

# An Error-Guided Correction Model for Chinese Spelling Error Correction

Rui Sun<sup>1,2</sup>, Xiuyu Wu<sup>1,2</sup>, Yunfang Wu<sup>1,3\*</sup>

<sup>1</sup>MOE Key Laboratory of Computational Linguistics, Peking University Beijing, China

<sup>2</sup>School of Software and Microelectronics, Peking University, Beijing, China

<sup>3</sup>School of Computer Science, Peking University, Beijing, China

{sunrui0720, xiuyu\_wu}@stu.pku.edu.cn, wuyf@pku.edu.cn

## Abstract

Although existing neural network approaches have achieved great success on Chinese spelling correction, there is still room to improve. The model is required to avoid over-correction and to distinguish a correct token from its phonological and visually similar ones. In this paper, we propose an error-guided correction model (EGCM) to improve Chinese spelling correction. By borrowing the powerful ability of BERT, we propose a novel zero-shot error detection method to do a preliminary detection, which guides our model to attend more on the probably wrong tokens in encoding and to avoid modifying the correct tokens in generating. Furthermore, we introduce a new loss function to integrate the error confusion set, which enables our model to distinguish easily misused tokens. Moreover, our model supports highly parallel decoding to meet real application requirements. Experiments are conducted on widely used benchmarks. Our model achieves superior performance against state-of-the-art approaches by a remarkable margin, on both the correction quality and computation speed.

## 1 Introduction

Chinese spelling correction (CSC) attracts wide attention in recent years, which is significant for many real applications, such as search engine (Martins and Silva, 2004), optical character recognition(OCR) (Afli et al., 2016) and automatic speech recognition(ASR) (Hinton et al., 2012).

Given an input sentence with spelling errors, the model is trained to detect and correct these errors and output a correct sentence. According to Liu et al. (2010), phonologically and visually similar characters are major contributing factors for errors in Chinese text. As shown in Figure 1, in the first example, the error is caused by the misuse of "派"(send) and "拍"(take) which have similar

\*Corresponding author.

wrong sentence	correct sentence	misused tokens	Type of Similarity
他派了很多照片。 He sent a lot of photos.	他拍了很多照片。 He took a lot of photos.	(派&拍)	phonological (sounds similar)
他门才开始考试。 His door starts the exam.	他们才开始考试。 They start the exam.	(门&们)	visual (looks similar)

Figure 1: Examples of Chinese spelling errors. Misspelled characters and their corresponding corrections are marked in red.

Chinese pronunciation. In the second example, the error is caused by the misuse of "门"(door) and "们"(they) which have similar shapes.

Recently, the advanced neural network models and pre-trained models have achieved great success in CSC, such as PLOME (Liu et al., 2021), REALISE (Xu et al., 2021), PHMOSpell (Huang et al., 2021), SpellBert (Ji et al., 2021), GAD (Guo et al., 2021), MLM-phonetics (Cheng et al., 2020), RoBERTa-DCN (Wang et al., 2021) and ECSpell (Lv et al., 2022). Although much progress has been made, there are still limitations in previous methods.

First, given an input sequence, only a small fragment might be misspelled. However, for most of the previous models, they are totally blind to the errors at start, and so they attend on all tokens equally in encoding and generate every token from left to right for inference. As a result, previous models are inefficient and might create over-correction. As these models obtain a stronger ability to correct the errors, they also tend to modify the correct tokens by mistake.

Second, the confusion set, where a set of phonological and visual similar tokens are defined for each Chinese token, provides valuable knowledge for spelling correction, as shown in Figure 2. But the methodology to use it should be further improved. For example, Liu et al. (2021) propose a Confusion Set based Masking Strategy, in which they remove a token and replace the token with a random character in the confusion set. As the

A Chinese Token $t$	Confusion set of $t$	
	visual similar tokens	phonetic similar tokens
大	犬 太 头	打 答 沓
义	艾 仪 议	以 亿 宜
早	早 异 阜	遭 澡 凿

Figure 2: An example of the confusion set.

model randomly chooses a token from the confusion set each time, some tokens might be ignored. Besides, this method can't pay more attention to the token that is more easily to be misused. Wang et al. (2019) propose to generate a character from the confusion set rather than the entire vocabulary. In this hard restriction, the model cannot generate tokens that are not in the confusion set.

Third, when a CSC model is deployed in real applications, the time cost of inference is a critical problem to be considered. However, most previous models try to improve the generation quality but ignore the computation speed.

To address these issues mentioned above, we propose an Error-Guided Correction Model (EGCM) for CSC. Firstly, taking advantage of the strong ability of BERT (Devlin et al., 2018), we propose a novel zero-shot error detection method to do a preliminary detection, which provides precise guidance signals to the correction model. Following the guidance, our model attends more on the probably wrong tokens in encoding, and fixes the probably correct tokens during generation to avoid over-correction. Furthermore, we introduce a new loss function that effectively integrates the confusion set. By applying this loss function, every similar token in the confusion set is learned to be distinguished from the target token, and the most similar token with a high possibility of being misused is given more attention. To speed up the inference, we apply a mask-predict strategy (Ghazvininejad et al., 2019) to support parallel decoding, where the tokens with low generation probability are masked and predicted iteratively.

We conduct extensive experiments on the widely used benchmark dataset SIGHAN (Wu et al., 2013; Yu and Li, 2014; Tseng et al., 2015). Experimental results show that our model significantly outperforms all previous approaches, achieving a new state-of-the-art performance for Chinese spelling correction. Moreover, our model has a distinct speed advantage over other models, which is 6.3 times faster than the standard Transformer and

1.5 times faster than the recent non-autoregressive model TtT (Li and Shi, 2021).

