PAPER Special Section on Information and Communication Technology to Support Hyperconnectivity

PSDSpell: Pre-Training with Self-Distillation Learning for Chinese Spelling Correction

Li HE[†,††a)], **Xiaowu ZHANG**[†,††b)], ***Nonmembers***, **Jianyong DUAN**[†,††c)], ***Member***, **Hao WANG**[†,††d)], **Xin LI**[†,††e)], ***and*** **Liang ZHAO**[†††,††††f)], ***Nonmembers***

SUMMARY Chinese spelling correction (CSC) models detect and correct a text typo based on the misspelled character and its context. Recently, Bert-based models have dominated the research of Chinese spelling correction . However, these methods only focus on the semantic information of the text during the pretraining stage, neglecting the learning of correcting spelling errors. Moreover, when multiple incorrect characters are in the text, the context introduces noisy information, making it difficult for the model to accurately detect the positions of the incorrect characters, leading to false corrections. To address these limitations, we apply the multimodal pretrained language model ChineseBert to the task of spelling correction. We propose a self-distillation learning-based pretraining strategy, where a confusion set is used to construct text containing erroneous characters, allowing the model to jointly learns how to understand language and correct spelling errors. Additionally, we introduce a single-channel masking mechanism to mitigate the noise caused by the incorrect characters. This mechanism masks the semantic encoding channel while preserving the phonetic and glyph encoding channels, reducing the noise introduced by incorrect characters during the prediction process. Finally, experiments are conducted on widely used benchmarks. Our model achieves superior performance against state-of-the-art methods by a remarkable gain.

key words: *spelling correction, ChineseBert, self-distillation, multimodal information*

1. Introduction

Chinese Spelling Check, an essential task in Chinese natural language processing, focuses on identifying and rectifying spelling errors in Chinese texts. With the advancement of technology, the trend towards paperless office practices has grown, making it worthwhile to explore methods for correcting erroneous characters present in text input via keyboards. Chinese input methods commonly include Pinyin and Wubi input methods. Consequently, during keyboard input, two types of errors are prone to occur: phonologically similar errors and visually similar errors, resulting from the misuse of Chinese characters with similar pronunciations or visual appearances. According to the study mentioned in the [1], about 83% of errors are related to phonological similarity, and 48% are related to visual similarity. Unlike English, Chinese is a logographic writing system, and it does not have misspelled words that are not present in the Chinese character dictionary; instead, it has homophonic characters. Chinese characters do not have clear word boundaries, and the meaning of each character can undergo significant changes when the context changes. Therefore, it is challenging to determine whether there are word-level errors in a sentence [2]. Table 1 illustrates two examples of Chinese spelling correction errors. Recently, pre-trained language models such as BERT (Devlin et al. [3]) have been successfully applied to Chinese spelling correction tasks. However, since BERT is trained based on the masked token recovering task, it can only treat all characters as potentially erroneous during the error detection phase, leading to lower efficiency and accuracy. When multiple errors exist in the text, BERT relies solely on contextual semantics for prediction, and erroneous context introduces noise to the model. As a result, the model may struggle to determine the positions of errors accurately and may lead to false corrections.

Texts typically contain multiple errors, as evidenced by our analysis of multi-error samples from the SIGHAN datasets. Specifically, in the SIGHAN2013 [4], SIGHAN2014 [5], and SIGHAN2015 [6] datasets, the percentage of multi-error samples reached 21%, 29%, and 22%, respectively. We observed that the performance of existing spelling correction models on multi-error samples is inferior to their performance on the entire dataset. This discrepancy can be attributed to the noise introduced by the contextual information containing erroneous characters in multi-error samples.

To enable the model to learn spelling error knowledge during the pre-training phase and improve its robustness to

Table 1 Examples of Chinese spelling errors. Misspelling characters are marked in red, and the corresponding phonics are given in brackets.

Type	Sentence	Correction
phonological（83%）	今天的天空真是太没（mei）了。 The sky today is really gone.	美 (mei) beautiful
visually（48%）	你需要有门票才能进人（ren）场馆。 You must have a ticket to human the venue.	入（ru） enter

Manuscript received May 27, 2023.
Manuscript revised September 1, 2023.
Manuscript publicized October 25, 2023.

†The authors are with College of Informatics, North China University of Technology, Beijing, China.
††The authors are with CNONIX National Standard Application and Promotion Lab, Beijing, 100144, China.
†††The author is with School of Information Management, Wuhan University, Wuhan, Hubei, 430072, China.
††††The author is with Key Laboratory of Semantic Publishing and Knowledge Service of the National Press and Publication Administration, Wuhan University, Wuhan, Hubei, China.
a) E-mail: heli@ncut.edu.cn
b) E-mail: zhangxw21@outlook.com
c) E-mail: duanjy@ncut.edu.cn
d) E-mail: wanghaomails@gmail.com
e) E-mail: lx@ncut.edu.cn
f) E-mail: liangzhao@whu.edu.cn
DOI: 10.1587/transinf.2023IHP0005

Table 2 The correction performance of various Chinese Spelling Check (CSC) models on the SIGHAN15 test set and a multi-error test set (consisting of 242 test instances extracted from SIGHAN15). We evaluated the models using character-level evaluation metrics.

Model	SIGHAN15			Multi-typo Set		
	Pre	Rec	F1	Pre	Rec	F1
Bert	97.0	79.3	87.3	96.4	67.4	79.3
SpellGCN	96.7	81.4	88.4	95.9	69.9	80.9
PLOME	97.2	85.0	90.7	94.0	75.9	84.0

the noise introduced by spelling error context, we utilize a Chinese character confusion set [4] to replace 15% of randomly masked characters with characters from the confusion set, ensuring that each sentence contains multiple errors. We employ self-distillation learning to guide the model in jointly learning semantics and spelling error knowledge during pre-training. Our proposed pre-training strategy is model-agnostic and can be applied to different models.

Furthermore, we have observed that incorporating phonetic and character shape information is beneficial for Chinese spelling correction tasks (PLOME [7], REALIZE [8], MLM-phonetics [9]). However, these models often fuse information from all channels and mask the information of erroneous characters, thereby preventing the model from utilizing the valuable information carried by the erroneous characters. To address this issue, we use Chinesebert [10], which combines Chinese character shape and phonetic features, to construct our correction network. Unlike other models that mask all channels, we retained the phonetic and visual features that are most beneficial to the model's final predictions. Subsequently, the denoised fused features are fed into the correction model. Since the correction model already masks the semantic features of erroneous characters during input, coupled with the constraints imposed by the visual and phonetic aspects of the model's predictions, PSDSpell is better equipped to handle Chinese spelling correction tasks.

