Training a Better Chinese Spelling Correction Model via Prior-knowledge Guided Teacher

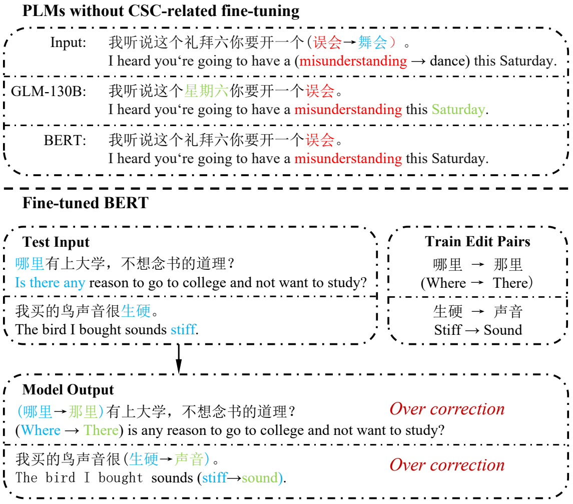
Chi Wei[1], Shaobin Huang[2]*, Rongsheng Li[3]*, Naiyu Yan, Rui Wang
Computer Science Department, Harbin Engineering University, Harbin, 150001
[1]weichi1207@163.com, {[2]Huangshaobin,[3]dasheng}@hrbeu.edu.cn

Abstract

Recent advancements in Chinese Spelling Correction (CSC) predominantly leverage pre-trained language models (PLMs). However, a notable challenge with fine-tuned PLM-based CSC models is their tendency to over-correct, leading to poor generalization for error patterns outside the standard distribution. To address this, we developed a teacher network guided by prior knowledge for distillation learning of CSC models. Unlike traditional teacher networks, which depend on task-related pre-training, our method infuses task-related prior information into the teacher network, offering guidance beyond mere labels to the student network. This strategy significantly enhances the CSC model's language modeling capabilities, crucial for minimizing over-correction. Importantly, our approach is model-independent and the teacher network does not require task-related pre-training, making it broadly applicable for enhancing various PLM-based CSC models with minimal additional computational resources. Extensive experiments on widely used benchmarks demonstrate that our method achieves new state-of-the-art results. Additionally, we explored the potential of generalizing our method to other non-autoregressive text-generation tasks.

1 Introduction

The primary objective of Chinese Spelling Correction is to detect and correct spelling errors in Chinese sentences, a task that holds substantial utility in various applications. CSC can play a pivotal role in enhancing tasks such as optical character recognition, text editing, and speech recognition (Afli et al., 2016; Gupta et al., 2021; Dong and Zhang, 2016). Moreover, it acts as an invaluable tool for individuals learning a new language, enabling quicker and more economical learning processes. The importance of CSC spans numerous fields within natural language processing (NLP), underscoring its wide-ranging utility.



PLMs without CSC-related fine-tuning

Input: 我听说这个礼拜六你要开一个(误会→舞会）。
I heard you‘re going to have a (misunderstanding → dance) this Saturday.

GLM-130B: 我听说这个星期六你要开一个误会。
I heard you‘re going to have a misunderstanding this Saturday.

BERT: 我听说这个礼拜六你要开一个误会。
I heard you‘re going to have a misunderstanding this Saturday.

Fine-tuned BERT

Test Input
哪里有上大学，不想念书的道理？
Is there any reason to go to college and not want to study?
我买的鸟声音很生硬。
The bird I bought sounds stiff.

Train Edit Pairs
哪里 → 那里
(Where → There)
生硬 → 声音
Stiff → Sound

Model Output
(哪里→那里)有上大学，不想念书的道理？ *Over correction*
(Where → There) is any reason to go to college and not want to study?
我买的鸟声音很(生硬→声音)。 *Over correction*
The bird I bought sounds (stiff→sound).

Figure 1: Examples of Chinese Spelling Correction. We mark the input errors/ ground truth/ over correction tokens in red/blue/ green. Here, the model is trained with edit pairs such as "生硬 → 声音" (hard → voice) and "哪里 → 那里" (where → there). During testing, the model overcorrected for "生硬 → 声音" (hard → voice) and "哪里 → 那里" (where → there) because the model tends to overfit to the training edit pairs.

In recent years, CSC models based on pre-trained language models (PLMs) have become increasingly popular (Zhu et al., 2022; Ji et al., 2021; Wu et al., 2023). However, the corpora used to pre-train these PLMs consist exclusively of correct tokens, and the tasks designed for pre-training do not account for spelling errors. As a result, even advanced large language models (LLMs) like GLM-130B (Du et al., 2022) face challenges in fully covering the CSC task, as illustrated in Figure 1. These limitations underscores the importance of fine-tuning CSC models, although this process encounters its own set of challenges, such as the tendency of fine-tuned BERT-based CSC models to memorize training edit pairs, leading to over-

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 13578–13589
August 11-16, 2024 ©2024 Association for Computational Linguistics

correction problem shown in Figure 1 during the prediction.

