CAR: Cost-Aware Adaptive Routing for Chinese Spelling Correction

Anonymous ACL submission

Abstract

Large language models (LLMs) demonstrate strong generalization for Chinese Spelling Correction (CSC), particularly on out-of-domain and rare error patterns. However, applying LLMs to every input is computationally expensive and redundant, as many instances can be reliably corrected by lightweight BERT-based models, and full LLM invocation may even introduce over-corrections. In this work, we propose CAR, a risk-aware adaptive routing framework that selectively invokes an LLM only for high-risk samples. CAR estimates correction reliability using multi-dimensional risk features, including edit-based metrics, model uncertainty, and semantic consistency, and employs a two-stage regularized decision model to identify cases where LLM verification is most beneficial. Extensive experiments on SIGHAN13 and CSCDNS demonstrate that CAR achieves superior correction performance compared to both the base model and full LLM invocation. It suggests that our method is a practical way to balance correction quality and LLM inference cost for deployable CSC systems.

1 Introduction

Chinese Spelling Correction (CSC) aims to detect and correct erroneous Chinese characters while preserving the intended meaning of the input sentence (Yu and Li, 2014; Xiong et al., 2015). It is widely used in search, intelligent writing, content moderation, and OCR/ASR post-processing. In realistic applications, CSC systems need to process large volumes of text under strict latency and cost constraints, while also handling domain shifts, informal expressions, and newly emerging error patterns.

Existing CSC methods can be broadly divided into lightweight pre-trained model (PLM)-based correctors and Large language model (LLM)-based correctors. BERT-style models, such as Soft-Masked BERT (Zhang et al., 2020) and MacBERT (Cui et al., 2020), are efficient and perform strongly on in-domain benchmarks (Wu et al., 2023; Hu et al., 2024). However, their performance often degrades under distributional shifts or with complex multi-character errors. LLMs, in contrast, leverage broader linguistic knowledge and stronger contextual reasoning, yielding better generalization on out-of-domain or rare error cases. Nevertheless, invoking an LLM for every input is computationally expensive, often redundant, and may even introduce over-corrections on simple instances(Li et al., 2023).

This raises a fundamental question for deployable CSC systems: should every CSC input be sent to an LLM? In many cases, the lightweight corrector already produces a reliable correction, and invoking the LLM is redundant. In other cases, especially when the base corrector is uncertain, edits too aggressively, or changes the sentence semantics, LLM verification can be valuable. The key challenge is therefore not only improving correction model accuracy, but also deciding when LLM resources should be allocated.

In this work, we propose CAR, a Cost-Aware Adaptive Routing framework for CSC. Given an input sentence, a lightweight base corrector first produces an initial correction and intermediate confidence evidence. A risk-aware router then estimates the reliability of the base output using edit, uncertainty, and semantic consistency features. Low-risk samples directly use the base output, while high-risk samples are routed to an LLM corrector. Unlike ACI-style collaborative training pipelines (Liu et al., 2025), our method focuses on inference-time selective invocation rather than iterative data generation. Unlike general LLM routing methods (Ding et al., 2024), our router is tailored to the structure of CSC errors and base-corrector outputs.

Our contributions are as follows:

1. We formulate cost-aware CSC as an adaptive routing problem that balances correction quality and LLM invocation cost.

2. We design a CSC-specific risk feature set covering edit behavior, uncertainty, and semantic consistency.

3. We propose a two-stage regularized routing model that combines L1-based feature selection and L2-based imbalanced classification to identify high-risk samples.

4. We conduct experiments and analyses on SIGHAN13 and CSCDNS, including cost-effectiveness, decision effectiveness, feature ablations, and case analysis.

2 Related Work

Base model for Chinese Spelling Correction Early CSC methods have been widely studied under rule-based, statistical, and neural paradigms. Some approaches rely on handcrafted rules, confusion sets, and statistical language models (Chiu et al., 2013; Yeh et al., 2014), while offering limited robustness to contextual and semantic errors. With the advent of pre-trained language models, recent methods such as Soft-Masked BERT (Zhang et al., 2020) and MacBERT (Cui et al., 2020) significantly improve CSC performance by leveraging contextual representation learning. Subsequent work further explores phonological and visual similarity modeling, as well as detector–corrector architectures to enhance robustness under noisy inputs. However, most PLM-based methods lack adaptability to varying error difficulty levels. As a result, they often struggle with under-correction in complex scenarios, especially under domain shift.