In summary, our contributions are as follows: 1. We propose a pre-training strategy based on self-distillation learning, allowing the model to jointly learn semantics and spelling error knowledge during the pre-training phase. 2. We introduce a single-channel masking mechanism that improves the utilization of phonetic and character shape information in existing models. This approach retains the phonetic and character shape features that help predict the output. Experimental results demonstrate that our model achieves improvements in error detection and correction compared to baseline models. It also performs well on multi-error samples. Overall, our contributions enhance the understanding and utilization of spelling error knowledge in pre-training and improve the performance of Chinese spelling correction models.

2. Related Work

Chinese spelling correction has received widespread attention over the past few decades. In the early stages, the focus was mainly on rule-based and statistical methods. Y. Jiang [11] proposed a new grammar rule system for addressing spelling and grammar errors. However, these rules are challenging to cover all types of spelling errors, and rule-based methods struggle to handle all Chinese spelling errors comprehensively. Wang [12] employed word embeddings and a conditional random field (CRF)-based error detector to identify potential spelling errors and provide correction suggestions. Huang [13] used an N-gram model based on word segmentation for error detection and combined it with heuristic rules for error correction. Statistical approaches often follow a pipeline correction pattern, which can lead to error propagation. Moreover, they typically rely on threshold-based criteria to judge sentence fluency, limiting the exploration of semantic information and potentially weakening the model's performance.

In recent years, pre-training models based on masking mechanisms have achieved significant success in various natural language processing tasks. Liu [4] fused semantic, phonetic, and character shape information at the embedding layer and predicted Chinese characters and phonetic outputs, combining their outputs during prediction. Xu [8] employed a multimodal approach that integrates semantic, phonetic, and character shape representations to enhance the error detection and correction performance of the model. Zhu [14] proposed a multitasking framework for Chinese spelling correction, using a late fusion strategy to combine the hidden states of the correction and detection modules, minimizing the misleading impact of spelling errors on character correction. Liu [15] constructed a noisy sample for each training sample, training the model to output outputs more similar to the original training data and the noisy sample. While these methods have improved the performance of the models to some extent, they essentially involve sorting and filtering the model's correction results, and the noisy information is still input to the model, causing certain interference in the model's predictions. In contrast, using a single-channel masking strategy, our approach reduces the interference caused by erroneous characters during the prediction process.

3. Approach

The Chinese spelling correction task aims to detect spelling errors at the character level in a given sentence $X = \{x_1, x_2, x_3, \cdots, x_n\}$ and generate the corrected sentence $Y = \{y_1, y_2, y_3, \cdots, y_n\}$. Existing methods based on pre-training models directly generate the target sentence based on the input sentence information. Although this simplifies the correction process, these methods often utilize the semantics of one erroneous character to predict another erroneous character, resulting in poor performance on texts with multiple errors. To address this issue, we utilize a confusion set to construct texts with multiple errors and employ self-distillation learning to pre-train the correction network. This allows the model to simultaneously learn both semantic knowledge and more spelling error knowledge.

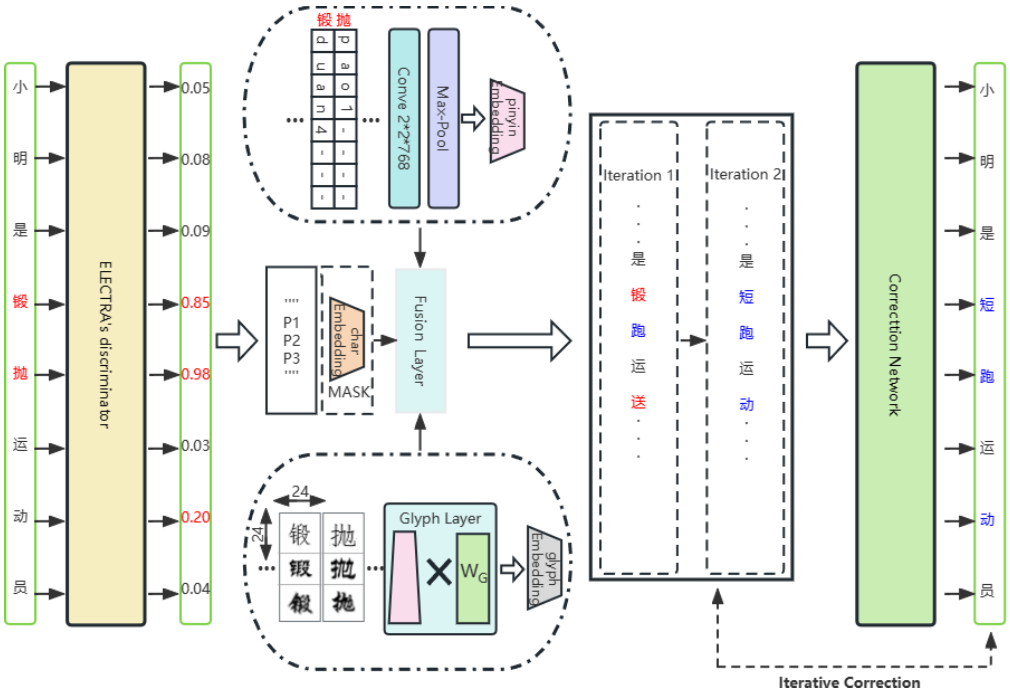


Fig. 1 The framework of the proposed PSDSpell, where the incorrect characters are marked in red, and the corrected characters are marked in blue. Left: the detection network detects potentially incorrect characters. Middle: based on the results of the detection network, potential erroneous characters (锻 (forge), 抛 (throw), 动 (move)) in the input sentence are identified. The semantic encoding channel of potentially incorrect characters is masked, while preserving the visual glyph encoding and pinyin encoding channels representing these potential errors. Subsequently, the denoised fused features are input into a correction model for refinement. Right: the correction network utilizes the iterative correction strategy to perform corrections and outputs the corrected results.