Driving from (Kernighan et al., 1990) and the Bayes rule, the spelling correction model $P(y_i|X)$ is formulated for an input sequence $X = \{x_1, x_2, ..., x_n\}$ and a corresponding output sequence $Y = \{y_1, y_2, ..., y_n\}$. A CSC model makes decisions through the collaboration of a *language model* and an *error model* (Wu et al., 2023).

$$P(y_i \mid X) \propto \underbrace{P(y_i \mid x_{-i})}_{language\ model} \cdot \underbrace{P(x_i \mid y_i, x_{-i})}_{error\ model} \quad (1)$$

Where x_{-i} denotes all input characters except x_i. Empirical research by (Wu et al., 2023) points out that the main reason for over-correction in PLM-based CSC models is they typically over-fit the *error model* and under-fit the *language model*.

Derived from this research finding, we argue that relying exclusively on ground truth complicates the task of avoiding over-correction, often leading to the development of less effective contextual spelling correction (CSC) models. Therefore, this paper employs a **P**rior-knowledge **G**uided **T**eacher (PGT) network as the auxiliary training target of the CSC model. The prior-knowledge Guided teacher network, also based on PLM, masks all spelling errors in input during training and outputs the distribution of y_i in a non-autoregressive manner. When x_i is the typo, the prior-teacher network can be formalized as language model $P(y_i|x_{-i})$. which enables the CSC model to circumvent over-fitting to the *error model* by integrating an additional language modeling goal.

Our approach does not require any structural or input modifications to the CSC model. It employs the output distribution of the prior-knowledge guided teacher network as soft tags, alongside the ground truth, to establish a new training objective for the CSC model. Contrary to methods that depend on input data augmentation (Zhao and Wang, 2020; Liu et al., 2021), PGT does not impose any error assumptions. Additionally, PGT does not result in any loss of input information for the CSC model, unlike the case with input masking CSC methods(Zhang et al., 2020; Li et al., 2021). Therefore, PGT can learn an unbiased CSC model from complete and real data. This exciting property allows PGT to set new state-of-the-art results across public CSC benchmarks.

Another significant attribute of PGT is its potential for model compression, as it can be viewed as a knowledge distillation (KD) method. However, PGT's effectiveness emanates from the incorporation of prior knowledge rather than task-related pretraining, distinguishing it in the context of Large Language Models (LLMs). Given the substantial computational and time costs consumed in task-related fine-tuning for LLMs, PGT stands out in the era of LLMs. A comprehensive analysis of PGT's model compression effectiveness will be presented in the experimental section.

In a word, our contributions are summarized in four-fold. (1) We demonstrate a simple prior-knowledge guided CSC training strategy significantly enhances PLM-based CSC methods, leading to new state-of-the-art results in CSC benchmarks. (2) We demonstrate that our method is model-independent and capable of improving the performance of various PLM-based CSC methods. (3) We perform empirical analyses that show PGT possesses significant model compression capabilities in CSC task, surpassing traditional KD schemes in both performance and efficiency. (4) A preliminary experiment has validated the potential of our approach to be generalized to other non-autoregressive generation tasks.

2 Related Work

2.1 Chinese Spelling Correction

In spelling correction, initial approaches used RNN and Bi-LSTM (Huang et al., 2015) networks (Wang et al., 2018), later enhanced by (Duan et al., 2019) with CRF layers for improved output. The transition to Lattice-LSTM by (Wang et al., 2021) marked a significant shift, as it integrated characters and token features in ways that went beyond the capabilities of Bi-LSTM.

The introduction of BERT (Devlin et al., 2019) led to new CSC models like deep denoising autoencoder for CSC based on BERT (Hong et al., 2019) and a combination model of BERT with GCN for CSC (Cheng et al., 2020). Addressing the impact of typos on context, the BERT-based detection-correction model emerged (Li et al., 2021; Zhu et al., 2022; Zhang et al., 2020). Furthermore, some works have enhanced BERT for spelling correction by integrating Chinese glyph, pronunciation, and character features (Huang et al., 2021; Liu et al., 2022; Xu et al., 2021).

However, the focus on model architecture and feature fusion has overshadowed the research on training strategies. Innovations like ECOPO by (Li et al., 2022), which employs a contrastive learning training strategy using prediction errors as negative examples, the random input mask strategy proposed by(Wu et al., 2023), and the training strategy of Chinese spelling correction model as a rewritten language model (Liu et al., 2023) have both significantly enhanced BERT-based CSC models. This underscores the need to explore more effective learning strategies to fully develop BERT's potential in spelling correction tasks, which is also the focus of this paper.

2.2 Knowledge Distillation

Recent knowledge distillation in natural language processing leverages large-scale pre-trained language models for task-specific learning. (Sun et al., 2019) proposed a method to distill multiple intermediate layers of the teacher network into the student network. TinyBERT (Jiao et al., 2020) employs a two-stage framework with BERT, initially using general corpora, and then fine-tuning for specific tasks. BERT-EMD (Li et al., 2020) introduced a versatile layer mapping distillation, enabling comprehensive learning from teacher networks. LAD (Lin et al., 2023) developed a hierarchical adaptive distillation method, iterative aggregating knowledge through multiple gate blocks. In a different approach, DynaBERT(Hou et al., 2020) created an adaptive student network that adjusts its size for varied tasks, while (Sun et al., 2020) added a bottleneck structure to student Transformer models for compactness.

However, these methods require specially designed or task-related fine-tuned teacher networks to achieve knowledge distillation and model compression. Different from them, we adopt a teacher network with prior knowledge, achieving better performance while incurring lower costs.