We summarize our contributions as follows:

- We propose a novel zero-shot error detection method, which guides the correction model to attend more on the probably wrong tokens in encoding and fix the probably correct tokens in inference to avoid over-correction.
- We propose a new loss function to take advantage of the confusion set, which enables our model to distinguish similar tokens and attach more importance to the easily misused tokens.
- We apply an error-guided mask-predict decoding strategy for spelling correction, which supports highly parallel decoding and greatly accelerates the computation speed.
- We integrate all modules into a unified model, which achieves a new state-of-the-art performance for both correction quality and inference speed.

## 2 Related work

CSC is a task that detect and correct wrong tokens in Chinese Sentences. It's an active topic that varieties of approaches have been proposed to tackle the task (Wang et al., 2019; Cheng et al., 2020; Li and Shi, 2021; Xu et al., 2021; Liu et al., 2021; Huang et al., 2021).

Earlier work in CSC focuses mainly on unsupervised methods, which typically adopts a confusion set to find correct candidates and employs a language model to select the correct one (Chen et al., 2013; Yu and Li, 2014). Recently, sequence translation and sequence tagging are the two most widely used methods in CSC. Wang et al. (2018) treats the CSC task as a sequence labeling problem, and use a bidirectional LSTM to predict the correct characters. Liu et al. (2021); Ji et al. (2021); Xu et al. (2021); Lv et al. (2022) try to enrich the representation generated by the encoder by introducing visual and phonetic features. Softmax operation is utilized to find a substitution for each token in the sentence. As the rapid development of neural machine translation (Vaswani et al., 2017), seq2seq encoder-decoder frameworks have been introduced to the CSC task in (Ji et al., 2017; Chollampatt et al., 2016; Wang et al., 2019).

Recent work tends to utilize character similarity as an external knowledge. The confusion set

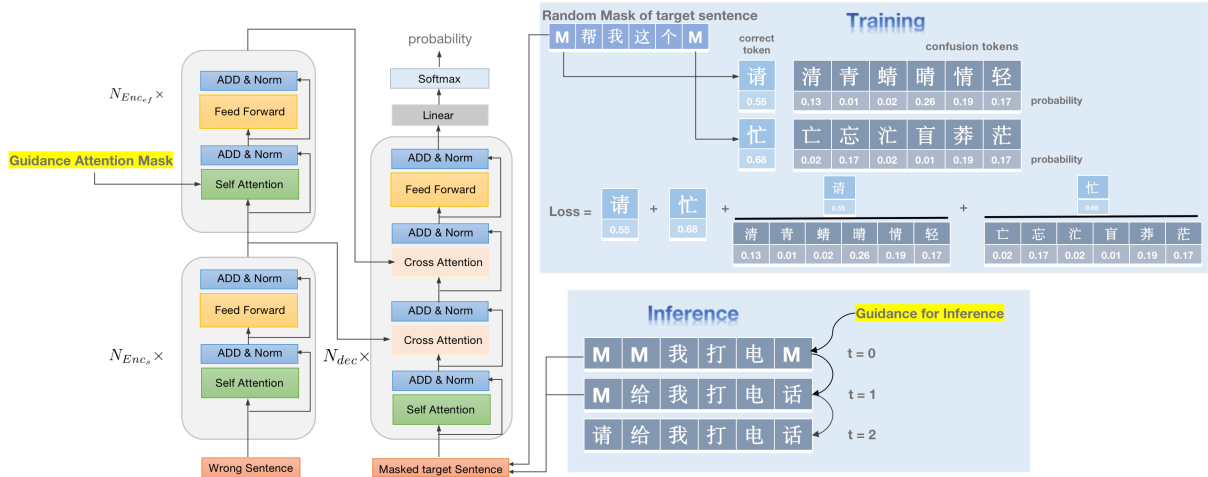


Figure 3: The architecture of our proposed model. "M" denotes the [MASK] token. *Guidance for Inference* and *Guidance Attention Mask* are generated from Zero-shot error detection as shown in Figure 4.

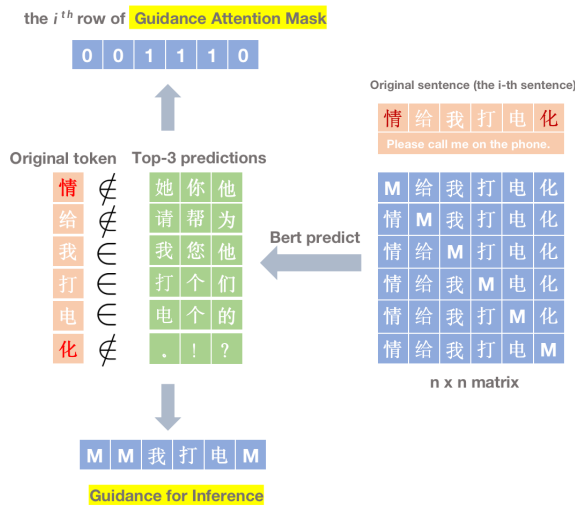


Figure 4: An example of how our model obtains guidance signals using a zero-shot method. The original wrong token is marked red.

where similar characters are stored is widely used (Liu et al., 2021; Zhang et al., 2020; Wang et al., 2019; Yu and Li, 2014; Cheng et al., 2020; Lv et al., 2022). There are several ways of using the confusion set. The first is to augment the training data by replacing the original token with its similar tokens (Liu et al., 2021; Zhang et al., 2020). Wang et al. (2019) proposes to generate a character from the confusion set rather than the entire vocabulary. Yu and Li (2014) proposes to produce candidates by retrieving the confusion set and then filter them via language models. Cheng et al. (2020) uses similarity graphs derived from the confusion set and use graph convolution operation to absorb the informa-

tion from neighboring characters in the graph.