As shown in Fig. 1, the proposed spelling correction model (PSDSpell) consists of two main components: the detection network and the correction network. The detection network predicts the error probability for each character, resulting in a probability sequence $P = \{P_1, P_2, P_3, \cdots, P_n\}$, which identifies potentially erroneous characters in the text. We then employ a single-channel masking mechanism to mask the semantic information of these characters while preserving the phonetic and character shape features that are helpful for the final model predictions. This allows us to effectively reduce the noise introduced by the erroneous characters during the correction process. Furthermore, we adopt a simple yet effective iterative correction strategy to avoid erroneous corrections. We progressively refine the correction results through two rounds of iteration, ensuring more accurate corrections. Ultimately, we obtain the corrected sentence Y, which represents the final output of our model.

3.1 Pre-Training Strategy Based on Self-Distillation Learning

We employed a substitution strategy guided by a Chinese character confusion set (including phonetically similar and visually similar errors) introduced by Wu [4] to construct sentence pairs for self-distillation learning. We replaced the fixed mask token "[MASK]" that does not exist in downstream tasks with characters from the confusion set. We abandoned the Next Sentence Prediction (NSP) task, which is irrelevant to Chinese spelling correction. We utilized a dynamic masking strategy, randomly masking 15% of different characters during each training iteration. Unlike the masking strategy of other Chinese spelling correction models, considering a higher proportion of phonetically similar errors, our masking strategy replaced 70% of characters with phonetically similar ones and 30% with visually similar ones, without retaining randomly generated characters. Therefore, we constructed an adequate amount of multi-error text for pre-training. The details are shown in Table 3.

Table 3 Examples of different masking strategies. The chosen token is marked in red, and the corresponding phonics is given in brackets.

Masking Strategies	Sentence
Original Sentence	人口负增长确实会造成不少困(kun,difficulty) 难，经济衰退(shuai tui,decline) 是其严重的结果。
Phonic Masking（70%）	人口负增长确实会造成不少困难，经济摔腿（shuai tui,fall leg）是其严重的结果。
Shape Masking（30%）	人口负增长确实会造成不少因（yin,because）难，经济衰退是其严重的结果。

In recent years, self-distillation learning has achieved impressive results in the fields of computer vision (CV) and natural language processing (NLP) (Gao [16], Zhang [17], Lee [18]). Through self-distillation, knowledge from deeper parts of the network can be distilled into shallower parts,

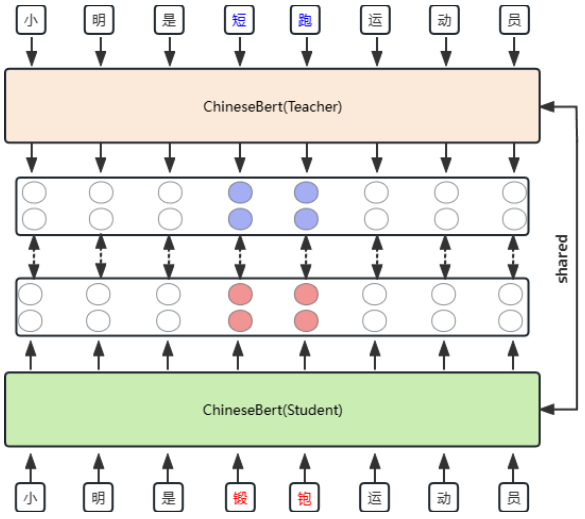


Fig. 2 Using the self-distillation pre-training strategy, we input sentences containing typos and their corresponding correct versions separately into the two sides of ChineseBert.

which significantly helps with data augmentation and improves model performance. By combining the substitution strategy using a confusion set, we further exploit the advantages of the pre-training-fine-tuning paradigm using self-distillation learning. We use ChineseBert to encode sentences with spelling errors and their corresponding correct sentences. Inspired by contrastive learning, we perform effective knowledge transfer using Wang's approach [19]. By using contrastive loss, we regularize the hidden states of sentences with errors to make them closer to the hidden states of correct sentences. The process is illustrated in Fig. 2.

We use an additional distillation loss to help ChineseBert establish a connection between incorrect characters and their correct counterparts. We aim to use this loss to make the hidden layer representations of sentences with misspelled characters and their corresponding correct sentences closer in output. We employ a self-distillation method using shared ChineseBert weights to construct positive and negative samples for contrastive learning. The specific loss calculation is as follows:

$$L_{kc} = -\sum_{i=1}^{n} \theta(\widetilde{x_i}) \log \frac{\exp(sim(\widetilde{h_i}, h_i)/\tau)}{\sum_{j=1}^{n} \exp(sim(\widetilde{h_i}, h_j)/\tau)} \tag{1}$$

Suppose x_i is an incorrect character, then $\theta(\widetilde{x_i}) = 1$. Otherwise, $\theta(\widetilde{x_i}) = 0$. $\widetilde{h_i}$ represents the hidden state from the teacher model with the correct input. τ is the distillation temperature hyperparameter, and $sim(\widetilde{h_i}, h_i)/\tau$ represents the cosine similarity between these two vectors. The objective of minimizing L_{kc} is to make the hidden state of the student model, which contains erroneous characters, similar to the corresponding correct state of the teacher model. We use stop gradient (sg) to decouple the gradient backpropagation to $\widetilde{h_i}$, ensuring stability during training. Pretraining is performed in conjunction with the cross-entropy loss between the student and teacher models. The specific loss is as follows:

$$L_s = -\sum_{i=1}^{n} \log \left(P\left(\hat{Y}_i = y_i | X\right)\right) \tag{2}$$

$$L_t = -\sum_{i=1}^{n} \log \left(P\left(\bar{Y}_i = y_i | Y'\right)\right) \tag{3}$$

$$L_p = L_s + \alpha L_t + \beta L_{kc} \tag{4}$$

Where α and β are hyperparameters, our model initializes using the parameters of ChineseBert[†].

3.2 Detection Network

We use the Discriminator part of ELECTRA (Base) (Clark et al.) [20] as our detection network. The input to the detection network is a sequence of embeddings $E = \{e_1, e_2, e_3, \cdots, e_n\}$, where e_i represents the feature vector of character x_i, which is the sum of word embeddings, position embeddings, and sentence embeddings. The output is a label sequence $E_p = \{e_{p_1}, e_{p_2}, e_{p_3}, \cdots, e_{p_n}\}$, where e_{p_i} represents the label of the i character. We use 1 to indicate that the character is incorrect and 0 to indicate correctness. We use the sigmoid function for each character to obtain the error probability P_i, where a higher error probability indicates a higher likelihood of the character being incorrect. It is defined as follows:

$$P_i = P_d\left(e_{p_i} = 1 \mid X\right) = \sigma\left(W_d H_{di} + b_d\right) \tag{5}$$

Where H_{di} represents the output of the last layer after the character has been processed by the detection network, and W_d and b_d are learnable parameters for binary classification.

