CSCFRL: Chinese Spelling Correction in Dynamic Environments with Feedback-Driven Retrieval-Augmented LLMs

Anonymous ACL submission

Abstract

Chinese Spelling Correction (CSC) becomes challenging in dynamic environments where new error types continuously emerge. Although fine-tuned BERT-based models perform well on closed benchmarks for routine corrections, they struggle to generalize to evolving errors without retraining, requiring costly and inconsistent expert manual review as a post-processing step. Large language models (LLMs) show generalization potential but suffer from instability and over-correction. In this paper, we propose CSCFRL, a feedback-driven, retrieval-augmented LLM-based CSC framework, designed to replace expert manual review. A LLM-based Corrector generates candidate corrections using internal knowledge and dynamically retrieved samples, and a LLM-based Evaluator enforces semantic and length constraints to evaluate quality, forming an iterative correction mechanism. Experiments on closed (SIGHAN2015) and open (CSCD-NS-T) benchmarks show that CSCFRL significantly improves recall and F1-score, outperforming the BERT-based and vanilla LLM baselines. These results demonstrate its superior reliability and adaptability for real-world high-accuracy applications.

1 Introduction

Chinese spelling correction (CSC) plays a fundamental role in multimedia text understanding, with broad applications in content moderation, intelligent editing, OCR and ASR post-processing. However, real-world text inputs are increasingly dynamic rather than static. Evolving informal linguistic norms, intentional manipulations to resist censorship, and AIGC-generated content (Yang et al., 2025b; Ji et al., 2025) have created constantly shifting text error distributions. Therefore, they pose a severe challenge to CSC tasks: how to rapidly handle newly emerging spelling errors without expensive model retraining in dynamic environments.

Traditional CSC pipelines rely on two-stage architectures: lightweight BERT-based models for fast initial correction, followed by expert manual review to handle unseen or ambiguous errors. While BERT-based models such as MacBert4csc (Ming, 2021) achieve excellent performance on closed benchmarks with fixed error distributions, their performance drops catastrophically when encountering novel error types not seen during training. This necessitates continuous human intervention, which is slow, expensive, and inconsistent across different reviewers.

The rapid progress of large language models (LLMs) (Xiong et al., 2025) has offered a promising alternative, as their vast internal knowledge and contextual reasoning abilities suggest potential for better generalization to unseen errors. They can effectively replace the expert manual review. However, vanilla LLMs suffer from two critical limitations that prevent their direct deployment in high-stakes CSC scenarios:

Outdated knowledge: The internal knowledge of both LLMs and BERT-based models is inherently bounded by their training data, which restricts adaptability to novel error types.

Instability in outputs: Inherent randomness prevents a single LLM to precisely control its outputs following prompt. It will result in over-correction, hallucinations, superfluous explanations or semantic drift.

To address these challenges, we redefine the CSC task as a reliable and adaptable generation problem, where the goal is to automate and replace expert manual review for dynamic error cases. We propose CSCFRL, a feedback-driven, retrieval-augmented LLM-based CSC framework. Our main contributions are:

- We reformulate CSC as a reliable and adaptable generation problem, shifting the focus from static input-output mapping to continual

adaptation under evolving error distributions.

- We propose CSCFRL, a unified multi-agent framework that tightly integrates retrieval augmentation for adaptability and feedback-driven evaluation for reliability.

- We conduct extensive experiments on both closed (SIGHAN2015) and open dynamic (CSCD-NS-T) benchmarks, demonstrating that CSCFRL outperforms BERT-based and vanilla LLM-based baselines, providing an effective automated alternative to expert manual review.

2 Related Work

Our work builds on three main lines of research: traditional BERT-based CSC methods, LLM-based CSC methods, and retrieval-augmented and feedback-driven NLP systems.

2.1 BERT-based CSC Method

Early CSC systems relied on rule-based and statistical methods, which were limited by their ability to capture contextual information. The advent of BERT (Devlin et al., 2019) revolutionized the field by enabling powerful contextual representation learning. Soft-Masked BERT (Zhang et al., 2020) introduced a soft-masking mechanism to jointly learn error detection and correction, achieving state-of-the-art performance on standard benchmarks. MacBERT (Cui et al., 2020) further improved performance by using masked language modeling with whole-word masking, leading to the widely used MacBert4csc model (Ming, 2021).

While these methods excel at handling common error patterns seen during training, they suffer from poor generalization to unseen errors. As noted in previous work (Li et al., 2023), their performance drops significantly when evaluated on out-of-distribution datasets, requiring frequent retraining to maintain effectiveness. This limitation makes them unsuitable for dynamic environments where new error types emerge continuously.

