Mixture of Small and Large Models for Chinese Spelling Check

Anonymous ACL submission

Abstract

In the era of large language models (LLMs), the Chinese Spelling Check (CSC) task has seen various LLM methods developed, yet their performance remains unsatisfactory. In contrast, fine-tuned BERT-based models, relying on high-quality in-domain data, show excellent performance but suffer from edit pattern overfitting. This paper proposes a novel dynamic mixture approach that effectively combines the probability distributions of small models and LLMs during the beam search decoding phase, achieving a balanced enhancement of precise corrections from small models and the fluency of LLMs. This approach also eliminates the need for fine-tuning LLMs, saving significant time and resources, and facilitating domain adaptation. Comprehensive experiments demonstrate that our mixture approach significantly boosts error correction capabilities, achieving state-of-the-art results across multiple datasets. Our code is available at https://anonymous.4open.science/r/MSLLM.

1 Introduction

The Chinese Spelling Check (CSC) task focuses on identifying and correcting spelling errors in given sentences, as demonstrated in Figure 2. Such errors may lead to comprehension difficulties and adversely affect various natural language processing applications, including machine translation and information retrieval. Given its practical significance, CSC has gained substantial research attention in recent years.

In the era of pre-trained language models, BERT-based approaches (Devlin et al., 2019) have emerged as the dominant solution for the CSC task (Zhang et al., 2020; Cheng et al., 2020; Li et al., 2022a). Based on the characteristic where the input and output are of equal length for the CSC task, they effectively treat CSC as a character-level classification problem. That is, for each character in the input sentence, they predict whether it needs

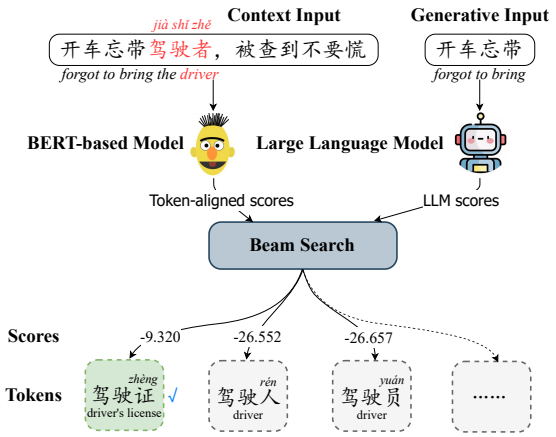

Figure 1: Overview of our approach. The correct sentence is "开车忘带驾驶证，被查到不要慌" (If you forgot to bring the driver's license while driving, don't panic when being checked).

to be corrected. During the fine-tuning process using in-domain data, either artificial or real-world, these BERT-based models can adeptly capture intricate relationships between edit pairs. However, this process sometimes leads to overfitting specific edit pairs and generating erroneous sentences.

On the other hand, the primary objective of the CSC task is to generate fluent and accurate sentences. As a result, generative models are particularly well-suited for this task. With the advent of large language models (LLMs), which boast extensive parameter sizes and vast training datasets, these models exhibit strong cross-domain generalization abilities. Researchers have explored various strategies for utilizing LLMs in the CSC task, focusing on whether to fine-tune these models (Li et al., 2023; Dong et al., 2024). Nevertheless, LLMs often over-polish the text for fluency, choosing expressions they consider better, which leads to inconsistencies between the prediction and the input lengths. Li et al. (2024) attempted to address this issue by employing a character-level tokenization and supervised fine-tuning (SFT) technique. How-

ever, this approach requires significant time and resources.

Unlike previous LLM strategies, Zhou et al. (2024) fully utilize the language modeling capabilities of LLMs. They treat open-source LLMs as pure language models and manually design a distortion model to ensure faithfulness between inputs and outputs by leveraging phonetic and glyph similarities. This approach is particularly effective in zero-shot scenarios.

In general, compared to fine-tuned small models, the correction performance of LLM approaches remains unsatisfactory. Liu et al. (2024a) tried using an unfine-tuned small model as an arbiter to choose between predictions from both small and large models, yet only achieved minimal improvements. Indeed, due to their different inference mechanisms, BERT-based models and LLMs inherently excel in different aspects of error correction: precision and domain adaptability for BERT-based models, and fluency for LLMs. We believe that a deeper integration of these two models at the inference stage could be a more effective strategy.

Motivated by these insights, this paper presents a novel dynamic mixture approach that strategically integrates a BERT-based model with an LLM. Specially, we incorporate the probability distribution from the BERT-based model into the LLM's beam search process, thereby preserving the correction capabilities of both models while mitigating the small model's overfitting tendencies through the LLM's robust language modeling. Furthermore, by fine-tuning the small model instead of the LLM, we significantly cut down on the resources and time needed for domain adaptation.

Our contributions are summarized as follows:

- We propose a novel and straightforward approach that combines a BERT-based model with an LLM, leveraging their complementary strengths to further enhance error correction performance.
- Our approach does not require fine-tuning the LLM, significantly reducing time and resource costs while preserving its strong generalization capability.
- Extensive experiments on multiple mainstream public benchmarks show that our mixture approach substantially boosts correction performance, achieving SOTA results on several datasets.

Type	Probability
`Identical`	0.962
`Same Pinyin`	0.023
`Similar Pinyin`	0.008
`Similar Shape`	0.004
`Unrelated`	0.003

Table 1: The distribution of the different distortion types extracted from Zhou et al. (2024).

2 The Basic Approaches

Given an input sentence comprising n characters, denoted as $\boldsymbol{x} = x_1 \cdots x_i \cdots x_n$, the objective of a CSC model is to generate a corresponding corrected sentence, represented as $\boldsymbol{y} = y_1 \cdots y_i \cdots y_n$, in which all erroneous characters in $\boldsymbol{x}$ are replaced with the correct ones. In other words, CSC models aim to find an optimal sentence $\boldsymbol{y}$ that maximizes $\texttt{score}(\boldsymbol{x}, \boldsymbol{y})$. Currently, there exist two representative CSC models, i.e., the generative LLM-based models, and the classification-based models.

2.1 The LLM-based Approach

Recently, Zhou et al. (2024) proposed a novel prompt-free training-free LLM-based approach for the CSC task. The key is treating LLM as a pure language model. They designed a distortion model (DM) to model the relationships between $\boldsymbol{x}$ and $\boldsymbol{y}$, and more precisely to ensure $\boldsymbol{y}$ is faithful to $\boldsymbol{x}$.

$$
\begin{aligned}
\texttt{score}(\boldsymbol{x}, \boldsymbol{y}) &= \log p_{\texttt{LLM}}(\boldsymbol{y}) + \log p_{\texttt{DM}}(\boldsymbol{x} \mid \boldsymbol{y}) \\
p_{\texttt{LLM}}(\boldsymbol{y}) &= \prod_{i=1}^{n} p_{\texttt{LLM}}(y_i \mid \boldsymbol{y}_{<i}) \\
p_{\texttt{DM}}(\boldsymbol{x} \mid \boldsymbol{y}) &= \prod_{i=1}^{n} p_{\texttt{DM}}(x_i \mid y_i) \\
p_{\texttt{DM}}(x_i \mid y_i) &= p(\texttt{type}(x_i, y_i))
\end{aligned}
$$

The LLM component generates a sentence in an auto-regressive manner, and gives us the probability, i.e., $p_{\texttt{LLM}}(\cdot)$.

The DM component first classifies each character pair (e.g., (c_1, c_2)), into five types, and then obtains the pre-defined corresponding probability, as shown in Table 1.