### 3 Methodology

The proposed Error-Guided Correction Model (EGCM) is illustrated in Figure 3. We apply the conditional masked language model (CMLM) (Ghazvininejad et al., 2019) as a backbone, which is an encoder-decoder architecture trained with a masked language model objective (Devlin et al., 2018; Conneau and Lample, 2019). In the CMLM architecture, the source wrong sentence with  $n$  tokens is denoted as  $X = (x_1, x_2, x_3, \dots, x_n)$ , the target sentence is denoted as  $Y = (y_1, y_2, y_3, \dots, y_n)$ . Several tokens in  $Y$  are replaced with [MASK]. These masked tokens construct the set  $Y_{mask}$ . And the rest of the tokens in  $Y$  that are unmasked construct the set of  $Y_{obs}$ . For Chinese spelling correction, given a source sentence  $X$  and the set of unmasked target tokens  $Y_{obs}$ , the object is to predict the probability  $P(y|X, Y_{obs})$  and generate token  $y$  for each  $y \in Y_{mask}$ .

We first propose a zero-shot spelling error detection method to provide two guidance signals to the correction model, as shown in Figure 4. The first guidance signal is the *Guidance Attention Mask* that is used in the error-focused encoder, in which the probably correct tokens are masked to push our model to attend more on the wrong tokens. The second guidance signal is the *Guidance for Inference* that serves as the start of decoding to avoid modifying correct tokens by mistake. Moreover, we introduce a new loss function to take advantage of the confusion set. During inference, we apply an error-guided mask-predict strategy in which the

correct tokens are fixed and the probably wrong tokens are masked and repredicted iteratively.

### 3.1 Zero-shot Error Detection

Given a sentence  $X = (x_1, x_2, x_3, \dots, x_n)$  that contains  $n$  tokens, we want to make a preliminary decision on which tokens are probably wrong and which are correct.

As shown in Figure 4, firstly, we construct a  $n \times n$  matrix by repeating the original sentence  $n$  times, where the  $k^{th}$  token is masked in the  $k^{th}$  row in the matrix ( $k$  is from 1 to  $n$ ). Then, we employ BERT (Devlin et al., 2018) to predict each masked position condition on the unmasked tokens in the same row. Thus, for each position from  $x_1$  to  $x_n$  in the sentence  $X$ , we obtain the predicted tokens along with their probabilities. The tokens with the top- $k$  probabilities are selected as candidates of modification. We assume that if the original token  $x_i$  occurs in the candidates list, the token is considered correct. Otherwise, the token is probably wrong and needs to be corrected.

Based on the output of error detection, we construct two guidance signals namely *Guidance Attention Mask* and *Guidance for Inference*, as shown in Figure 4. The *Guidance Attention Mask* (GAM) is a matrix constructed by:

$$GAM_{ij} = \begin{cases} 0, & x_{ij} \text{ is probably wrong} \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

where  $x_{ij}$  denotes the  $j^{th}$  token in the  $i^{th}$  sentence.  $GAM_{ij}$  denotes the element of the  $i^{th}$  row and the  $j^{th}$  column in GAM. The *Guidance for Inference* (GFI) is constructed by masking all the probably wrong tokens in the original sentence. Further, GAM will be projected into the error-focused encoder, and GFI will be utilized to initialize the decoder.

### 3.2 Error-aware Encoder

We adopt the Transformer (Vaswani et al., 2017) encoder-decoder framework for Chinese spelling correction. We deviate from the standard Transformer encoder by fusing an error-focused encoder, as shown in the left part of Figure 3.  $Encoder_s$  is a standard Transformer encoder, and on top of that we introduce an error-focused encoder  $Encoder_{ef}$ , which utilizes *Guidance Attention Mask* to expose the probably wrong tokens and divert the attention of our model from the correct tokens. The output of  $Encoder_s$  is input into the error-focused encoder  $Encoder_{ef}$ . The *Guidance Attention Mask*

is used as an extra attention mask in calculating self-attention in  $Encoder_{ef}$ , which informs the model which error part of the sentence should be focused on. Concretely, the output of the  $Encoder_s$  and  $Encoder_{ef}$  is calculated respectively as:

$$H^s = Encoder_s(Embedding(X)) \quad (2)$$

$$H^{ef} = Encoder_{ef}(H^s, atten\_mask = GAM) \quad (3)$$

### 3.3 Integrating Error Confusion Set for Training

During training, the tokens in  $Y_{mask}$  are randomly selected among the target correct sentence as shown in Figure 3. To better fit the requirements of correcting both single-character errors and multi-character errors in Chinese spelling correction, we adopt two masking strategies, namely mask-separate and mask-range. In mask-separate, we first sample the number of masked tokens from a uniform distribution between  $[1, \text{len}(X)]$ , and then randomly choose that number of tokens. For mask-range, we select  $l \in [2, 3]$ , and randomly select a span with length  $l$ . We replace the tokens in  $Y_{mask}$  with a special [MASK] token, which is the generation object of the model.

