Enhancing Chinese Spelling Correction by Mitigating Suboptimal Corrections with Lexical Background Knowledge

Anonymous ACL submission

Abstract

Language models (LMs) trained on large-scale, error-free corpora can be fine-tuned to exhibit strong spelling correction capabilities. However, the limited availability of annotated data and the sparse occurrence of errors introduce biases within the datasets, which can severely affect the model's generalization, often resulting in suboptimal corrections. To address this issue, we propose a knowledge-guided candidate reranking method, KG-Reranking, that leverages lexical background knowledge to select optimal corrections. This method reframes the correction of erroneous words as a two-step process: candidate generation followed by re-ranking. A k-recursive correction algorithm leveraging local and global information is introduced to generate reliable candidates. We further propose a knowledge enhancement framework which generates targeted lexical background knowledge to help the LLM-based reranker to make accurate selections among candidates. We additionally design a knowledge boundary identification method to ensure the accuracy and efficiency of the knowledge generation process. Extensive experiments show that our approach significantly mitigates the suboptimal correction problem in Chinese Spelling Correction (CSC). Moreover, we assess its performance in zero-shot scenarios, demonstrating robust handling of domain shifts.

1 Introduction

Chinese Spelling Correction (CSC) aims to identify and correct errors within sentences, playing a crucial role in natural language processing (NLP) tasks (Hládek et al., 2020). By correcting user inputs or augmented data, CSC enhances recommendation systems' ability to accurately interpret user intent and improves the reliability of search engine inputs (Zhang et al., 2020; Sharma et al., 2023; Wang et al., 2024). The CSC task consists of two core subtasks: error detection, which identifies erroneous words within a sentence, and error correction, which replaces these words to ensure both fluency and grammatical correctness (Wei et al., 2023). Pre-trained language models (PLMs) have demonstrated strong language modeling capabilities, and most current CSC methods leverage PLMs to jointly train these two subtasks in an end-to-end fashion (Huang et al., 2021; Liu et al., 2021; Li et al., 2022; Zhu et al., 2022; Liu et al., 2022; Lv et al., 2023).

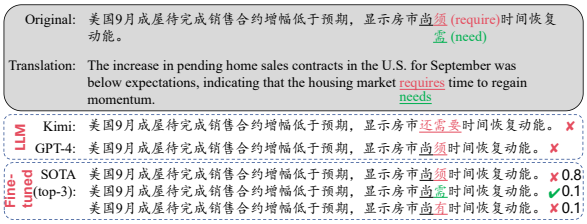


Figure 1: In this example, directly applying LLMs for CSC proves unreliable. Fine-tuned PLMs can effectively detect dissonant collocations and generate a reliable set of correction candidates. However, selecting the optimal correction from these candidates remains a significant challenge.

CSC datasets are often domain-specific, leading models trained on these parallel sentences to replicate the correction patterns observed in the training data when addressing errors (Liu et al., 2022; Lv et al., 2023). To mitigate this issue, researchers have developed increasingly sophisticated methods, such as augmenting training data with confusion word lists (Zhang et al., 2020; Liu et al., 2021, 2022), incorporating phonetic and glyph features (Liu et al., 2021; Huang et al., 2021; Wei et al., 2023), and training on more challenging samples (Li et al., 2022; Lv et al., 2023). While these approaches adjust the model output logits to enhance correction accuracy, they often impose a predefined prior distribution, rather than dynamically adapting to the reasoning context and scenario. As a result, the suboptimal correction problem arising from this

remains challenging to resolve. This issue becomes even more pronounced in real-world applications, as illustrated in Fig. 1.

In fact, the model generates a distribution over the vocabulary, instead of the only word with the highest probability. We conducted a pilot experiment on CSCD-IME (Hu et al., 2022) using the state-of-the-art (SOTA) CSC model (Ming, 2021) to generate the top-3 candidates, then examined whether the correct answers were among them. Our findings revealed that if there were a mechanism to select the correct answer from these candidates, the performance metrics could improve by 2-16%. It suggests that while fine-tuned CSC models can generate reliable candidates, relying solely on the highest-ranked output for correction is arbitrary. The candidate with the highest logit simply reflects the optimal correction according to the model's training data distribution, making the CSC system vulnerable in practical applications. Expanding the candidate search range to the top-k during inference could significantly enhance accuracy.

To address this, we propose KG-Reranking, a lexical background knowledge-guided candidate reranking method. KG-Reranking leverages relevant background knowledge associated with potential corrections to enable the CSC system to select the optimal correction from the candidate set without solely relying on the model's highest logit. This approach effectively raises the upper performance limit of SOTA CSC models.

We developed a novel model-agnostic knowledge enhancement framework to facilitate knowledge generation. In this framework, we categorize lexical background knowledge into two types: factual knowledge, typically related to the physical world, is retrieved by querying a general knowledge base, i.e. large language models (LLMs) in this paper. For non-factual knowledge, we generate contrastive samples with similar text stems, guiding the LLM-based reranker to identify and utilize subtle lexical differences. By injecting the generated knowledge into the ranker, we enable the selection of the optimal correct answers. Additionally, we introduced a knowledge boundary identification method to enhance the precision and efficiency of the generated knowledge.