2.2 LLM-based CSC Method

The emergence of large language models has opened new possibilities for CSC. Early studies (Li et al., 2023) showed that vanilla LLMs underperform fine-tuned BERT models on closed benchmarks due to issues with mixed character-word tokenization and lack of task-specific training. To address this, some researchers (Li et al., 2024; Dong et al., 2024; Liu et al., 2024) proposed character-level mappings, semantic knowledge, alignment-and-replacement module to enhance correction performance. In addition, 11,021 variant and homophonic test cases from ChineseSafe (Zhang et al., 2024) were compiled for large-scale evaluation of 22 LLMs.

However, most existing LLM-based methods focus on improving performance on static benchmarks rather than addressing the dynamic adaptation problem. They also suffer from the same instability issues as vanilla LLMs, requiring manual review to ensure reliability.

2.3 RAG and Feedback

Retrieval-augmented generation (RAG) (Zhao et al., 2024) has emerged as a powerful technique for enhancing LLMs with external knowledge, enabling them to access up-to-date information without retraining. Retrieval-based methods have been used to provide reference examples for correcting rare or unseen errors (Dong et al., 2025). Feedback-driven mechanisms, inspired by self-refinement and multi-agent systems (Song et al., 2025), have also been applied to improve LLM output quality by iteratively refining generations based on evaluation feedback.

Our work integrates RAG and feedback mechanisms in a unified framework tailored for the dynamic CSC scenario, enabling continual adaptation without retraining.

3 Method

To address the dual challenges of adaptability and reliability in the CSC task, our framework replaces the expert review through Corrector, Evaluator and dynamic knowledge base.

3.1 Problem setting

The traditional CSC task can be formally defined as the task of mapping an input sequence x containing potential spelling errors to a predicted corrected output sequence y that preserves the intended semantics. That is,

$$y = \zeta(x), \tag{1}$$

where $\zeta(x)$ represents the judgment and correction process. Formally, we attempt to find an optimal

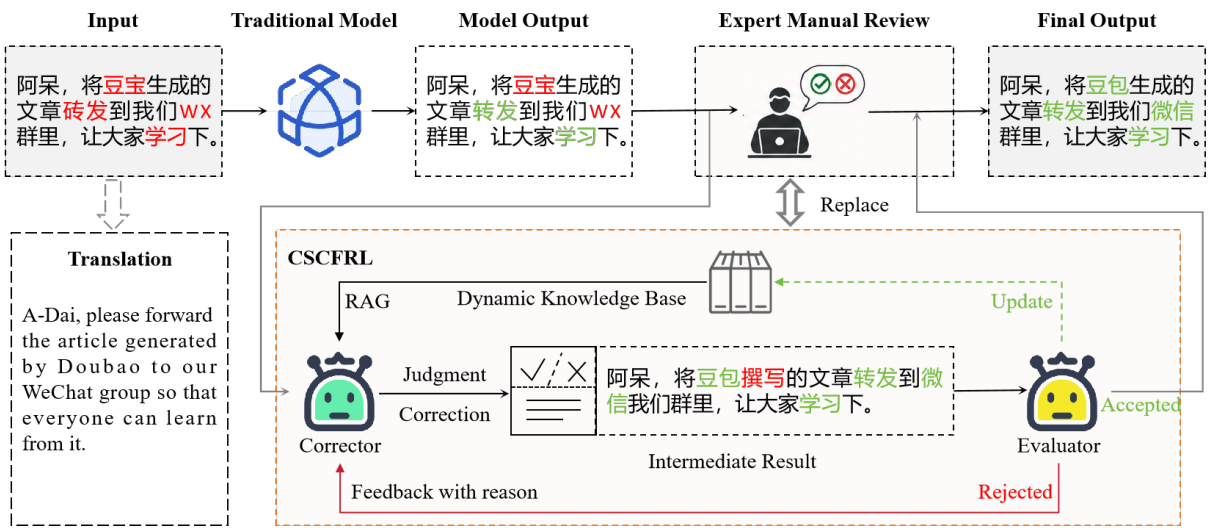


Figure 1: CSCFRL Framework. A user intends to express the sentence "A-Dai, please forward the article generated by Doubao to our WeChat group, so that everyone can learn from it", while the Chinese input may contain spelling errors. Traditional correction models can only detect partial errors, leaving the rest for manual double-checking. The proposed framework replaces manual work for this part. It integrates a Corrector, an Evaluator, and a dynamic knowledge base into a feedback-driven multi-agent optimization pipeline, improving correction quality via a constrained iterative mechanism.

correction function,

$$\zeta^* = \arg\max_{\zeta} \mathrm{S}\big(\zeta(x), y^*\big), \tag{2}$$

where $\mathrm{S}(y, y^*)$ represents the cosine similarity of Sentence-BERT embeddings between the predicted output y and the true target y^*. ζ^* represents the most suitable correction function.