2.2 The Classification Approach

In the pre-trained model era, most mainstream CSC models follow a BERT-based approach (Zhang et al., 2020; Xu et al., 2021; Li et al., 2022a; Liu et al., 2024b). These models treat CSC as a local

classification problem, i.e., for each character, they determine whether it needs to be corrected and, if so, which character it should be modified to:

$$p_{\texttt{SM}}(y \mid \boldsymbol{x}, i) = \texttt{softmax}(\ \texttt{MLP}(\boldsymbol{h_i})\ )[y]$$

where $\boldsymbol{h_i}$ represents the contextual representation of the i-th character obtained from the BERT-based encoder, and SM is the abbreviation for the small classification model. By selecting the character with the highest probability at each position, i.e., $y^* = \arg\max_{y \in \mathcal{V}} p(y \mid \boldsymbol{x}, i)$, where $\mathcal{V}$ represents the vocabulary, we can obtain the final correction result $\boldsymbol{y}$. This classification-based approach demonstrates strong fitting capabilities in specific domains with high-quality training datasets, and it also offers fast decoding speed. However, it lacks a global $\texttt{score}(\boldsymbol{x}, \boldsymbol{y})$, which results in the model tending to memorize specific edit pairs, leading to locally optimal solutions.

3 Our Mixture Approach

In this work, we propose a straightforward mixture approach to integrate the power of both small and large models. On the one hand, the training-free LLM-based approach of Zhou et al. (2024) exhibits remarkable ability in domain generalization. On the other hand, the small classification models can be effectively trained on in-domain labeled data, if available, and thus can dramatically improve in-domain performance.

$$\begin{aligned}\texttt{score}(\boldsymbol{x}, \boldsymbol{y}) &= \log p_{\texttt{LLM}}(\boldsymbol{y}) + \log p_{\texttt{DM}}(\boldsymbol{x} \mid \boldsymbol{y}) \\ &\quad + \log p_{\texttt{SM}}(\boldsymbol{y} \mid \boldsymbol{x}) \\ p_{\texttt{SM}}(\boldsymbol{y} \mid \boldsymbol{x}) &= \prod_{i=1}^{n} p_{\texttt{SM}}(y_i \mid \boldsymbol{x}, i)\end{aligned}$$

3.1 Incremental Decomposition

In the inference phase, our model follows the LLM component, and produces $\boldsymbol{y}$ from left to right in an auto-regressive manner. Thus, we give an incremental decomposition of a partial output sentence as follows:

$$\begin{aligned}\texttt{score}(\boldsymbol{x}, \boldsymbol{y}_{\leqslant i}) = \texttt{score}(\boldsymbol{x}, \boldsymbol{y}_{<i}) \\ + \log p_{\texttt{LLM}}(y_i \mid \boldsymbol{y}_{<i}) \\ + \log p_{\texttt{DM}}(x_i \mid y_i) \\ + \log p_{\texttt{SM}}(y_i \mid \boldsymbol{x}, i)\end{aligned}$$

3.2 Token-based Generation and Beam Search Decoding

Current LLMs usually generate a sentence token by token, i.e., using tokens as the basic units. Therefore, the output sentence can also be denoted as $\boldsymbol{y} = \boldsymbol{t}_1 \cdots \boldsymbol{t}_k \cdots \boldsymbol{t}_o$, where a token is composed of $\ell \geqslant 1$ characters, i.e., $\boldsymbol{t}_k = y_{i-\ell+1} \ldots y_i$.

Moreover, we follow Zhou et al. (2024) and employ their proposed faithfulness reward to further encourage that $\boldsymbol{y}$ retains the same meaning as $\boldsymbol{x}$.

The full model. Combining the above two factors, we give the incremental decomposition of our full model.

$$\begin{aligned}&\texttt{score}(\boldsymbol{x}, \boldsymbol{y}_{\leqslant i} = \boldsymbol{t}_{\leqslant k}) = \texttt{score}(\boldsymbol{x}, \boldsymbol{t}_{<k}) \\ &+ \log p_{\texttt{LLM}}(\boldsymbol{t_k} \mid \boldsymbol{t}_{<k}) \\ &+ (1 + H_{\texttt{LLM}}(\cdot)) \times \begin{pmatrix} \alpha \times \log p_{\texttt{DM}}(\boldsymbol{x}, i \mid \boldsymbol{t}_k) \\ + \\ \beta \times \log p_{\texttt{SM}}(\boldsymbol{t}_k \mid \boldsymbol{x}, i) \end{pmatrix}\end{aligned} \tag{1}$$

where α and β are the weights of the distortion component in the generative LLM-based model and the small BERT-based model, respectively; $H_{\texttt{LLM}}(\cdot)$ corresponds to the faithfulness reward of Zhou et al. (2024), and represents the entropy of $p_{\texttt{LLM}}(\boldsymbol{t}_k \mid \boldsymbol{t}_{<k})$, i.e., the probability distribution of the LLM component regarding the generation of $\boldsymbol{t}_k$. Higher entropy means the LLM is more uncertain about the selection of the next token, and thus the other two components obtain higher weights.

The token-level formulations for the distortion component and the small model are defined as follows:

$$\begin{aligned}p_{\texttt{DM}}(\boldsymbol{x}, i \mid \boldsymbol{t}_k) &= \prod_{j=1}^{\ell} p_{\texttt{DM}}(x_{i-\ell+j} \mid \boldsymbol{t}_k[j-1]) \\ p_{\texttt{SM}}(\boldsymbol{t}_k \mid \boldsymbol{x}, i) &= \prod_{j=1}^{\ell} p_{\texttt{SM}}(\boldsymbol{t}_k[j-1] \mid \boldsymbol{x}, i-\ell+j)\end{aligned}$$

Beam search. During inference, we follow Zhou et al. (2024) and employ beam search, in order to explore a larger search space. At each decoding step, we retain the top K candidates with the highest scores, and based on them build candidates for the next step.

4 Experimental Setup

4.1 Datasets

Chinese Learner Texts. Following the conventions of previous works, we employ the **SIGHANs** datasets (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015) as our benchmarks, which are derived from Chinese learner texts. We utilize a revised version of SIGHANs (Yang et al., 2023b), which was

Model	rSIGHANs			CSCD-NS			MCSCSet			ECSpell			LEMON		
	S-F↑	C-F↑	FPR↓	S-F↑	C-F↑	FPR↓	S-F↑	C-F↑	FPR↓	S-F↑	C-F↑	FPR↓	S-F↑	C-F↑	FPR↓
Previous SOTAs															
BERT†	70.8	**81.9**	12.1	**77.4**	79.3	10.9	**87.6**	**94.0**	1.8	91.0	94.6	**3.8**	48.1	49.3	13.1
ReLM†	**71.8**	81.8	**11.5**	77.3	**79.7**	**10.5**	87.0	93.7	**1.6**	**92.3**	**94.9**	7.4	**50.6**	**51.6**	**11.7**
LLMs (Zhou et al., 2024)															
Baichuan2	**59.1**	**70.1**	**15.4**	**62.7**	**65.7**	**16.8**	**67.2**	**78.0**	**2.0**	**85.4**	**90.2**	**5.1**	**53.2**	**56.7**	**9.9**
Qwen2.5	55.6	68.6	17.9	58.6	62.6	23.5	63.3	74.1	3.2	81.3	88.1	6.6	48.6	53.5	12.8
IL2.5	52.5	66.2	20.1	55.3	59.1	26.5	51.9	62.8	5.3	80.3	87.0	6.9	45.3	50.0	15.6
Ours															
BERT + BC2	72.5	83.4	7.4	78.1	79.1	10.1	91.2	96.0	1.4	94.4	96.3	**1.7**	56.4	58.2	6.9
BERT + QW2.5	73.6	**83.9**	**7.0**	76.8	78.6	11.3	91.6	95.9	1.6	94.0	95.5	2.7	56.1	58.7	10.1
BERT + IL2.5	72.8	83.8	7.9	77.3	78.3	11.5	90.7	95.7	1.7	94.3	94.9	2.2	53.8	55.1	6.7
ReLM + BC2	73.9	83.0	9.2	**79.9**	**81.8**	**8.4**	91.7	96.2	**1.1**	**97.1**	**98.3**	2.1	**61.0**	**61.6**	**5.6**
ReLM + QW2.5	73.9	82.6	9.1	78.9	79.8	10.0	**92.1**	**96.3**	1.3	96.6	97.8	3.0	60.5	61.4	7.9
ReLM + IL2.5	**74.4**	83.6	9.5	79.3	81.0	10.1	90.4	95.4	1.5	96.7	98.0	2.6	58.9	60.2	7.9