We conduct ablation experiments on CSCD-IME and SIGHAN15, verifying that our knowledge enhancement framework can effectively mitigate the suboptimal choice problem of existing CSC models. Also, we perform zero-shot evaluations on AlipaySEQ, proving that such a model-agnostic post-processing method is still effective when faced with significant domain changes.

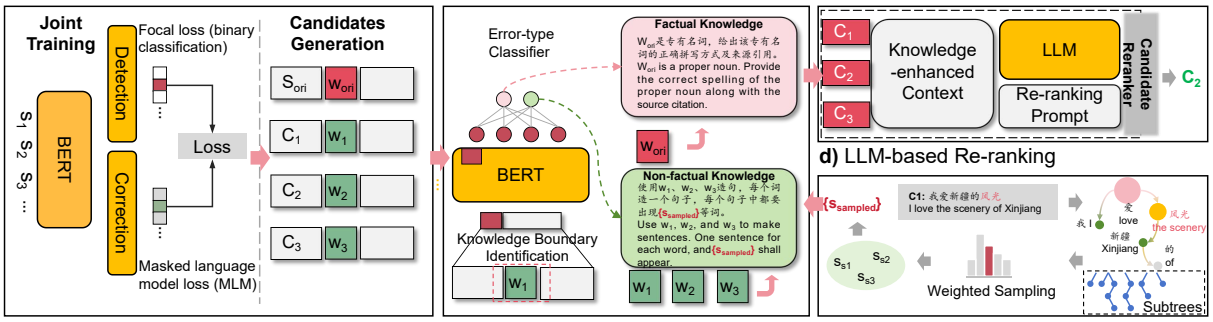
The main contributions of this paper are as follows:

- We propose KG-Reranking, a novel method that frames CSC as a candidate generation and re-ranking process. By incorporating lexical background knowledge into the re-ranking phase, KG-Reranking effectively mitigates the suboptimal correction problem.
- We design a model-agnostic knowledge enhancement framework to produce lexical background knowledge and further optimized this process leveraging a knowledge boundary identification method.
- Through a series of extensive experiments, we validated the effectiveness of KG-Reranking. Additionally, we conducted zero-shot experiments, further demonstrating that KG-Reranking has strong generalization ability across different scenarios.

2 Related Work

Chinese spelling errors frequently arise from incorrect pinyin input or the selection of incorrect candidate words (Hu et al., 2022). Due to the inherent imbalance in the distribution of such errors, existing research has focused extensively on improving the integration of contextual features to better identify and correct erroneous characters during model training (Hládek et al., 2020). Phonological and morphological features have been extensively studied (Zhang et al., 2021; Ji et al., 2021; Liu et al., 2021; Zhu et al., 2022).

With the widespread adoption of PLMs in the NLP field, researchers have explored incorporating task-specific features into both the pre-training and fine-tuning stages to achieve more targeted improvements (Liu et al., 2021; Zhu et al., 2022; Lv et al., 2023; Wei et al., 2023). PLOME (Liu et al., 2021) introduces a confusion masking strategy and incorporates a pronunciation prediction loss during pre-training. Similarly, Li et al. (Li et al., 2022) introduce a character prediction pre-training process that utilizes error-prone characters to generate training samples. While these methods have improved performance to some extent, their effectiveness remains highly dependent on the distribution of both the training and augmented datasets. Moreover,



a) Candidate Generation by CSC Models **b)** Knowledge Enhancement Framework **c)** Sentence Stem Generation **d)** LLM-based Re-ranking

Figure 2: KG-Reranking workflow. As illustrated in sub-figure (a), the input text is first processed by the CSC model, which generates a top-k set of candidate corrections. Required knowledge boundaries are then identified, and a classifier assigns the appropriate type of knowledge to be generated, as depicted in sub-figure (b). For non-factual knowledge, we employ a custom sampling method based on the syntactic parse tree to generate text stems, as shown in sub-figure (c). Finally, the optimal correction candidate is selected using an LLM-based re-ranker, as demonstrated in sub-figure (d).

recent studies have highlighted that even LLMs, despite being trained on vast amounts of data, still encounter significant challenges when addressing CSC tasks (Li et al., 2023).

Another category of approaches focuses on enhancing the post-processing phase (Xiong et al., 2015; Gou and Chen, 2021; Li, 2023). Earlier methods relied on various linguistic features to rerank candidate words, including dictionary-based features, edit distance, segmentation features, and web-based resources, all of which required careful design and balancing of different components (Xiong et al., 2015). More recent research has shifted towards developing smarter methods for refining candidate word filtering, integrating features such as pronunciation and glyph information (Hong et al., 2019), and narrowing the search space by utilizing enhanced confusion sets (Lv et al., 2023). Unlike these approaches, which target improvements through prior knowledge, our proposed method shifts the focus to the knowledge level. The core idea is to generate knowledge that can precisely distinguish subtle differences and to design efficient mechanisms to inject this knowledge into the post-processing re-ranking stage.

