uChecker: Masked Pretrained Language Models as Unsupervised Chinese Spelling Checkers

Piji Li
College of Computer Science and Technology,
Nanjing University of Aeronautics and Astronautics
MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
Nanjing, Jiangsu, China
pjli@nuaa.edu.cn

Abstract

The task of Chinese Spelling Check (CSC) is aiming to detect and correct spelling errors that can be found in the text. While manually annotating a high-quality dataset is expensive and time-consuming, thus the scale of the training dataset is usually very small (e.g., SIGHAN15[1] only contains 2339 samples for training), therefore supervised-learning based models usually suffer the data sparsity limitation and over-fitting issue, especially in the era of big language models. In this paper, we are dedicated to investigating the **unsupervised** paradigm to address the CSC problem and we propose a framework named **uChecker** to conduct unsupervised spelling error detection and correction. Masked pretrained language models such as BERT are introduced as the backbone model considering their powerful language diagnosis capability. Benefiting from the various and flexible MASKing operations, we propose a Confusionset-guided masking strategy to fine-train the masked language model to further improve the performance of unsupervised detection and correction. Experimental results on standard datasets demonstrate the effectiveness of our proposed model uChecker in terms of character-level and sentence-level Accuracy, Precision, Recall, and F1-Measure on tasks of spelling error detection and correction respectively.

1 Introduction

Chinese Spelling Check (CSC) is a crucial and essential task in the area of natural language processing. It aims to detect and correct spelling errors in the Chinese text (Chang, 1995; Wang et al., 2020b). Generally, sequence translation (Wang et al., 2018; Ge et al., 2018; Wang et al., 2019, 2020a; Kaneko et al., 2020) and sequence tagging (Omelianchuk et al., 2020; Liang et al., 2020; Mallinson et al., 2020; Parnow et al., 2021) are the two most typical technical paradigms to tackle the problem. Benefiting from the development of pretraining techniques, many researchers fine-tune the pretrained language models such as BERT (Devlin et al., 2019) on the task of CSC and obtain encouraging performance (Zhao et al., 2019; Hong et al., 2019; Zhang et al., 2020; Liu et al., 2021; Li et al., 2021; Huang et al., 2021; Guo et al., 2021; Zhang et al., 2021; Li and Shi, 2021; Dai et al., 2022). Meanwhile, it should be emphasized that almost all of the above mentioned models are trained via the **supervised learning** paradigm.

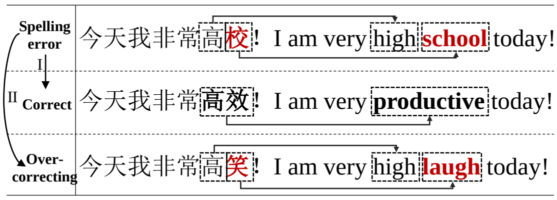
Spelling error
I
II
Correct
Over-correcting
今天我非常高校！ I am very high school today!
今天我非常高效！ I am very productive today!
今天我非常高笑！ I am very high laugh today!

Figure 1: Illustration for the task of Chinese spelling check (operation path I) as well as the over-fitting phenomenon existing in the current supervised learning based models (operation path II).

However, during the investigating stage about those newly typical state-of-the-art models, we observe some spiny and serious phenomenons: (1) Occasionally those models may generate some special **over-correcting** results. As shown in Figure 1, operation path **I** is the regular spelling error detection and correction path, while operation path **II** is also observable in the inference stage where the models can detect the errors correctly but rectify them using some other error tokens in the correction stage. (2) The spelling error detection and correction performance will drop dramatically when those models did not see the spelling error cases in the training dataset or the text are from different genres and domains. This issue tells us that the **generalization capability** of those models are limited and need to be enhanced.

Then what are the causes of these phenomenons?

[1] http://ir.itc.ntnu.edu.tw/lre/sighan8csc.html

Proceedings of the 29th International Conference on Computational Linguistics, pages 2812–2822
October 12–17, 2022.

Corpus	#Train	#ErrTrain	AvgLen	#Test	#ErrTest	AvgLen
SIGHAN13	350	350	49.2	1,000	996	74.1
SIGHAN14	6,526	3,432	49.7	1,062	529	50.1
SIGHAN15	3,174	2,339	30.0	1,100	550	30.5

Table 1: Statistics of the SIGHAN series datasets.

Since some of the models are already strong enough (which are constructed based on big pretrained models), then we shift our eyes to the data perspective. In real practical scenarios, natural human-labeled spelling error corpus are difficult and expensive to obtain. Although some works such as Wang et al. (2018) employ OCR and ASR based techniques to automatically synthetic the paired samples by replacing the correct tokens using visually or phonologically similar characters, obviously, the constructed data is unrealistic and far from the real and objective scenarios. Therefore, actually, the scale of the typical corpus for the task of Chinese spelling check is very small. Considering that almost all the research works have used SIGHAN series datasets (Tseng et al., 2015) to train and evaluate their algorithms, we conduct counting on those three corpora, and the statistics results are shown in Table 1. From the results we can observe that there are only 2k∼3k sentences with spelling errors in the training dataset and really far from the practical requirements.