There are three attention blocks in the Transformer decoder layer. After the self-attention block, the decoder will first attend to  $H^s$ , the representation of the source wrong sentence. Then, the decoder will attend to  $H^{ef}$ , the representation of the sentence with correct tokens being masked. The output of the previous decoder layer is then input into the next decoder layer.

$$H_{d_1}^l = \text{selfAttention}(H^{l-1}) \quad (4)$$

$$H_{d_2}^l = \text{Attention}(Q = H_{d_1}^l, K = H^s, V = H^s) \quad (5)$$

$$H^l = \text{Attention}(Q = H_{d_2}^l, K = H^{ef}, V = H^{ef}) \quad (6)$$

where  $H^0 = Embedding(Y_{obs})$ . Q, K, V represents the Query, Key, Value matrix.  $Y_{obs}$  is the set of unmasked tokens in the target sentence. The output probability distribution  $P$  is generated from the decoder over the vocabulary  $V$ :

$$P = \text{softmax}(H^l W + b) \quad (7)$$

where  $H^l \in \mathbb{R}^{t \times d}$ ,  $W \in \mathbb{R}^{d \times |v|}$ ,  $b \in \mathbb{R}^{t \times |v|}$ .  $t$  denotes the sequence length.

We optimize the model over every token in  $Y_{mask}$ . Besides the traditional loss function, we introduce a new loss to integrate the confusion set knowledge.



We employ Maximum Likelihood Estimation (MLE) to conduct parameter learning and utilize negative log-likelihood (NLL) as the loss function, which is computed as:

$$L_{nll} = - \sum_{y_i \in Y_{mask}} \log P(y_i | X, Y_{obs}) \quad (8)$$

To make full use of the confusion set knowledge, we introduce a new loss function  $L_{cs}$ . We adopt the confusion set constructed by Lv et al. (2022). For each token  $y_i$  in  $Y_{mask}$ , we find out the set of the similar tokens of  $y_i$  based on the confusion set, namely  $Y_{conf}$ . The tokens in  $Y_{conf}$  are regarded as negative samples of  $y_i$ . We use these negative samples to help our model better learn the difference between the target token and its similar ones. The optimization objective for the confusion loss  $L_{cs}$  is defined as:

$$L_{cs} = - \sum_{y_i \in Y_{mask}} \frac{\log P(y_i | X, Y_{obs})}{\sum_{y_c \in Y_{conf}} \log P(y_c | X, Y_{obs})} \quad (9)$$

where  $y_c$  denotes the similar token of  $y_i$  in the confusion set.

Overall, the final optimization objective of our model is:

$$L_f = L_{nll} + \gamma \times L_{cs} \quad (10)$$

where  $\gamma$  is a hyperparameter to balance two loss functions.

### 3.4 Error-Guided Generation

In the inference stage, we apply a mask-predict approach (Ghazvininejad et al., 2019), where the tokens with low probability are masked and predicted within a constant number of iterations.

To provide the model a good start point for generation, we exploit the *Guidance for Inference* (GFI) as an initialization for decoding. GFI produces a draft sentence, where the probably wrong tokens are masked and the probably correct ones are remained unmasked. During generation, the unmasked tokens will be fixed, and only the masked tokens are taken into consideration for modification in each iteration. Fixing these correct tokens will effectively teach our model to avoid over-correction. Figure 5 shows how does our model correct a wrong sentence in 3 iterations.

The model runs for a pre-determined number of iterations  $T$ . The number of [MASK] in the draft sentence is denoted as  $N_{ori}$ . Accordingly, the number of tokens that are masked in the  $t_{th}$

	We door American movies, I think it's a thing.
The wrong sentence	我 门 看 美 国 电 影 ， 我 觉 得 很 有 意 事 。
Guidance for Inference	我 门 看 美 国 电 影 ， 我 觉 得 很 有 意 事 。
t = 1	我 门 看 美 国 电 影 ， 我 觉 得 很 有 意 事 。
t = 2	我 们 看 美 国 电 影 ， 我 觉 得 很 有 意 事 。
t = 3	我 们 看 美 国 电 影 ， 我 觉 得 很 有 意 思 。
	We watch American movies, I think it's interesting.

Figure 5: An example of Error-Guided Generation. In Guidance for Inference, the masked tokens are highlighted. In later iterations, the highlighted tokens are of lowest probabilities and are masked and repredicted. The wrong tokens are marked in red.

iteration is defined as  $N_t = N_{ori} \times \frac{T-t}{T}$ . Formally,  $Y_{mask}^{(0)}$  is the set of masked tokens in the *Guidance for Inference*. At a later iteration  $t$ , we choose  $N_t$  tokens among the masked tokens in the previous iteration  $t-1$  that has the lowest probability scores:

$$Y_{mask}^{(t)} = \arg \min_{y_i \in Y_{mask}^{(t-1)}} (p_i, N_t) \quad (11)$$

$$Y_{obs}^{(t)} = Y \setminus Y_{mask}^{(t)} \quad (12)$$

Where  $p_i$  is the probability score of  $y_i$  calculated in Equation 13, 14.  $Y_{mask}^{(t)}$  is the set of masked tokens that are probably wrong at the  $t_{th}$  iteration, and  $Y_{obs}^{(t)}$  is the set of unmasked tokens that are considered correct and fixed in later iterations. At each iteration, the model predicts the probably wrong tokens in  $Y_{mask}^{(t)}$  conditioned on the source text  $X$  and  $Y_{obs}^{(t)}$ . We select the prediction with the highest probability for each masked token  $y_i \in Y_{mask}^{(t)}$ , and update its probability score accordingly:

$$y_i^{(t)} = \arg \max_{w \in V} P(y_i = w | X, Y_{obs}^{(t)}) \quad (13)$$

$$p_i^{(t)} = \max_{w \in V} P(y_i = w | X, Y_{obs}^{(t)}) \quad (14)$$

where  $P(y_i = w | X, Y_{obs}^{(t)})$  is the conditional probability of  $y_i$  being predicted as the token  $w$  in the vocabulary set  $V$ .