The lack of CSC datasets also hinders the development of models with greater generalizability (Yin, 2023). For instance, the SIGHAN series (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015) remains widely used for evaluation, despite its relatively simplistic and somewhat outdated nature. Recently, researchers have focused on developing more targeted datasets (Hu et al., 2022; Yang et al., 2023; Wang et al., 2024).

3 Approach

In this section, we begin with a formal description of the CSC task, followed by an introduction to the proposed method, as illustrated in Fig. 2.

3.1 Preliminaries

In the CSC task, given a sentence $w_1, w_2, ..., w_n$ composed of n tokens[1], it is often unclear whether the sentence contains errors, how many errors are present, and where those errors occur. This uncertainty results in a significantly large search space for model optimization. We adopt the sequence classification paradigm (Liu et al., 2021; Li et al., 2022; Lv et al., 2023) rather than the sequence generation approach (Li et al., 2023; Deng et al., 2024), as the former offers a relatively smaller search space. Consequently, the CSC task can be primarily divided into two subtasks: error detection and error correction. The corresponding training objectives can be formally expressed as follows:

$$DET: \max_{c \in \{0,1\}} \prod_{i=1}^{n} P(c_i = \hat{c}_i \mid w_1, w_2, \ldots, w_n), \quad (1)$$

$$COR: \max_{w_i \in V_{err}} \prod_{i=1}^{n} P(w_i = \hat{w}_i \mid w_1, w_2, \ldots, w_n). \quad (2)$$

Here c_i denotes the detection label of the i_{th} token and $c_i = 1$ represents w_i should be corrected and

[1]The division of tokens typically varies depending on the language and methodology employed. In CSC, it generally follows the vocabulary defined by the selected PLM.

vice versa. $\hat{c}_i$ and $\hat{w}_i$ denote the ground truths. V_{err} represents the error word collection. The category of c_i in Eq. 1 is extremely unbalanced. We will talk about how to deal with it in § 3.2.

3.2 Candidate Generation

The error correction process in CSC aims to identify optimal substitutions. We have observed that considering the top-k candidate answers, rather than just the top-1, significantly enhances model performance. Thus, a key component of our method involves generating an effective top-k candidate set.

Our approach adopts the model architecture of MacBERT4CSC (Ming, 2021), which is jointly trained for error detection and error correction, but differs in that we apply an adaptive loss function to dynamically balance the importance of the two tasks. We additionally develop a k-recursive correction algorithm to maximize the utility of the top-k candidate set.

Error Detection and Correction. The detection task is conducted by extracting features using BERT (Kenton and Toutanova, 2019) and performing binary classification based on the representation from the last hidden layer of each token. Note that there are only two types of labels and the positive and negative samples are severely imbalanced. Thus focal loss (Lin et al., 2017) is employed.

Error correction is implemented utilizing the inline position prediction ability of BERT (Kenton and Toutanova, 2019). Obtaining the error positions in the detection process, BERT with the language model header predicts the most suitable token based on the last hidden states. The loss function for this component directly adopts the language model loss, i.e., the masked language model (MLM) loss in BERT.

Joint Learning Loss Function. A straightforward joint training method is to add the loss functions with weighted summation, which can be formally expressed as follows:

$$\mathcal{L} = w * \mathcal{L}_{cor} + (1 - w) * \mathcal{L}_{det}, \tag{3}$$

where w is a settled weight parameter. We performed grid search for w. Additionally, we design a dynamic weight $w = \mathcal{L}_{cor}/(\mathcal{L}_{cor} + \mathcal{L}_{det})$ that can automatically scale based on the magnitude of the loss terms.

K-recursive Correction. Multi-round correction (Li, 2023) has been employed to tackle the challenge of multiple errors within a given context. Some errors can be corrected using only partial context, while additional information may introduce redundancy, potentially impairing the model's inference capabilities. We introduce a k-recursive correction algorithm that integrates both global and local information to mitigate this issue. The corresponding pseudo-code is presented in Algorithm 1.

Algorithm 1 Algorithm of K-recursive Correction

Input: CSC model M, text t, candidate set size k
Output: Candidate set $\mathcal{C}$

```
error_poss = M_det(t) //detection part
error_segs = get_err_segs(t, error_poss) //get segments containing errors
sens = [t]
sens_p = error_segs
while k > 0 do
    for each sen ∈ sens do
        sens = M_cor(sen, k)
        C ← sens
    end for
    for each sen_p ∈ sens_p do
        sens_p = M_cor(sen_p, k)
        C ← complete(sens_p) //complete local corrections and add to C
    end for
    k = k − 1
end while
C ← filter(C) //use a dictionary to narrow the candidate set
return C
```

3.3 Knowledge Enhancement Framework

Due to the inductive bias inherent in autoregressive models (Lin et al., 2021), widely used CSC models based on PLMs often face challenges in identifying subtle distinctions among correction candidates. To address this limitation, we develop a model-agnostic knowledge enhancement framework capable of automatically generating specific knowledge to effectively differentiate between these words.