Thus, sticking to train the supervised learning models based on those scale-limited resources might not be a wise direction. Therefore, in this paper, we are dedicated to exploring **unsupervised** frameworks to conduct Chinese spelling error detection and correction. Fortunately, masked pretrained language models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), etc. can satisfy the needs of detecting and correcting spelling errors in an unsupervised manner. First, the masked training strategy is naturally a convenient and perfect shortcut for us to conduct token-grained detection and correction. For example, we can mask any token and predict it based on the bi-directional context to see if the current token is appropriate or not. Second, the pretrained language models are usually trained using large-scale corpora, thus the language diagnosis capability is very strong. Intuitively, these models can also guarantee the generalization capability considering the corpora may contain text from a wide range of domains and genres.

Therefore, based on the masked pretrained language models, we propose a framework named **uChecker** to conduct unsupervised spelling error detection and correction respectively. uChecker is a two-stage framework and it will detect the text token-by-token first and then correct the abnormal tokens. Models such as BERT are introduced as the backbone model. Inspired by the previous works (Wang et al., 2019; Liu et al., 2021), benefiting from the various and flexible masking operations, we also introduce a confusionset-guided masking strategy to fine-train the masked language model to further improve the performance of unsupervised detection and correction. Though uChecker is a two-stage framework, we design an elegant method to let the information pass BERT only once to guarantee the time efficiency. Moreover, interestingly, in unsupervised settings, we experimentally find the performance of error detection is crucial to the global and general performance. It means that the correction capability of the pretrained language models are strong enough, then the key-point is how to improve the performance of detection. Therefore, in uChecker, we also design several algorithms to improve the performance of seplling error detection. Yasunaga et al. (2021) employ GPT2-like models to conduct unsupervised English grammatical error correction which also verifies the feasible of our direction.

In summary, our contributions are as follows:

- We propose an unsupervised framework named **uChecker** to conduct Chinese spelling error detection and correction.
- Benefiting from flexible masking operations, we introduce s confusionset-guided masking strategy to fine-train BERT to further improve the performance.
- We experimentally find that error detection is crucial to the global SCS performance. Therefore, we also design some algorithms to improve the capability of error detection.
- Extensive experiments on several benchmark datasets demonstrate the effectiveness of the proposed approach. And the results also show that uChecker can even outperform some strong supervised models.

Rate	Strategy	今天我非常高**效**！
20%:80%	<MASK>	今天我非常高<MASK>！
20%:10%	Random	今天我非常高**语**！
20%:10%	Unchange	今天我非常高**效**！

Figure 2: Illustration of the masking strategies of BERT during the pretraining stage.

2 Background: Masked Language Models

BERT (Devlin et al., 2019) is the most typical masked language model, and it is regarded as the backbone model of our proposed framework uChecker, therefore in this section we introduce the technical details of this model, especially the masking strategies.

BERT is constructed based on the model of Transformer (Vaswani et al., 2017). After preparing the input samples, an embedding layer and a stack of Transformer layers are followed to conduct the bi-directional semantic modeling. Specifically, for the input, we first obtain the representations by summing the word embeddings with the positional embeddings:

$$\mathbf{H}_t^0 = \mathbf{E}_{w_t} + \mathbf{E}_{p_t} \tag{1}$$

where 0 is the layer index and t is the state index. $\mathbf{E}_w$ and $\mathbf{E}_p$ are the embedding vectors for tokens and positions, respectively. Then the obtained embedding vectors $\mathbf{H}^0$ are fed into several Transformer layers. Multi-head self-attention is used to conduct bidirectional representation learning:

$$\begin{aligned}
\mathbf{H}_t^1 &= \textsc{Ln}\left(\textsc{Ffn}(\mathbf{H}_t^1) + \mathbf{H}_t^1\right) \\
\mathbf{H}_t^1 &= \textsc{Ln}\left(\textsc{Slf-Att}(\mathbf{Q}_t^0, \mathbf{K}^0, \mathbf{V}^0) + \mathbf{H}_t^0\right) \\
\mathbf{Q}^0 &= \mathbf{H}^0\mathbf{W}^Q \\
\mathbf{K}^0, \mathbf{V}^0 &= \mathbf{H}^0\mathbf{W}^K, \mathbf{H}^0\mathbf{W}^V
\end{aligned} \tag{2}$$

where $\textsc{Slf-Att}(\cdot)$, $\textsc{Ln}(\cdot)$, and $\textsc{Ffn}(\cdot)$ represent self-attention mechanism, layer normalization, and feed-forward network respectively (Vaswani et al., 2017). After L Transformer layers, we obtain the final output representation vectors $\mathbf{H}^L \in \mathbb{R}^{T\times d}$, where T is the input sequence length and d is the vector dimension.

