, as detailed in Section III-A, as well as on whisper-small and whisper-large-v3 to process the DoAD speeches. Specifically, we measure the CER of the PERL on these systems. The results in Table III demonstrate that PERL substantially reduces CER compared to the 1-best baselines, highlighting its strong robustness in various ASR models.

Length Prediction. To assess the significance of the length prediction module, we evaluate our model's length correction performance, as shown in Table IV. The pre-trained length predictor demonstrates improved accuracy in predicting the correct length compared to relying solely on the 1-best result. This enhancement enables the model to determine the masked tokens, improving overall prediction.

Ablation Experiments. We evaluate PERL on the Aishell-1 and DoAD datasets under different ablation settings as shown in Table VI: (1) removing the length predictor and using only the length of the 1-best prediction, (2) removing the Pinyin encoder, and (3) using only the 1-best prediction during correction. Our findings show that removing the Pinyin encoder or relying solely on the N-best results significantly degrades performance. Furthermore, removing the length predictor during correction results in even more significant performance degradation, as the language model needs to rephrase the sentence into shorter or longer sequences.

TABLE V
EXAMPLE OF ASR N-BEST CORRECTION FOR DIFFERENT MODELS, WHERE WE USE RED/BLUE/ORANGE COLORS TO REPRESENT WRONG/CORRECT/WRONG EDIT TOKENS, BEST VIEWED IN COLOR.

5-Best Sentences	**Model Outputs**
目前挂牌的只有几松土地	目前挂牌的只有几宗土地 (PERL)
目前挂排的只有几松土地	目前挂牌的只有几松土地 (1-best)
目前挂牌的只有几松土	目前挂牌的只有几块土地 (ReLM)
目前挂牌的只有几宋土地	目前挂牌的只有几宗土地 (GPT-4o)
目前挂牌的只有几宗土地	目前挂牌的只有几宗土地 (Gold)
Translation: Currently, only a few sects of land are listed for sale.	

TABLE VI
ABLATION OF PERL MODEL ON CHINESEHP/AISHELL-1 AND DOAD DATASETS WITH DIFFERENT SETTINGS.

Method	**CHP/Aishell-1**	**Law**	**Med**	**Odw**
PERL	4.10	3.41	3.91	4.87
- w/o len	7.28	5.08	5.47	7.12
- w/o pho	4.78	4.27	4.67	5.83
- w/o N-best	4.31	3.69	4.62	5.15

Case Study. ASR N-best Correction example is shown in Table V, PERL could calculate the correct length and incorporate Pinyin knowledge to recover the wrong token 松 (sōng, pine) into 宗 (zōng, sect) since they share similar Pinyin. At the same time, the ReLM model failed by correcting this in 块 (kuài, piece) since it cannot use the Pinyin information, and the 1-best result is unable to provide potential corrections compared to the N-best results.

Latency Analysis. Beyond correction accuracy, inference efficiency is also evaluated, a key requirement for deployment in real-world ASR scenarios. On a single NVIDIA 3090 GPU, lightweight encoder-based PLMs achieve the fastest speeds per input (MacBERT: 1.67 ms, ReLM: 2.87 ms). The PERL pipeline introduces only minor overhead (3.09 ms) while lowering CER, offering a strong balance between speed and accuracy. In contrast, large generative LLMs are substantially slower (hundreds of ms) and thus less practical for latency-sensitive ASR correction.

V. CONCLUSION

PERL integrates a Pinyin module to reduce phonetic errors and a length predictor to address sequence inconsistencies in N-best Chinese ASR error correction. To evaluate performance under challenging conditions, we introduce the highly noisy DoAD dataset. Experiments demonstrate that PERL effectively leverages semantic and phonetic features to correct ASR errors while maintaining low latency, making it both accurate and practical for real-world applications.

REFERENCES

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022. [Online]. Available: https://arxiv.org/abs/2212.04356

[2] H.-D. Xu, Z. Li, Q. Zhou, C. Li, Z. Wang, Y. Cao, H. Huang, and X.-L. Mao, "Read, listen, and see: Leveraging multimodal information helps chinese spell checking," 2021. [Online]. Available: http://arxiv.org/abs/2105.12306

[3] J. Li, Q. Wang, Z. Mao, J. Guo, Y. Yang, and Y. Zhang, "Improving Chinese spelling check by character pronunciation prediction: The effects of adaptivity and granularity," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4275–4286. [Online]. Available: https://aclanthology.org/2022.emnlp-main.287

[4] X. Wu and Y. Wu, "From spelling to grammar: A new framework for Chinese grammatical error correction," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 889–902. [Online]. Available: https://aclanthology.org/2022.findings-emnlp.63

[5] J. Liang, Z. Junnan, F. Zhai, N. Cheng, C. Zong, and Y. Zhou, *A Hybrid Approach towards Chinese Spelling and Splitting Error Correction*. ECAI, 10 2024, pp. 3883–3890.