Table 2: Sentence- and character-level results on the mainstream CSC test sets. F_1 and FPR scores are reported (%). BC2 stands for Baichuan2, QW2.5 for Qwen2.5, and IL2.5 for InternLM2.5. All LLMs use the 7B model size version. Note that the performance metrics for rSIGHANs, ECSpell, and LEMON are presented as macro averages. "†" indicates that the models are pre-trained on 34 million synthetic data and fine-tuned with MFT strategy (LEMON cannot be further fine-tuned due to the lack of in-domain training data).

manually annotated for errors and noise present in the original SIGHANs. We refer to this version as **rSIGHANs**. In the training stage, we use Wang271K (Wang et al., 2018) + SIGHANs as our training set.

Chinese Native-speaker Texts. Due to the lack of domain diversity in the SIGHAN datasets, we further conduct experiments on more diverse datasets, including **LEMON** (Wu et al., 2023), **ECSpell** (Lv et al., 2023), **CSCD-NS** (Hu et al., 2024), and **MCSCSet** (Jiang et al., 2022), all of which were written by native speakers. Notably, LEMON includes test sets from seven different domains but does not provide in-domain training sets, making it an ideal benchmark for evaluating a model's cross-domain generalization capability. Detailed information about all datasets can be found in Appendix A.

4.2 Baseline Models

We select several BERT-based models and LLMs as our baselines.

Small BERT-based Models. For benchmarks across various domains, we select BERT (Devlin et al., 2019) and ReLM (Liu et al., 2024b) for experiments. Additionally, on the rSIGHAN15 test set, we select several mainstream SOTA models for experiments and report the results, such as ReaLiSe (Xu et al., 2021) and SCOPE (Li et al., 2022a). Detailed information about all baselines can be found in Appendix B.

Open-source LLMs. We use Baichuan2 (Yang et al., 2023a), Qwen2.5 (Bai et al., 2023), and InternLM2.5 (Cai et al., 2024) as our open-source LLMs for experiments. All LLMs use the base version. In the main experiments, we fix the model size to 7B. For comprehensive and robust experiments, in Section 6.1, we select LLMs with different sizes ranging from 0.5B to 20B.

Mixture Strategy. ARM (Liu et al., 2024a) attempts to make trade-offs between the correction results of small models and GPT-3.5-Turbo.[1] Due to the lack of open-source code and differences in experimental settings on LEMON, we only list their results in Table 8 and Table 11.

4.3 Evaluation Metrics

Following the mainstream evaluation metrics for CSC tasks, we report the Precision (P), Recall (R), and F_1 scores of the correction subtask at both the sentence- and character-level, denoted as S-P/R/F and C-P/R/F, respectively. To comprehensively evaluate the model's correction capability, we also include the false positive rate (FPR) as an additional metric.

[1] https://platform.openai.com

Input: 水饺和新鲜的空气都很重要。
Dumplings and fresh air are both important.
Reference: 水饺 ➔ 睡觉 (*shuǐjiǎo ➔ shuìjiào, sleep*)
ReLM: 水饺 ➔ 水觉 (*shuǐjiǎo ➔ shuǐjiào, water sleep*)
LLM: NONE
ReLM+LLM: 水饺 ➔ 睡觉 (*shuǐjiǎo ➔ shuìjiào, sleep*)

(a) Correct errors from ReLM and LLM

Input: 开车忘带驾驶者,被查到不要慌，...
If you forgot to bring the driver while driving, don't panic when being checked,...
Reference: 驾驶者 ➔ 驾驶证(*zhě ➔ zhèng, driver's license*)
ReLM: 驾驶者 ➔ 驾驶证(*zhě ➔ zhèng, driver's license*)
LLM: NONE
ReLM+LLM: 驾驶者 ➔ 驾驶证(*zhě ➔ zhèng, driver's license*)

(b) Correct errors from LLM

Input: 不仅赖账，还提无力要求！
Not only do they refuse to pay, but they also make powerless demands!
Reference: 无力 ➔ 无理(*wúlì ➔ wúlǐ, unreasonable*)
ReLM: 提 ➔ 是(*tí ➔ shì, are*)
LLM: 无力 ➔ 无理(*wúlì ➔ wúlǐ, unreasonable*)
ReLM+LLM: 无力 ➔ 无理(*wúlì ➔ wúlǐ, unreasonable*)

(c) Correct errors from ReLM

Figure 2: Cases from rSIGHAN15 and LEMON-*New* test sets. The LLM used is `Baichuan2(7B)`.

4.4 Implementation Details

For ReaLiSe, we employ the pre-trained checkpoint available from its official GitHub repository.[2] For SCOPE, we adopt their official implementation for fine-tuning.[3] Both BERT and ReLM utilize the framework from Liu et al. (2024b), with their pre-trained models (trained on a corpus of 34 million synthetic sentences) being fine-tuned in our experiments.[4] During training, we set the batch size to 128 and the learning rate to 3×10^{-5}, while incorporating the MFT strategy (Wu et al., 2023). All experiments are conducted on an NVIDIA A100-PCIE-40GB GPU.

5 Main Results

Table 2 presents experimental results across five benchmark datasets: rSIGHANs, CSCD-NS, MCSCSet, ECSpell, and LEMON. When integrated with LLMs, nearly all models exhibit substantial enhancements across sentence- and character-level correction metrics (Precision, Recall, and F_1). We also provide the results of the **original SIGHAN15** dataset in Appendix C.1.

To evaluate cross-domain generalization capability, we specifically test on ECSpell and LEMON datasets. Our approach demonstrates consistent performance gains on both test sets, with detailed subdomain results provided in Appendix C.2. Notably, ReLM integrated with `Baichuan2` achieves remarkable improvements: a 10.4% average increase in sentence-level F_1 across all seven LEMON domains, and 4.8% enhancement on ECSpell. During our experiments, we discovered that ECSpell has a serious issue of target sentence leakage, leading to overly high performance of fine-tuned small models. We will explain this issue and present the results on the cleaned ECSpell dataset in Appendix C.3.

Due to different experimental setups, we also list another mixture strategy, ARM (Liu et al., 2024a), in Appendix C.1 and C.2. In comparison, our approach achieves greater improvements across all datasets.