However, as spelling errors continually evolve in real world environments, the domain of x keeps expanding, making it impossible for a fixed correction function $\zeta()$ to cover all possible inputs. Therefore, we define this issue as a reliable and adaptive generation problem and attempt to find appropriate solutions to enhance the following abilities.

Reliability: The ability to produce stable, faithful corrections that avoid over-correction or semantic drift, addressing the inherent instability in LLM outputs.

Adaptability: The ability to comprehend newly emerging error-correction patterns without retraining, thereby overcoming the constraint of bounded knowledge.

3.2 Framework

The CSCFRL integrates three core components, as illustrated in Figure 1 : (1) a retrieval-augmented Corrector that leverages a dynamic knowledge base (DKB) to adapt to emerging errors without retraining; (2) a dedicated Evaluator that enforces semantic and length constraints to ensure output reliability; and (3) an iterative feedback mechanism that refines corrections until they meet predefined quality standards.

3.3 Corrector

The LLM-based Corrector agent judges and corrects errors in the input text under constrained prompting. We improve the correction function by incorporating reference samples as conditional inputs into the correction process. It ensures that the Corrector's output is not solely dependent on the LLM's static internal knowledge. The Corrector produces a correction intermediate result for each input sequence. That is,

$$c = \zeta(x, \epsilon(x, m)), \tag{3}$$

where $\epsilon(x)$ denotes m reference samples retrieved from the DKB through RAG method.

For clearer understanding, we present the initial correction prompt of corrector in Figure 2, which specifies the scope of error types, provides illustrative examples, enforces formatting constraints, and defines the JSON format of the output. The evaluation and feedback-driven prompts are similar, so we will not elaborate on them further.

3.4 Evaluator

The LLM-based Evaluator agent evaluates whether the intermediate results satisfy predefined constraints. We compute correction reliability via quantifying the degree of two strict constraints.

Semantic Consistency $f_{\mathbf{sc}}(x, c)$ The corrected text preserves the original meaning, which is approximated using embedding-based semantic similarity between the input x and the corrected c. That is,

$$f_{\text{sc}}(x, c) = \frac{\boldsymbol{E}(x) \cdot \boldsymbol{E}(c)}{\|\boldsymbol{E}(x)\|_2 \, \|\boldsymbol{E}(c)\|_2}, \tag{4}$$

where $\boldsymbol{E}(\cdot)$ denotes the sentence embedding vector generated by the Chinese pre-trained SimCSE model, and $f_{\text{sc}}(x, c) \in [0, 1]$.

Length Stability $f_{\mathbf{ls}}(x, c)$ The corrected text introduces only necessary edits, operationalized via length consistency as a proxy for avoiding unnecessary insertions or deletions. That is,

$$f_{\text{ls}}(x, c) = \frac{\min(\boldsymbol{L}(x), \boldsymbol{L}(c))}{\max(\boldsymbol{L}(x), \boldsymbol{L}(c))}, \tag{5}$$

where $\boldsymbol{L}(\cdot)$ represents the length of text, and $f_{\text{ls}}(x, c) \in [0, 1]$.

Both $f_{\text{sc}}(x, c)$ and $f_{\text{ls}}(x, c)$ serve as lightweight heuristic indicators rather than theoretically optimal metrics. Their primary role is to provide a computationally efficient signal for the LLM-based Evaluator to reason over, rather than to serve as standalone decision criteria. Formally, the reliability score $e(x, c)$ and analysis result $a(x, c)$ are achieved through the comprehensive analysis of constraint indicators by the Evaluator. That is,

$$\{e(x, c), a(x, c)\} = \text{Eva}\big(f_{\text{sc}}(x, c), f_{\text{ls}}(x, c), x, c\big), \tag{6}$$

where Eva() represents the LLM's analysis process and the detailed scoring criteria is set through the evaluation prompt, and $e(x, c) \in [0, 10]$.

To ensure the quality of outputs, we define E as a reliability threshold. If $e(x, c) > E$, the intermediate results will be accepted as final results. Otherwise, the Evaluator will prompt recorrecting with corresponding explanations to the Corrector. Notably, $e(x, c)$ is also related to the performance of the model baselines and the design of the prompt.