The masking strategies used in BERT are shown in Figure 2. There are 20% tokens will be masked, and among them there are 80% tokens are replaced with a special symbol such as <MASK>, and 10% are replaced with a random token, and the left 10% keep unchanged. What should be emphasized here is the ***random replacing operation*** which plays an crucial role in the following model designs about the unsupervised detection and correction as well as the confusionset-guided fine-training.

Finally, a linear function g with $\mathrm{softmax}$ activation is used to predict the masked token x_t via:

$$p\left(x_t | x_{\le t-1}, x_{\ge t+1}\right) = \mathrm{softmax}\left(g\left(\mathbf{h}_t\right)\right) \tag{3}$$

3 The Proposed uChecker Framework

3.1 Overview

Figure 3 depicts the basic components of our proposed framework uChecker. The backbone model is masked language model, say BERT. Note that all the parameters of BERT are frozen. Input is an incorrect sentence $X = (x_1, x_2, \ldots, x_T)$ which contains spelling errors, where x_i denotes each token (Chinese character) in the sentence, and T is the length of X. The objective of the task Chinese spelling check is to detect and correct all errors in X and obtain a new sentence $Y = (y_1, y_2, \ldots, y_{T'})$. Benefiting from the various and flexible masking operations, we introduce the confusionset-guided masking strategy to fine-train BERT to further improve the performance of CSC. We also design several algorithms such as unsupervised detection (UnsupDetection), supervised detection (SupDetection), and Ensemble of UnsupDetection and SupDetection to improve the performance of error detection to further improve the overall performance.

3.2 Unsupervised Spelling Error Detection

Given a pretrained BERT model, for a sentence $X = (x_1, x_2, \ldots, x_T)$ which need to be checked, the preconceived first guess is to mask the tokens one each time from left to right diagonally, as shown in Figure 4, and then input the masked sequence X' into BERT to conduct prediction. Assume token x_t is masked, the predicted distribution at position t is $p(x'_t)$, then we can obtain the probability of the corresponding original input token x_t by:

$$p^u_{x_t} = p_{x'_t = x_t}(x'_t) \tag{4}$$

Intuitively, if x_t is just the correct token, then $p^u_{x_t}$ will be very large (say 0.99). Otherwise, error may hide in this position. Therefore, the simple

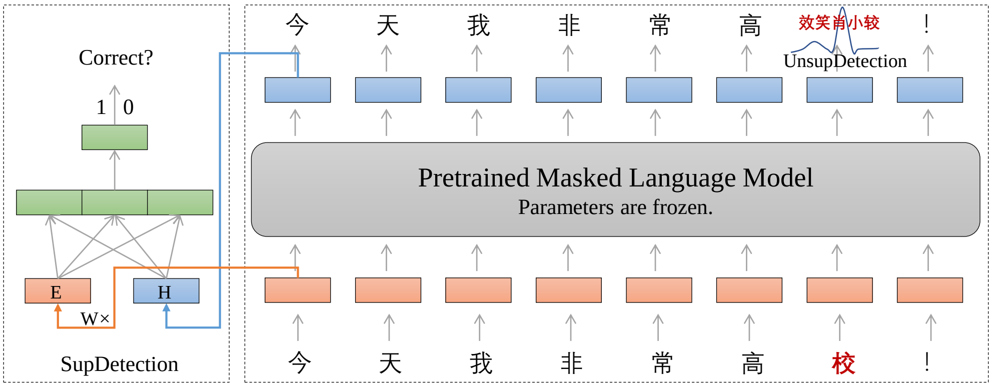

Figure 3: The proposed uChecker framework for unsupervised Chinese spelling error detection and correction.

<CLS><MASK>天我非常高校！ <SEP>
<CLS>今<MASK>我非常高校！ <SEP>
<CLS>今天<MASK>非常高校！ <SEP>
······
<CLS>今天我非常高<MASK>！ <SEP>
<CLS>今天我非常高校<MASK> <SEP>

Figure 4: Illustration of the first guess **diagonal masking strategy** for unsupervised error detection and correction, which is actually **time-consuming and not necessary**.

approach to conduct detection is to find a threshold θ^u and diagnose the results by:

$$error_t = \begin{cases} 1, & \text{if } p^u_{x_t} < \theta^u \\ 0, & otherwise \end{cases} \tag{5}$$

where $error_t = 1$ means that token x_t is not correct in sentence X.