## 4 Experimental Setup

### 4.1 Dataset and Metrics

**Training dataset** Following Liu et al. (2021), the training data is composed of 10K manually annotated samples from SIGHAN (Wu et al., 2013) and 271K automatically generated samples from (Wang et al., 2018).

**Evaluation dataset** Following previous works, the SIGHAN15 test dataset (Tseng et al., 2015) is used to evaluate the proposed model. Statistics of the used datasets please refer to Appendix A.

**Evaluation Metrics** We evaluate model performance of detection and correction at sentence-level, with accuracy, precision, recall and F1 scores. We evaluate these metrics using the script from Cheng et al. (2020)<sup>1</sup>. Moreover, following Liu et al. (2021), we also report the sentence-level results evaluated by SIGHAN official tool<sup>2</sup>.

## 4.2 Comparing Methods

We compare the performance of our model with several strong baseline methods as follows:

**Confusionset** introduces a copy mechanism into seq2seq and generates characters from the confusionset (Wang et al., 2019).

**FASpell** utilizes a denoising autoencoder to generate candidates (Hong et al., 2019).

**SpellGCN** incorporates phonological and visual knowledge via a graph convolutional network (Cheng et al., 2020).

**Chunk** proposes a chunk-based decoding method with global optimization (Bao et al., 2020).

**SM BERT** uses soft-masking technique to connect the network of detection and correction (Zhang et al., 2020).

**TtT** employs a Transformer Encoder with a Conditional Random Fields layer stacked (Li and Shi, 2021).

**PLOME** proposes a confusion set based masking strategy (Liu et al., 2021).

**REALISE** leverages the multimodal information and mixes them electively (Xu et al., 2021).

**PHMOSpell** integrates pinyin and glyph with a multi-modal method (Huang et al., 2021).

**ECSpell** adopts the Error Consistent masking strategy for pretraining (Lv et al., 2022).

**MLM-phonetics** integrates phonetic features by leveraging pre-training and fine-tuning (Zhang et al., 2021).

**RoBERTa-DCN** generates the candidates via a Pinyin Enhanced Generator (Wang et al., 2021).

**SpellBert** employs a graph neural network to introduce visual and phonetic features (Ji et al., 2021).

**GAD** learns the global relationships of the potential correct input characters and the candidates of potential error characters (Guo et al., 2021).

**BERT** We also implement classical methods for comparison. We fine-tune the Chinese BERT model (Devlin et al., 2018) on the CGEC corpus directly.

## 4.3 Hyperparameter Setting

We follow most of the standard hyperparameters for transformers in the base configuration (Vaswani et al., 2017) and follow the weight initialization scheme from BERT (Devlin et al., 2018). For regularization, we use 0.3 dropout, 0.01 L2 weight decay. The hyperparameter  $\gamma$  which is used to weight the confusion loss is set to 2 after tuning. Adam optimizer (Kingma and Ba, 2014) with  $\beta = (0.9, 0.999)$ ,  $\epsilon = 1e^{-6}$  is used to conduct the parameter learning. The learning rate is set to  $5e^{-5}$ , and the model is trained with learning rate warming up and linear decay.

## 5 Results and Analysis

### 5.1 Overall Performance

Table 1 reports the performance of our proposed EGCM model and baseline models on the SIGHAN15 test set. For a fair comparison, we also employ the pre-trained model cBERT (Liu et al., 2021) which has the same architecture with BERT and pre-trained via the confusion set based masking strategy. Our model with pretrained cBERT (Pre-Tn EGCM) outperforms all existing approaches, achieving a 81.6 F1 at detection and 79.9 F1 at correction. Compared with the BERT baseline, Pre-Tn EGCM achieves 5.5% performance gain on detection F1 and 6.5% gain on correction F1. Among un-pretrained methods, EGCM also outperforms all competitor models by a wide margin.

We also evaluate the model performance using the official tool, and report the results in Table 2. Our model Pre-Tn EGCM obtains the best results for both detection and correction. Especially, it greatly outperforms previous methods in precision.

It should be emphasized that, our model EGCM is trained on 270k HybridSet and outperforms several models that are pre-trained on a big size of synthetic data, such as PLOME (Liu et al., 2021) which is pre-trained using 162 million sentences. This demonstrates that our model effectively learns to correct spelling errors without relying on heavy-weight data. An example output of our EGCM comparing with BERT is listed in Appendix B.

### 5.2 Ablation Study

We explore the contribution of each component in our EGCM model by conducting ablation studies with the following settings: (1) Removing the error-focused encoder mentioned in 3.2. (2) Removing