You are a professional Chinese spelling correction system. Your task is:
1. Determine whether the original text contains character errors such as visually similar characters or phonologically similar characters.
2. If there are no errors, output "Correct" along with the original text.
3. If errors are detected, correct each erroneous character one-to-one and output "Incorrect" along with the corrected text.

Instructions:
1. Do not make modifications to sentence case or modal particles.
2. Phonetically similar character errors refer to substitutions involving homophones that are semantically incorrect (e.g., "洗碗" wrongly written as "希碗").
3. Visually similar character errors refer to substitutions involving characters with similar visual forms but different meanings (e.g., "普遍" wrongly written as "普偏").
4. Output the final result with JSON format, without additional explanations.
5. If the original text is correct, the corrected text should be identical to it.

Input Format:
{ "Original_text": "..." }

Output Format:
{ "Original_text": "...",
"Judgment": "Correct/Incorrect",
"Corrected_text": "..." }

Reference Example::
These are reference examples indicating whether errors are present and the corrected results. That is: { RAG_context }

Figure 2: Initial Correction Prompt

3.5 Dynamic knowledge base

A fundamental limitation of traditional CSC methods lies in their inability to adapt to emerging error types without retraining. To address this issue, We transform CSC from a static supervised task into a continual adaptation process through RAG (Zhao et al., 2024), which incorporates external knowledge bases to provide LLMs with additional reference samples. When the knowledge base is dynamically updated with new error-correction pairs, LLMs can continuously access useful information, enhancing the adaptability to emerging error types.

The DKB maintains a repository $(x_j, y_j)_{j=1}^{M}$ with M data, where each pair represents an observed error instance and its accepted correction. Accepted pairs are managed through the batch update policy described in Section 4.3. Given a new input x, the retriever queries the knowledge base to obtain the top-m most relevant samples. That is,

$$\epsilon(x, m) = \{(x'_1, y'_1), ..., (x'_m, y'_m)\}. \tag{7}$$

Then the samples will be integrated into the Corrector's generation prompt.

3.6 Iterative mechanism

The core strategy of constraint iterative correction mechanism is feedback-driven multi-agent evaluation optimization (Song et al., 2025). It facilitates information exchange between the Corrector and the Evaluator. During the next correction cycle, the Corrector is provided with richer reference information, denoted as

$$c_t = \zeta(x, \epsilon(x, m), e(x, c_{t-1}), a(x, c_{t-1})), \tag{8}$$

Table 1: Dataset

Dataset	Training		Testing		Verification
	True	False	True	False	
SIGHAN 2015	66	2272	548	552	Reliability
CSCD-NS-T	2436	2064	261	239	Adaptability

where t represents the number of times that the Corrector has been executed. c_t is the t-th intermediate result and $c_0 = null$. The loop terminates when a candidate satisfies the reliability constraint or when a maximum iteration bound T is reached. We obtain the final converging result as,

$$\begin{aligned} y &= \{c_{t^*}, a(x, c_{t^*})\}, \\ t^* &= \min(\{t \mid e(x, c_t) > E\} \cup \{T\}). \end{aligned} \tag{9}$$

Regardless of whether the final iteration corrects the errors, the output includes the latest intermediate result c_{t^*} and supplementary analysis $a(x, c_{t^*})$.

Finally, the mechanism solves the reliable and adaptive generation problem. The correction function that maximizes the similarity between predictions and correct texts can be defined as,

$$\begin{aligned} \zeta^* = \arg\max_{\zeta} \mathbb{E}_{(x,y^*)\sim\mathcal{D}} \\ \mathrm{S}\big(\zeta(x, \epsilon(x, m), e(x, c_{t^*-1}), a(x, c_{t^*-1})), y^*\big), \end{aligned} \tag{10}$$

where $\mathcal{D}$ represents the entire testing dataset.

4 Experiments

4.1 Dataset

We evaluate CSCFRL on both closed-domain and dynamic-domain CSC benchmarks, which are presented in Table 1.

SIGHAN2015 dataset (Tseng et al., 2015) It is adopted as the closed-domain benchmark to evaluate correction reliability under relatively stable error distributions. It comprises 3,438 short texts, including 1,100 for testing, and serves as a standard closed dataset to analysis the reliability.

CSCD-NS-T dataset (Hu et al., 2024) It originates from the CSCD-NS dataset, which contains 40,000 diverse social-media texts with emerging spelling variants. We incrementally incorporate 4,500 samples into the dynamic knowledge base and reserve 500 samples for testing, simulating continual adaptation without retraining.