However, although diagonal masking strategy is a natural way to conduct token diagnose, considering that for each sentence with length T, we need prepare T sequences to feed BERT to conduct prediction, which is a time-consuming procedure with low efficiency, and it is also difficult to be deployed and executed concurrently. Recall the masking strategies shown in Figure 2, besides replacing the tokens with <MASK> symbol, *BERT also uses* ***random tokens*** *to conduct masking*. Therefore, we do not need to rigidly obey the <MASK> based masking approach. On the contrary, we can briefly regard the potential error tokens as the random masking strategy and just feed the original input sentence X into the BERT model to conduct probability estimation. And this approach can execute with high concurrency because we can process hundreds or even thousands of sentences in a batch based on the parallel computing capability of GPUs. Moreover, since random masking is feasible, then *how about conduct random masking using the corresponding tokens from the* ***confusionset***? This is the **inspiration** of confusionset-guided fine-training strategy which will be introduced in the following sections.

For each sentence X, the spelling error detection stage will return a list containing the indices of the wrong tokens $\mathcal{I}^e = [2, 5, i, \dots]$, ranked by the predicted corresponding probability in an ascending order. The order indicates that the most worse token will be corrected firstly.

3.3 Unsupervised Spelling Error Correction

Given the detected wrong token indices $\mathcal{I}^e$, the unsupervised error correction component then scans the list and chooses the most appropriate tokens from the probability distribution to conduct correction. Specifically, for any index $i \in \mathcal{I}^e$, the predicted distribution is $p(x'_i)$, then we can straightforwardly select the token with the largest score as the correct result:

$$x_j = \text{argmax}_j \ p_{x'_i=x_j}(x'_i) \tag{6}$$

The operation of unsupervised spelling error correction is simple, therefore the correction performance will completely depend on the capability of the pretrained backbone language models.

Confusionset-guided Token Selection Due to the special input methods such as Pinyin and Wubi, many Chinese characters are similar either in phonology or morphology. There are about 76% of Chinese spelling errors belong to phonological similarity error and 46% belong to visual similarity error (Liu et al., 2011). Intuitively, incorporating the Confusionset with the token selection procedure may improve the performance. Therefore, we

Algorithm 1: Confusionset-guided Token Correction

```
Data: Wrong token indices $\mathcal{I}^e$; The
      predicted distributions for all the
      positions $\mathcal{P}$; The predefined
      confusionset $\mathcal{C}$ (hashmap<string,
      list>).
Result: The correct tokens $Y$ for $\mathcal{I}^e$.
Y = [] ;
for $i \in \mathcal{I}^e$ do
    $W_i$ = top_k($\mathcal{P}_i$);
    for $w_i \in W_i$ do
        if $w_i \in \mathcal{C}(x_i)$ then
            Y.insert($w_i$);
            break;
        end
        Y.insert($W_i[0]$);
    end
end
```

further build a simple confusionset-guided token selection approach as shown in Algorithm 1.

Specifically, for each detected index i, we fist fetch the top_k tokens according to the distribution $\mathcal{P}_i$). Then if the top_k tokens are also from the corresponding confusionset, then we get the result. Otherwise, we still select the best predicted result.

3.4 Self-Supervised Spelling Error Detection

Surprisingly and interestingly, during the experiments stage, we find that **the performance of error detection plays an crucial role** in affecting the global checking performance. Therefore, improving the capability of error detection can benefit the whole system. But, there is a precondition that we cannot adjust the original backbone BERT parameters because unsupervised error correction is the essential component of our uChecker framework, and we do not want to let the BERT parameters collapse to some special areas or domains. So the BERT parameters need to be frozen when designing the error detection strategies.

As shown in Figure 3, after examining the model carefully, we create a **smart but straightforward self-supervised detection method** to tackle the problem. The basic observation is that, based on the masked language models, the information in the output hidden states $\mathbf{H}$ will be more closer to the true tokens because the model will use them ($\mathbf{H}$) to conduct masked prediction (Eq. 3). Moreover, the information in the embedding layer $\mathbf{E}$ also contains the token information. Then we assume that for any correct token x_i and error token x_j, pair $(\mathbf{e}_i, \mathbf{h}_i)$ for normal token holds a more tight relationship than the pair $(\mathbf{e}_j, \mathbf{h}_j)$ for wrong token, where $\mathbf{e}$ is the learnt token embedding and $\mathbf{h}$ is the output layer of BERT:

$$\mathcal{M}(\mathbf{e}_i, \mathbf{h}_i) > \mathcal{M}(\mathbf{e}_j, \mathbf{h}_j) \tag{7}$$

where $\mathcal{M}$ is metric to represent the interaction relationship, and here we use the following calculations to conduct the interaction modeling:

$$\mathbf{h}_i^s = \mathbf{W}_s(\mathbf{e}_i^{\prime}; \mathbf{h}_i; \mathbf{e}_i^{\prime} \odot \mathbf{h}_i; |\mathbf{e}_i^{\prime} - \mathbf{h}_i|) + \mathbf{b}_s \tag{8}$$

where ; is the concatenation operation and $\mathbf{e}_i^{\prime}$ is a transformation of $\mathbf{e}_i$ using:

$$\mathbf{e}_i^{\prime} = \mathbf{W}_e(\mathbf{e}_i) + \mathbf{b}_e \tag{9}$$

This transformation is **essential and cannot be ignored** because that $\mathbf{e}$ and $\mathbf{h}$ are in different vector space. *Otherwise the training will not converge*.