LLM for Chinese Spelling Correction Recently, LLMs have been explored for Chinese text correction through various approaches, including character-level constrained decoding to preserve sentence length (Li et al., 2024), training-free prompt-free methods that leverage LLMs' inherent language modeling ability (Zhou et al., 2024), alignment-and-replacement modules that align LLM outputs with original sentences (Liu et al., 2024), and retrieval-augmented generation for few-shot scenarios (Dong et al., 2025). Due to their strong reasoning and world knowledge, LLMs demonstrate improved generalization on rare or out-of-distribution error patterns. However, LLMs are not always reliable for CSC tasks, as they may introduce hallucinated corrections or over-correction issues, especially when the input is already correct or only contains minor perturbations. More critically, full LLM invocation for all inputs leads to substantial computational overhead, making it impractical for real-world high-throughput deployment scenarios.

Model Collaboration and Routing To balance efficiency and performance, recent studies explore model collaboration and routing strategies. For example, ACI (Liu et al., 2025) uses an LLM verifier to iteratively optimize a BERT-based corrector through high-quality training data construction. Another line of work explores combining small and large models during decoding, such as MSLLM (Qiao et al., 2025), which dynamically mixes the output probability distributions of lightweight models and LLMs. General routing methods allocate inputs to models of different capacities based on estimated difficulty or utility (Ding et al., 2024). However, they lack explicit modeling of the task-specific correction risk inherent in CSC.

Summary of Research Gap In summary, Existing CSC systems suffer from three fundamental limitations: lightweight models have limited generalization, LLM-based systems are expensive and prone to over-correction, and generic routing strategies lack CSC-specific risk modeling. This motivates the need for a cost-aware, CSC-specific adaptive routing framework that can explicitly model correction risk and dynamically determine whether LLM intervention is necessary.

3 Method

3.1 Task Formulation

Given an input sentence $X = (x_1, x_2, \ldots, x_n)$, CSC predicts a corrected sentence $Y = (y_1, y_2, \ldots, y_n)$ that should match the gold correction $C = (c_1, c_2, \ldots, c_n)$ while preserving the original meaning. Let M denote a lightweight base corrector, L denote an LLM corrector, and $\mathcal{D}$ denote the binary routing decision function. The final prediction is defined as:

$$Y = \begin{cases} M(X), & \mathcal{D}(X, M(X), \tilde{M}(X)) = 0, \\ L(X), & \mathcal{D}(X, M(X), \tilde{M}(X)) = 1, \end{cases} \tag{1}$$

where $\tilde{M}(X)$ denotes intermediate evidence from the base corrector, such as top-k candidate probabilities and edit operations. The routing function outputs 1 for high-risk samples that require LLM invocation and 0 for low-risk samples that can directly accept the base correction.

The objective is to maximize correction quality under an LLM invocation budget:

$$\begin{aligned} \max_{D} \quad & \mathbb{E}\left[\mathbb{I}(Y_i = C_i)\right] \\ s.t. \quad & \mathbb{E}\left[D(X_i, M(X_i), \tilde{M}(X_i))\right] \leq \beta \end{aligned} \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function for successful correction, and β is the upper bound of the allowed LLM invocation rate.

3.2 Overall Framework

Figure 1 illustrates the overall architecture of our CAR framework, which consists of three sequentially executed modules: a lightweight base corrector, a multi-dimensional risk-aware router, and an optional LLM corrector. The inference pipeline proceeds as follows:

(1) Initial Correction: The input sentences are first processed by the base corrector, which generates an initial correction $M(X)$ and all necessary intermediate evidence in a single forward pass.

(2) Risk Assessment and Adaptive Routing: The router extracts three categories of risk features from the base corrector's outputs: edit-based features, uncertainty-based features, and semantic consistency features. Then, a two-stage decision model processes the feature vector to produce a risk score, which determines whether $M(X)$ constitutes a high-risk sample.