4.2 Baselines

We design three groups of baselines addressing different aspects to compare with the CSCFRL framework, i.e. *DeepSeek-R1-671B-RAG-Feedback*, where the Corrector and the Evaluator all take *DeepSeek-R1-671B* as the base model.

Fine-tuned BERT-based models *BERT-Base-Chinese*, *SoftMask-BERT* and *MacBert4csc-Base-Chinese* are selected as representative SOTA BERT-Based approaches. They achieve strong performance on conventional static benchmarks and serve as the primary comparison targets for reliability evaluation.

Vanilla LLM-based methods *DeepSeek-R1* with different parameter scales (1.5B, 70B, and 671B) (Guo et al., 2025) are examined to analyze the influence of model size on correction reliability.

Modular variants of CSCFRL To investigate the contribution of different components, we further evaluate *DeepSeek-R1-671B* equipped with retrieval augmentation only (RAG), evaluator feedback only (Feedback), and the complete CSCFRL framework (RAG-Feedback). In addition, the Evaluator is replaced with Qwen-Plus-20250728 (Yang et al., 2025a) to analyze the function of heterogeneous evaluator architectures.

4.3 Implementation details

To validate the reliability and consistency of different methods, we make initial settings from multiple aspects.

Parameters The *DeepSeek-R1* and *Qwen* are all configured with a temperature of 0.3, a maximum token length of 2,048. The RAG sample size is set to $m = 5$. The reliability threshold is $E = 8$, which is discussed below.

DKB format Each short text in the training set is reformatted into a unified sample. That is *"The original text is .... It contains or doesn't contain errors. The correct text is ..."*. The samples are transformed into dense vectors using the *Bge-Small-Zh-v1.5* Embedding Model and indexed in the *Chroma* Vector Database for efficient similarity search. We simulate continual adaptation by incrementally adding the CSCD-NS-T training samples into the DKB in batch, rather than performing online per-sample updates.

Evaluation metrics The approaches are evaluated from the sentence-level perspectives, focusing on two core tasks: error judgment and correc-

tion. For each task, four standard evaluation metrics are computed: accuracy, precision, recall, and F1-score. In addition, three independent runs are averaged for reproducibility, with standard deviations less than 0.5% for all metrics.

5 Results and Analysis

5.1 Reliability performance

We conduct experiments on SIGHAN2015 to evaluate the reliability of different methods. The results in Table 2 compare both traditional BERT-based and LLM-based methods under closed-domain settings.

Comparison with fine-tuned BERT-based models For *DeepSeek-R1* without additional enhancement modules, even with large parameter sizes, all evaluation metrics are consistently lower than *MacBert4csc-Base-Chinese*. When *DeepSeek-R1* is equipped with the CSCFRL framework, *DeepSeek-R1-671B-RAG-Feedback* outperforms *MacBert4csc-Base-Chinese* in terms of accuracy, recall, and F1-score for both judgment and correction. The results indicate that CSCFRL can effectively enhance the correction stability of LLM-based models, which can achieve comparable or even superior performance to fine-tuned BERT-based models.

Comparison of LLMs under different parameters Regardless of judgment or correction, results across three different *DeepSeek-R1* models (1.5B, 70B, and 671B) demonstrate a clear upward trend in all performance metrics as the parameter size increases. Specifically, the F1-Score improves from only 1.80 with *DeepSeek-R1-1.5B* to 64.21 with *DeepSeek-R1-671B*. The finding suggests that larger scale LLMs possess stronger capabilities, benefiting from the enhanced representation and reasoning capacities brought by increased parameter size.

Comparison of Evaluators constructed by different LLMs To mitigate single model cognitive biases, we introduce *DeepSeek-R1-671B* as the Corrector and *Qwen-Plus-20250728* as the Evaluator (*DeepSeek-RAG_Qwen-Feedback*). The result shows that *Qwen* counterproductively degrades the whole correction performance compared with *Deepseek*. It suggests that, model capability may be a more critical factor than model type when selecting the Evaluator, though further investigation is needed to generalize this finding.

Comparison with RAG-based methods The RAG-enhanced LLM baseline (Dong et al., 2025) achieves judgment F1-scores of 51.07 under the few-shot setting, which is significantly lower than zero-shot DeepSeek-R1-671B and even lower than DeepSeek-R1-70B. It demonstrates that larger LLMs with stronger general capabilities have inherent advantages in CSC tasks even without task-specific samples.