We use cross entropy as the optimization objective:

$$\begin{aligned} \mathbf{y}_i^s &= softmax(\mathbf{h}_i^s) \\ \mathcal{L}^{\mathrm{s}} &= -\sum_{i=0}^{1} \log P^{\mathrm{s}}(\mathbf{y}_i^t | \mathbf{h}_i^s) \end{aligned} \tag{10}$$

For the self-supervised learning, we still employ the masked training strategy to conduct training, where we assign the label for random masking is 1 (position with errors), and 0 for those unchanged positions. Let $p_{x_t}^s = \mathbf{y}_t^s[1]$ be the self-supervised probability of error, then we also set up a threshold θ^s to conduct diagnose as well:

$$error_t = \begin{cases} 1, & \text{if } p_{x_t}^s >= \theta^s \\ 0, & otherwise \end{cases} \tag{11}$$

where $error_t = 1$ means that token x_t is not correct in sentence X.

Note that BERT parameters are **frozen** during the self-supervised learning procedure, therefore we only conduct optimization for a small group of parameters in Eq. 8 and Eq. 9, which is a light-scale training stage.

3.5 Ensemble Detection Methods

Obvious, we can collect all the detected error positions ($\mathcal{I}_u^e$ and $\mathcal{I}_s^e$) by unsupervised and self-supervised detectors respectively, which we name it ensemble detection operation.

TestSet	Model	Detection			Correction		
		PREC.	REC.	F1	PREC.	REC.	F1
SIGHAN13	**Supervised Methods**						
	LMC (Xie et al., 2015)	79.8	50.0	61.5	77.6	22.7	35.1
	Hybird (Wang et al., 2018)	54.0	69.3	60.7	-	-	52.1
	Confusionset (Wang et al., 2019)	66.8	73.1	69.8	71.5	59.5	69.9
	SpellGCN (Cheng et al., 2020)	82.6	88.9	85.7	98.4	88.4	93.1
	Unsupervised Methods						
	uChecker (Sec.3)	81.6	**93.0**	**86.9**	95.8	93.1	94.4
	w/o self-supervised detection	83.3	90.3	86.7	96.6	**93.2**	**96.8**
	w/o confusionset	**84.3**	89.0	86.6	89.8	86.4	88.1
SIGHAN14	**Supervised Methods**						
	LMC (Xie et al., 2015)	56.4	34.8	43.0	71.1	50.2	58.8
	Hybird (Wang et al., 2018)	51.9	66.2	58.2	-	-	56.1
	Confusionset (Wang et al., 2019)	63.2	82.5	71.6	79.3	68.9	73.7
	SpellGCN (Cheng et al., 2020)	83.6	78.6	81.0	97.2	76.4	85.5
	Unsupervised Methods						
	uChecker (Sec.3)	75.9	73.3	74.6	91.7	**84.9**	85.0
	w/o self-supervised detection	72.4	66.1	69.2	92.9	81.4	**86.8**
	w/o confusionset	78.0	68.5	72.9	84.3	78.2	78.3
SIGHAN15	**Supervised Methods**						
	LMC (Xie et al., 2015)	56.4	34.8	43.0	71.1	50.2	58.8
	Hybird (Wang et al., 2018)	56.6	69.4	62.3	-	-	57.1
	Confusionset (Wang et al., 2019)	66.8	73.1	69.8	71.5	59.5	69.9
	SpellGCN (Cheng et al., 2020)	88.9	87.7	88.3	95.7	83.9	89.4
	PLOME (Liu et al., 2021)	94.5	87.4	90.8	97.2	84.3	90.3
	Unsupervised Methods						
	uChecker (Sec.3)	85.6	79.7	82.6	91.6	**84.8**	88.1
	w/o self-supervised detection	75.8	71.3	73.5	92.6	84.5	88.4
	w/o confusionset	87.4	75.9	81.2	84.6	77.7	81.0

Table 2: The character-level performance on both detection and correction level. *We notice that character-level detection performance of scrips from Hong et al. (2019) and Wang et al. (2019) are same. But the correction performance is different. Usually the scrip from Wang et al. (2019) is used to conduct the correction evaluation.

3.6 Confusionset-Guided Fine-Training

As mentioned in Section 3.2, inspired by the random masking strategy in the pretraining stage of BERT, we tailor design a confusionset-guided random masking strategy where the target token x_t will probably be replaced using its corresponding tokens in the confusionset $\mathcal{C}(x_t)$. The masking rate will also be adjusted slightly. Recently we find that confusionset-guided fine-training strategy has been deployed in some related works (Liu et al., 2021; Guo et al., 2021).