(3) Final Output: Low-risk samples directly use the base corrector's output, while high-risk samples use the LLM's corrected output.

3.3 Base Corrector

The base corrector is a lightweight PLM-based CSC model that provides fast inference and reliable performance on common error patterns. Besides the initial corrected sentence, the base corrector also output: (1) top-k candidate characters and their softmax probabilities at each position, (2) the edit operations (insertion, deletion, substitution) applied to the input, (3) the token-level hidden states from the last layer.

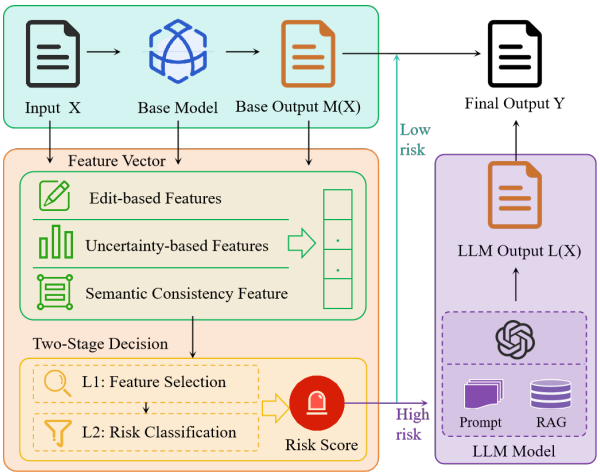

Figure 1: Overall architecture of CAR. The base corrector handles all inputs first and the routing module invokes the LLM only for high-risk samples.

3.4 Feature Construction

We design a comprehensive set of 9 CSC-specific risk features that capture different aspects of the base corrector's reliability. These features are grouped into three complementary categories, as summarized in Table 1.

Edit-based Features The features capture how much the base corrector changes the original sentence, which are computed by aligning X with $M(X)$. We use the number of modified characters, a binary flag indicating whether digits or symbols are modified, and the number of abnormal repeated characters in the corrected output. Intuitively, aggressive or unusual edits are more likely to be incorrect.

Uncertainty-based Features The features quantify the base model's internal confidence in its predictions. For each position i, we compute the Shannon entropy of the top-k candidate distribution $H_i = -\sum_{j=1}^{k} p_{ij} \log p_{ij}$ to measure the local prediction uncertainty. We then derive five uncertainty features: average top-1 probability, minimum top-1 probability, average entropy reduction, minimum entropy reduction, and the number of low-confidence positions.

Semantic Consistency Feature The feature measures whether the base correction preserves the meaning of the input. We compute cosine similarity between vector representations of X and $M(X)$, which is $\frac{v_X^\top v_{M(X)}}{\|v_X\| \|v_{M(X)}\|}$.

Table 1: Multi-dimensional risk feature set for CSC.

Category	Feature	Definition
Edit-based	Modified char count	Number of characters changed by the base corrector
Edit-based	Digit/symbol flag	Binary indicator if any digit or symbol was modified
Edit-based	Repetition count	Number of consecutive repeated characters in $M(X)$
Uncertainty-based	Avg. top-1 prob.	Mean of the maximum probabilities across all positions
Uncertainty-based	Min. top-1 prob.	Minimum of the maximum probabilities across all positions
Uncertainty-based	Avg. entropy drop	Average reduction in entropy from the input to the output
Uncertainty-based	Min. entropy drop	Minimum reduction in entropy across all positions
Uncertainty-based	Low-conf. count	Number of positions with top-1 probability
Semantic Consistency	Similarity	Cosine similarity between X and $M(X)$

3.5 Two-Stage Decision Model

The routing decision model is trained to predict whether the base corrector's output is correct. For each labeled CSC instance (X, C), we automatically generate the routing label as:

$$r_i = \mathbb{I}(M(X_i) \neq C_i), \tag{3}$$

where $r_i = 1$ indicates a high-risk sample where the base corrector produces an incorrect prediction, and $r_i = 0$ indicates a low-risk sample where the base corrector yields a correct prediction.

Then, we propose a two-stage regularized decision model to address two critical challenges (i.e. feature redundancy and severe class imbalance) in routing model training.