5.2 Adaptability performance

To evaluate the adaptability of CSCFRL to emerging errors, We conduct experiments on CSCD-NS-T testing set in Table 3. The DKB is constructed from the SIGHAN2015 training data and dynamically augmented with the CSCD-NS-T training data. The results can identify three key characteristics.

(1) Without incorporating the CSCD-NS-T as either part of the training data, *MacBert4csc-Base-Chinese* exhibits extremely low performance on the CSCD-NS-T testing set, with the judgment F1-Score of only 32.73, indicating almost no generalization capability.

(2) When *DeepSeek-R1-671B-RAG* employs SIGHAN2015 training dataset as the basic knowledge database, it maintains relatively strong performance, achieving an judgment F1-Score of 79.75. When the CSCD-NS-T training data is integrated into the DKB, the judgment F1-Score of *DeepSeek-R1-671B-RAG* further increases by 3.27 points. The results demonstrate that the DKB enables LLMs to rapidly adapt to evolving error patterns without additional retraining.

(3) With the addition of the feedback module, the performance of *DeepSeek-R1-671B-RAG-Feedback* shows a slight but insignificant improvement. This is expected, since the Evaluator mainly improves correction reliability by reducing over-correction and semantic drift rather than directly enhancing correction capability.

5.3 Ablation Study

The four configurations also serve as an ablation study of CSCFRL's components.

(1) The Corrector without other modules (*DeepSeek-R1-671B*) is capable of spelling error correction, but its accuracy and precision remain lower than fine-tuned BERT-based models.

(2) The Corrector with RAG (*DeepSeek-R1-671B-RAG*) yields notable gains in accuracy and precision over the vanilla baseline.

Table 2: Sentence-level correction performance on SIGHAN2015 testing set.

Model	Judgment				Correction			
	Acc.	Pre.	Rec.	F1	Acc.	Pre.	Rec.	F1
Bert-Base-Chinese (Devlin et al., 2019)	74.82	83.85	60.44	70.25	57.45	68.34	25.14	36.76
SoftMask-BERT (Zhang et al., 2020)	76.64	75.47	79.17	77.27	75.18	74.09	**77.72**	75.86
MacBert4csc-Base-Chinese (Ming, 2021)	**88.27**	**83.99**	**94.09**	**88.75**	**77.45**	**80.08**	72.09	**75.88**
DeepSeek-R1-1.5B	53.91	52.59	68.93	59.67	20.55	2.31	1.47	1.80
DeepSeek-R1-70B	71.73	68.61	78.37	73.17	45.64	41.39	25.32	31.42
DeepSeek-R1-671B	**79.91**	**73.67**	**92.05**	**81.84**	**65.55**	**65.64**	**62.85**	**64.21**
GPT3.5-RAG (Dong et al., 2025)	51.27	51.50	50.65	51.07	44.27	44.36	43.62	43.99
DeepSeek-R1-671B-RAG	85.09	82.44	88.54	85.38	72.64	77.03	63.22	69.44
DeepSeek-R1-671B-Feedback	81.55	75.85	92.75	83.46	67.55	68.71	64.86	66.73
DeepSeek-RAG_Qwen-Feedback	80.00	**88.12**	68.58	77.13	70.18	**84.03**	48.61	61.59
DeepSeek-R1-671B-RAG-Feedback	**88.51**	81.43	**99.32**	**89.49**	**78.46**	77.69	**78.87**	**78.27**

Table 3: Sentence-level correction performance on CSCD-NS-T testing set.

Model	Training Data/Knowledge Base	Judgment		Correction	
		Acc.	F1	Acc.	F1
MacBert4csc-Base-Chinese	SIGHAN + Wang271k	55.60	32.73	50.40	18.42
DeepSeek-R1-671B-RAG	SIGHAN2015	80.60	79.75	75.20	72.57
DeepSeek-R1-671B-RAG	SIGHAN2015 + CSCD-NS-T	**83.80**	**83.02**	**77.20**	**74.32**
DeepSeek-R1-671B-RAG-Feedback	SIGHAN2015	82.00	81.48	76.60	74.51
DeepSeek-R1-671B-RAG-Feedback	SIGHAN2015 + CSCD-NS-T	**83.80**	**83.23**	**77.40**	**74.94**

(3) The Evaluator feedback (*DeepSeek-R1-671B-Feedback*) leads to moderate improvements across all four metrics, reflecting its role in enhancing reliability.

(4) The full framework (*DeepSeek-R1-671B-RAG-Feedback*) produces the largest overall gain, with all four metrics improving by more than 10 points, suggesting that RAG and feedback mechanism together contribute to the overall performance gains.