After fine-training, we can use the new BERT model to conduct self-supervised/unsupervised detection and unsupervised correction.

4 Experimental Setup

4.1 Settings

The core technical components of our proposed uChecker is a pre-trained Chinese BERT-base model (Devlin et al., 2019). The most important parameters in our framework are the two thresholds θ^u and θ^s and we set them to be 0.1 and 0.4 for unsupervised detection and supervised detection respectively. For the supervised error detection training, Adam optimizer (Kingma and Ba, 2015) is used to conduct the parameter learning and partial of the training dataset from SIGHAN series are employed as the trainset.

4.2 Datasets

The overall statistic information of the datasets used in our experiments are depicted in Table 1. As did in the previous works, we also conduct evaluation on those three datasets: SIGHAN13, SIGHAN14, and SIGHAN15 (Tseng et al., 2015)[2].

4.3 Comparison Methods

Considering that we did not notice some typical unsupervised methods with good results. Therefore, in this Section we introduce several classical and stage-of-the-art supervised approaches for comparisons.

HanSpeller++ employs Hidden Markov Model

[2] http://ir.itc.ntnu.edu.tw/lre/sighan8csc.html

TrainSet	Model	Detection				Correction			
		Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
SIGHAN13	**Supervised Methods**								
	FASPell (Hong et al., 2019)	-	76.2	63.2	69.1	-	73.1	60.5	66.2
	SpellGCN (Cheng et al., 2020)	-	80.1	74.4	77.2	-	78.3	72.7	75.4
	DCN (Wang et al., 2021)	-	86.8	79.6	83.0	-	84.7	77.7	81.0
	Unsupervised Methods								
	uChecker (Sec.3, Ours)	73.4	75.4	73.4	74.4	70.8	72.6	70.8	71.7
	w/o self-supervised detection	73.9	78.0	73.7	75.8	72.0	75.9	71.8	73.8
	w/o confusionset	73.5	78.2	73.3	75.7	65.5	69.4	65.1	67.2
SIGHAN14	**Supervised Methods**								
	FASPell (Hong et al., 2019)	-	61.0	53.5	57.0	-	59.4	52.0	55.4
	SpellGCN (Cheng et al., 2020)	-	65.1	69.5	67.2	-	63.1	67.2	65.3
	DCN (Wang et al., 2021)	-	67.4	70.4	68.9	-	65.8	68.7	67.2
	Unsupervised Methods								
	uChecker (Sec.3, Ours)	73.3	61.7	61.5	61.6	71.3	57.6	57.5	57.6
	w/o self-supervised detection	68.4	55.3	52.1	53.7	66.7	51.6	48.7	50.1
	w/o confusionset	72.5	62.3	57.3	59.7	58.3	52.9	48.7	50.7
SIGHAN15	**Supervised Methods**								
	*FASPell (Hong et al., 2019)	74.2	67.6	60.0	63.5	73.7	66.6	59.1	62.6
	*Confusionset (Wang et al., 2019)	-	66.8	73.1	69.8	-	71.5	59.5	64.9
	*SoftMask-BERT (Zhang et al., 2020)	80.9	73.7	73.2	73.5	77.4	66.7	66.2	66.4
	*Chunk (Bao et al., 2020)	76.8	88.1	62.0	72.8	74.6	87.3	57.6	69.4
	SpellGCN (Cheng et al., 2020)	-	74.8	80.7	77.7	-	72.1	77.7	75.9
	DCN (Wang et al., 2021)	-	77.1	80.9	79.0	-	74.5	78.2	76.3
	Unsupervised Methods								
	uChecker (Sec.3, Ours)	82.2	75.4	72.0	73.7	79.9	70.6	67.3	68.9
	w/o self-supervised detection	74.0	65.7	61.1	63.3	72.6	62.5	58.1	60.2
	w/o confusionset	81.4	76.2	68.5	72.1	76.5	65.1	58.5	61.6

Table 3: The sentence-level performance on both detection and correction level. Evaluation script is from Hong et al. (2019). * indicates the supervised methods which our unsupervised methods can outperform.

with a reranking strategy to conduct the prediction (Zhang et al., 2015).

LMC presents a model based on joint bi-gram and tri-gram LM and Chinese word segmentation (Xie et al., 2015).

Hybrid utilizes LSTM-based seq2seq framework to conduct generation (Wang et al., 2018) and **Confusionset** introduces a copy mechanism into seq2seq framework (Wang et al., 2019).

FASPell incorporates BERT into the seq2seq for better performance (Hong et al., 2019).

SoftMask-BERT firstly conducts error detection using a GRU-based model and then incorporating the predicted results with the BERT model using a soft-masked strategy (Zhang et al., 2020). Note that the best results of **SoftMask-BERT** are obtained after pre-training on a large-scale dataset with 500M paired samples.