<sup>1</sup><https://github.com/ACL2020SpellGCN/SpellGCN>

<sup>2</sup><http://nlp.ee.ncu.edu.tw/resource/csc.html>

Method	Detection Level				Correction Level			
	Acc.	Pre.	Rec.	F1.	Acc.	Pre.	Rec.	F1
Confusionset (2019)	-	66.8	73.1	69.8	-	71.5	59.5	64.9
FASpell (2019)	74.2	67.6	60.6	63.5	73.7	66.6	59.1	62.6
SpellGCN (2020)	-	74.8	80.7	77.7	-	72.1	77.7	75.9
Chunk2020 (2020)	76.8	88.1	62.0	72.8	74.6	87.3	57.6	69.4
SM BERT (2020)	80.9	73.7	73.2	73.5	77.4	66.7	66.2	66.4
RoBERTa-DCN (2021)	-	76.6	79.8	78.2	-	74.2	77.3	75.7
ECSpell (2022)	83.4	76.4	79.9	78.1	82.4	74.4	77.9	76.1
PLOME* (2021)	-	77.4	81.5	79.4	-	75.3	79.3	77.2
REALISE* (2021)	84.7	77.3	81.3	79.3	84.0	75.9	79.9	77.8
PHMOSpell* (2021)	-	<b>90.1</b>	72.7	80.5	-	<b>89.6</b>	69.2	78.1
MLM-phonetics* (2021)	-	77.5	<b>83.1</b>	80.2	-	74.9	<b>80.2</b>	77.5
GAD* (2021)	-	75.6	80.4	77.9	-	73.2	77.8	75.4
SpellBert* (2021)	-	87.5	73.6	80.0	-	87.1	71.5	78.5
BERT-finetune	82.4	74.2	78.0	76.1	81.0	71.6	75.3	73.4
Our EGCM	86.4	82.7	77.6	80.0	85.8	80.6	74.7	77.5
Pre-Tn EGCM*	<b>87.2</b>	83.4	79.8	<b>81.6</b>	<b>86.3</b>	81.4	78.4	<b>79.9</b>

Table 1: Performance on the SIGHAN15 test set. Best results are in bold. The first group lists the models that are not pretrained, and the second group lists the methods that are pretrained (denoted with "\*" ).

Method	Detection level			Correction level		
	Pre	Rec	F1	Pre	Rec	F1
SpellGCN	85.9	<b>80.6</b>	83.1	85.4	77.6	81.3
ECSpell	85.7	78.4	81.9	85.4	76.6	80.7
TiT	85.4	78.1	81.6	85.0	75.6	80.0
GAD	86.0	80.4	83.1	85.6	<b>77.8</b>	81.5
Pre-Tn EGCM	<b>93.5</b>	76.7	<b>84.3</b>	<b>91.4</b>	74.5	<b>82.1</b>

Table 2: Performance on the SIGHAN15 test evaluated by the official tools. Best results are in bold.

Method	Detection Level			Correction Level		
	Pre.	Rec.	F1.	Pre.	Rec.	F1.
EGCM	82.7	77.6	80.0	80.6	74.7	77.5
- EFEnc	80.5	75.2	77.8	78.7	72.3	75.4
- CFL	79.2	72.7	75.8	77.3	69.9	73.4
- GFI	77.5	76.9	77.2	77.1	73.9	75.7

Table 3: Ablation study on SIGHAN15. "-EFEnc" means removing the error-focused encoder. "-CFL" means removing the confusion set loss. "-GFI" means not using *Guidance for Inference* in the inference stage.

the confusion set loss  $L_{cs}$  in equation 9. (3) Initialize the start sequence of inference with all [MASK] instead of using the *Guidance for Inference*. The results are shown in Table 3.

Specifically, the confusion set loss leads to the biggest improvement to our model with 4.2 points for detection and 4.1 points for correction. By removing the error-focused encoder, the drop of performance indicates that this encoder does learn to pay attention to the probably wrong tokens of the sentence and impel our model to correct the wrong tokens actively. Also, without the use of *Guidance for Inference* as the start of decoding for

top-k	$P_{error\&mask/error}$	$P_{correct/unmask}$
k=1	94%	99.8%
k=2	90%	99.7%
k=3	88%	99.7%

Table 4: An evaluation on Zero-shot error detection.

inference, the performance drops especially on precision, which indicates that by fixing the tokens that are correct can effectively avoid over-correction and improve precision.

### 5.3 Evaluation on Zero-shot Error Detection

We employ a zero-shot detection approach to do a preliminary detection, in which all the tokens are divided into two groups, the probably wrong tokens and the probably correct one. In the inference stage, the probably correct ones are unmasked and will not be modified to avoid over-correction, while the probably wrong ones are masked and repredicted. We want to ensure that unmasked tokens are truly correct that don't need to be modified, and at the same time, the errors in the sentences are masked as many as possible.

As shown in Table 4,  $P_{error\&mask/error}$  denotes the percentage of errors that are masked,  $P_{correct/unmask}$  denotes the percentage of truly correct tokens in the unmasked tokens. In our zero-shot error detection, the BERT predicted tokens with top- $k$  probabilities are selected as candidates, if the original token is not in the candidates list, it is considered as wrong. We try different  $k$  and conduct experiments. Our method achieves promising results with high accuracy, which guarantees cor-

Confusion set	Method	Detection level			Correction level		
		Pre.	Rec.	F1.	Pre.	Rec.	F1.
[1] (2022)	ECSpell	76.4	79.9	78.1	74.4	77.9	76.1
	EGCM	82.7	77.6	80.0	80.6	74.7	77.5
[2] (2013)	SpellGCN	74.8	80.7	77.7	72.1	77.7	75.9
	EGCM	80.6	78.2	79.4	78.2	76.3	77.2
	PLOME*	77.4	81.5	79.4	75.3	79.3	77.2
[3] (2018)	Pre-Tn EGCM*	81.6	79.6	80.6	79.8	76.4	78.1
	Confusionset EGCM	66.8	73.1	69.8	71.5	59.5	64.9
		79.5	74.7	77.0	77.4	71.7	74.4

Table 5: Effects of different confusion sets.

rect signals for further processing. Obviously, the smaller  $k$  is, the more tokens are masked and less tokens are fixed, this might lead to over-correction. We want the errors are masked as many as possible, and at the same time, fewer tokens are masked. Therefore, in our model, we set  $k = 2$ .