Stage 1: L1-Regularized Feature Selection We first use L1-regularized logistic regression to select informative features and reduce redundancy:

$$\min_{w,b} \quad -\frac{1}{N}\sum_{i=1}^{N}[r_i \log q_i + (1-r_i)\log(1-q_i)] + \gamma\|w\|_1, \tag{4}$$

where $q_i = \sigma(w^\top F_i + b)$ is the predicted probability of being high-risk, F_i is the 9-dimensional CSC-specific risk feature vector, and γ is the L1 regularization strength. Features with non-zero weights are retained for the second stage.

Stage 2: Weighted L2 Classification Because high-risk samples are relatively sparse, we train a second weighted logistic regression model with L2 regularization on the selected feature subset:

$$\min_{w_2,b_2} \quad -\frac{1}{N}\sum_{i=1}^{N}[\alpha r_i \log q_{2i} + (1-r_i)\log(1-q_{2i})] + \lambda\|w_2\|_2^2, \tag{5}$$

where $q_{2i} = \sigma(w_2^\top F_i' + b_2)$, F_i' is the selected feature subset, $\alpha > 1$ is the class weight, and λ is the L2 regularization strength. α ensures that the model prioritizes correctly identifying high-risk samples, as missing these samples directly degrades correction performance.

The final routing decision is made by thresholding the predicted probability:

$$\mathcal{D}(X_i, M(X_i), \tilde{M}(X_i)) = \mathbb{I}(q_{2i} \geq \tau), \tag{6}$$

where τ is tuned on the development set to satisfy different cost budgets (i.e. the desired LLM invocation rate β).

3.6 LLM Corrector

For samples classified as high-risk, we invoke an LLM to perform secondary correction and verification. The LLM leverages its extensive world knowledge and strong contextual reasoning ability to correct complex errors that the base model fails to handle. To ensure the LLM produces high-quality and consistent correction results, we design a task-specific prompt and integrate retrieval-augmented generation (RAG).

Task-Specific Prompt Engineering We construct a detailed prompt that defines the LLM as a professional Chinese spelling correction expert, with three explicit requirements. (1) Error Scope: Clearly define the types of errors to be corrected, including phonetic errors, glyph errors, symbol interference errors, and semantic drift

Dataset	Samples	Error Ratio
SIGHAN13-Train	700	48.57%
SIGHAN13-Test	1,000	97.10%
CSCDNS-Train subset	5,000	63.92%
CSCDNS-Test	5,000	46.06%
Wang271k subset	5,000	98.86%

Table 2: Dataset statistics used in our experiments.

caused by over-correction. (2) Semantic Preservation: Strictly preserve the original semantic meaning and text length, and avoid unnecessary modifications. (3) Output Format: Only output the corrected text without any additional explanations or reasoning processes, to ensure compatibility with the framework.

Retrieval-Augmented Generation To improve the LLM's performance on domain-specific and rare error patterns, we build a knowledge base consisting of high-quality CSC sample pairs. For each input sample, we first retrieve the top-j most similar error-correction pairs from the knowledge base. These examples are then inserted into the prompt as in-context learning demonstrations to guide the LLM's correction behavior.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate on two widely used CSC benchmarks. SIGHAN13 is standard in-domain benchmark and we train the base corrector with SIGHAN-style data. CSCDNS (Hu et al., 2024) is a recent large-scale dataset, which can be treated as an out-of-domain benchmark for testing generalization. For routing model training, we use SIGHAN13-Train, 5,000 random samples from CSCDNS-Train, and 5,000 random samples from Wang271k (Wang et al., 2018). Dataset statistics are shown in Table 2.

Baselines We compare CAR with the following baselines. (1) **Base Only**: MacBERT4CSC is adopted as the base model, which is fine-tuned on SIGHAN and Wang271k datasets and achieves perfect performance on phonetic and glyph error correction. (2) **LLM Only**: DeepSeek-v3-671B is used with a generation temperature of 0.3 and a maximum generation token count of 2048. The knowledge base is built from the SIGHAN13, Wang271k, and CSCDNS datasets and the top-j candidate size is set to 5. (3) **Fixed Threshold Routing**: The routing method invokes the LLM based on a fixed threshold (0.95) of the base model's minimum top-1 probability. (4) **Random Routing**: The routing method randomly invokes the LLM with a fixed ratio.