SpellGCN proposes to incorporate phonological and visual similarity knowledge into language models via a specialized graph convolutional network (Cheng et al., 2020).

Chunk proposes a chunk-based decoding method with global optimization to correct single character and multi-character word typos in a unified framework (Bao et al., 2020).

PLOME also employs a confusionset to conduct training of BERT. Besides character prediction, PLOME also introduces pronunciation prediction to learn the misspelled knowledge on phonic level (Liu et al., 2021).

DCN generates the candidate Chinese characters via a Pinyin Enhanced Candidate Generator and then utilizes an attention-based network to model the dependencies between two adjacent Chinese characters (Wang et al., 2021).

4.4 Evaluation Metrics

Following the above mentioned works, we employ character-level and sentence-level **Accuracy**, **Precision**, **Recall**, and **F1-Measure** as the automatic metrics to evaluate the performance of all systems. Besides the official java-based evaluation toolkit (sentence-level) (Tseng et al., 2015)[3], as did in the previous works, we also report and compare the results evaluated by the tools from FASPell (character-level and sentence-level) (Hong et al.,

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

Parameter	Value	Detection				Correction			
		Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
θ^s	0.2	76.3	99.0	76.4	86.3	65.5	98.9	65.4	78.7
	0.4	76.8	99.3	76.8	86.6	66.4	99.2	66.2	**79.4**
	0.5	75.3	99.3	75.3	85.6	64.9	99.1	64.6	78.3
	0.6	72.2	99.5	71.9	83.4	62.5	99.4	62.0	76.4
	0.8	61.2	99.4	60.7	75.3	54.6	99.3	54.8	69.8
θ^u	0.0001	21.9	99.1	20.2	33.5	21.9	99.1	20.2	33.5
	0.01	44.3	99.6	43.2	60.2	41.9	99.6	41.7	57.8
	0.1	53.3	98.2	53.0	68.9	49.7	98.1	49.4	**65.7**
	0.5	53.1	97.9	53.0	68.8	48.2	97.7	48.1	64.5
	0.9	44.2	96.7	44.3	60.8	38.4	96.2	38.4	54.9

Table 4: Parameter tuning on devset of SIGHAN2015 (sentence-level). Due to the limited computing resource, we only conduct parameter tuning independently. Finally, we let $\theta^u = 0.1$ and $\theta^s = 0.4$.

2019)[4] and Confusionset (character-level) (Wang et al., 2019)[5].

5 Results and Discussions

5.1 Main Results

Character-level Evaluation Table 2 depicts the evaluation results on character-level on the datasets of SIGHAN13, SIGHAN14, and SIGHAN15. It is obvious most of the baseline methods are published recently and their performance are very strong. More importantly, almost all of the models are supervised learning based approaches and some of them are even trained using external large-scale datasets. Nevertheless, **surprisingly**, our proposed unsupervised framework uChecker has obtained comparable or even better results than those strong baseline methods. Moreover, during the investigation about the evaluation methods, we notice that character-level detection performance of scrips from Hong et al. (2019) and Wang et al. (2019) are same. But the correction performance is different. Usually Wang et al. (2019) is used to conduct the correction evaluation and results reporting.

Sentence-level Evaluation Figure 3 depicts the evaluation results in sentence-level on those three datasets. Evaluation script is employed from Hong et al. (2019) in order to align the results and to conduct comparing fairly. We also find that the official evaluation tool will output large values though the predicted results are same. We are trying to figure out the reasons.

From Table 3 we can observe that our proposed unsupervised framework uChecker has obtained comparable or even better results than those strong baseline methods in the sentence-level as well.

5.2 Parameter Tuning

Considering that the proposed unsupervised model uChecker is simple and straightforward, there are only two hyperparameters in our framework, θ^u and θ^s, which are the threshold values to conduct spelling diagnosis for unsupervised detectors and supervised detectors respectively. And we only need to tune those two parameters. The tuning is conducted on the validation set of SIGHAN2015. Due to the limited computing resource, we only conduct tuning independently. Finally, we let $\theta^u = 0.1$ and $\theta^s = 0.4$.

5.3 Performance on small datasets

It is surprising that uChecker outperforms all the strong supervised baselines on datasets SIGHAN13 in character-level evaluation, as shown in Figure 2. After investigations we believe that the main reason is that the scale of trainset of SIGHAN13 (350) is much smaller than the other two corpora (6,526 and 3,174). This interesting phenomenon also verify the advantages of the unsupervised learning based methods, especially for the task of CSC which is very difficult for collecting real labelled datasets.

5.4 Ablation Analysis

In the main results tables Table 2 and Table 3, we also provide the results of our model uChecker without the components of self-supervised detection and confusionset guided fine-training and decoding. Generally, the experimental results demonstrate that the corresponding components can indeed improve the performance.