To recall more incorrect characters, we set the threshold to 0.1. That is if $P_i \geq 0.1$, the character is classified as incorrect, and if $P_i < 0.1$, it is classified as correct. Finally, for the detection model, we optimize the detection network using the binary cross-entropy loss function.

$$L_d = -\frac{1}{N}\sum_{i=1}^{N} \left[e_{pi} \cdot \log\left(P_i\right) + \left(1 - y_i\right) \cdot \log\left(1 - P_i\right)\right] \tag{6}$$

3.3 Correction Network

The correction network is built based on ChineseBert, a Chinese pretraining language model that integrates phonetic and visual information about Chinese characters. Since Chinese is an ideographic writing system, both visual and phonetic features contain crucial information that is highly important for language comprehension. ChineseBert takes each Chinese character and concatenates its semantic, visual, and phonetic features. These features are then mapped to the same dimensionality through a fully connected layer, forming fused features. Finally, the fused feature vectors are combined with position encoding vectors and used as input to the Bert model. Considering the characteristics of Chinese spelling errors, incorporating ChineseBert as the correction

[†]https://github.com/ShannonAI/ChineseBert

network is highly suitable.

The encoder first generates character embeddings, phonetic embeddings, and visual embeddings, all of which have a size of D. These three embeddings are then concatenated and mapped to a fused embedding of size D through a fully connected layer. Similar to other pretraining language models, the fused embedding is added to the position embedding and passed through a stack of consecutive transformer layers. This process generates the contextual representation $h_i \in \mathbb{R}^D$ for the input character x_i. We denote the resulting character representations as $H = \{h_1, h_2, h_3, \cdots, h_n\}$. To project h_i into a specific feature space, we use learnable parameters $W^{(c)} \in \mathbb{R}^{D\times D}$ and $b^{(c)} \in \mathbb{R}^D$ for the character-specific feature projection layer.

$$h_i^{(c)} = GeLU\left(W^{(c)}h_i + b^{(c)}\right) \tag{7}$$

Then, based on the projected output, we predict the corresponding correct character y_i. Here, $W^{(y)} \in \mathbb{R}^{V\times D}$ and $b^{(y)} \in \mathbb{R}^V$ are the learnable parameters of the character prediction layer, where V represents the vocabulary size.

$$P\left(\hat{y}_i \mid X\right) = softmax\left(W^{(y)}h_i^{(c)} + b^{(y)}\right) \tag{8}$$

We optimize the correction model using cross-entropy loss.

$$L_c\left(\hat{y}_i, y\right) = -\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) \tag{9}$$

Single-channel masking mechanism: After obtaining the position information of potentially incorrect characters from the detection network, we adopt a single-channel masking mechanism to reduce the noise impact of incorrect characters. By preserving the phonetic and morphological encoding channels through masking, we impose constraints on the model predictions using phonetic and morphological information. This allows the model to effectively utilize the denoised information and better handle texts with multiple errors. For example, although the characters "困 (tired)" and "因 (reason)" have significant semantic differences, their morphological information extracted through CNN is similar. Similarly, although "县 (county)" and "鲜 (fresh)" have significant differences in morphological information and semantics, they share similar phonetic encodings. Therefore, by leveraging the related information of the incorrect characters' morphology and phonetics, we enhance the model's performance on texts with multiple errors.

After obtaining the error position information from the detection network, we only mask the semantic information at the corresponding positions, while preserving the channels for phonetic and morphological information modeling. This ensures that we provide the model with more plausible information without introducing additional noise. Specifically, when the detection network identifies an incorrect character, our masking strategy transitions from Eq. (10) to Eq. (11).

$$e_{fi} = W_F\left[e_{wi} \otimes e_{gi} \otimes e_{si}\right] \tag{10}$$

$$e_{fi} = W_F\left[e_{mi} \otimes e_{gi} \otimes e_{si}\right] \tag{11}$$

Where e_{wi} represents the semantic encoding, e_{gi} represents the glyph encoding, e_{si} represents the phonetic encoding, and e_{mi} denotes the semantic mask.

Iterative Correction Strategy: SCOPE [21] employs a simple yet effective constrained iterative correction strategy to address the tendency of Chinese spelling correction models to rectify accurate expressions into more frequent ones. Similarly, in PSDSPell, a similar approach is adopted, correcting erroneous positions through two rounds of iterative correction. We progressively correct the errors within a specified window around the previously corrected positions. Considering the characteristics of error samples, we set the window size to 3, which means one position on the left and one on the right of the current position. We set the number of iterations to 2 to ensure sufficient error correction while avoiding over correction. After one round of iteration, if a position has been modified in each iteration round, we restore it to the original character, making no further modifications.

3.4 Learning

The training process of PSDSpell is driven by two objectives, namely the loss function of the detection network and the loss function of the correction network. We combine these two loss functions linearly to form the overall training objective.

$$L = \lambda \cdot L_c + (1 - \lambda) \cdot L_d \tag{12}$$

Here, L_d and L_c represent the loss functions of the detection network and correction network, respectively. L represents the joint training loss function of the entire model, and $\lambda \in [0, 1]$ is the parameter for linear combination.

4. Experimental Results

4.1 Pre-Training

Dataset: During the pre-training phase, to enhance the effectiveness of the training strategy based on self-distillation learning, we utilize the wiki2019zh[†] corpus as the foundation. This corpus encompasses one million pages from Chinese Wikipedia[††]. Additionally, it incorporates a pretraining corpus of three million news articles collected by PLOME [7]. These pages and articles are segmented into sentences, resulting in a total of 162.1 million sentences. Then we concatenate consecutive sentences to obtain text fragments with at most 510 characters, which are used as the training instances.

Parameter Settings: We set the distillation temperature $\tau = 0.9$, $\alpha = 1$, and $\beta = 0.05$. The learning rate is set to 5e-5. The batch size is set to 32, and the number of epochs is set to 30. The learning rate warmup steps are set to 5000, and the Adam optimization algorithm is used.