5.4 Effect of RAG sample size

To validate the trade-off between knowledge sufficiency and noise introduction, we vary the number of retrieved samples m referenced by the Corrector in *DeepSeek-R1-671B-RAG-Feedback*. The objective is to identify the optimal RAG retrieved sample size for maximal correction performance based on engineering experience. The results are shown in Figure 4.

(1) As the number of retrieved samples increases, both judgment and correction performance metrics exhibit a fluctuating trend. It suggests that m has a noticeable impact on correction quality. Additionally, excessive retrieval introduces noise, which in turn degrades the overall judgment and correction quality.

(2) When $m = 5$, the precision, recall, and F1-Score of judgment, achieve relatively high values. Similarly, the accuracy and precision of correction also reach optimal levels. The findings indicate that retrieving 5 samples represents a relatively good configuration for the RAG-enhanced Corrector.

5.5 Reliability constraint verification

To further govern the balance between false acceptance and excessive refinement, we analyze the constraint effect of reliability threshold to identify an optimal threshold setting E. we conduct experiments with different threshold using *DeepSeek-R1-671B-RAG-Feedback* in Figure 5.

(1) Both judgment and correction metrics exhibit an initial increase followed by a decline as the threshold increases. A low threshold increases the probability of false acceptances, while an excessively high threshold causes excessive re-correcting and additional computational overhead.

(2) When $E = 8$, the method yields the best overall performance in all aspects of indicators, and approximately 6% of intermediate results are sent back to the corrector through evaluator feedback for re-correction.

5.6 Case study

To gain deeper insights into how CSCFRL improves correction reliability and adaptability, we analyze two representative cases in Figure 3.

Handling unseen errors 'DiDi' is a homophonic error, where *MacBert4csc-Base-Chinese* and *DeepSeek-R1-671B* fail due to unseen variants in training data. *DeepSeek-R1-671B-RAG* successfully corrects this via retrieved samples. It demonstrates that the DKB plays a crucial role in helping the Corrector understand new errors.

Preventing over-correction *DeepSeek-R1-671B-RAG* incorrectly produce over-correction

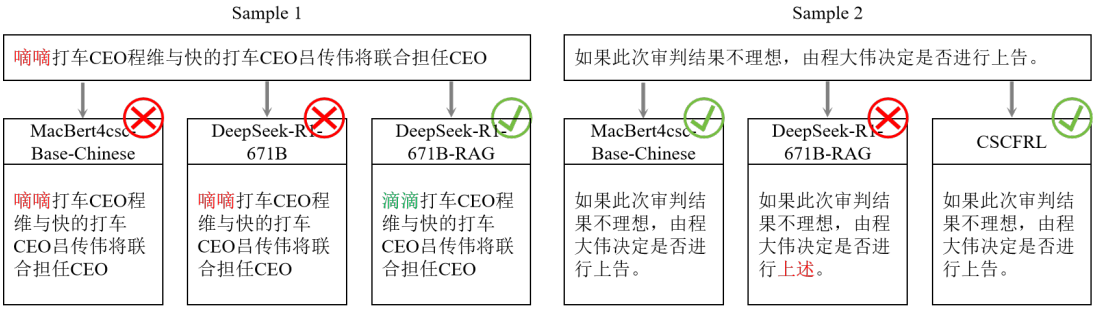
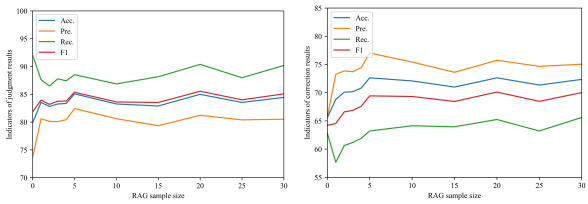
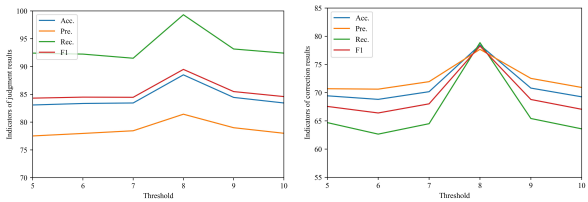


Figure 3: Case analysis



(a) Judgment (b) Correction

Figure 4: Different RAG sample size



(a) Judgment (b) Correction

Figure 5: Different reliability thresholds

result. Under CSCFRL, the Evaluator identifies this discrepancy as violating the semantic constraint, rejects the incorrect modification, preserving the original text via feedback. It illustrates how reliability constraints prevent over-correction, a common failure mode of unconstrained Corrector.