LLM	Size	rSIGHAN15		ECSpell-*Odw*		LEMON-*Nov*	
		S-F↑	C-F↑	S-F↑	C-F↑	S-F↑	C-F↑
BC2	7B	**79.6**	**85.4**	96.5	98.2	50.0	50.4
	13B	77.9	84.6	**96.7**	**98.3**	**50.8**	**50.9**
QW2.5	0.5B	78.4	84.2	95.7	97.3	45.4	46.3
	1.5B	78.2	83.3	95.7	97.6	48.1	49.0
	3B	**79.3**	83.7	96.7	98.1	48.8	49.9
	7B	77.9	84.2	95.9	97.7	49.4	50.6
	14B	78.6	**84.3**	**97.3**	**98.5**	**50.5**	**51.5**
IL2.5	1.8B	79.0	84.5	95.5	97.6	45.1	46.7
	7B	**79.5**	**85.3**	96.9	**98.3**	47.7	**48.8**
	20B	76.8	83.6	**97.1**	**98.3**	**48.1**	48.6

Table 3: Sentence- and character-level F_1 scores of different model families and sizes. All LLMs are combined with ReLM.

5.1 Case Study

Figure 2 presents comparative cases demonstrating our approach's effectiveness in error correction.

In Figure 2(a), while ReLM provides partial corrections and the LLM fails to detect the error, our mixture approach achieves complete error resolution. This is because compared to Zhou et al. (2024), we reduce the weight of the distortion model in Eq. (1), which allows the language model to play a greater role.

Figure 2(b) illustrates a typical under-correction scenario. Here, the LLM's under-correction of "驾驶者" (driver) is successfully revised to "驾驶证" (driver's license) through ReLM's complementary intervention.

The third case in Figure 2(c) reveals an over-correction scenario where ReLM introduces an er-

[2] https://github.com/DaDaMrX/ReaLiSe
[3] https://github.com/jiahaozhenbang/SCOPE
[4] https://github.com/gingasan/lemon

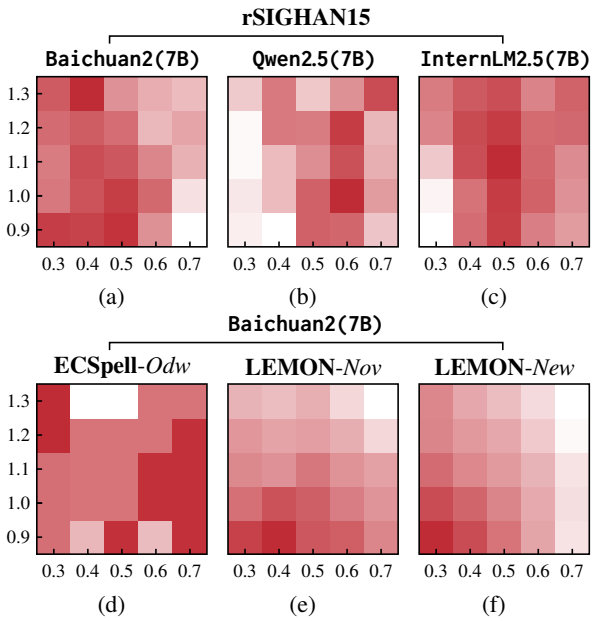

Figure 3: Model performance (sentence-level F_1) of ReLM + LLMs on rSIGHAN15, ECSpell, and LEMON with different α and β. The x-axis is α and y-axis is β. The red cells ■ denote superior performance of ReLM + LLM compared to ReLM. The blue cells ▨ represent inferior performance. The darker the color, the larger the performance gap.

Model	S-P↑	S-R↑	S-F↑	C-P↑	C-R↑	C-F↑	FPR↓
rSIGHAN15							
ReLM	76.8	73.8	75.2	87.9	78.0	82.6	8.3
BC2	67.1	55.8	61.0	78.7	61.0	68.7	8.3
Ours	**83.5**	**76.1**	**79.6**	**93.8**	78.3	**85.4**	**4.1**
- DM	-4.3	-0.2	-2.1	-4.2	**+1.3**	-1.1	+3.4
- FR	-3.6	-0.9	-2.1	-3.1	+0.3	-1.2	+2.3
- both	-4.7	-0.5	-2.5	-4.6	+1.1	-1.4	+3.8
ECSpell-*Odw*							
ReLM	89.4	91.5	90.4	93.1	95.7	94.4	5.8
BC2	92.0	89.1	90.5	95.0	92.6	93.8	1.6
Ours	**97.2**	**95.7**	**96.5**	**98.7**	**97.7**	**98.2**	**0.8**
- DM	-5.2	-1.5	-3.4	-3.2	-0.8	-2.0	+3.3
- FR	**+0.4**	**+0.4**	**+0.4**	-0.2	**+0.3**	**+0.0**	**+0.0**
- both	-4.4	-0.8	-2.7	-3.0	-0.5	-1.7	+3.3
Lemon-*Nov*							
ReLM	46.3	32.2	38.0	48.9	31.0	37.9	17.6
BC2	49.8	37.0	42.4	53.5	**42.0**	47.1	14.9
Ours	**64.0**	**41.0**	**50.0**	**67.2**	40.3	**50.4**	**9.8**
- DM	-15.8	-4.2	-8.2	-16.1	-4.3	-8.1	+7.8
- FR	-6.2	**+3.1**	**+0.0**	-6.2	**+4.5**	**+1.3**	+4.3
- both	-16.4	-3.0	-7.8	-16.5	-2.6	-7.1	+8.6

Table 4: Ablation results of distortion model (DM) and faithfulness reward (FR) on ReLM + Baichuan2(7B). "ReLM" and "BC2" represent using ReLM and Baichuan2(7B) model alone respectively. "-both" represents that we remove the intervention of both DM and FR on the results.

roneous modification, which is effectively identified and corrected by the LLM through beam search decoding.

In summary, LLMs demonstrate superior fluency preservation, while small models are better at making accurate corrections. The synergistic integration of both capabilities through our framework yields optimal correction results.

6 Discussion

Following Zhou et al. (2024), we evaluate our approach on three benchmark datasets: rSIGHAN15, ECSpell-*Odw*, and LEMON-*Nov*.

6.1 Impact of Different LLM Families and Model Sizes

To systematically analyze how LLM families and model sizes affect correction performance, we conduct comprehensive experiments with multiple open-source LLMs scaled from 0.5B to 20B parameters, including Baichuan2 (7B-13B), Qwen2.5 (0.5B-14B), and InternLM2.5 (1.8B-20B). All models are integrated with ReLM and evaluated across all three datasets. Table 3 reveals two key findings:

- Within the same LLM family, larger models do not necessarily lead to better performance. For example, on rSIGHAN15, the correction performance of all LLMs does not show an increasing trend with the growth of model size.
- Different LLM families exhibit varying strengths across different domains. For example, InternLM2.5 outperforms Qwen2.5 on the rSIGHAN15 dataset, whereas the opposite is true for LEMON-*Nov*. Overall, the combination of ReLM and Baichuan2 shows more stable correction performance.