[†]https://github.com/suzhoushr/nlp_chinese_corpus
[††]https://zh.wikipedia.org/wiki/

4.2 Fine-Tuning

Training Data: This paper uses the SIGHAN dataset (Wu et al. [4]; Yu et al. [5]; Tseng et al. [6]) and 271K training data collected from Wang et al. [22]. The test sets from SIGHAN 13, SIGHAN 14, and SIGHAN 15 are used. The training samples are converted to simplified Chinese characters using OpenCC[†]. Additionally, we extracted multiple error samples from the SIGHAN 2015 and SIGHAN 2014 test sets, which include 552 sentences with multiple errors.
Parameter Settings: In the specific fine-tuning process, all feature vectors are set to have a dimension of 768. The learning rate is set to 5e-5 with linear decay. Dropout is set to 0.1. The batch size is set to 32, and the number of epochs is set to 30. The learning rate warm-up steps are set to 5000, and the Adam optimization algorithm is used.

4.3 Baseline Model and Evaluation Metrics

We use widely adopted sentence-level accuracy, recall, and F1 score as our main evaluation metrics. Compared to character-level evaluation metrics, sentence-level metrics are more stringent. To demonstrate the effectiveness of PSDSpell approach, this paper selects the following models as baseline models for comparison:

(1) SpellGCN (Cheng et al.) [23]: This method learns the pronunciation/shape relationships between characters by applying graph convolutional networks on two similarity graphs. It combines graph representations with semantic representations from BERT to predict correction candidates.

(2) MLM-phonetics (Zhang et al.) [9]: This method combines a language model with phonetic features for pre-training. It further fine-tunes the model with a joint detection module and correction module.

(3) REALIZE (Xu et al.) [8]: This method models the semantic, phonetic, and visual (glyph) information of input characters and selectively combines information from these modalities for the final correction task.

(4) PLOME (Liu et al.) [7]: This method utilizes GRU networks to extract phonetic and visual (glyph) features of characters. It combines semantic information, phonetic information, and glyph information through direct summation and predicts the pronunciation of the target character in a coarse-grained manner.

(5) MDCSpell (Zhu et al.) [14]: This method utilizes BERT to capture the visual and phonetic features of each character in the original sentence. It employs a post-fusion strategy to combine the hidden states of the corrector with the hidden states of the detector, reducing the impact of misspelled characters.

[†]https://github.com/BYVoid/OpenCC

4.4 Main Results

Table 4 presents the evaluation results of PSDSpell and baseline methods in terms of detection and correction performance on three test sets. The boldface font in the table represents the best results. Table 5 shows the results of the model on our extracted multi-error test set.

Table 4 shows the performance of PSDSpell and the baseline models on the test sets. In most cases, our improvements have yielded promising results. The F1 scores for detection and correction on the SIGHAN15 dataset have improved by 3.4/3.1, respectively. On the SIGHAN2014 dataset, the F1 scores for detection and correction have improved by 0.8/1.1, respectively. PSDSpell also performs competitively with the previous best model, REALIZE, on the SIGHAN2014 dataset. Compared to previous models, we have employed a more refined self-distillation learning pre-training strategy, enabling PSDSpell to jointly learn semantic and spelling error knowledge during pre-training and better adapt to multi-error text correction.

In addition, we also evaluated the performance of our model on a multi-error test set. The bold font in Table 5 represents the best results. Compared to the state-of-the-art methods, PSDSpell performs significantly better on the multi-error test set. While both PLOME and REALIZE achieved good F1 scores at the detection level, their F1 scores dropped noticeably at the correction level, indicating that although these models can identify errors in noisy text, they struggle to correct them accurately. Our approach achieves an improvement of 1.9/0.7 in terms of F1 scores for detection and correction, respectively, compared to the optimal results of the baseline.

4.5 Effects of Pre-Training Strategy

To verify the effectiveness of our self-distillation-based pre-training strategy, we adopt cBert [7], a Bert model pre-trained using a confusion set-guided approach. In this approach, 15% of the characters are masked, of which 60% are replaced using a phonetic substitution strategy, 15% are replaced using a shape substitution strategy, 15% are kept unchanged, and 10% are randomly replaced. We directly evaluate the model on the constructed multi-error test data. The results are shown in Table 6.

The results show that cBert, which utilizes confusion set-guided pre-training, shows an overall improvement compared to Bert's direct error correction. However, our self-distillation strategy, where semantic and spelling error knowledge is jointly learned during pre-training, achieves a higher F1 score improvement of 3.1/4.3 compared to cBert. This demonstrates the effectiveness of our pre-training strategy.

Table 4 Sentence-level performance on the test sets of SIGHAN13, SIGHAN14, and SIGHAN15, where precision (Pre), recall (Rec), F1 (F1) for detection, and correction are reported (%). The "*" symbol indicates that we applied post-processing (following the same preprocessing steps as REALIZE). Before evaluation, we eliminated all instances of the characters "的 (de)", "得 (de)", and "地 (de)" in both the detection and correction tasks. This was done to the model outputs for the SIGHAN13 dataset. The experimental results for other baselines are sourced from their respective literature.

Dataset	Method	Detection Level			Correction Level		
		Pre	Rec	F1	Pre	Rec	F1
SIGHAN13	SpellGCN	80.1	74.4	77.2	78.3	72.7	75.4
	SpellGCN*(Our reimplementation)	85.8	78.8	82.2	84.2	77.4	80.7
	MLM-phonetics	82.0	78.3	80.1	79.5	77.0	78.2
	REALIZE*	88.6	82.5	85.4	87.2	81.2	84.1
	MDCSpell	**89.1**	78.3	83.4	**87.5**	76.8	81.8
	PSDSpell(Ours)*	87.7	**83.4**	**85.5**	86.6	**81.9**	**84.2**
SIGHAN14	SpellGCN	65.1	69.5	67.2	63.1	67.2	65.3
	MLM-phonetics	66.2	**73.8**	69.8	64.2	**73.8**	68.7
	REALIZE	67.8	71.5	69.6	66.3	70.0	68.1
	MDCSpell	**70.2**	68.8	69.5	69.0	67.7	68.3
	PSDSpell(Ours)	69.4	71.8	**70.6**	**69.6**	70.1	**69.8**
SIGHAN15	SpellGCN	74.8	80.7	77.7	72.1	77.7	75.9
	MLM-phonetics	77.5	83.1	80.2	74.9	80.2	77.5
	REALIZE	77.3	81.3	79.3	75.9	79.9	77.8
	PLOME	77.4	81.5	79.4	75.3	79.3	77.2
	MDCSpell	80.8	80.6	80.7	78.4	78.2	78.3
	PSDSpell(Ours)	**82.8**	**85.4**	**84.1**	**80.1**	**82.7**	**81.4**

Table 5 Results on the multi-error test set, extracted from SIGHAN2014 and SIGHAN2015, consisting of 552 test instances. We evaluated the baseline model and PSDSpell using sentence-level evaluation metrics.