6 Conclusion

To address the challenges of adaptability and reliability in CSC for dynamic environments, we propose CSCFRL, a feedback-driven, retrieval-augmented LLM-based CSC framework to replace expert manual review. A correction function is defined to describe the constraint-aware iterative correction mechanism. It incorporates external knowledge through the DKB to enhance the adaptability of the Corrector. Additionally, an Evaluator is employed to evaluate the correction outputs: satisfactory results are output, while unsatisfactory ones are iteratively refined. The results demonstrate that the proposed method contributes to the possibility of robustness improvement and continual adaptation.

7 Limitations

Our work has several key limitations. First, the Corrector relies on a large 671B model for optimal generalization, making it more suitable for high-accuracy scenarios. As LLM performance improves, smaller models may suffice, and targeted distillation can further balance accuracy and speed. Second, the Evaluator's heuristics may occasionally penalize valid edits requiring insertions/deletions. Third, the dynamic knowledge base in its current form relies on batch updates rather than fully automated online insertion, and long-term error accumulation under continuous deployment remains an open challenge. Finally, our experiments focus exclusively on Chinese.

References

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1*, pages 4171–4186.

Ming Dong, Yujing Chen, Miao Zhang, Hao Sun, and Tingting He. 2024. Rich semantic knowledge enhanced large language models for few-shot Chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7372–7383, Bangkok, Thailand. Association for Computational Linguistics.

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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yong Hu, Fandong Meng, and Jie Zhou. 2024. Cscd-ns: 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.

Haojing Huang, Jingheng Ye, Qingyu Zhou, Yinghui Li, Yangning Li, Feng Zhou, and Hai-Tao Zheng. 2023. A frustratingly easy plug-and-play detection-and-reasoning module for chinese spelling check. *arXiv preprint arXiv:2310.09119*.

Cheng Ji, Dong Lv, Yiwen Xiong, Baoliu Ye, and Bing Cai. 2025. Sir-d based information processing for multiplatform complex social networks. *IEEE Transactions on Computational Social Systems*.

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 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 160–171, Bangkok, Thailand. Association for Computational Linguistics.

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*.

Yinghui Li, Shirong Ma, Shaoshen Chen, Haojing Huang, Shulin Huang, Yangning Li, Hai-Tao Zheng, and Ying Shen. 2025. Correct like humans: Progressive learning framework for chinese text error correction. *Expert Systems with Applications*, 265:126039.

Yinghui Li, Shirong Ma, Qingyu Zhou, Zhongli Li, Yangning Li, Shulin Huang, Ruiyang Liu, Chao Li, Yunbo Cao, and Haitao Zheng. 2022. Learning from the dictionary: Heterogeneous knowledge guided fine-tuning for Chinese spell checking. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 238–249, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Miami, Florida, USA. Association for Computational Linguistics.

Xu Ming. 2021. pycorrector: Text error correction tool.

Rui Song, Yingji Li, Mingjie Tian, Hanwen Wang, Fausto Giunchiglia, and Hao Xu. 2025. Causal keyword driven reliable text classification with large language model feedback. *Information Processing & Management*, 62(2):103964.

Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. Introduction to sighan 2015 bake-off for chinese spelling check. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 32–37.

Cheng Xiong, Gengfeng Zheng, Xiao Ma, Chunlin Li, and Jiangfeng Zeng. 2025. Delphiagent: A trustworthy multi-agent verification framework for automated fact verification. *Information Processing & Management*, 62(6):104241.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Shujian Yang, Shiyao Cui, Chuanrui Hu, Haicheng Wang, Tianwei Zhang, Minlie Huang, Jialiang Lu, and Han Qiu. 2025b. Exploring multimodal challenges in toxic Chinese detection: Taxonomy, benchmark, and findings. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14382–14396, Vienna, Austria. Association for Computational Linguistics.

Hengxiang Zhang, Hongfu Gao, Qiang Hu, Guanhua Chen, Lili Yang, Bingyi Jing, Hongxin Wei, Bing Wang, Haifeng Bai, and Lei Yang. 2024. Chinesesafe: A chinese benchmark for evaluating safety in large language models. *arXiv preprint arXiv:2410.18491*.

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, Online. Association for Computational Linguistics.

Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K Qiu, and Lili Qiu. 2024. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely. *arXiv preprint arXiv:2409.14924*.

Chenxi Zhu, Ziqiang Ying, Boyu Zhang, and Feng Mao. 2022. MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction. In *Findings of the Association for Computational*

Linguistics: ACL 2022, pages 1244–1253, Dublin, Ireland. Association for Computational Linguistics.