



Article MLSL-Spell: Chinese Spelling Check Based on Multi-Label Annotation

Liming Jiang ^(D), Xingfa Shen *, Qingbiao Zhao and Jian Yao

School of Computer Science, Hangzhou Dianzi University, Hangzhou 310018, China; 211050118@hdu.edu.cn (L.J.); sad@hdu.edu.cn (Q.Z.); yaojian_0105@hotmail.com (J.Y.) * Correspondence: shenxf@hdu.edu.cn

Abstract: Chinese spelling errors are commonplace in our daily lives, which might be caused by input methods, optical character recognition, or speech recognition. Due to Chinese characters' phonetic and visual similarities, the Chinese spelling check (CSC) is a very challenging task. However, the existing CSC solutions cannot achieve good spelling check performance since they often fail to fully extract the contextual information and Pinyin information. In this paper, we propose a novel CSC framework based on multi-label annotation (MLSL-Spell), consisting of two basic phases: spelling detection and correction. In the spelling detection phase, MLSL-Spell uses the fusion vectors of both character-based pre-trained context vectors and Pinyin vectors and adopts the sequence labeling method to explicitly label the type of misspelled characters. In the spelling correction phase, MLSL-Spell uses Masked Language Mode (MLM) model to generate candidate characters, then performs corresponding screenings according to the error types, and finally screens out the correct characters through the XGBoost classifier. Experiments show that the MLSL-Spell model outperforms the benchmark model. On SIGHAN 2013 dataset, the spelling detection F1 score of MLSL-Spell is 18.3% higher than that of the pointer network (PN) model, and the spelling correction F1 score is 10.9% higher. On SIGHAN 2015 dataset, the spelling detection F1 score of MLSL-Spell is 11% higher than that of Bert and 15.7% higher than that of the PN model. And the spelling correction F1 of MLSL-Spell score is 6.8% higher than that of PN model.

Keywords: Chinese spelling check (CSC); sequence labeling; Masked Language Model (MLM); XGBoost

1. Introduction

In practical scenarios such as Optical Character Recognition (OCR) and input methods, Chinese spelling errors frequently occur due to the similarities in both pronunciation and visual appearance of Chinese characters. Specifically, misspelled characters might arise when utilizing optical character recognition, speech recognition, or Chinese input methods such as Pinyin input (phonetic-based), Wubi input (shape-based), and handwriting input. An important step to tackle the problem of Chinese spelling errors is the CSC model's performance in computational efficiency and accuracy. According to Liu et al. [1], sound similarity and shape similarity account for 83% and 48% of spelling errors, respectively. Therefore, the main task of the CSC is aiming to detect and correct misspelled characters with similar pronunciation or similar shape. Currently, the CSC is still a challenging task. In the existing CSC models, the effects of spelling check are far from satisfactory, owing to the model's inability to learn Chinese semantics and the difficulty to screen out the correct characters from a large number of candidate characters.

Compared with the effective spelling check based on rules and vocabulary in English, the CSC task is more arduous since there is no clear delimiter between words in a Chinese sentence. For a single English word, multiple Chinese characters might be required to represent the same idea. For example, when expressing the English word "China" in Chinese,



Citation: Jiang, L.; Shen, X.; Zhao, Q.; Yao, J. MLSL-Spell: Chinese Spelling Check Based on Multi-Label Annotation. *Appl. Sci.* **2024**, *14*, 2541. https://doi.org/10.3390/app14062541

Academic Editor: Christos Bouras

Received: 26 January 2024 Revised: 8 March 2024 Accepted: 8 March 2024 Published: 18 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). we must employ a combination of two Chinese characters. The former pronunciation is " $zh\bar{o}ng$ " which means 'middle', and the latter pronunciation is "guó" which means 'country'. If feature vectors based on character granularity are used, the relationship between the current character and other characters in the sentence cannot be fully extracted.

While the semantic integrity of words can be ensured by using word granularitybased feature vector extraction, it involves text segmentation, which is prone to errors. For example, the Chinese phrase "yī gè tiān zhēn de hái zi" translates to 'an innocent child' in English. This phrase conveys the idea of a child who is pure and without guilt. More specifically, the Chinese term " $y\bar{i}$ gè" translates to 'an' in English, while " $t\bar{i}an \ zhen \ de$ " and "hái zi" translate to 'innocent' and 'child', respectively. However, if the Chinese word "tiān zhēn de" is split into "tiān" and "zhēn de", their English translations are 'sky' and 'true' respectively. This can result in a meaning that is significantly different from the original sentence. As shown in Figure 1, if this Chinese phrase is segmented incorrectly as 'yi ge/tian/zhen de/hai zi' (lit. 'an/sky/true/child'), the resulting sentence will be semantically unintelligible and difficult to understand. Therefore, "tiān zhēn de" (meaning 'innocent') should be used as an inseparable phrasal adjective. If training is carried out with the incorrect word segmentation results, the semantic output of the word vectors after training will have a certain deviation, which will, to some extent, influence the CSC task. Therefore, Chinese spelling detection is pivot to Chinese spelling correction. For the detection phase, it is significantly more difficult to detect spelling errors at the character level than it is at the sentence and phrase level.