Knowledge Boundary Identification. To acquire effective knowledge, we extract meaningful short phrase fragments that include erroneous characters through a word segmentation process utilizing a domain-specific dictionary. This dictionary

is constructed using a novel word discovery algorithm (Shoujian et al., 2022) based on left-right information entropy and coherence, which ensures the knowledge's relevance and effectiveness in new domains. Additionally, we implement a voting mechanism across multiple CSC models to filter out corrections with high confidence, thus refining the scope of knowledge generation.

Knowledge Generation. Within this framework, we classify lexical background knowledge into two categories. The first is factual knowledge, typically associated with named entities. We perform named entity recognition[2] on sentence t to obtain the label l of the differentiating phrase p, then construct samples in the form of $([CLS]p[SEP]t, l)$, and train a binary classifier based on BERT. Relevant knowledge is extracted by querying general knowledge bases, specifically LLMs. We generate contrastive samples featuring similar stems for t that include p and utilize these to guide LLMs in producing knowledge related to word usage. The stems for generating these samples are derived from the root of t's syntactic parse tree, p, and words sampled with weights based on the size of each subtree, as illustrated in Fig. 2. Prompts design is illustrated in § 4.5.

3.4 Candidate Reranker

We propose a straightforward yet effective LLM-based reranker for candidate selection. To mitigate the issue of hallucination (Tonmoy et al., 2024) in LLMs, we incorporate specific explicit constraints into the prompts. The re-ranking process can be formally described as follows:

$$\hat{y} = LLM(x|candidates, c_{know}, c_{task}, c_{gen}, \theta), \quad (4)$$

where $candidates$ are generated following § 3.2, c_{know} denotes the lexical background knowledge, c_{task} explicitly declares the task type, and c_{gen} formats the output. While c_{task} and c_{gen} define the task in a natural language form, c_{know} injects knowledge into the reranker.

4 Experiments

4.1 Experimental Setup

Datasets. Dataset statistics are summarized in Table 1. The CSCD-IME dataset (Hu et al., 2022) was developed to address the limitations outlined in § 2, and we use it as one of the primary evaluation bases. The CSCD-IME pseudo data is combined with the training set for fine-tuning. Additionally, SIGHAN15 (Tseng et al., 2015) is evaluated, with the corresponding training set augmented following the methodology outlined in (Ming, 2021). To assess the generalization capability of our method, we also include AlipaySEQ (Wang et al., 2024), a CSC dataset derived from real-world search engine queries.

Table 1: Statistics of evaluation datasets. #Sen refers to the number of sentences, while #Len denotes the average sentence length.

Dataset	#Sen	#Len	Misspellings
CSCD-IME (pseudo+train)	2059942	31.2	2949251
CSCD-IME (dev)	5000	57.5	2554
CSCD-IME (test)	5000	57.6	2528
SIGHAN15 (train)	251835	42.5	351217
SIGHAN15 (dev)	27981	42.8	39114
SIGHAN15 (test)	1100	30.7	715
AlipaySEQ (test)	1000	7.1	553

Baselines. We select several strong baselines for comparison. **BERT** (Kenton and Toutanova, 2019) directly fine-tunes on the training set and performs MLM prediction on every position. **Soft-Masked BERT** (Zhang et al., 2020) designs a detection network based on bidirectional GRU to differentiate the embedding representations of incorrect characters, simultaneously training both the detection and correction tasks. **MacBERT4CSC** (Ming, 2021) incorporates a fully connected layer to explicitly handle the error detection task, which is jointly trained with the error correction task. **T5** (Raffel et al., 2020) fine-tunes on the sequence-to-sequence task using the training set and generates text during inference. **Kimi**[3] is a commercial LLM. We use Kimi as a baseline in a question-answering manner, and it is also the LLM employed in our proposed method.

Implementation Details. We utilize the standard base configuration for all transformer-based models (Kenton and Toutanova, 2019). Training is conducted using 8 *GeForce RTX 3090* graphics cards. Consistent with previous studies (Li et al., 2022; Hu et al., 2022; Wang et al., 2024), we report the evaluation metrics at the sentence level, including precision, recall, and F1, which are averaged over three experiments.

[2]https://spacy.io/.

[3]https://www.moonshot.cn/.

4.2 Validity of Candidates

Pilot Experiment. This paper conceptualizes the error correction process in CSC as a candidate generation and re-ranking procedure. To demonstrate the effectiveness of this approach, we conducted a pilot experiment using MacBERT4CSC to generate the top-k correction candidates. If the ground truth is included among these candidates, it is deemed the optimal correction; if not, the first candidate is selected. The sentence-level Precision, Recall, and F1 scores are presented in Fig. 3.