For the decision model, the positive-to-negative class weight ratio is $\alpha = 10$.

4.2 Evaluation Metrics

We report three categories of metrics to comprehensively evaluate our method.

Correction Performance Metrics We use the standard sentence-level detection F1 and correction F1. Det. F1 measures the ability to identify sentences containing errors, while Corr. F1 measures the ability to produce the exact correct sentence.

Cost-Effectiveness Metrics LLM Invocation Rate (IR), as the main cost metric, calculates the percentage of samples routed to the LLM:

$$\mathrm{IR} = \frac{\sum_{i=1}^{K} \mathcal{D}(X_i, M(X_i), \tilde{M}(X_i))}{K}, \quad (7)$$

where K is the number of test samples.

F1 Gain Ratio (FGR), as the main effectiveness metric, calculates the percentage of the performance gain achieved by the routed system.

$$\mathrm{FGR} = \frac{\mathrm{F1}_{CAR} - \mathrm{F1}_{\mathrm{Base}}}{\mathrm{F1}_{\mathrm{LLM}} - \mathrm{F1}_{\mathrm{Base}}}, \quad (8)$$

where $\mathrm{F1}_{\mathrm{Base}}$ and $\mathrm{F1}_{CAR}$ represent the performance of each standalone corrector.

Routing Quality Metrics Decision Effectiveness (DE) calculates the percentage of truly high-risk samples that are correctly identified by the router.

$$\mathrm{DE} = \frac{K_{\mathrm{hit}}}{K_{\mathrm{pos}}}, \quad (9)$$

where K_{pos} is the number of truly high-risk samples that the base model fails to correct, and K_{hit} is the number of such samples selected by the router.

5 Results and Analysis

5.1 Correction Effect

Table 3 shows the main experimental results. On both datasets, CAR achieves an excellent balance between correction performance and LLM invocation cost.

(1) On the in-domain SIGHAN13 dataset, CAR improves the base corrector's correction F1 from

Dataset	Method	Det. F1	Corr. F1	FGR	IR
SIGHAN13	Base Only	0.8541	0.7022	0.0000	0.0000
	LLM Only	0.9809	0.8379	1.0000	1.0000
	CAR	0.9851	0.8815	1.3213	0.5360
CSCDNS	Base Only	0.5820	0.2570	0.0000	0.0000
	LLM Only	0.7669	0.6899	1.0000	1.0000
	CAR	0.7645	0.6758	0.9674	0.8792

Table 3: Main experimental results, including correction effect and cost-effectiveness.

0.7022 to 0.8815, uses 53.60% LLM calls and even outperforms LLM Only invocation. The remarkable result demonstrates that adaptive routing not only reduces cost but also mitigates the over-correction problem of LLMs.

(2) On the out-of-domain CSCDNS dataset, the base corrector exhibits limited generalization, achieving a correction F1 of only 0.2570. By contrast, CAR attains 0.6758, close to LLM Only performance of 0.6899. CAR uses 87.92% LLM calls and achieves an FGR of 0.9674. This higher invocation rate is expected, as the base model is less reliable under distribution shift.

5.2 Routing Quality Effectiveness

We conduct routing effectiveness evaluations on the CSCDNS dataset, because its error ratio (46.06%) provides a sufficient number of high-risk samples and represents a challenging out-of-domain scenario where the base corrector performs poorly.

Decision Effectiveness Across Invocation Budgets We vary the LLM invocation budgets to generate a series of operating points with different routing methods. A higher DE indicates that the router is better at prioritizing samples that actually need LLM intervention. As shown in Figure 2, the CAR consistently selects more truly high-risk samples than random routing and fixed-threshold routing.

Performance Comparison at Fixed Invocation Rate To ensure a fair comparison under identical cost constraints, we adjust the threshold of random routing method to match CAR's LLM invocation rate of 87.92% on CSCDNS. As shown in Table 4, CAR achieves a correction F1 of 0.6758, which is 94.3% higher than fixed threshold (0.3479) and 6.7% higher than random routing (0.6332).