Figure 1. The graph's upper section illustrates the correct segmentation of words, while the lower section depicts incorrect word partitioning.

Furthermore, the CSC requires some background information and contextual reasoning ability to obtain a desirable spelling correction performance. Two sets of examples are shown in Table 1. In the first example, the word "vector" is a mathematical term that is wrongly transcribed as "elephant" because they have similar pronunciations "xiang" in Chinese. This example shows that the CSC requires certain background information to detect and correct spelling errors. In the second example, the words "late" and "equator" are pronounced "chi dao" in Chinese, so that they are misused. By incorporating the Chinese context "shang xué" (lit. 'go to school'), it can be deduced that the correct word should be "late".

Table 1. Examples of Chinese spelling errors. Words added in red are incorrect and should be corrected to words added in green.

| Wrong | Correct | Similar Pronunciation in Chinese |
|----------------------------------------------------------------------|--------------------------------------------------------------------|-------------------------------------|
| A elephant refers to a quantity with magnitude and direction. | A vector refers to a quantity with magnitude and direction. | xiàng |
| He is always <mark>equator</mark> for school. | He is always late for school. | chi dao |

The CSC models can be roughly divided into two categories: the traditional CSC models and those based on deep learning. The traditional CSC models share a similar pipeline scheme: first, use the word segmentation tool to segment the sentence, then replace the suspicious Chinese characters with the confusion set, and finally score the sentence via language models and select the sentence with the highest score as the CSC result. Liu et al. [2] developed a hybrid CSC model, in which candidate characters for the misspelled character are generated first by using models based on statistical machine translation and language, and then the correct candidate character is screened out via SVM [3] classifier. Based on N-gram, traditional models tend to have limitations since the N-gram language model can only extract limited history information and cannot capture the future information. For the CSC models based on deep learning, Wang et al. [4] proposed a sequence labeling-based method for Chinese spelling detection. Their spelling detection approach is innovative, but the feature vectors they adopted cannot adequately convey the relationship between the current characters and other characters in a sentence.

There are a lot of work using the sequence-to-sequence models, and they achieved good performance. However, most of the sequence-to-sequence models based on Bert are faced with the problem of overcorrection. When using the form of "error check first, then error correction", the error correction model will obtain a priori information about the position of wrong characters. It can enable the model to avoid misjudgment that may be caused by context information.

In this paper, we propose a novel CSC framework Based on Multi-label Annotation (MLSL-Spell) that divides the CSC task into two sub-tasks: spelling detection and spelling correction. First, the MLSL-Spell detection model takes Chinese characters as the unit, fuses the pre-trained context vectors and the Pinyin vector, and delivers them to the neural network model for sequence labeling. After fully learning the contextual information between the characters, the detection model outputs the labeled character sequence, which corresponds to the Chinese characters in the original sentence. If the current Chinese character is correct, it will be labeled as "T". If the current Chinese character is detected as an error of similar pronunciation, it will be labeled as "P"; if it is detected as an error of similar pronunciation, it will be labeled as "B"; if the detected error belongs both to similar pronunciation and shape, it is labeled as "B"; if it is an error of other types, it will be labeled as "O".

As for the spelling correction module of MLSL-Spell, it receives the character sequence output from the spelling detection module, and corrects the misspelled Chinese characters that are detected. MLSL-Spell employs the Masked Language Model (MLM) [5] model to replace the wrong character with "[MASK]", infers the character at the wrong position through the contextual information, generates possible candidates for the misspelled character, and then performs corresponding screenings according to the types of error to generate the final candidates. For different types of spelling errors, MLSL-Spell adopts different candidate strategies to extract character features. In the end, MLSL-Spell uses XGBoost [6] classifier to screen out the correct characters.

We conducted experiments on three public datasets. The results reveal that the MLSL-Spell model outperforms all other contrast models in terms of CSC's indicators. The contributions of this paper are as follows:

1. In order to fully extract the contextual information and correct the errors according to the corresponding error types, we propose MLSL-Spell, a CSC model based on pre-training context vectors and multi-label annotation.

2. Taking into account the Pinyin information of Chinese characters and the contextual information between characters, the spelling detection module of MLSL-Spell fuses the pre-trained context vectors and Pinyin vectors and uses multiple tags for sequence labeling. Moreover, its spelling correction module uses MLM model and the XGBoost classifier to screen out the correct characters.

3. Compared with the CSC model proposed recently by Wang et al. [7], MLSL-Spell model has better CSC performance on two public datasets. On SIGHAN 2013 dataset, the

spelling detection F1 score of MLSL-Spell is 18.3% higher than that of the pointer network (PN) model, and the spelling correction F1 score is 10.9% higher. On SIGHAN 2015 dataset, the spelling detection F1 score of MLSL-Spell is 15.7% higher than that of the PN model, and the spelling correction F1 score is 6.8% higher.