3 Methodology

In this section, we provide a comprehensive description of our proposed Prior-knowledge Guided Teachers network (PGT), as depicted in Figure 2. PGT adopts a teacher network guided by task-related prior knowledge instead of task-related pre-trained teacher networks. PGT offers several key advantages. Firstly, our approach does not necessitate any alterations to the input or structure of the CSC model, allowing for the training of an unbiased CSC model from real human data. Secondly, the model-independent of PGT facilitates superior performance across various PLM-based CSC models. Finally, compared with other KD schemes, PGT does not require task-related pre-training and fine-tuning of the teacher network, resulting in a significant reduction in its computational cost.

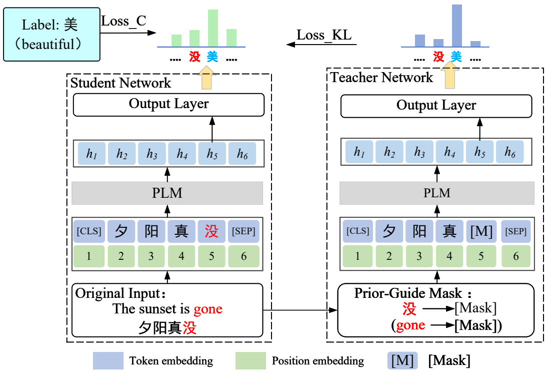


Figure 2: Overview of PGT. The Prior-knowledge Guided Teacher network processes the same input as the CSC model except masks spelling errors under the guidance of prior-knowledge.

3.1 Prior-knowledge Guided Teacher

PGT is compatible with various PLMs-based CSC models, and this section uses BERT as an example to provide a detailed description of the PGT framework. The prior-knowledge guided teacher network, requiring no task-related pre-training and its parameters are frozen during the fine-tuning. It is important to note that to prevent label information leakage, the output of the teacher network is employed solely as a soft label and is not used during the prediction.

The input of the teacher is a natural language sequence $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, consisting of n characters. Its output is a sequence of probability distributions $\mathbf{P} = \{p_1, p_2, ..., p_n\}$, where each p_i represents the probability distribution of the correction characters predicted by the teacher network.

For any given input sequence $\mathbf{X}$, the teacher network first embeds it by Equation 2

$$\mathbf{E} = BertEmbedding(\mathbf{X}) \tag{2}$$

Then, the teacher network masks $\mathbf{E} = \{e_1, e_2, ..., e_n\}$ based on the error position labels as Equation 3:

$$e_i' = b_i \cdot e_{mask} + (1 - b_i) \cdot e_i \tag{3}$$

Where b_i is the error position label whose value range is $\{0, 1\}$. When x_i and golden character are equal, $b_i = 0$, Otherwise $b_i = 1$. e_{mask} denotes the embedding of '[MASK]'.

Then feed $\mathbf{E}' = \{e'_1, e'_2, \cdots, e'_n\}$ to the BERT model, and use the hidden states $\mathbf{H} = \{h_1, h_2, \cdots, h_n\}$ of the last layer of BERT to predict the output probability distribution as Equation 4:

$$p_{ci} = p_c(y_i \mid \mathbf{X}) = softmax(\mathbf{W}\mathbf{h}_i + \mathbf{b}) \quad (4)$$

Where $\mathbf{W}$ and $\mathbf{b}$ are trainable parameters. $\mathbf{P}_c = \{p_{c1}, p_{c2}, \cdots, p_{cn}\}$ is the output of the teacher network.

3.2 Learning

The loss function used in this study consists of two terms. Contrasted with the traditional CSC training strategy that solely relies on cross-entropy loss, our method incorporates an additional Kullback-Leibler (KL) losses. This is designed to steer the CSC model towards a greater focus on language modeling capabilities, thereby mitigating over-correction issues that arise from memorizing training edit pairs. For a training sample $(\mathbf{X}, \mathbf{Y})$, equations 5 and 6 define these two loss function terms, respectively.

$$\mathcal{L}_{KL} = \sum_{i=1}^{n} D_{KL}(p_c(y_i \mid \mathbf{X}) \| p(y_i \mid \mathbf{X})) \quad (5)$$

$$\mathcal{L}_c = -\sum_{i=1}^{n} \log(p(y_i \mid \mathbf{X})) \quad (6)$$

Where $p_c(y_i \mid \mathbf{X})$ represents the probability distribution modeled by the teacher network, $p(y_i \mid \mathbf{X})$ denotes the probability distribution predicted by the CSC model and D_{KL} denotes the KL loss. Finally, we use the linear combination of the two loss functions as the overall objective, as Equation 7.

$$\mathcal{L} = \beta \cdot \mathcal{L}_c + (1 - \beta) \cdot \mathcal{L}_{KL} \quad (7)$$

Where β is the hyperparameter that balances the two loss functions.

4 Experiments

This section will give a comprehensive overview of the data, settings, and outcomes of the experiments in this paper. Moreover, we will perform essential analysis and discussion to show the merits of our approach.