6.2 Impact of Hyperparameters

We conduct an analysis of two critical hyperparameters in Eq. (1): the distortion model weight α and the small BERT-based model weight β. Figure 3 demonstrates their effects on three 7B-scale LLM families evaluated on rSIGHAN15. For α, most LLMs achieve optimal performance around 0.5. Compared with Baichuan2, both Qwen2.5 and InternLM2.5 require higher β values for peak performance. This difference likely stems from their weaker performance on rSIGHAN15 compared to Baichuan2. This hypothesis is further supported by experiments on different test sets. On

Model	rSIGHAN15						
	S-P↑	S-R↑	**S-F↑**	C-P↑	C-R↑	**C-F↑**	**FPR↓**
LLMs (Zhou et al., 2024)							
BC2	67.1	55.8	61.0	78.7	61.0	68.7	8.3
Previous SOTAs & Ours							
ReaLiSe	75.7	70.2	72.9	83.4	73.9	78.4	8.1
+ BC2	80.5	72.0	76.0	92.9	74.6	82.7	5.2
BERT	75.3	74.9	75.1	86.2	**79.6**	82.8	10.8
+ BC2	82.6	74.4	78.3	93.3	78.4	**85.2**	3.5
ReLM	76.8	73.8	75.2	87.9	78.0	82.6	8.3
+ BC2	83.1	**75.6**	79.2	**93.8**	78.1	**85.2**	4.1
SCOPE	78.7	73.5	76.0	83.9	76.7	80.1	7.0
+ BC2	**84.9**	75.4	**79.9**	**93.8**	77.2	84.7	**3.3**

Table 5: Ablation results of SOTA BERT-based models combined with Baichuan2(7B).

ECSpell-*Odw*, where the small and large models demonstrate closer correction capabilities, we observe greater flexibility in β selection. Conversely, for LEMON, the superior performance of LLMs reduces the significance of small models.

It is important to note that in Table 2, the weights are set at fixed values ($\alpha = 0.5$, $\beta = 0.9$). In practice, tuning these weights can further enhance model performance. Importantly, our approach consistently surpasses the ReLM baseline across all hyperparameter settings, highlighting its robustness.

6.3 Impact of Distortion Model and Faithfulness Reward

Table 4 investigates the individual contributions of the distortion model (DM) and faithfulness reward (FR) through an ablation study. The results show that while the removal of DM leads to performance degradation, our approach still outperforms the ReLM baseline. This demonstrates that even when stripped of LLM-specific correction strategies, an LLM retaining core language modeling capability can still provide effective corrective guidance to the small model.

Moreover, removing the DM and FR has a larger impact on Precision than on Recall. This is because the DM's similarity information and FR's constraints help the model avoid unnecessary modifications.

6.4 Impact of Small BERT-based Models

To evaluate the impact of small models on experimental outcomes, we conducted experiments with several classic and high-performing BERT-based models in combination with Baichuan2(7B). As demonstrated in Table 5 and Table 8, the integration with LLMs leads to substantial performance improvements across all small models. Notably, this enhancement is reflected in both Precision and Recall metrics, while simultaneously reducing the FPR. Furthermore, our analysis reveals a strong positive correlation between the performance of the selected small models and the overall correction performance of our mixture approach.

Model	Speed (ms/sent)	Slowdown
BC2	1,276.0	1.00×
BERT	13.7	–
+ BC2	1,470.7	1.15×
ReLM	14.1	–
+ BC2	1,498.1	1.17×

Table 6: The decoding time per sentence on rSIGHAN15. The LLM used is Baichuan2(7B).

6.5 Impact of Beam Size

During beam search inference, beam size determines the number of candidate sequences maintained during decoding. While larger beam sizes enhance output diversity and better approximate the global optimal $\texttt{Score}(\boldsymbol{x}, \boldsymbol{y})$, this comes at the expense of decoding speed. Beam search can alleviate the limitation where generative models can only access previously generated tokens. In our approach, incorporating BERT-based models supplies additional contextual information to the LLMs, thereby reducing the dependence on beam search compared to using LLMs alone. As shown in Figure 4, as beam size increases, the performance of all models generally shows an upward trend.

6.6 Inference Speed

We compare the inference speeds of our approach and baseline models in Table 6, with all experiments conducted on a single NVIDIA A100-PCIE-40GB GPU. The evaluation uses a batch size of 1 and beam size of 12 for LLMs. The results reveal that our approach exhibits marginally slower inference speed compared to using the LLM alone, which can be attributed to the computational overhead introduced by integrating the small model's probability distribution at each beam search step.

7 Related Work

7.1 BERT-based Approaches.

Since the advent of BERT, BERT-based CSC models have shown strong correction capabilities. To

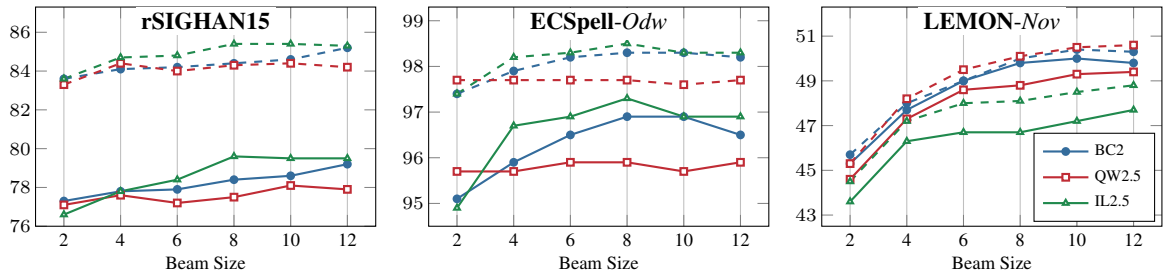

Figure 4: F_1 scores of different LLMs (model size is 7B) with varying beam sizes. All models are combined with ReLM. Solid lines show sentence-level results, while dashed lines show character-level results.

better capture the relationships between misspelled and correct characters, phonetic and glyph similarities have been incorporated into the encoder. Techniques include using confusion sets for data augmentation (Liu et al., 2021) and employing neural networks to encode phonetic and glyph features (Xu et al., 2021; Huang et al., 2021). Additionally, new training objectives like pinyin prediction tasks have been developed to boost model performance (Liu et al., 2021; Li et al., 2022a,b; Liang et al., 2023).

Other strategies involve modifying the model pipeline, for example, by adding a detection layer to enhance detection capability (Zhang et al., 2020; Huang et al., 2023); and employing decoding intervention strategies to improve result selection (Wang et al., 2019; Bao et al., 2020; Lv et al., 2023; Qiao et al., 2024).

7.2 LLM Approaches

In the era of LLMs, researchers have been actively exploring their potential in CSC tasks. A key consideration in these studies is whether LLMs require fine-tuning. Li et al. (2023) pioneered the exploration of prompt-based methods for correction tasks. They also experimented with SFT techniques, though the results were not satisfactory. Dong et al. (2024) enhanced the correction capabilities of LLMs by incorporating pinyin and radical information of Chinese characters into prompts. However, these prompt-based strategies struggle to maintain consistency between the prediction and input lengths.

Further advancements were made by Li et al. (2024), who replace mixed tokenization with character-level tokenization. They fine-tune the LLMs to perform corrections in a character-by-character way, thereby resolving over 99% of the length inconsistency issues.

In contrast to these methods, Zhou et al. (2024) introduced a novel approach that is both prompt-free and training-free, treating LLMs as pure language models. This innovative approach opens a new research direction, aiming to fully leverage the intrinsic language modeling potential of LLMs.

7.3 Mixture Approaches

In the field of Grammatical Error Correction (GEC), model ensemble is commonly used. Multiple small models vote on edits to produce a final prediction (Zhang et al., 2022). Zhou et al. (2023) employed a language model, GPT-2, and a Grammatical Error Detection (GED) model, BART, as critics to dynamically guide the token selection in a Seq2Seq GEC model. They did not use a LLM due to the absence of a token-alignment method.

In the CSC domain, Liu et al. (2024a) introduced an ARM method. Similar to the model ensemble technique, they replaced the voting process with an unfine-tuned small model that selects between the outputs of the LLM and a fine-tuned small model. Unlike their method, which focuses on post-processing predictions, our approach integrates both models during the beam search decoding process.