#### 5.4 Analysis on Different Confusion Sets

To further prove the effectiveness of the confusion loss we proposed, and to show that this loss function can be generalized, we conduct experiments on three different confusion sets, including the confusion set proposed by Lv et al. (2022)<sup>3</sup>, Wu et al. (2013)<sup>4</sup>, and Wang et al. (2018)<sup>5</sup>. For each confusion set, we compare our model with the models that use the same confusion set but in different way.

As shown in table 5. For all three confusion sets, our model outperforms the model that utilizes the same confusion set. Compared with previous methods, our model takes every token in the confusion set into consideration by computing its possibility of being misused. Moreover, the results indicate that our model has strong generalization ability and is not limited to any specific confusion set.

#### 5.5 Analysis on Decoding Iterations

With a predefined decoding iteration  $T = 10$ , we show the F1 score of previous iterations  $t(t < T)$  to illustrate how the mask-predict strategy detects and corrects the wrong tokens step by step. As shown in figure 6, the F1 score of detection and correction improves as the decoding iteration goes up. This indicates that, by masking and repredicting the tokens of low probability in each iteration, our model corrects the tokens that are wrongly predicted during previous iterations. And as the number of unmasked tokens increases, more informa-

<sup>3</sup><https://github.com/Aopolin-Lv/ECSpell>

<sup>4</sup><http://nlp.ee.ncu.edu.tw/resource/csc.html>

<sup>5</sup><https://github.com/wdimmy/Automatic-Corpus-Generation>

Model	Time(ms)	Speedup
Transformer (2017)	63ms	1x
PLOME (2021)	45ms	1.4x
REALISE (2021)	17ms	3.7x
TtT (2021)	15ms	4.2x
EGCM	10ms	6.3x

Table 6: Comparisons of the computing efficiency.

tion is given to help predict the hard masked tokens. With 8 iterations, our model achieves state-of-the-art performance.

#### 5.6 Analysis on Computing Efficiency

Chinese spelling correction can be applied in many real-life applications, such as writing assistant and search engine. Therefore, the time cost efficiency of models is a key point to be considered. We implement both the baseline models and our model on the single NVIDIA RTX 2080 GPU. Table 6 depicts the time cost per sample of our model comparing with some previous approaches. Our model runs faster than all previous approaches.

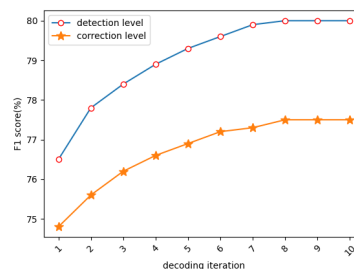


Figure 6: Results of different decoding iterations

## 6 Conclusion

We propose an error-guided correction model for the CSC task. A zero-shot error detection method is proposed to provide guidance signals for training and inference. We apply a conditional masked language model as a backbone, where we improve the encoder-decoder architecture by adding an error-focused encoder which pushes our model to focus on the wrong tokens. During training, we introduce a new confusion loss to help the model distinguish similar tokens. During inference, the error-guided mask-predict decoding strategy is adopted to mask and repredict the tokens that are probably wrong. Experimental results show that our model not only achieves superior performance against state-of-the-art approaches but also is cost-saving and green.



## Limitation

In this paper, we use the results from zero-shot spelling error detection as a guidance signal. The sentence with probably wrong tokens masked and the other tokens fixed are used as a start of decoding. This means that if a wrong token is not assigned with a [MASK] token, it will never be corrected in later iterations. Even though we conduct experiments and the result shows that up to 94% of the wrong tokens are masked in the guidance signal, there are still some wrong tokens missed by our model. To limit the number of tokens that are free to be modified is one of our ways to improve precision, but we are also looking forward to a way to further improve recall.

What's more, even though we make full use of the confusion set, we still think that's not enough. Now we are using the confusion set in which every token has a set of predefined similar tokens. And these sets of similar tokens are isolated with each other. However, Chinese has various kinds of spelling errors, the target token might not be in the predetermined similar tokens set of the original token. And this kind of mistakes can never be learned to correct by the model. We think a better design for the data structure of the confusion set needs to be proposed, in which the sets are not isolated and we are able to calculate the similarity distance between each pair of tokens using particular algorithms, for example, UnionFind on a dynamic Graph. This kind of dynamic confusion knowledge can help avoid ignoring the probably misused tokens.

## Acknowledgement

This work is supported by the National Hi-Tech RD Program of China (No.2020AAA0106600), the National Natural Science Foundation of China (62076008) and the Key Project of Natural Science Foundation of China (61936012).

## References

Haithem Afli, Zhengwei Qiu, Andy Way, and Páiraic Sheridan. 2016. [Using smt for ocr error correction of historical texts](#). In *LREC*. European Language Resources Association (ELRA).

Z. Bao, C. Li, and R. Wang. 2020. Chunk-based chinese spelling check with global optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang, and Hsin-Hsi Chen. 2013. [A study of language modeling for chinese spelling check](#). In *SIGHAN@IJCNLP*, pages 79–83. Asian Federation of Natural Language Processing.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. [Spellgen: Incorporating phonological and visual similarities into language models for chinese spelling check](#). *CoRR*, abs/2004.14166.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016. [Neural network translation models for grammatical error correction](#). *CoRR*, abs/1606.00189.