4.1 Baselines

In order to demonstrate the superiority of this method, we selected different types of strong baseline methods, which represent different CSC model paradigms. **BERT** (Devlin et al., 2019) is the cornerstone of existing PLMs-based CSC models, and we fine-tune BERT directly on the training set. **Soft-Masked BERT**[1] (Zhang et al., 2020) is a detection-correction CSC model, it masks the detected error characters and fed the masked input into the BERT-based model for correction. **Spell-GCN** (Cheng et al., 2020) promotes the performance of CSC with additional information, which integrates confusing characters into CSC models via GCNs. **MDCSpell**[2] (Zhu et al., 2022) is a detection-correction CSC model too, which fuses the hidden states of the detection module and the correction module to predict the correct characters. **ECOPO**(Li et al., 2022) is a competitive training strategy designed for the CSC task driven by past predict errors. **Masked-FT** (Wu et al., 2023) achieved start-of-the-art work on SIGHAN14/15 by randomly masking 20% non-error tokens from the input sequence during fine-tuning.

4.2 Experiments Setup

According to the work of (Li et al., 2022), these evaluation metrics were computed by different algorithms, which could be grouped into two groups: character-level scores evaluated based on the algorithms from (Cheng et al., 2020; Wang et al., 2019), sentence-level scores evaluated based on the algorithms from (Hong et al., 2019; Liu et al., 2021)[3]. We report the sentence-level metrics for evaluation, which are more challenging.

Our code is based on BERT[4]. More detail experimental settings are in the Appendix A. Source code are available at this URL.

4.3 Experimental Results on SIGHAN

Following existing works, our training data consists of two group samples. The first group samples come from the SIGHAN13/14/15[5] (Tseng et al., 2015; Wu et al., 2013; Yu and Li, 2014) training set and were written by humans, and the other is 271k

[1] https://github.com/hiyoung123/SoftMaskedBert

[2] https://github.com/iioSnail/MDCSpell_pytorch

[3] https://github.com/liushulinle/PLOME

[4] https://huggingface.co/bert-base-chinese

[5] http://nlp.ee.ncu.edu.tw/resource/csc.html

Test set	**Method**	**Detection**			Correction		
		P	**R**	**F1**	**P**	**R**	**F1**
SIGHAN14	SpellGCN(Li et al., 2021)	65.1	69.5	67.2	63.1	67.2	65.3
	ECOPO (REALISE) (Li et al., 2022)	68.8	**72.1**	70.4	67.5	**71.0**	69.2
	BERT (Xu et al., 2021)	64.5	68.6	66.5	62.4	66.3	64.3
	ECOPO (BERT) (Li et al., 2022)	65.8	69.0	67.4	63.7	66.9	65.3
	PGT (BERT)	**70.4**↑	67.7	69.0↑	68.6↑	66.0	67.3↑
	MDCSpell (Zhu et al., 2022)	70.2	68.8	69.5	69.0	67.7	68.3
	PGT (MDCSpell)	**70.4**↑	71.3↑	**70.9**↑	**69.1**↑	70.0↑	**69.5**↑
SIGHAN15	SpellGCN (Li et al., 2021)	74.8	80.7	77.7	72.1	77.7	75.9
	ECOPO (REALISE) (Li et al., 2022)	77.5	**82.6**	80.0	76.1	81.2	78.5
	BERT (Xu et al., 2021)	74.2	78.0	76.1	71.6	75.3	73.4
	Masked-FT (BERT) (Wu et al., 2023)	-	-	-	76.7	79.1	77.9
	ECOPO (BERT) (Li et al., 2022)	78.2	82.3	80.2	76.6	80.4	78.4
	PGT(BERT)	**81.6**↑	80.4	81.0↑	**80.1**↑	79.0	**79.6**↑
	Soft-Masked BERT (Zhang et al., 2020)	78.1	82.1	80.0	74.9	78.8	76.8
	Masked-FT (Soft-Maked) (Wu et al., 2023)	-	-	-	76.3	**81.8**	79.0
	PGT (Soft-Masked)	80.4↑	81.7	**81.1**↑	78.0↑	79.3	78.7
	MDCSpell (Zhu et al., 2022)	80.8	80.6	80.7	78.4	78.2	78.3
	PGT (MDCSpell)	80.4	81.7↑	**81.1**↑	78.3	80.1↑	79.2↑

Table 1: The performance of PGT and baselines on SIGHAN test sets. The baseline results are from published papers, except Soft-Masked BERT (The training set of the original Soft-Masked BERT is different from other works, so we reproduced it by the same training set with other works). PGT (Models_X) denotes applying PGT to Models_X. The best results are in **bold**. "↑" signifies that the corresponding baseline method achieves a performance improvement after optimization by PGT.

public augmented training samples[6]. The data style of the two group samples is consistent.

In order to test the effectiveness of the methods, the widely used SIGHAN14/15 (Tseng et al., 2015; Yu and Li, 2014) test set is used as the evaluation benchmark. The example of data samples, training data statistics, and testing data statistics are shown in the Appendix B.

We report the experimental results in Table 1. From the experimental results, we draw the following conclusions through observation.

For the BERT model, PGT (BERT) outperforms the original BERT in both detection and correction performance on all test sets. Compared to BERT, PGT (BERT) achieved a 6% absolute improvement in correction F1 on the SIGHAN 15 test set, reaching state-of-the-art levels. Note that we did not change the structure of the BERT model or use any features other than the token feature. Such experimental results demonstrate the capabilities of PGT. Although PGT (BERT) achieved a 3% absolute improvement in BERT correction F1 on the SIGHAN 14 test set, it has not yet achieved state-of-the-art results. The current best on the SIGHAN 14 test set is PGT (MDCSpell), highlighting the model-independent advantage of PGT.