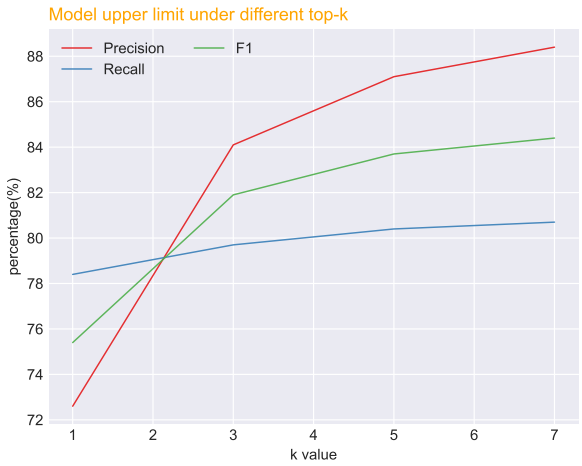


Figure 3: Sentence-level correction precision, recall, and F1 under different k value.

We observed that employing a method capable of effectively selecting the optimal correction from the top-k candidates can significantly raise the upper limit of existing CSC models, resulting in metric improvements ranging from 2-16%. Furthermore, as k increases, the rate of metric improvement tends to diminish, which informs our choice of k values in subsequent experiments.

Hyperparameters in Loss Functions. We employed a grid search to optimize the loss weights for joint training. The sentence-level precision, recall, and F1 at various w values are depicted in Fig. 4. The results indicate that a weight of $w = 0.3$ yielded the best performance. Interestingly, the adaptive weights did not reach optimal levels. Setting the correction loss weight (i.e., the language model loss weight) to zero resulted in failure to train the model correctly. Conversely, when the detection loss weight was set to zero, the model still demonstrated strong error detection capabilities. We attribute this to the initialization with *shibing624/macbert4csc-base-chinese* which has been fine-tuned on other CSC datasets. Overall, while the language model loss is essential, the effects of the two loss components are not equally balanced. For specific datasets, employing fixed weights can lead to optimal outcomes.

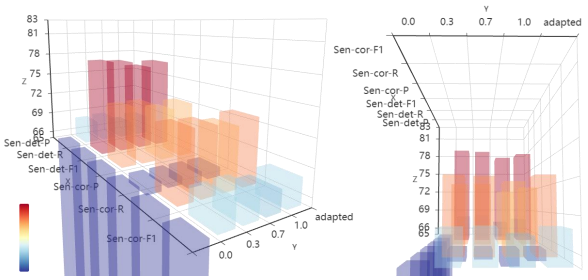


Figure 4: Sentence-level correction precision, recall, and F1 under different loss weight w.

K-recursive Correction. We utilized a well-trained MacBERT4CSC model to evaluate the effectiveness of the k-recursive algorithm. Our assessment focused on the impact of local context correction and various values of k value, with the experimental results presented in Fig. 5. When incorporating local context information, a k value of 3 achieves the same effectiveness as a k value of 7 without local context, while also requiring less computational time. As k increases, both the number of candidates and the computational time rise accordingly.

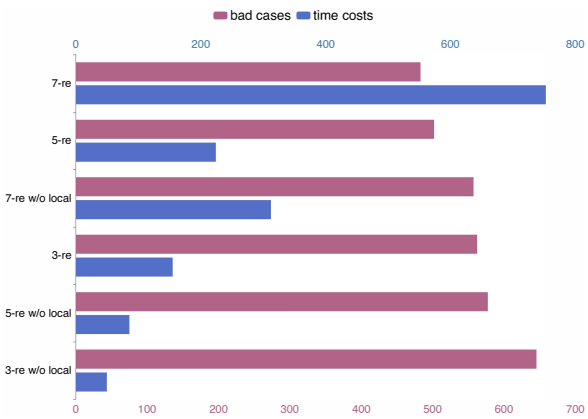


Figure 5: Evaluation of k-recursive correction under different settings. The blue bar indicates the number of erroneous cases identified on the test set, while the red bar represents the inference time measured in seconds.

Theoretically, for a text containing a errors, assuming the reasoning time of the CSC model for a given text as $O(b)$, the upper bound of time complexity for the k-recursive algorithm with local context correction is $O(abk!n)$, while the upper bound without local context is $O(bk!n)$. Given the

Table 2: Main evaluation results on CSCD-IME and SIGHAN15. The best results are shown in bold.