The results confirm that our CAR strategy makes more intelligent use of LLM resources by allocating them to the samples where they will have the greatest impact.

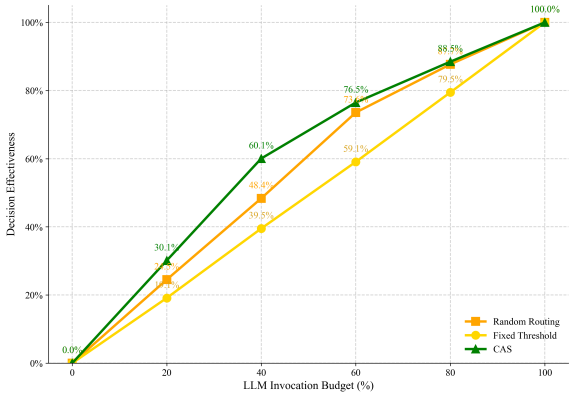

Figure 2: Routing decision effectiveness under different LLM invocation budgets.

Dataset	Method	Det. F1	Corr. F1
CSCDNS	Fixed Threshold	0.5866	0.3479
	Random Routing	0.7474	0.6332
	CAR	0.7645	0.6758

Table 4: Routing Decision methods comparison on CSCDNS-Test.

5.3 Ablation Study

Feature Selection Table 5 shows feature selection and weights after the two-stage decision process. The most important feature by far is the modified character count, with a weight of +7.7366. This confirms our intuition that samples with more aggressive edits are much more likely to be incorrect. The positive weight indicates that as the number of modified characters increases, the risk of base model failure increases. The sentence similarity feature has a negative weight, indicating that lower similarity (greater semantic change) is associated with higher risk.

Feature Category analysis Table 6 shows the effect of removing feature categories. Using all three feature categories achieves the highest correction F1. Removing edit-based features causes the largest drop, but removing uncertainty-based or semantic consistency also hurts performance. This confirms that a comprehensive multi-dimensional risk assessment is essential for effective routing in CSC.

5.4 Case Analysis

As shown in Figure 3, two representative cases that illustrate the strengths of our CAR framework.

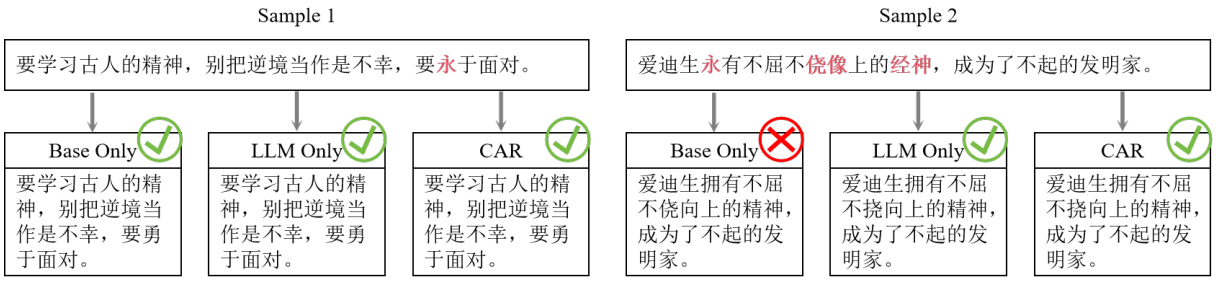

Figure 3: Qualitative case analysis.

Category	Feature	Selected	Weight
Edit	Modified char count	✓	+7.7366
Edit	Digit/symbol flag	✓	+0.8435
Edit	Repetition count	×	–
Uncertainty	Avg. entropy drop	✓	+0.0917
Uncertainty	Min. entropy drop	✓	-1.0826
Uncertainty	Avg. top-1 prob.	×	–
Uncertainty	Min. top-1 prob.	✓	-0.6062
Uncertainty	Low-conf. count	✓	+0.0603
Semantic	Similarity	✓	-0.4200

Table 5: Feature selection and weight distribution.