Compared to Masked-FT, which adopts a random masking strategy on the input of the CSC model, PGT (BERT) demonstrates superior performance in terms of correction F1. Masked-FT employs a strategy that randomly masks 20% non-error tokens from the input during the fine-tuning phase, yet it utilizes the complete input for prediction. In contrast, the CSC model in PGT consistently uses complete input throughout both the fine-tuning and prediction stages. This consistency may be a crucial factor in PGT's superior performance over Masked-FT.

The absolute improvement achieved by PGT varies for different works. This is partly because the basic performance of Soft-Masked BERT and MDCSpell far exceeds that of BERT, making it difficult to improve them with a BERT-based prior teacher network, and instead, a stronger PLM should be used as the teacher network.

4.4 Experimental Results on ECSpell

The ECSpell benchmark consists of corpora from three domains: LAW (1,960 training and 500 test examples), MED (medical treatment, 3,000 training and 500 tests), and ODW (official document writing, 1,728 training and 500 tests). Following the setup of the (Wu et al., 2023), we divided the test set of each domain into two subsets. One sub-

[6] https://github.com/wdimmy/Automatic-Corpus-Generation

	Method	INC-F1	EXC-F1	F1
LAW	BERT	68.4	10.0	40.2
	MASK-FT (BERT)	84.9	65.9	76.8
	PGT(BERT)	81.9	66.2	76.6
	MDCSpell	69.0	13.7	42.2
	MASK-FT (MDCSpell)	86.1	73.2	81.1
	PGT (MDCSpell)	83.2	67.1	77.6
MED	BERT	35.6	5.7	26.9
	MASK-FT (BERT)	46.7	43.2	63.8
	PGT(BERT)	78.3	50.7	66.7
	MDCSpell	32.1	7.4	25.7
	MASK-FT (MDCSpell)	47.9	47.8	72.4
	PGT (MDCSpell)	77.1	45.4	64.2
ODW	BERT	54.4	7.4	26.7
	MASK-FT (BERT)	71.3	42.4	62.9
	PGT(BERT)	80.9	72.5	63.3
	MDCSpell	55.9	6.7	27.5
	MASK-FT (MDCSpell)	75.1	51.2	72.0
	PGT (MDCSpell)	79.2	61.0	70.4

Table 2: The performance of PGT on ECSpell test sets.

set contains editing pairs not seen in the training set (EXC, shorthand for exclusive), and the remaining data constitute another subset (INC, shorthand for inclusive).

From Table 2, it can be observed that PGT (BERT) and PGT (MDCSpell) got better performance across most metrics in three domains, compared to the baselines. PGT (BERT) and PGT (MDCSpell) are significantly better than their own basic models on EXC-F1, INC-F1, and F1 scores.

Unlike SIGHAN, the test set of ECSpell contains a high proportion ($\approx 70\%$) of edit pairs that are not seen in the training set. **The performance of PGT (BERT) and PGT(MDCSpell) on ECSpell demonstrates that although improving the model's performance on unseen edit pairs was not the primary focus of this paper, PGT still displayed impressive capabilities in this regard**. This is attributed to PGT's ability to prevent overfitting of the student network to the training set, thereby helping the student network to make more contextual decisions.

4.5 Analysis and Discussion

In this section, a detailed analysis and discussion of PGT will be conducted on the SIGHAN 15 test set, chosen for its superior annotation quality compared to other test sets.

	Detection			Correction		
Method	P	R	F1	P	R	F1
BERT	74.2	78.0	76.1	71.6	75.3	73.4
PGT_S	79.8	55.6	65.6	78.5	54.7	64.5
PGT_L	81.2	78.2	79.7	78.4	75.5	76.9
PGT	81.6	80.4	81.0	80.1	79.0	79.6
MDCSpell	80.8	80.6	80.7	78.4	78.2	78.3
PGT_S (MDCSpell)	79.4	65.5	71.8	78.1	64.4	70.6
PGT_L (MDCSpell)	78.2	81.2	79.6	75.7	78.6	77.1
PGT (MDCSpell)	80.4	81.7	81.1	78.3	80.1	79.2

Table 3: Evaluating the Contribution of Prior Knowledge to the PGT.

4.5.1 The Contribution of Prior Knowledge

In this subsection, an ablation study will be conducted to evaluate the contribution of prior knowledge to the PGT. Given the unique nature of the CSC task, where the input may contain spelling errors, we experimented with two distinct types of input schemes. **PGT_S**: The inputs of the teacher and CSC networks both use original texts as input. **PGT_L**: The student network uses the original texts as input and the teacher network uses the ground truth texts as input.

As shown in Table 3, we observed that PGT_S significantly reduced model performance. This indicates that the strength of PGT comes from the prior knowledge added to the Prior teacher network. PGT_L, which uses label text as the input to the teacher network, can obtain completely correct input information.

Additionally, experimental results using MDCSpell as the base model show that the performance of both PGT_S and PGT_L settings is inferior to MDCSpell, while the PGT setting can improve the performance of MDCSpell. This indicates that using only PLM as the teacher network is insufficient; the use of prior information about the position of misspellings in the input is crucial. On the other hand, this result also suggests that when combining PGT with strong baseline CSC models, consideration should be given to using a teacher network larger than the student network (the setting in this paper is that the sizes of the teacher and the student network are the same, but general KD settings typically uses a larger teacher than the student).