8 Conclusions

In this paper, we propose a novel dynamic mixture approach that effectively combines small models and LLMs during the beam search decoding phase. By leveraging the strong error correction capabilities of fine-tuned BERT-based models and the language modeling strengths of LLMs, we achieve significant performance improvements. The advantage of not requiring fine-tuning of LLMs enhances the domain adaptability of our method. Experiments on mainstream public datasets demonstrate that our mixture approach achieves SOTA performance across multiple datasets.

Limitations

Different Tasks. Our method is primarily designed for the CSC task. However, this mixture approach can also be adapted to other tasks, such as GEC, which is another type of text error correction, and even other areas that can be modeled simultaneously as classification and generation tasks.

Different Languages. Our method is currently limited to Chinese. However, for other languages, such as English, Korean, and Japanese, by modifying the token alignment between small models and LLMs, the same mixture approach can also achieve collaborative error correction.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen Technical Report. *ArXiv preprint*, abs/2309.16609.

Zuyi Bao, Chen Li, and Rui Wang. 2020. Chunk-based Chinese Spelling Check with Global Optimization. In *Findings of EMNLP*, pages 2031–2040, Online.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, Aijia Guo, Qipeng Guo, Conghui He, Yingfan Hu, Ting Huang, Tao Jiang, Penglong Jiao, Zhenjiang Jin, Zhikai Lei, Jiaxing Li, Jingwen Li, Linyang Li, Shuaibin Li, Wei Li, Yining Li, Hongwei Liu, Jiangning Liu, Jiawei Hong, Kaiwen Liu, Kuikun Liu, Xiaoran Liu, Chengqi Lv, Haijun Lv, Kai Lv, Li Ma, Runyuan Ma, Zerun Ma, Wenchang Ning, Linke Ouyang, Jiantao Qiu, Yuan Qu, Fukai Shang, Yunfan Shao, Demin Song, Zifan Song, Zhihao Sui, Peng Sun, Yu Sun, Huanze Tang, Bin Wang, Guoteng Wang, Jiaqi Wang, Jiayu Wang, Rui Wang, Yudong Wang, Ziyi Wang, Xingjian Wei, Qizhen Weng, Fan Wu, Yingtong Xiong, Chao Xu, Ruiliang Xu, Hang Yan, Yirong Yan, Xiaogui Yang, Haochen Ye, Huaiyuan Ying, Jia Yu, Jing Yu, Yuhang Zang, Chuyu Zhang, Li Zhang, Pan Zhang, Peng Zhang, Ruijie Zhang, Shuo Zhang, Songyang Zhang, Wenjian Zhang, Wenwei Zhang, Xingcheng Zhang, Xinyue Zhang, Hui Zhao, Qian Zhao, Xiaomeng Zhao, Fengzhe Zhou, Zaida Zhou, Jingming Zhuo, Yicheng Zou, Xipeng Qiu, Yu Qiao, and Dahua Lin. 2024. InternLM2 Technical Report. *ArXiv preprint*, abs/2403.17297.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. SpellGCN: Incorporating Phonological and Visual Similarities into Language Models for Chinese Spelling Check. In *Proceedings of ACL*, pages 871–881, Online.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, Minneapolis, Minnesota.

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.

Yong Hu, Fandong Meng, and Jie Zhou. 2024. CSCD-NS: a Chinese Spelling Check Dataset for Native Speakers. In *Proceedings of ACL*, pages 146–159, Bangkok, Thailand.

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. In *Findings of EMNLP*, pages 11514–11525, Singapore.

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, Online. Association for Computational Linguistics.

Wangjie Jiang, Zhihao Ye, Zijing Ou, Ruihui Zhao, Jianguang Zheng, Yi Liu, Bang Liu, Siheng Li, Yujiu Yang, and Yefeng Zheng. 2022. MCSCSet: A Specialist-annotated Dataset for Medical-domain Chinese Spelling Correction. In *Proceedings of CIKM*, pages 4084–4088.

Jiahao Li, Quan Wang, Zhendong Mao, Junbo Guo, Yanyan Yang, and Yongdong Zhang. 2022a. Improving Chinese Spelling Check by Character Pronunciation Prediction: The Effects of Adaptivity and Granularity. In *Proceedings of EMNLP*, pages 4275–4286, Abu Dhabi, United Arab Emirates.

Jiahao Li, Quan Wang, Zhendong Mao, Junbo Guo, Yanyan Yang, and Yongdong Zhang. 2022b. Improving Chinese Spelling Check by Character Pronunciation Prediction: The Effects of Adaptivity and Granularity. In *Proceedings of the 2022 Conference on*

Empirical Methods in Natural Language Processing, pages 4275–4286, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kunting Li, Yong Hu, Liang He, Fandong Meng, and Jie Zhou. 2024. C-LLM: Learn to Check Chinese Spelling Errors Character by Character.

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.

Zihong Liang, Xiaojun Quan, and Qifan Wang. 2023. Disentangled Phonetic Representation for Chinese Spelling Correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13509–13521, Toronto, Canada. Association for Computational Linguistics.

Changchun Liu, Kai Zhang, Junzhe Jiang, Zirui Liu, Hanqing Tao, Min Gao, and Enhong Chen. 2024a. 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.

Linfeng Liu, Hongqiu Wu, and Hai Zhao. 2024b. Chinese Spelling Correction as Rephrasing Language Model. In *Proceedings of AAAI*, pages 18662–18670, Vancouver, Canada.

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 ACL-IJCNLP*, pages 2991–3000, Online.

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. *TALLIP*, pages 1–18.

Ziheng Qiao, Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2024. DISC: Plug-and-Play Decoding Intervention with Similarity of Characters for Chinese Spelling Check.

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 SIGHAN*, pages 32–37, Beijing, China.

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 EMNLP*, pages 2517–2527, Brussels, Belgium.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided Pointer Networks for Chinese Spelling Check. In *Proceedings of ACL*, pages 5780–5785, Florence, Italy.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023. Rethinking Masked Language Modeling for Chinese Spelling Correction. In *Proceedings of ACL*, pages 10743–10756, Toronto, Canada.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. In *Proceedings of SIGHAN*, pages 35–42, Nagoya, Japan.

Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, Heyan Huang, and Xian-Ling Mao. 2021. Read, Listen, and See: Leveraging Multimodal Information Helps Chinese Spell Checking. In *Findings of ACL-IJCNLP*, pages 716–728, Online.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023a. Baichuan 2: Open Large-scale Language Models.

Liner Yang, Xin Liu, Tianxin Liao, Zhenghao Liu, Mengyan Wang, Xuezhi Fang, and Erhong Yang. 2023b. Is Chinese Spelling Check ready? Understanding the correction behavior in real-world scenarios. *AI Open*, pages 183–192.

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 CIPS-SIGHAN*, pages 126–132, Wuhan, China.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling Error Correction with Soft-Masked BERT. In *Proceedings of ACL*, pages 882–890, Online.

Yue Zhang, Zhenghua Li, Zuyi Bao, Jiacheng Li, Bo Zhang, Chen Li, Fei Huang, and Min Zhang. 2022. MuCGEC: a Multi-Reference Multi-Source Evaluation Dataset for Chinese Grammatical Error Correction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3118–3130, Seattle, United States. Association for Computational Linguistics.

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.

Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023. Improving Seq2Seq Grammatical Error Correction via Decoding Interventions. In *Findings of EMNLP*, pages 7393–7405, Singapore.