Model	Detection Level			Correction Level		
	Pre	Rec	F1	Pre	Rec	F1
SpellGCN	71.2	73.5	72.3	70.1	72.0	71.0
REALIZE	75.7	74.6	75.1	72.9	75.4	74.1
PLOME	73.1	74.0	73.5	73.5	74.9	74.2
PSDSpell(Ours)	**76.3**	**77.8**	**77.0**	**74.3**	**75.6**	**74.9**

Table 6 A comparison between self-distillation pre-training and confusion set-guided pre-training, with the pre-training and fine-tuning datasets kept consistent. The evaluation is performed using sentence-level evaluation metrics.

Model	Detection Level			Correction Level		
	Pre	Rec	F1	Pre	Rec	F1
Bert	30.4	31.7	31.0	29.7	26.9	28.2
cBert	35.9	46.3	40.4	32.1	42.0	36.4
Bert（Using our pre-trained method）	**39.8**	**48.4**	**43.5**	**36.1**	**46.7**	**40.7**

4.6 Effects of the Threshold Value "Err" on the Model Performance

We evaluated the impact of different thresholds (0.5, 0.4, 0.3, 0.2, 0.1, 0.01) on the detection network and the correction network separately, as shown in Fig. 3. The experiments were conducted on the SIGHAN13, SIGHAN14, and SIGHAN15 datasets.

As shown in Figs. 3 (a)–(c), with the decrease of the threshold, the precision (DN-P) value of the detection network decreases, while the recall rate of erroneous characters improves. However, since the recall rate (DN-R) has already approached its maximum value, the reduction in Err has a diminishing effect on the improvement of recall rate (DN-R) gain, while the precision (DN-P) decreases rapidly. This results in a continuous decline in the overall performance F1 (DN-F1) value. Therefore, in the experiment, we select a relatively optimal Err, namely 0.1.

As shown in Fig. 3 (d), To further investigate the impact of the hyperparameter "Err" on the correction model, we delve into the variations in model performance under different hyperparameter settings. Based on the experimental results, it can be observed that as the threshold value "Err" decreases, the F1 score of the model tends to increase. The highest F1 score is achieved when Err = 0.1, followed by a decreasing trend. Setting the threshold value too low can introduce more noise to the correction model. Through the preceding experiments, it can be observed that: Although lowering Err can improve the recall of the model, the decrease in precision becomes more significant. Therefore, when Err = 0.1, the performance of the model starts to decline. Consequently, we choose Err = 0.1 as the threshold value for the detection model.

4.7 The Impact of the Loss Function Hyperparameter λ on the Model Performance

As shown in Fig. 4, when we set λ to 0.85, we achieve the best F1 score. This setting is reasonable because the convergence of the correction task is more challenging than the detection task, requiring higher weight during learning. However, setting λ too high would reduce the learning of the detection network and diminish its contribution. Therefore, selecting a relatively higher λ can achieve a better balance between the two tasks and achieve optimal results.

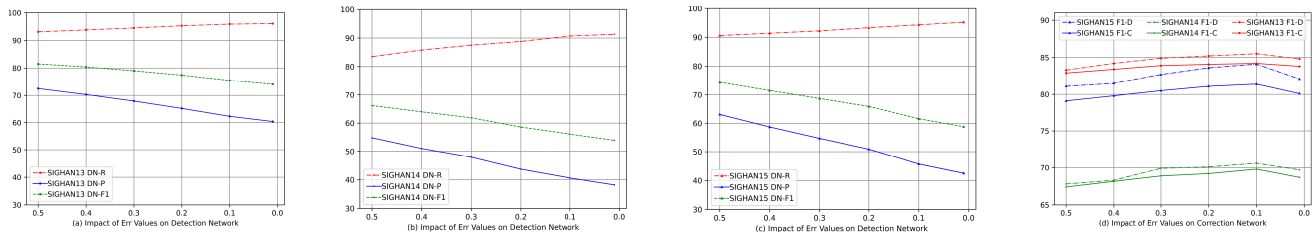


Fig. 3 The threshold value "Err" impacts model performance. There are four images in total, labeled from left to right as Figs. (a), (b), (c), and (d). Figures (a)–(c) illustrate the impact of different thresholds on the detection network, using character-level evaluation metrics (DN-R for recall, DN-P for precision, DN-F1 for F1 score). Figure (d) presents the influence of various thresholds on the correction network, utilizing sentence-level evaluation metrics. The experiments were conducted on the test sets of SIGHAN13, SIGHAN14, and SIGHAN15.

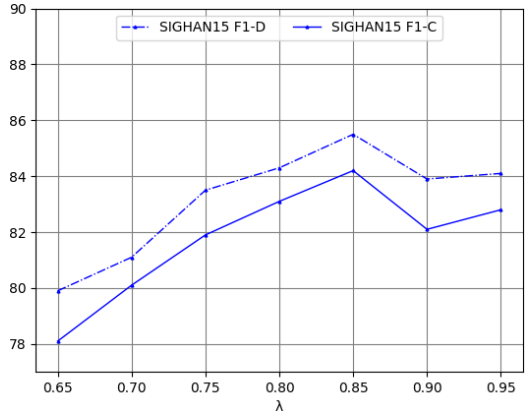


Fig. 4 Impact of the loss function hyperparameter on model performance.

Table 7 Results of the ablation study.

Model	Detection Level			Correction Level		
	Pre	Rec	F1	Pre	Rec	F1
(w/o single-channel masking)	82.8	84.9	83.8	78.3	80.7	79.5
(w/o iterative correction)	80.0	83.7	81.8	77.4	81.2	79.3
(w/o pretraining)	77.5	84.1	80.7	76.9	82.2	79.5
PSDSpell(Ours)	**82.8**	**85.4**	**84.1**	**80.1**	**82.7**	**81.4**

Table 8 Case study analysis on dataset examples.