4.5.2 Over Correction

In the Introduction section, we explored the motivation behind PGT, which is primarily aimed at addressing the over-correction issue in BERT-based

Test set	Method	Over	Total	ratio
SIGHAN14	BERT	189	371	50.9%
	PGT	125	315	39.7%
SIGHAN15	BERT	125	259	48.3%
	PGT	81	210	38.6%

Table 4: Statistics results on over-correction characters. In the table header, 'Over' represents the number of over-correction characters, and 'Total' refers to the overall number of failed prediction characters.

Method	Detection			Correction		
	P	R	F1	P	R	F1
BERT	74.2	78.0	76.1	71.6	75.3	73.4
KD_S	74.6	78.2	76.4	72.5	76.0	74.2
KD_L	77.5	80.3	78.9	74.0	76.6	75.2
PGT	**81.6**	**80.4**	**81.0**	**80.1**	**79.0**	**79.6**

Table 5: Results of different distillation methods on SIGHAN15

CSC models. This subsection presents an analysis that determines whether PGT achieves this objective.

Firstly, we clarify the definition of over-correction and failed prediction. A *failed prediction* refers to instances where the model's prediction does not align with the ground truth. An *over-correction* refers to the model wrong predicting a correct input character as another character. The 'ratio' is defined as the percentage of over-correction characters relative to all failed prediction characters. The statistics results are displayed in Table 4.

The statistics results lend considerable support to the argument of this paper. About half of BERT's failed predictions on two widely used benchmarks are over-corrections, highlighting the over-correcting propensity of BERT-based CSC models. When we run PGT, both the number and ratio of over-correction characters decrease significantly. This demonstrates the efficacy of the PGT in overcoming the over-correction challenge.

4.5.3 Comparison with General Knowledge Distillation

In this section, we will conduct a comparative analysis of PGT against the general two-stage KD scheme. This two-stage KD involves an initial teacher network fine-tuning stage, followed by the KD stage. In Fine-Tuning Stage, training the teacher network with the training set. During the KD stage, the fine-tuned teacher network is employed to distill knowledge into the student network, with the teacher network's gradient being frozen. Therefore, we tried two different types of input schemes during the KD stage: **KD_S**: The inputs of the teacher network and the student network are both using the original text as input. **KD_L**: The student network uses the original texts as input, while the teacher network uses the ground truth texts as input.

Table 5 presents the performance of various KD schemes, with each scheme employing BERT as the student network. The experimental findings indicate that, in comparison to PGT, KD_S has limited improvement on the student network in the CSC task. Both KD_L and KD_S utilize the same teacher network, differing solely in their input. Yet, KD_L exhibits a substantial performance improvement over KD_S. This observation underscores that a significant factor limiting the effectiveness of classical knowledge distillation methods in spell correction tasks is the disruption caused by typos in the input. Furthermore, it is noteworthy that PGT outperforms both KD_S and KD_L in terms of detection and correction precision. This proves from another perspective that PGT can avoid the over-correction problem of the BERT-based CSC model, and this is guided by prior knowledge.

4.5.4 The Contribution of PGT to Model Compression

In previous experiments, the scales of the CSC model and the teacher network in PGT were aligned. To evaluate the model compression capabilities of PGT, this section employs a CSC model composed of N×Transformer encoder blocks, replacing the BERT-based CSC model. The experimental results are presented in Table 6.

The results presented in Table 6 demonstrate the efficient model compression capabilities of PGT. The experiments indicate that when the student network's scale is increased to 66.6% of BERT's, PGT surpasses BERT's performance and outperforms the general KD approach. Furthermore, even when the student network consists of only 4×Transformer encoder blocks, PGT's detection and correction precision exceed those of both the BERT model. This finding implies that PGT consistently avoids over-corrections of correct input

Method	P	R	F1	P	R	F1
4×Transformer	76.8	66.7	71.4	74.9	65.1	69.6
6×Transformer	79.3	70.6	74.7	77.4	68.9	72.9
8×Transformer	79.7	77.8	78.8	77.7	75.8	76.7
10×Transformer	80.3	79.9	80.1	78.4	78.0	78.2
BERT	74.2	78.0	76.1	71.6	75.3	73.4
KD_S	74.6	78.2	76.4	72.5	76.0	74.2

Table 6: The results of PGT in model compression

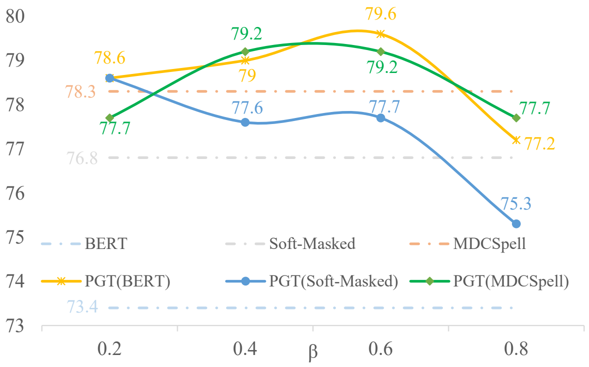


Figure 3: The impact of β

tokens, a trait that remains unaffected by the scale of the student network.