	Method	Sentence-level					
		Detection-level			Correction-level		
		P	R	F1	P	R	F1
CSCD-IME	BERT (Kenton and Toutanova, 2019)	79.0	73.6	76.2	75.6	70.5	73.0
	Soft-Masked BERT (Zhang et al., 2020)	79.2	74.9	77.0	75.8	71.6	73.6
	MacBERT4CSC (Ming, 2021)	71.7	**79.1**	75.2	68.6	**75.7**	72.0
	T5 (Raffel et al., 2020)	60.6	65.1	62.8	57.8	62.2	59.9
	Kimi	62.4	44.4	51.9	59.2	42.2	49.3
	Soft-Masked BERT+re-ranking	81.0	73.3	77.0	77.1	70.0	73.4
	MacBERT4CSC+re-ranking	78.3	77.5	77.9	74.4	73.5	74.0
	KG-Reranking	**81.5**	76.3	**78.8**	**77.3**	72.9	**75.0**
SIGHAN15	BERT (Kenton and Toutanova, 2019)	74.2	78.0	76.1	71.6	75.3	73.4
	Soft-Masked BERT (Zhang et al., 2020)	73.7	73.2	73.5	66.7	66.2	66.4
	MacBERT4CSC (Ming, 2021)	72.3	**79.1**	75.5	69.5	**76.1**	72.7
	T5 (Raffel et al., 2020)	68.0	67.7	67.8	65.7	65.3	65.5
	Kimi	31.0	49.6	38.2	27.4	43.7	33.7
	Soft-Masked BERT+re-ranking	74.8	71.6	73.2	70.5	64.8	67.5
	MacBERT4CSC+re-ranking	76.1	77.4	76.7	74.4	74.2	74.3
	KG-Reranking	**77.3**	76.8	**77.0**	**76.0**	73.7	**74.8**

high inference cost associated with the LLM-based reranker, we selected $k = 3$.

4.3 Main Results

We conducted a series of experiments on the CSCD-IME and SIGHAN15 datasets to evaluate the effectiveness of the proposed method as shown in Table 2. Baseline models (excluding Kimi) were trained using CSCD-IME (pseudo+train) or SIGHAN (train) as the training set. We selected the best model parameters tested on the development set as the final parameters and evaluated various metrics on the test set.

The experimental results indicate that KG-Reranking achieves a new SOTA. The application of candidate generation and re-ranking processes to existing CSC models (MacBERT4CSC and Soft-Masked BERT) resulted in a significant increase in precision, with an improvement of up to 7%. Consequently, there is a slight decline in recall. From the F1 perspective, re-ranking provided a substantial gain for MacBERT4CSC, while it remains relatively unchanged for Soft-Masked BERT. We further adopt knowledge enhancement on MacBERT4CSC+re-ranking, achieving an additional improvement of 1-3% across all metrics.

Comparing all baselines, MacBERT4CSC exhibits a higher recall rate than other models. We argue this is due to MacBERT4CSC's explicit separation in jointly training the error detection and correction tasks. CSC models with higher recall rates generate more valid candidates, thus providing larger space for improvement. Notably, through hyperparameter searching, we pushed the recall rate to new heights, laying the ground for maximizing the effectiveness of our method. Soft-Masked BERT also demonstrates good overall metrics, but due to its characteristic of high precision and low recall, candidate generation and re-ranking does not yield significant improvements. This phenomenon further highlights that our method is more suited for high-recall models. Generative methods, whether the fine-tuned T5 or the commercial LLM Kimi, exhibits lower performance. We argue that current generative methods still face considerable challenges in handling tasks that require precise control.

Table 3: Seven categories of situations inconsistent with ground truths and 5-7 are unavoidable.

	Desc.	Prop. (%)	E.g. (src/pre/gt)
1	Named entity bias	13.7	簧门后街/前门后街/簧门后街
2	Prioritizing phonetic similarity	8.3	湘西/清晰/详细
3	Rare word selection	1.7	沁凉/清凉/沁凉
4	Prediction error	41.2	直观映像/直观印响/直观印象
5	Labeling error	17.1	西安是/西安市/西安是
6	No standard answer	14.3	今日/近日/今日
7	Original sentence illegal	3.7	四个碌/四个佬/四个碌

The extension description src/pre/gt denotes the original sentence, the corrected result, and the ground truth respectively.
The green represent correct expressions and cannot be further improved in the current context; the red represent incorrect expressions with grammatical or semantic errors; the blue are also correct expressions, but can be further improved by combining external knowledge, such as pinyin, knowledge base, etc.

Furthermore, the experimental results show that the metrics of KG-Reranking are far from reaching the theoretical upper limits. We conducted a manual evaluation of the samples and attributed the reasons for inference errors, as illustrated in Table 3. We found that approximately 1/3 of the