Input	人口越少，管理整个国家越困难，哪里(where) 都缺人才。
Baseline	人口越少，管理整个国家越困难，那里(there) 都缺人才。
PSDSpell(Ours)	人口越少，管理整个国家越困难，哪里(where) 都缺人才。
Input	我会说一点儿，不过一个汉子(man) 也看不懂，所以我迷路了。
Baseline	我会说一点儿，不过一个汉子(Chinese character) 也看不懂，所以我迷路了。
PSDSpell(Ours)	我会说一点儿，不过一个汉字(Chinese character) 也看不懂，所以我迷路了。
Input	还有几天才到我的生日，等到姓青号(name QingHao) 的时候再去。
Baseline	还有几天才到我的生日，等到姓青号(name QingHao) 的时候再去。
PSDSpell(Ours)	还有几天才到我的生日，等到心情好(a good mood) 的时候再去。

4.8 Ablation Study

We conducted a series of ablation study to evaluate the effectiveness of each method in PSDSpell. The experiments were performed on the SIGHAN15 dataset, and the parameters for all ablation experiments were kept the same. The specific experiments are as follows:

(1) Removal of single-channel masking mechanism: After obtaining the positions of potential errors detected by the detection network, the information from all channels is masked.

(2) Removal of iterative correction strategy: The proposed step-by-step correction strategy is not utilized during the correction process. Instead, the correction network directly performs the correction.

(3) Removal of pretraining strategy: The proposed pretraining strategy is not applied, and instead, the original task of Bert is used for pretraining.

As shown in Table 7, (1) Removing the single-channel masking mechanism prevents the correction model from utilizing the phonetic and glyph information of erroneous characters during the spelling correction task. Due to the influence of erroneous context, the model introduces additional noise, decreasing the correction performance. (2) If the iterative correction strategy is removed, with the low threshold of the detection network, many initially correct characters are mistakenly identified as errors. Without the step-by-step iteration, the model is easily influenced by these erroneous positions, resulting in erroneous or excessive corrections and decreased overall performance. (3) By removing the pretraining strategy, we can observe that utilizing self-distillation learning for pretraining is beneficial for the error correction task, allowing the model to learn Chinese spelling correction knowledge during the pretraining phase.

4.9 Case Study

We show several correction results to demonstrate the properties of PSDSpell. Several prediction results are given in Table 8.

The results show that PSDSpell performs well in avoiding interference when the context contains erroneous characters, effectively correcting them to the correct characters. As shown in Example 1, PSDSpell avoids mistakenly changing the correct character "哪里" (where) to the more common character "那里" (there), while the baseline model tends to make this substitution, resulting in incorrect correction. In Example 2, the baseline model is more inclined not to make any changes, but "汉子" (man) and "汉字" (Chinese character) are homophones, and "汉字" (Chinese character) is more consistent with the context. Therefore, PSDSpell modifies

Table 9 Some special cases that the model is unable to correct include instances such as errors in proper nouns and errors related to common sense.

Input	耳室(room) 症如何治疗。
Baseline	耳室(room) 症如何治疗。
PSDSpell(Ours)	耳室(room) 症如何治疗。
Input	氨基已(already) 酸是一种有机物
Baseline	氨基已(already) 酸是一种有机物
PSDSpell(Ours)	氨基已(already) 酸是一种有机物
Input	中国的首都是上海(Shanghai)
Baseline	中国的首都是上海(Shanghai)
PSDSpell(Ours)	中国的首都是上海(Shanghai)

the erroneous character, demonstrating a higher sensitivity to erroneous characters. In Example 3, there are three consecutive erroneous chracters, and PSDSpell successfully avoids the influence of the erroneous character context, changing the sequence of incorrect characters "姓青号" (name Qinghao) to "心情好" (a good mood), maintaining a smooth semantic context. This is also attributed to our pretraining strategy and the single-channel masking mechanism.

PSDSpell achieved promising results on the SIGHAN test dataset. However, as shown in Table 9, we observed that in certain specialized domains, such as "耳室 (shi, room) 症" (Otolithiasis, correct spelling: "耳石 (shi, stone) 症", a medical condition), and "氨基已 (yi, already) 酸" (Aminocaproic Acid, correct spelling: "氨基己 (ji, oneself) 酸", an organic compound), neither PSDSpell nor the baseline were able to correct the erroneous characters. Furthermore, both PSDSpell and the baseline also struggled with addressing common knowledge, for example: "中国的首都是上海" (which means "The capital of China is Shanghai", the correct expression: "The capital of China is Beijing"). How to enable the model to acquire knowledge in specialized domains remains an intriguing question worthy of exploration.

5. Conclusions

This paper proposes a Chinese spelling correction model called PSDSpell. We employ the self-distillation learning strategy to learn the contextual distribution from a teacher model, enabling the model to encounter a more significant number of multi-error samples during pretraining. We utilize a single-channel masking mechanism and an iterative correction strategy to enhance the model's performance on multi-error samples. The model employs a detection network to identify potential erroneous characters' positions and iteratively corrects them using a correction network. Experimental results on the SIGHAN dataset demonstrate that PSDSpell outperforms the baseline model. In the future, we plan to explore integrating external knowledge to enable the model to handle errors in specialized domains.

Acknowledgments

This work is supported by National Key Research and Development Program of China (2020AAA0109700), National Natural Science Foundation of China (62076167), the National Natural Science Foundation of China (61972003), R&D Program of Beijing Municipal Education Commission (KM202210009002), and the Beijing Urban Governance Research Base of North China University of Technology (2023CSZL16). We would also like to thank the anonymous reviewers for their helpful comments. We would like to thank the referees for their comments, which helped improve this paper considerably.

References

[1] C.L. Liu, M.H. Lai, Y.H. Chuang, and C.Y. Lee, "Visually and phonologically similar characters in incorrect simplified Chinese words," Coling 2010: Posters, Beijing, China, pp.739–747, Aug. 2010.

[2] C. Li, C. Zhang, X. Zheng, and X. Huang, "Exploration and exploitation: Two ways to improve Chinese spelling correction models," Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Online, pp.441–446, Association for Computational Linguistics, Aug. 2021.

[3] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, pp.4171–4186, June 2019.

[4] S.H. Wu, C.L. Liu, and L.H. Lee, "Chinese spelling check evaluation at SIGHAN bake-off 2013," Proc. Seventh SIGHAN Workshop on Chinese Language Processing, Nagoya, Japan, pp.35–42, Oct. 2013.