[4] https://github.com/iqiyi/FASPell

[5] https://github.com/sunnyqiny/Confusionset-guided-Pointer-Networks-for-Chinese-Spelling-Check

6 Conclusion

In this paper, we propose a framework named **uChecker** to conduct unsupervised spelling error detection and correction. Masked pretrained language models such as BERT are introduced as the backbone model. We also propose a confusionset-guided masking strategy to fine-train the model to further improve the performance. Experimental results on standard datasets demonstrate the effectiveness of our proposed model uChecker.

Acknowledgement

We thank the anonymous reviewers whose suggestions helped clarify this work. This research is supported by the National Natural Science Foundation of China (No.62106105), the CCF-Tencent Open Research Fund (No.RAGR20220122), the Scientific Research Starting Foundation of Nanjing University of Aeronautics and Astronautics (No.YQR21022), and the High Performance Computing Platform of Nanjing University of Aeronautics and Astronautics.

References

Zuyi Bao, Chen Li, and Rui Wang. 2020. Chunk-based chinese spelling check with global optimization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, EMNLP 2020, Online Event, 16-20 November 2020*, pages 2031–2040. Association for Computational Linguistics.

Chao-Huang Chang. 1995. A new approach for automatic chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, pages 278–283. Citeseer.

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, ACL 2020, Online, July 5-10, 2020*, pages 871–881. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yong Dai, Linyang Li, Cong Zhou, Zhangyin Feng, Enbo Zhao, Xipeng Qiu, Piji Li, and Duyu Tang. 2022. "is whole word masking always better for Chinese BERT?": Probing on Chinese grammatical error correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1–8, Dublin, Ireland. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *CoRR*, abs/1807.01270.

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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1419–1428. 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@EMNLP 2019, Hong Kong, China, November 4, 2019*, pages 160–169. Association for Computational Linguistics.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5958–5967. Association for Computational Linguistics.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 4248–4254. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Chong Li, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2021. Exploration and exploitation: Two ways to improve chinese spelling correction models.

In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 441–446. Association for Computational Linguistics.

Piji Li and Shuming 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4973–4984. Association for Computational Linguistics.

Deng Liang, Chen Zheng, Lei Guo, Xin Cui, Xiuzhang Xiong, Hengqiao Rong, and Jinpeng Dong. 2020. BERT enhanced neural machine translation and sequence tagging model for Chinese grammatical error diagnosis. In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 57–66, Suzhou, China. Association for Computational Linguistics.

C.-L. Liu, M.-H. Lai, K.-W. Tien, Y.-H. Chuang, S.-H. Wu, and C.-Y. Lee. 2011. Visually and phonologically similar characters in incorrect chinese words: Analyses, identification, and applications. *ACM Trans. Asian Lang. Inf. Process.*, 10(2):10:1–10:39.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2991–3000. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 1244–1255. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem N. Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector - grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2020, Online, July 10, 2020*, pages 163–170. Association for Computational Linguistics.

Kevin Parnow, Zuchao Li, and Hai Zhao. 2021. Grammatical error correction as gan-like sequence labeling. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3284–3290. 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, SIGHAN@IJCNLP 2015, Beijing, China, July 30-31, 2015*, pages 32–37. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2437–2446. 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, Brussels, Belgium, October 31 - November 4, 2018*, pages 2517–2527. 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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5780–5785. Association for Computational Linguistics.

Hongfei Wang, Michiki Kurosawa, Satoru Katsumata, and Mamoru Komachi. 2020a. Chinese grammatical correction using bert-based pre-trained model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, AACL/IJCNLP 2020, Suzhou, China, December 4-7, 2020*, pages 163–168. Association for Computational Linguistics.

Yu Wang, Yuelin Wang, Jie Liu, and Zhuo Liu. 2020b. A comprehensive survey of grammar error correction. *CoRR*, abs/2005.06600.

Weijian Xie, Peijie Huang, Xinrui Zhang, Kaiduo Hong, Qiang Huang, Bingzhou Chen, and Lei Huang. 2015.

Chinese spelling check system based on n-gram model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 128–136, Beijing, China. Association for Computational Linguistics.

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2021. Lm-critic: Language models for unsupervised grammatical error correction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7752–7763. Association for Computational Linguistics.

Ruiqing Zhang, Chao Pang, Chuanqiang Zhang, Shuohuan 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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2250–2261. 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, ACL 2020, Online, July 5-10, 2020*, pages 882–890. Association for Computational Linguistics.

Shuiyuan Zhang, Jinhua Xiong, Jianpeng Hou, Qiao Zhang, and Xueqi Cheng. 2015. Hanspeller++: A unified framework for chinese spelling correction. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2015, Beijing, China, July 30-31, 2015*, pages 38–45. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 156–165. Association for Computational Linguistics.