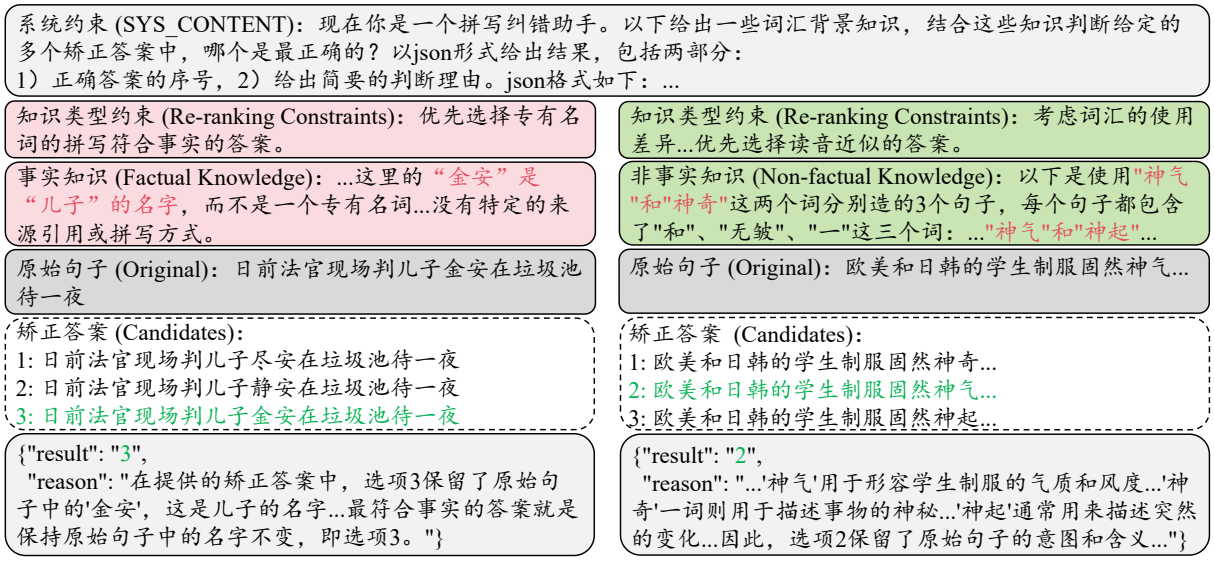


Figure 6: The re-ranking process is informed by lexical background knowledge.

evaluation errors were unavoidable. The long-tail phenomenon prevalent in CSC remains challenging to overcome, even with LLMs that possess rich world knowledge.

4.4 Zero-shot Evaluation

We performed a zero-shot evaluation on the Alipay-SEQ dataset to assess the generalization capability of the proposed method. Sentence-level metrics are reported in Table 4.

Table 4: Zero-shot evaluation results on AlipaySEQ. SOTA (fine-tuned) results are reported individually.

Method	Detection-level			Correction-level		
	P	R	F1	P	R	F1
SOTA (trained) (Wang et al., 2024)	88.4	82.2	85.2	80.1	74.2	77.0
ChatGPT (Wang et al., 2024)	-	-	-	23.0	**35.9**	27.6
MacBERT4CSC (Ming, 2021)	51.7	**49.1**	50.4	34.6	32.8	33.7
KG-Reranking	**77.2**	42.5	**54.8**	**58.1**	32.0	**41.2**

The results indicate that when there is a substantial shift in the distribution of test data—from Weibo media posts to search engine queries—MacBERT4CSC experiences a notable decline in performance. In contrast, KG-Reranking demonstrates an improvement of approximately 25% in accuracy, highlighting the effectiveness of our method in zero-shot scenarios. However, since the candidates are generated by MacBERT4CSC, the recall rate remains at a comparatively lower level. Nonetheless, both methods outperform ChatGPT.

4.5 Case Study

We demonstrate how our method transforms the initial suboptimal corrections into the optimal through two cases. As shown in the left half of Fig. 6, the k-recursive algorithm guides the CSC model to output three reliable candidates, but the optimal correction does not gain the highest logit. After knowledge enhancement, a factual knowledge is generated for this group of candidates. By injecting this knowledge into the context, the LLM-based reranker is correctly guided to select the optimal correction. The right half of Fig. 6 shows a similar process, except that it is achieved based on non-factual knowledge. These two cases demonstrate that our method can effectively leverage lexical background knowledge to solve real-world problems. Furthermore, the constraints imposing to the reranker also assist in regulating the generation process of the LLM.

5 Conclusions

We present a method called KG-Reranking to address the issue of suboptimal corrections in CSC tasks. The core idea involves utilizing a k-recursive algorithm to generate reliable candidates, then identifying the lexical differences and produce targeted background knowledge. By injecting this differential knowledge into the context, the LLM-based reranker can effectively select the optimal correction. Overall, our approach offers a novel perspective for overcoming the inherent inductive bias present in PLMs. Experiments have demonstrated that our method is both effective and robust. A promising direction for future research is to extend and adapt this methodology to other tasks similar to CSC that require precise control.

6 Limitations

Our method involves multiple processing stages, which may introduce implementation complexity.

Additionally, the knowledge generation and candidate re-ranking processes are based on commercial LLMs, coming with high costs. Using smaller LLMs or distilled models with comparable performance might help alleviate this issue.

Since our method relies on a fine-tuned CSC model to generate candidate sets, the choice of the CSC model is also a factor to consider when applying this method in practice.