A Details of Datasets

SIGHANs. The SIGHAN collection comprises three Chinese learner corpora (SIGHAN 13/14/15) (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015), which serve as standard benchmarks for CSC research. To address the noise and annotation errors in the original datasets, we adopt the revised version (Yang et al., 2023b) (**rSIGHANs**), which was manually re-annotated. Following prior setups, we combine SIGHAN datasets with Wang271K (Wang et al., 2018) (containing 271K synthetic training instances) to form the composite training set.

LEMON. Containing over 22K sentences across seven distinct domains (game, encyclopedia, contract, medical care, car, novel, and news) (Wu et al., 2023), LEMON provides a cross-domain evaluation framework for CSC systems. Due to the absence of in-domain training sets, LEMON is employed to assess the cross-domain generalization capability of CSC models.

ECSpell. ECSpell consists of three small-scale datasets from the domains of law, medical treatment, and official document writing (Lv et al., 2023), offering domain-specific evaluation scenarios for CSC models.

CSCD-NS. CSCD-NS (Hu et al., 2024) contains 40K annotated samples sourced from real posts on Sina Weibo, effectively reflecting the real-world error correction performance of CSC models.

MCSCSet. Developed for medical domain applications (Jiang et al., 2022), this large-scale corpus contains approximately 200K professionally annotated sentences. The significant domain gap between medical texts and open-domain datasets makes MCSCSet particularly valuable for evaluating domain adaptation capabilities in CSC systems.

Detailed statistics are presented in Table 9.

B Details of Baselines

In our experiments, we selected four BERT-based models as baselines: BERT, ReaLiSe, SCOPE, and ReLM.

Source
From Test set:
行政机关实施行政管理都应当公开，这是程序正档原则的要求
From Training set:
行政机关实施行政管理都应当公开，这是程序正当原则的要求
行政机关实施行政管理都应当公开，这是程序争当原则的要求
行政机关实施行政管理都应当公开，这是程序正当原则的要求
行政机关实施行政管理都应当公开，这是程序止当原则的要求
行政机关实私行政管理都应当公开，这是程序正当原则的要求

Table 7: Examples of ECSpell-*Law*.

Model	SIGHAN15						
	S-P↑	S-R↑	**S-F↑**	C-P↑	C-R↑	**C-F↑**	**FPR↓**
LLMs (Zhou et al., 2024)							
BC2	61.2	58.2	59.7	69.0	65.4	67.2	14.3
Previous SOTAs & Ours							
ReaLiSe	75.9	79.9	77.8	83.4	83.8	83.6	12.0
+ BC2	80.7	82.4	81.5	87.2	85.3	86.3	9.5
BERT	75.2	83.4	79.1	81.9	89.2	85.4	14.1
+ BC2	81.7	84.1	82.9	86.8	**89.6**	**88.2**	7.9
ReLM	76.8	83.9	80.2	83.2	88.6	85.8	12.7
+ BC2	82.8	**85.9**	84.3	87.6	88.7	88.1	**7.8**
SCOPE	78.6	83.5	81.0	83.3	86.6	84.9	11.3
+ BC2	**83.8**	85.0	**84.4**	**88.1**	87.6	87.9	7.9
+ ARM	79.5	83.1	81.3	–	–	–	–

Table 8: Sentence- and character-level results on the SIGHAN15 test set. ARM represents the latest approach that integrates the results of small models and GPT-3.5-Turbo (Liu et al., 2024a).

- **BERT**: The vanilla BERT architecture implemented using the bert-chinese-base[5] pre-trained weights.
- **ReaLiSe**: This model employs GCN and CNN respectively to encode the pinyin and visual features (font images) of each Chinese character. These additional representations help the model capture the intrinsic phonetic and glyph relationships between characters. It uses chinese-roberta-wwm-ext[6] as its backbone.
- **SCOPE**: As one of the current SOTA CSC models, SCOPE incorporates an extra training task for character pronunciation prediction (CPP). The model is initialized from ChineseBERT-base[7].
- **ReLM**: ReLM reframes the CSC task as a rephrasing problem rather than a tagging task. In practice, it still adopts non-autoregressive inference and remains essentially a BERT-based model, pre-trained and fine-tuned based on bert-chinese-base.

[5] https://huggingface.co/bert-base-chinese
[6] https://huggingface.co/hfl/chinese-roberta-wwm-ext
[7] https://huggingface.co/ShannonAI/ChineseBERT-base

Training Sets	SIGHANs	Wang271K	CSCD-NS	MCSCSet	ECSpell		
Subsets	–	–	–	–	*Law*	*Med*	*Odw*
#Sent.	6,479	271,329	30,000	157,194	1,960	3,000	1,720
Avg. Length	42.1	42.6	57.4	10.9	30.7	50.2	41.2
Avg. Error/Sent.	1.0	1.4	0.5	0.9	0.9	0.8	0.9

Test Sets	rSIGHANs			CSCD-NS	MCSCSet	ECSpell			LEMON
Subsets	*15*	*14*	*13*	–	–	*Law*	*Med*	*Odw*	–
#Sent.	1,100	1,062	1,000	5,000	19,650	500	500	500	22,252
Avg. Length	30.6	50.0	74.3	57.6	10.9	29.7	49.6	40.5	35.4
Avg. Error/Sent.	0.8	0.9	1.5	0.5	0.9	0.8	0.7	0.8	0.5

Table 9: Detailed statistics of all datasets used in our experiments.

Datasets	ECSpell (cleaned)		
Subsets	*Law*	*Med*	*Odw*
Previous SOTA			
ReLM	71.0	70.3	77.7
LLMs (Zhou et al., 2024)			
Baichuan2	83.7	**82.0**	**90.5**
Qwen2.5	83.7	73.1	89.1
IL2.5	**85.8**	66.2	89.0
Ours			
ReLM + BC2	86.4	**87.9**	93.1
ReLM + QW2.5	**89.0**	86.7	93.5
ReLM + IL2.5	86.4	87.1	**93.7**

Table 10: Experiments on cleaned ECSpell.

C More Experiments

In this section, we present detailed results across different domains and the results of some other CSC models, such as ARM (Liu et al., 2024a) and C-LLM (Li et al., 2024).

C.1 Experiments on Original SIGHAN15

Following the experimental settings of previous studies, we use the original SIGHAN15 test set for comparison.[8] In line with rSIGHAN15, we also adopt ReaLiSe and SCOPE, as shown in Table 8. Evidently, our method significantly improves the performance of all baseline models, surpassing the current SOTA performance. We also show the latest approach that ensembles the results of small models and GPT-3.5-Turbo (SCOPE + ARM) (Liu et al., 2024a), compared with which our method achieves a more effective performance improvement through deep integration of the small and large models during the inference stage.

[8]Since the dataset includes traditional characters, we use the preprocessed version by Xu et al. (2021).

C.2 Detailed Results

Due to space limitations, we present the detailed results of the subsets of rSIGHANs, ECSpell, and LEMON in Table 11, Table 12, and Table 13. We also provide the results of ARM and C-LLM in the corresponding tables. The results show that our method outperforms another mixture approach and the current SOTA SFT-based LLM strategy.

C.3 Experiments on Cleaned ECSpell

Our analysis reveals that the strong performance of fine-tuned small models on ECSpell originates from substantial data leakage caused by homogeneous sentence pairs. Although ECSpell avoids including identical source-target sentence pairs, it introduces different synthesized errors for the same correct sentences that appear in both the training and test sets. These overlapping sentences account for 52.7%, 19.3%, and 28.2% of the ECSpell-*Law*/*Med*/*Odw* training sets, respectively. For example, the first sentence in the ECSpell-*Law* test set appears five times in the training set as the same sentence, as illustrated in Table 7.