[5] L.C. Yu, L.H. Lee, Y.H. Tseng, and H.H. Chen, "Overview of SIGHAN 2014 bake-off for Chinese spelling check," Proc. Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, pp.126–132, Oct. 2014.

[6] Y.-H. Tseng, L.-H. Lee, L.-P. Chang, and H.-H. Chen, "Introduction to SIGHAN 2015 bake-off for Chinese spelling check," Proc. Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, pp.32–37, Association for Computational Linguistics, July 2015.

[7] S. Liu, T. Yang, T. Yue, F. Zhang, and D. Wang, "PLOME: Pre-training with misspelled knowledge for Chinese spelling correction," Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, pp.2991–3000, Association for Computational Linguistics, Aug. 2021.

[8] H.-D. Xu, Z. Li, Q. Zhou, C. Li, Z. Wang, Y. Cao, H. Huang, and X.-L. Mao, "Read, listen, and see: Leveraging multimodal information helps Chinese spell checking," Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, pp.716–728, Association for Computational Linguistics, Aug. 2021.

[9] R. Zhang, C. Pang, C. Zhang, S. Wang, Z. He, Y. Sun, H. Wu, and H. Wang, "Correcting Chinese spelling errors with phonetic pre-training," Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, pp.2250–2261, Association for Computational Linguistics, Aug. 2021.

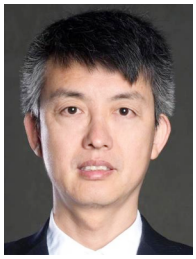
[10] Z. Sun, X. Li, X. Sun, Y. Meng, X. Ao, Q. He, F. Wu, and J. Li, "ChineseBERT: Chinese pretraining enhanced by glyph and pinyin information," Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, pp.2065–2075, Association for Computational Linguistics, Aug. 2021.

[11] Y. Jiang, T. Wang, T. Lin, F. Wang, W. Cheng, X. Liu, C. Wang, and W. Zhang, "A rule based Chinese spelling and grammar detection system utility," 2012 International Conference on System Science

and Engineering (ICSSE), pp.437–440, 2012.
[12] Y.-R. Wang and Y.-F. Liao, "Word vector/conditional random field-based Chinese spelling error detection for SIGHAN-2015 evaluation," Proc. Eighth SIGHAN Workshop on Chinese Language Processing, Beijing, China, pp.46–49, Association for Computational Linguistics, July 2015.
[13] Q. Huang, P. Huang, X. Zhang, W. Xie, K. Hong, B. Chen, and L. Huang, "Chinese spelling check system based on tri-gram model," Proc. Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, pp.173–178, Association for Computational Linguistics, Oct. 2014.
[14] C. Zhu, Z. Ying, B. Zhang, and F. Mao, "MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction," Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, pp.1244–1253, Association for Computational Linguistics, May 2022.
[15] S. Liu, S. Song, T. Yue, T. Yang, H. Cai, T. Yu, and S. Sun, "CRASpell: A contextual typo robust approach to improve Chinese spelling correction," Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, pp.3008–3018, Association for Computational Linguistics, May 2022.
[16] Y. Gao, J.-X. Zhuang, S. Lin, H. Cheng, X. Sun, K. Li, and C. Shen, "DisCo: Remedying self-supervised learning on lightweight models with distilled contrastive learning," Computer Vision – ECCV 2022, Cham, pp.237–253, Springer Nature Switzerland, 2022.
[17] L. Zhang, J. Song, A. Gao, J. Chen, C. Bao, and K. Ma, "Be your own teacher: Improve the performance of convolutional neural networks via self distillation," Proc. IEEE/CVF International Conference on Computer Vision (ICCV), Oct. 2019.
[18] H. Lee, S.J. Hwang, and J. Shin, "Rethinking data augmentation: Self-supervision and self-distillation," arXiv preprint arXiv:1910.05872, 2019.
[19] Y. Wang, S. Lin, Y. Qu, H. Wu, Z. Zhang, Y. Xie, and A. Yao, "Towards compact single image super-resolution via contrastive self-distillation," arXiv preprint arXiv:2105.11683, 2021.
[20] K. Clark, M. Luong, Q.V. Le, and C.D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," arXiv preprint arXiv:2003.10555, 2020.
[21] J. Li, Q. Wang, Z. Mao, J. Guo, Y. Yang, and Y. Zhang, "Improving Chinese spelling check by character pronunciation prediction: The effects of adaptivity and granularity," Proc. 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, pp.4275–4286, Association for Computational Linguistics, Dec. 2022.
[22] D. Wang, Y. Song, J. Li, J. Han, and H. Zhang, "A hybrid approach to automatic corpus generation for Chinese spelling check," Proc. 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, pp.2517–2527, Association for Computational Linguistics, Oct.-Nov. 2018.
[23] X. Cheng, W. Xu, K. Chen, S. Jiang, F. Wang, T. Wang, W. Chu, and Y. Qi, "SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check," Proc. 58th Annual Meeting of the Association for Computational Linguistics, Online, pp.871–881, Association for Computational Linguistics, July 2020.

Li He is an associate professor, graduated from Yanshan University in 2002 with a master's degree. Now she works in the Department of Computer Science, North China University of Technology. The main research interests include data warehouse and data mining, large database processing.

Xiaowu Zhang is a master student in College of Informatics, North China University of Technology. His major research field is Natural Language Processing and Knowledge Graph.



Jianyong Duan is a professor, born in 1978. He graduated from Department of computer science, Shanghai Jiao Tong University by 2007. His major research field includes natural language processing and information retrieval.

Hao Wang received the Ph.D. degree in Computer Application Technology from Tsinghua University in 2013. He is now an associate professor in College of Informatics, North China University of Technology. His research interests include machine learning and data analysis.

Xin Li received the Ph.D. degree in Physics, Electrical and Computer Engineering from Yokohama National University in 2020. He is now a lecturer in College of Informatics, North China University of Technology. His research interests include knowledge extraction from nonuniform skewed data, deep learning, and artificial intelligence applications.

Liang Zhao received the Bachelor's degree from Xi'dian University, RXi'an, China, in 2011, and then received the Ph.D. degree from Tsinghua University, Beijing, China, in 2017. Now she is an Associate Professor in School of Information Management, Wuhan University, Hubei, China. Her research interests include context-aware data management toward ambient intelligence, computational psychology in social network, and digital humanities.