4.5.5 Parameter Analysis

There is an important hyperparameter β in this paper. In order to evaluate the impact of the hyperparameter β, we assigned different values to β and observed its impact.

Figure 3 shows the changes in the correction F1 values of three models. We found that when β is around 0.5, the model is more likely to achieve better results. This indicates that the CSC model has roughly equal requirements for the two types of labels. On the other hand, SoftMasked BERT and MDCSpell are more dependent on ground truth than teacher networks. This is because the structure of Soft-Masked BERT and MDCSpell is more complex and requires clearer labels to prevent underfitting.

4.6 Generalize

The core of the PGT method proposed in this paper is the prior-knowledge guided teacher network. However, the error position information is a task-specific prior knowledge for CSC, which may raise concerns about the generality of the PGT method. Therefore, in this subsection, we will explore whether we can generalize the PGT method to other non-autoregressive text generation tasks. To generalize PGT, we use a random masking strategy, which randomly masks 15% tokens from the input of the teacher network, and we call this setting as PGT_F. The experimental results are shown in Table 7.

Method	Detection			Correction		
	P	R	F1	P	R	F1
BERT	74.2	78.0	76.1	71.6	75.3	73.4
PGT_L	81.2	78.2	79.7	78.4	75.5	76.9
PGT	81.6	80.4	81.0	80.1	79.0	79.6
PGT_F	79.9	81.3	80.6	76.6	78.0	77.3

Table 7: The results of the generalization experiment

As shown in Table 7, PGT_F performs between PGT and PGT_L (the definition of PGT_L is detailed in Section 4.5.1) on the detection and correction level. This result indicates that PGT can still improve the model performance without task-related prior knowledge. Therefore, the implications of the PGT proposed in this paper are not limited to the CSC task. PGT has the potential to generalize to other non-autoregressive text generation tasks. However, when viewed from another perspective, the performance of PGT_F falls short compared to PGT. This suggests that a crucial factor in the generalization of PGT lies in the utilization of task-related prior knowledge.

5 Conclusion

Building on the idea of augmenting the language modeling capabilities of the CSC model, this paper proposes to use a Prior-knowledge Guided Teacher (PGT) network to distill the CSC model. Extensive experiments conducted on widely used benchmarks affirm that our method achieves new state-of-the-art results on the CSC task. Furthermore, the experiment results validate that PGT indeed enhances the language modeling capabilities of the CSC model.

In terms of knowledge distillation, PGT outperforms general KD schemes in both performance and training efficiency for CSC tasks, while also demonstrating notable model compression efficiency. Lastly, we verify the potential of PGT to generalize to other non-autoregressive text-generation tasks through a simple experiment.

Limitations

Our method uses an additional teacher network. While this teacher network does not require task-related pre-training and parameter updates, it still incurs an additional computational overhead of approximately 25% during the fine-tuning. Although the focus of this work is on the Chinese, it does not utilize any Chinese-specific features. Therefore, in theory, other languages, such as English could also benefit from the same technique. However, empirical studies in these languages have not been conducted in this paper.

References

Haithem Afli, Zhengwei Qiu, Andy Way, and Páraic Sheridan. 2016. Using SMT for OCR error correction of historical texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 962–966, Portorož, Slovenia. European Language Resources Association (ELRA).

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 the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring – an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, Austin, Texas. Association for Computational Linguistics.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335.

Jianyong Duan, Bing Wang, Zheng Tan, Xiaopeng Wei, and Hao Wang. 2019. Chinese spelling check via bidirectional lstm-crf. In *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, pages 1333–1336.

Harsh Gupta, Luciano Del Corro, Samuel Broscheit, Johannes Hoffart, and Eliot Brenner. 2021. Unsupervised multi-view post-OCR error correction with language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8647–8652, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic bert with adaptive width and depth. In *Advances in Neural Information Processing Systems*, volume 33, pages 9782–9793. Curran Associates, Inc.

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.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *Computer Science*.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Mark D. Kernighan, Kenneth W. Church, and William A. Gale. 1990. A spelling correction program based on a noisy channel model. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.

Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020. BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural*

Language Processing (EMNLP), pages 3009–3018, Online. Association for Computational Linguistics.

Jing Li, Gaosheng Wu, Dafei Yin, Haozhao Wang, and Yonggang Wang. 2021. Dcspell: A detector-corrector framework for chinese spelling error correction. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1870–1874, New York, NY, USA. Association for Computing Machinery.

Yinghui Li, Qingyu Zhou, Yangning Li, Zhongli Li, Ruiyang Liu, Rongyi Sun, Zizhen Wang, Chao Li, Yunbo Cao, and Hai-Tao Zheng. 2022. The past mistake is the future wisdom: Error-driven contrastive probability optimization for Chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3202–3213, Dublin, Ireland. Association for Computational Linguistics.

Ying-Jia Lin, Kuan-Yu Chen, and Hung-Yu Kao. 2023. Lad: Layer-wise adaptive distillation for bert model compression. *Sensors*, 23(3).

Linfeng Liu, Hongqiu Wu, and Hai Zhao. 2023. Chinese spelling correction as rephrasing language model. *ArXiv*, abs/2308.08796.

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, Dublin, Ireland. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

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, Beijing, China. Association for Computational Linguistics.