Undoubtedly, the fine-tuning process of small models leads to results that are significantly higher than those of the LLMs without fine-tuning. Therefore, we removed the leaked sentences from the training sets and reorganized the experiments in Table 10. It is worth noting that all hyperparameter settings remain identical to those in the main experiments. Clearly, our mixture approach continues to achieve significant and stable improvements. In fact, due to the significantly reduced size of the training set, the performance of fine-tuned small models lags behind that of LLMs, and appropriately lowering the weight of the small model β can yield even more significant performance improvements.

Datasets	rSIGHANs			ECSpell			LEMON						
Subsets	*15*	*14*	*13*	*Law*	*Med*	*Odw*	*Car*	*Cot*	*Enc*	*Gam*	*Med*	*New*	*Nov*
Previous SOTAs													
BERT	75.1	65.6	71.6	95.9	88.0	89.1	52.0	63.8	45.3	32.9	50.8	56.0	35.8
ReLM	75.2	65.9	74.4	96.2	90.2	90.4	53.6	67.7	47.7	34.6	53.9	58.8	38.0
MDCSpell⋆	–	–	–	–	–	–	34.1	49.2	32.8	14.8	29.5	34.4	14.3
LLMs (Zhou et al., 2024)													
Baichuan2	**61.0**	**53.1**	**63.1**	83.7	**82.0**	**90.5**	**54.2**	**63.3**	**51.4**	**36.9**	**60.6**	**63.9**	**42.4**
Qwen2.5	58.3	50.7	57.9	83.7	73.1	89.1	48.2	59.9	46.5	34.7	54.2	59.5	37.1
IL2.5	55.6	45.9	56.0	**85.8**	66.2	89.0	44.4	54.7	45.2	33.0	50.2	57.2	32.1
Mixture													
MDCSpell + ARM⋆	–	–	–	–	–	–	37.1	52.7	35.2	15.3	33.0	36.4	15.6
BERT + BC2	78.3	67.1	72.1	95.7	93.8	93.6	56.0	68.0	49.4	41.2	63.9	68.7	47.6
BERT + QW2.5	77.7	**69.5**	73.7	96.5	94.1	91.5	57.7	67.0	52.4	37.9	65.2	67.5	44.7
BERT + IL2.5	78.4	67.7	72.3	96.3	94.7	92.0	55.8	67.4	51.6	38.3	55.6	64.8	43.1
ReLM + BC2	**79.6**	67.6	74.6	**98.3**	**96.6**	96.5	62.7	**74.5**	56.5	**45.1**	66.3	**72.0**	**50.0**
ReLM + QW2.5	77.9	67.7	**76.0**	98.1	95.9	95.9	**63.6**	71.2	**57.4**	44.0	**66.7**	71.4	49.4
ReLM + IL2.5	79.5	68.3	75.3	97.3	95.9	**96.9**	61.3	72.0	56.1	39.5	64.7	71.0	47.7

Table 11: Sentence-level F_1 (S-F) scores across different domains. "⋆" indicates that the results are extracted from Liu et al. (2024a) and the small model is trained on Wang271K + SIGHANs.

Datasets	rSIGHANs			ECSpell			LEMON						
Subsets	*15*	*14*	*13*	*Law*	*Med*	*Odw*	*Car*	*Cot*	*Enc*	*Gam*	*Med*	*New*	*Nov*
Previous SOTAs													
BERT	82.8	77.3	85.7	97.4	93.3	93.2	52.7	65.3	46.1	35.6	52.0	57.4	36.3
ReLM	82.6	76.7	86.1	93.6	94.4	94.4	54.3	67.4	48.1	37.9	54.9	60.5	37.9
LLMs (Zhou et al., 2024; Li et al., 2024)													
Baichuan2	**68.7**	**64.9**	**76.8**	**88.0**	**93.8**	**93.8**	**58.8**	**65.3**	56.3	40.9	61.6	**66.8**	**47.1**
Qwen2.5	66.1	64.7	75.0	81.6	93.0	93.0	54.1	62.5	52.4	**41.8**	56.6	63.2	43.8
IL2.5	64.6	60.6	73.4	78.3	92.4	92.4	49.9	58.1	51.6	37.5	53.5	61.0	38.6
C-LLM	–	–	–	–	–	–	57.5	60.4	**56.5**	38.0	**65.3**	64.5	43.9
Ours													
BERT + BC2	85.2	78.3	86.6	96.2	95.6	95.6	58.3	68.4	51.8	43.9	64.4	70.6	50.1
BERT + QW2.5	84.6	**80.1**	87.0	95.4	93.4	93.4	60.5	67.3	55.5	43.4	65.7	70.0	48.7
BERT + IL2.5	84.8	79.3	**87.4**	92.7	94.4	94.4	57.4	68.0	52.3	41.5	56.0	66.2	44.0
ReLM + BC2	**85.4**	78.0	85.7	**97.8**	98.2	98.2	62.5	**74.2**	56.7	47.8	66.5	**73.1**	50.4
ReLM + QW2.5	84.2	77.1	86.6	96.9	97.7	97.7	**64.0**	69.8	**57.8**	**48.2**	**66.8**	72.7	**50.6**
ReLM + IL2.5	85.3	78.7	86.8	97.5	**98.3**	**98.3**	62.2	71.8	57.4	44.1	65.1	72.1	48.8

Table 12: Character-level F_1 (C-F) scores across different domains.

Datasets	rSIGHANs			ECSpell			LEMON						
Subsets	*15*	*14*	*13*	*Law*	*Med*	*Odw*	*Car*	*Cot*	*Enc*	*Gam*	*Med*	*New*	*Nov*
Previous SOTAs													
BERT	10.8	13.0	12.5	3.7	5.1	2.5	12.2	7.8	13.8	22.4	8.5	9.4	17.3
ReLM	8.3	13.6	12.6	6.5	9.8	5.8	12.0	4.9	12.7	20.6	5.8	8.4	17.6
LLMs (Zhou et al., 2024)													
Baichuan2	**8.3**	**15.7**	**22.3**	4.9	**8.7**	**1.6**	**6.9**	**7.8**	**10.2**	**19.8**	**3.4**	**6.0**	**14.9**
Qwen2.5	10.2	19.4	24.2	**4.1**	12.9	2.9	10.7	6.7	13.2	23.9	5.4	9.2	20.7
IL2.5	13.3	21.4	25.5	**4.1**	14.0	2.5	13.0	10.1	15.2	28.3	7.9	9.8	24.8
Ours													
BERT + BC2	**3.5**	10.0	**8.7**	**1.6**	**2.6**	**0.8**	6.0	2.3	7.5	14.5	2.4	4.2	11.4
BERT + QW2.5	3.7	**8.7**	**8.7**	**1.6**	4.4	2.1	8.4	2.5	10.9	22.5	3.1	6.7	16.9
BERT + IL2.5	5.6	9.3	**8.7**	2.0	3.3	1.3	5.7	2.2	7.2	14.1	2.5	3.5	11.9
ReLM + BC2	4.1	10.0	13.4	2.4	3.1	**0.8**	**4.5**	**1.3**	**6.6**	**11.6**	2.3	**3.1**	**9.8**
ReLM + QW2.5	5.4	11.0	10.8	3.3	4.5	1.2	6.4	2.9	9.2	17.3	**2.2**	5.1	12.2
ReLM + IL2.5	5.2	10.6	12.7	3.3	3.8	**0.8**	6.8	2.2	9.1	15.7	3.3	4.4	13.9

Table 13: False positive rate (FPR) scores across different domains.