Alexis Conneau and Guillaume Lample. 2019. [Cross-lingual language model pretraining](#). In *NeurIPS*, pages 7057–7067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [Bert: Pre-training of deep bidirectional transformers for language understanding](#). Cite arxiv:1810.04805.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. [Mask-predict: Parallel decoding of conditional masked language models](#). In *EMNLP/IJCNLP (1)*, pages 6111–6120. Association for Computational Linguistics.

Zhao Guo, Yuan Ni, Keqiang Wang, Wei Zhu, and Guotong Xie. 2021. Global attention decoder for chinese spelling error correction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1419–1428.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97.

Y. Hong, X. Yu, N. He, N. Liu, and J. 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)*.

L. Huang, J. Li, W. Jiang, Z. Zhang, and J. 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)*.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. [A nested attention neural hybrid model for grammatical error correction](#). *CoRR*, abs/1707.02026.

- 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.
- D. Kingma and J. Ba. 2014. Adam: A method for stochastic optimization. *Computer Science*.
- P. Li and S. Shi. 2021. Tail-to-tail non-autoregressive sequence prediction for chinese grammatical error 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)*.
- C. L. Liu, M. H. Lai, Y. H. Chuang, and C. Y. Lee. 2010. Visually and phonologically similar characters in incorrect simplified chinese words. In *COLING 2010, 23rd International Conference on Computational Linguistics, Posters Volume, 23-27 August 2010, Beijing, China*.
- 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.
- Q. Lv, Z. Cao, L. Geng, C. Ai, X. Yan, and G. Fu. 2022. General and domain adaptive chinese spelling check with error consistent pretraining.
- Bruno Martins and Mário J. Silva. 2004. **Spelling correction for search engine queries**. In *EstAL*, volume 3230 of *Lecture Notes in Computer Science*, pages 372–383. Springer.
- Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen. 2015. **Introduction to sighthan 2015 bake-off for chinese spelling check**. In *SIGHAN@IJCNLP*, pages 32–37. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. **Attention is all you need**. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, page 5998–6008. Curran Associates, Inc.
- Baoxin Wang, Wanxiang Che, Dayong Wu, Shijin Wang, Guoping Hu, and Ting Liu. 2021. Dynamic connected networks for chinese spelling check. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2437–2446.
- 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.
- Dingmin Wang, Yi Tay, and Li Zhong. 2019. **Confusionset-guided pointer networks for chinese spelling check**. In *ACL (1)*, pages 5780–5785. Association for Computational Linguistics.
- Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013. **Chinese spelling check evaluation at sighthan bake-off 2013**. In *SIGHAN@IJCNLP*, pages 35–42. Asian Federation of Natural Language Processing.
- H. D. Xu, Z. Li, Q. Zhou, C. Li, and X. L. Mao. 2021. Read, listen, and see: Leveraging multimodal information helps chinese spell checking.
- Junjie Yu and Zhenghua Li. 2014. **Chinese spelling error detection and correction based on language model, pronunciation, and shape**. In *CIPS-SIGHAN*, pages 220–223. Association for Computational Linguistics.
- Ruiqing Zhang, Chao Pang, Chuanqiang Zhang, Shuo-huan Wang, Zhongjun He, Yu Sun, Hua Wu, and Haifeng Wang. 2021. Correcting chinese spelling errors with phonetic pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2250–2261.
- Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. **Spelling error correction with soft-masked bert**. In *ACL*, pages 882–890. Association for Computational Linguistics.

## A Statistics of Datasets

Training Set	#Sent	Avg.Length	#Errors
SIGHAN13	700	41.8	343
SIGHAN14	3437	49.6	5122
SIGHAN15	2338	31.3	3037
Wang271K	271329	42.6	381962
Test Set	#Sent	Avg.Length	#Errors
SIGHAN15	1100	30.6	704

Table 7: Statistics of used datasets.

## B Case Study

We list several cases of Chinese Spelling Correction. We present the source wrong sentence and the target correct sentence. We also present the corrections made by BERT and our EGCM.

Type	Sample
src : (wrong sentence)	大明说: 请座, 请座. Daming said " please <b>seat</b> down"
tgt : (correct sentence)	大明说: 请坐, 请坐. Daming said " please sit down"
BERT prediction	大明说: 请座, 请座. Daming said " please <b>seat</b> down"
EGCM prediction	大明说: 请坐, 请坐. Daming said " please sit down"
src : (wrong sentence)	可是你现在不在宿舍, 所以我留了一枝条. But you are not in the dormitory, so I leave a <b>branch</b> .
tgt : (correct sentence)	可是你现在不在宿舍, 所以我留了一纸条. But you are not in the dormitory, so I leave a note.
BERT prediction	可是你现在不在宿舍, 所以我留了一只条. But you are not in the dormitory, so I leave a <b>piece of slip</b> .
EGCM prediction	可是你现在不在宿舍, 所以我留了一纸条. But you are not in the dormitory, so I leave a note.
src : (wrong sentence)	我以前想要高诉你, 可是我忘了, 我真户秃. I wanted to <b>accuse you highly</b> , but I forgot, I was so <b>bald</b> .
tgt : (correct sentence)	我以前想要告诉你, 可是我忘了, 我真糊涂. I wanted to tell you, but I forgot. I was so muddled.
BERT prediction	我以前想要告诉你, 可是我忘了, 我真户秃. I wanted to tell you, but I forgot. I was so <b>bald</b> .
EGCM prediction	我以前想要告诉你, 可是我忘了, 我真糊涂. I wanted to tell you, but I forgot. I was so muddled.

Figure 7: Case Study. The wrong tokens are marked red.