References

Hua Deng, Kang Xu, Rongsheng Li, and Yifei Qi. 2024. Chinese text error correction based on pe-t5 model. In *Proceedings of the 2024 5th International Conference on Computing, Networks and Internet of Things*, pages 223–227, Tokyo Japan. ACM.

Wei Gou and Zheng Chen. 2021. Think twice: A post-processing approach for the chinese spelling error correction. *Applied Sciences*, 11(13):5832.

Daniel Hládek, Ján Staš, and Matúš Pleva. 2020. Survey of automatic spelling correction. *Electronics*, 9(10):1670.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. Faspell: A fast, adaptable, simple, powerful chinese spell checker based on dae-decoder paradigm. In *Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019)*, pages 160–169.

Yong Hu, Fandong Meng, and Jie Zhou. 2022. Cscd-ime: Correcting spelling errors generated by pinyin ime. *arXiv preprint arXiv:2211.08788*.

Li Huang, Junjie Li, Weiwei Jiang, Zhiyu Zhang, Minchuan Chen, Shaojun Wang, and Jing Xiao. 2021. Phmospell: Phonological and morphological knowledge guided chinese spelling check. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5958–5967.

Tuo Ji, Hang Yan, and Xipeng Qiu. 2021. Spellbert: A lightweight pretrained model for chinese spelling check. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3544–3551.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Xiang Li. 2023. Towards robust chinese spelling check systems: Multi-round error correction with ensemble enhancement. In Fei Liu, Nan Duan, Qingting Xu, and Yu Hong, editors, *Natural Language Processing and Chinese Computing*, volume 14304, pages 325–336. Springer Nature Switzerland, Cham.

Yinghui Li, Haojing Huang, Shirong Ma, Yong Jiang, and Yangning Li. 2023. On the (in)effectiveness of large language models for chinese text correction. *arXiv preprint*.

Yinghui Li, Qingyu Zhou, Yangning Li, Zhongli Li, and Ruiyang Liu. 2022. The past mistake is the future wisdom: Error-driven contrastive probability optimization for chinese spell checking. *arXiv preprint*.

Chu-Cheng Lin, Aaron Jaech, Xin Li, Matthew R. Gormley, and Jason Eisner. 2021. Limitations of Autoregressive Models and Their Alternatives.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2980–2988.

Shulin Liu, Shengkang Song, Tianchi Yue, Tao Yang, Huihui Cai, TingHao Yu, and Shengli Sun. 2022. Craspell: A contextual typo robust approach to improve chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3008–3018.

Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, and Di Wang. 2021. Plome: Pre-training with misspelled knowledge for chinese spelling correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2991–3000.

Qi Lv, Ziqiang Cao, Lei Geng, Chunhui Ai, Xu Yan, and Guohong Fu. 2023. General and domain-adaptive chinese spelling check with error-consistent pretraining. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(5):1–18.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, and Sharan Narang. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Sanat Sharma, Josep Valls-Vargas, Tracy Holloway King, Francois Guerin, and Chirag Arora. 2023. Contextual multilingual spellchecker for user queries. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3395–3399, Taipei Taiwan. ACM.

Y. U. Shoujian, Wang Baoying, and L. U. Ting. 2022. Sentiment Lexicon Construction Based on Improved Left-Right Entropy Algorithm. *Journal of Donghua University (English Edition)*, 39(1).

SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

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.

Yue Wang, Zilong Zheng, Zecheng Tang, Juntao Li, and Zhihui Liu. 2024. Towards better chinese spelling check for search engines: A new dataset and strong baseline. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 769–778, Merida Mexico. ACM.

Xiao Wei, Jianbao Huang, Hang Yu, and Qian Liu. 2023. Ptcspell: Pre-trained corrector based on character shape and pinyin for chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6330–6343.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese spelling check evaluation at sighan bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 35–42.

Jinhua Xiong, Qiao Zhang, Shuiyuan Zhang, Jianpeng Hou, and Xueqi Cheng. 2015. Hanspeller: A unified framework for chinese spelling correction. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language*.

Liner Yang, Xin Liu, Tianxin Liao, Zhenghao Liu, Mengyan Wang, Xuezhi Fang, and Erhong Yang. 2023. Is chinese spelling check ready? understanding the correction behavior in real-world scenarios. *AI Open*, 4:183–192.

Xunjian Yin. 2023. Overview of the nlpcc 2023 shared task: Chinese spelling check. In Fei Liu, Nan Duan, Qingting Xu, and Yu Hong, editors, *Natural Language Processing and Chinese Computing*, volume 14304, pages 337–345. Springer Nature Switzerland, Cham.

Liang-Chih Yu, Lung-Hao Lee, Yuen-Hsien Tseng, and Hsin-Hsi Chen. 2014. Overview of sighan 2014 bake-off for chinese spelling check. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 126–132.

Ruiqing Zhang, Chao Pang, Chuanqiang Zhang, Shuohuan Wang, and Zhongjun He. 2021. Correcting chinese spelling errors with phonetic pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2250–2261.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. *arXiv preprint*.

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.