Edit	Uncertainty	Semantic	Det. F1	Corr. F1
✓	✓	✓	0.7645	0.6758
×	✓	✓	0.7486	0.6321
✓	×	✓	0.7497	0.6333
✓	✓	×	0.7502	0.6413

Table 6: Feature category ablation on CSCDNS-Test.

(1) For simple phonetic errors, the base corrector can often produce the correct output, and invoking the LLM is redundant. CAR identifies such low-risk cases and directly accepts the base output.

(2) For sentences with multiple interacting errors, the base corrector may only fix part of the sentence. In these cases, risk features such as high edit count, low confidence, and semantic inconsistency cause the router to invoke the LLM, leading to a more complete correction.

6 Conclusion

We propose CAR, a cost-aware adaptive routing framework for Chinese Spelling Correction. Instead of invoking an LLM for every input, CAR estimates the risk of the base corrector output using edit, uncertainty, and semantic consistency features. A two-stage regularized decision model then routes only high-risk samples to the LLM. Experiments on SIGHAN13 and CSCDNS show that CAR achieves strong correction performance while reducing redundant LLM calls, and decision analyses confirm that the proposed router identifies high-risk samples more effectively than fixed-threshold and random routing baselines.

7 Limitations

This work focuses on inference-time routing rather than improving the internal correction ability of the LLM itself. More advanced LLM correction modules are orthogonal to our router and can be explored in future work. In addition, the current router is trained offline on static datasets; online adaptation to continuously emerging error patterns remains an important direction. Finally, while we report invocation rate as a cost metric, future work should also report token cost and latency under real deployment constraints.

References

Hsun-wen Chiu, Jian-cheng Wu, and Jason S. Chang. 2013. Chinese spelling checker based on statistical machine translation. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 49–53.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks Lakshmanan, and Ahmed H. Awadallah. 2024. Hybrid llm : Cost-efficient and quality-aware query routing. In *The Twelfth International Conference on Learning Representations*.

Ming Dong, Zhiwei Cheng, Changyin Luo, and Tingting He. 2025. Retrieval-augmented generation for large language model based few-shot chinese spell checking. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10767–10780.

Yong Hu, Fandong Meng, and Jie Zhou. 2024. Cscdns: A chinese spelling check dataset for native

speakers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 146–159.

Kunting Li, Yong Hu, Liang He, Fandong Meng, and Jie Zhou. 2024. C-llm: Learn to check chinese spelling errors character by character. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5944–5957.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, Yangning Li, Feng Zhou, Hai-Tao Zheng, and Qingyu Zhou. 2023. On the (in)effectiveness of large language models for chinese text correction. *arXiv preprint arXiv:2307.09007*.

Changchun Liu, Kai Zhang, Junzhe Jiang, Zirui Liu, Hanqing Tao, Min Gao, and Enhong Chen. 2024. Arm: An alignment-and-replacement module for chinese spelling check based on llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10156–10168.

Linfeng Liu, Hongqiu Wu, and Hai Zhao. 2025. Evolving chinese spelling correction with corrector-verifier collaboration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29022–29028.

Ziheng Qiao, Houquan Zhou, and Zhenghua Li. 2025. Mixture of small and large models for chinese spelling check. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28298–28311.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2517–2527.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023. Rethinking masked language modeling for chinese spelling correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10743–10756.

Jinhua Xiong, Qiao Zhang, Shuiyuan Zhang, Jianpeng Hou, and Xueqi Cheng. 2015. Hanspeller: A unified framework for chinese spelling correction. *International Journal of Computational Linguistics and Chinese Language Processing*, 20(1).

Jui-Feng Yeh, Yun-Yun Lu, Chen-Hsien Lee, Yu-Hsiang Yu, and Yong-Ting Chen. 2014. Chinese word spelling correction based on rule induction. In *Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 139–145.

Junjie Yu and Zhenghua Li. 2014. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In *Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 220–223.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 882–890.

Houquan Zhou, Zhenghua Li, Bo Zhang, Chen Li, Shaopeng Lai, Ji Zhang, Fei Huang, and Min Zhang. 2024. A simple yet effective training-free prompt-free approach to chinese spelling correction based on large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17446–17467.