[6] Z. Tang, D. Wang, S. Huang, and S. Shang, "Pinyin regularization in error correction for chinese speech recognition with large language models," in *Interspeech 2024*. ISCA, pp. 1910–1914.

[7] P. Xing, Y. Li, S. Ma, X. Liang, H. Huang, Y. Li, H.-T. Zheng, W. Jiang, and Y. Shen, "Mitigating catastrophic forgetting in multi-domain chinese spelling correction by multi-stage knowledge transfer framework," 2024.

[8] K. Li, Y. Hu, L. He, F. Meng, and J. Zhou, "C-llm: Learn to check chinese spelling errors character by character," 2024. [Online]. Available: https://arxiv.org/abs/2406.16536

[9] J. Liang and Y. Zhou, "Rair: Retrieval-augmented iterative refinement for chinese spelling correction," 2025. [Online]. Available: https://arxiv.org/abs/2504.18938

[10] Y. Li, H. Huang, S. Ma, Y. Jiang, Y. Li, F. Zhou, H.-T. Zheng, and Q. Zhou, "On the (in)effectiveness of large language models for chinese text correction," 2023. [Online]. Available: https://arxiv.org/abs/2307.09007

[11] Y. Zhang, Z. Li, Z. Bao, J. Li, B. Zhang, C. Li, F. Huang, and M. Zhang, "MuCGEC: a multi-reference multi-source evaluation dataset for chinese grammatical error correction," in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022, pp. 3118–3130. [Online]. Available: https://aclanthology.org/2022.naacl-main.227

[12] S. Zhang, H. Huang, J. Liu, and H. Li, "Spelling error correction with soft-masked bert," 2020.

[13] R. Ma, M. J. F. Gales, K. M. Knill, and M. Qian, "N-best t5: Robust asr error correction using multiple input hypotheses and constrained decoding space," in *INTERSPEECH 2023*, ser. Interspeech 2023. ISCA, August 2023, pp. 3267–3271. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2023-1616

[14] Y. Leng, X. Tan, R. Wang, L. Zhu, J. Xu, W. Liu, L. Liu, X.-Y. Li, T. Qin, E. Lin, and T.-Y. Liu, "FastCorrect 2: Fast error correction on multiple candidates for automatic speech recognition," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, M.-F. Moens, X. Huang, L. Specia, and S. W.-t. Yih, Eds. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4328–4337. [Online]. Available: https://aclanthology.org/2021.findings-emnlp.367

[15] J. Pu, T.-S. Nguyen, and S. Stuker, "Multi-stage large language model correction for speech recognition," 2024. [Online]. Available: https://arxiv.org/abs/2310.11532

[16] L. Liu, H. Wu, and H. Zhao, "Chinese spelling correction as rephrasing language model," 2024. [Online]. Available: https://arxiv.org/abs/2308.08796

[17] Q. Lv, Z. Cao, L. Geng, C. Ai, X. Yan, and G. Fu, "General and domain adaptive chinese spelling check with error consistent pretraining," vol. 22, no. 5, pp. 1–18. [Online]. Available: http://arxiv.org/abs/2203.10929

[18] BELLEGroup, "Belle: Be everyone's large language model engine," https://github.com/LianjiaTech/BELLE, 2023.

[19] D. Wang, Y. Song, J. Li, J. Han, and H. Zhang, "A hybrid approach to automatic corpus generation for Chinese spelling check," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 2517–2527. [Online]. Available: https://aclanthology.org/D18-1273

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, "Revisiting pre-trained models for Chinese natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. Online: Association for Computational Linguistics, Nov. 2020, pp. 657–668. [Online]. Available: https://www.aclweb.org/anthology/2020.findings-emnlp.58

[22] X. Ming, "pycorrector: Text error correction tool," 2021. [Online]. Available: https://github.com/shibing624/pycorrector

[23] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2023. [Online]. Available: https://arxiv.org/abs/1910.10683