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

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided pointer networks for Chinese spelling check. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785, Florence, Italy. Association for Computational Linguistics.

Hao Wang, Bin Wang, Jianyong Duan, and Jiajun Zhang. 2021. Chinese spelling error detection using a fusion lattice lstm. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(2).

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

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, Nagoya, Japan. Asian Federation of Natural Language Processing.

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*.

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

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

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI*

Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 1226–1233. AAAI Press.

Chenxi Zhu, Ziqiang Ying, Boyu Zhang, and Feng Mao. 2022. MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1244–1253, Dublin, Ireland. Association for Computational Linguistics.

A Experiments Details

When fine-tuning, we used the AdamW optimizer with default hyper-parameters. Unless otherwise stated, all experimental results reported in this paper are the average of four experiments. Following the MDCSpell, we first use all training data to fine-tune the model, where the learning rate is 2e-5. Then, we use the SIGHAN training data to fine-tune the model, where the learning rate is 1e-5. The detailed experimental settings are as follows:

- We do not use the dynamic learning rate strategy.
- We always freeze the gradient of the teacher network during fine-tuning.
- The batch size is set to 32.
- The training epochs are set to 30, and use an early stopping strategy.
- For baselines, we use the best hyperparameters reported by the author.
- Using RTX3090 GPU for training.

B Datasets Details

A data sample is shown in Table 8. Where 'id' represents the sample number of the data, 'original_text' represents the original text needs to be corrected, 'wrong_ids' represents location labels of spelling errors in the original text, and if there are no typos in the original text, 'wrong_ids' is empty. 'correct_text' represents the ground truth text.

id:	A2-0023-1
original_text:	下个星期，我跟朋游打算去法国
wrong_ids:	[9]
Correct_text:	下个星期，我跟朋友打算去法国

Table 8: Example of Sample.

Table 10 shows the detailed statistics of the datasets that we use. We report the number of sentences in the datasets (Line), the average sentence length of the datasets (Avg. Length), and the number of typos the datasets contain (Errors).

Train and Val Data	Line	Avg. Length	Errors
(Wang et al., 2018)	271,329	44.4	382,704
SIGHAN13	350	49.2	350
SIGHAN14	6,526	49.7	5284
SIGHAN15	3,174	30.0	3143
Test Data	**Line**	**Avg. Length**	**Errors**
SIGHAN14	1062(531)	50.1	782
SIGHAN15	1100(550)	30.5	715

Table 9: Statistics of the datasets that we use.

C Case Study

The correction results of PGT and baseline are shown in Table 10, and the cases we use are the same as ECOPO. In Table 10, we mark the input error/confusion/golden/wrong correction characters in red/green/blue/yellow.

Input:	与其自暴自气 (弃) 不如往好处想。 It's better to think for the good than to be angry (give up).
BERT:	[己(own), 大(big), 利(benefit)]
ECOPO:	[弃(give up), 尊(respect), 强(strong)]
ORPO:	[弃(give up), 气(angry), 起(raise)]
Input:	我努力打败数不进(尽)的风雨。 I try to beat the enter (endless) storms.
BERT:	[起(raise), 上(up), 得(get)]
ECOPO:	[尽(endless), 得(get), 完(end)]
ORPO:	[尽(endless), 进(enter), 近(near)]

Table 10: Examples of spelling errors and corresponding output (Top 3 candidates) of different methods. We mark the input error/confusion/golden/wrong correction characters in red/green/blue/yellow.

BERT fails in two cases. For the first case, BERT assigns the most predicted probability to the common character "己" instead of the golden character "弃". The statistics of the pre-training corpus Wiki2019ZH show that "自己" appears 136,318 times, and "自弃" only appears 119 times. For the second case, the output of BERT such as "起", "上" and "得" all are far away from the golden character in terms of pronunciation and glyph. This supports the core motivation of this paper that there is a gap

Method	CAR	ENC	GAM	MEC	NEW	NOV	COT	AVG
BERT	15.1	13.6	14.3	12.6	16.6	15.1	17.3	14.9
PGT(BERT)	32.8	32.8	25.5	30.5	33.3	24.3	45.0	32.0

Table 11: The performance of PGT and baseline on LEMON.

between the knowledge representation of BERT and the knowledge required by spelling correction.

ECOPO (BERT) and PGT (BERT) make correct corrections in both cases. However, the other two of the top three candidate characters for ECOPO (BERT) are not confusing characters. In comparison, the top three candidate characters for PGT are all confusing characters, with the same phonics as the golden characters “弃(qì)” and “尽(Jìn)”. This result shows that PGT can not only accurately assign the maximum probability to the golden character and make correct corrections, but also pay more attention to confusing characters rather than common characters. This supports the central argument of this paper that PGT can bridge the gap between PLMs and spelling correction task by teaching the PLMs model how to fully utilize the learned knowledge for spelling correction task.

D Experimental Results on LEMON

For the LEMON benchmark (Wu et al., 2023), we opted to directly test on the LEMON dataset using the model without benchmark-specific fine-tuning. (The model was fine-tuned on the training set described in Section 4.1)

LEMON consists of corpora from seven different domains. As shown in the table mentioned earlier, PGT (BERT) achieved significant improvements in each domain, even without domain-specific adaptation. This experiment conducted on the high-quality benchmark revalidated the effectiveness of PGT.