The rest of the paper is organized as follows. In Section 2, the literature on the existing CSC models, including traditional ones and deep learning-based ones, is given. In Section 3, we describe our proposed MLSL-Spell model. Section 4 presents the experimental setup and experimental results. Finally, we provide conclusions and a summary in Section 6.

2. Related Work

Due to the significance of CSC tasks in downstream processes such as OCR and input methods, an increasing number of researchers are dedicating their efforts to this field. For traditional CSC models, Xie et al. [8] proposed a model of joint bigram, trigram, and Chinese word segmentation. Liu et al. [2] combined the candidate sets selected respectively by the language model based on word segmentation and the statistical translation model and re-scored the correct characters with the SVM classifier. Yu et al. [9] employed a characterbased N-gram language model to detect potential misspelled characters with probability below the predefined threshold, and to generate a candidate set of similar pronunciation and shape for each potential misspelled character; then they screened out the candidate characters with the highest probability through the language model. Chiu et al. [10] devised a CSC method based on similar pronunciation or shape. It relies on a Web corpus to classify similar characters and uses a character-based language model in the channel model and noise model to correct spelling errors. Jia et al. [11] applied a graph model to the CSC task and performed a single-source shortest path algorithm on the graph to correct spelling errors. Han et al. [12] approached CSC by training a maximum entropy model on a large corpus, treating CSC as a binary classification task. Xiong et al. [13] proposed a method based on Logistic Regression (LR), which used confusion sets to replace text to generate new sentences, then extracted the text features of the new sentences, and used the LR model to screen out the correct sentences.

For CSC models based on deep learning, Duan et al. [14] introduced a new neural network architecture integrating bidirectional LSTM model and CRF model, which took the character sequence of the sentence as input. The bidirectional LSTM layer first learns the character sequence information before sending the probability vectors to CRF layer, which then outputs best-predicted label sequence as the spelling detection result. And then there is a FL-LSTM-CRF model (Wang et al. [15]). As an extension of the LSTM-CRF model, it combines word lattice, character, and Pinyin information to perform Chinese spelling detection. Wang et al. [4] put forward a novel method CSC dataset construction approach based on OCR and ASR. Then, in order to verify the validity of the dataset, they employed a sequence labeling-based approach to detect Chinese spelling. Moreover, Wang et al. [7] designed an end-to-end pointer network model, through which the correct character could be copied from the input character list or generated from the confusion set instead of from the entire vocabulary. Hong et al. [16] implemented a CSC model composed of an autoencoder and a decoder. The model has a simple structure, and faster calculation speed, and is easier to adapt to simplified or traditional Chinese text generated by humans or machines. Zhang et al. [17] based their CSC model on BERT. The spelling detection network is connected to the spelling correction network through soft-masking technology. Chen et al. [18] integrated Pinyin and character similarity knowledge into the language model via a customized graph convolutional neural network. Later, an end-to-end trainable model emerged (Huang et al. [19]), which uses multichannel information to improve the performance of CSC. A chunk-based framework for uniformly correcting single-character and multi-character word errors was presented by Bao et al. [20]. Gou et al. [21] implemented performing a post-processing operation on the error correction tasks. Nguyen et al. [22] proposed a scalable adaptable filter that exploits hierarchical character embeddings. A significant amount of work emerged

subsequently, such as Zhang et al. [23], Wang et al. [24], PLOME [25], REALISE [26], SpellBERT [27], LEAD [28], Liang et al. [29], and PTCSpell [30], attempting to integrate glyph or pronunciation information into models. For example, Zhang et al. [23] designed an end-to-end model that integrated phonetics into the language model by leveraging the powerful pre-training and fine-tuning method. REALISE [26] used a gating mechanism to fuse visual and pronunciation information. PTCSpell [30] designed two novel pre-training objectives to capture pronunciation and shape information in Chinese characters. Another segment of research, including ECOPO [31], CRASpell [32], CoSPA [33], EDMSpell [34], and Wu et al. [35], focused on addressing the issue of the overcorrection in the model. For instance, ECOPO [31] introduced contrast learning to the CSC task. EDMSpell [34] reduced overcorrection of the model through post-processing. Wu et al. [35] employed different masking and substitution strategies to obtain a better language model.

3. Approach

In this section, we will introduce our CSC model in detail. Here are the steps of performing CSC tasks in MLSL-Spell model: firstly, initialize randomly the Pinyin vectors and then fuse the pre-trained context vectors and the Pinyin vectors; secondly, send the fusion vectors to the sequence labeling framework composed of bidirectional GRU [36] neural network and CRF [37] model for multi-label annotation.Compared with LSTM, GRU has fewer parameters and uses a simpler structure to achieve the same effect. And the calculation efficiency of GRU is higher. For the original RNN, it is difficult to obtain the relevant information about the long distance in the sentence.; thirdly, generate candidate characters for the misspelled characters through MLM model; fourthly, perform corresponding screening according to the error types, and generate the final candidate characters. In the end, for different error types, MLSL-Spell adopts varied feature extraction strategies of candidate character. After extracting the features of the candidate characters, MLSL-Spell employs XGBoost classifier to screen out the correct characters. The framework of MLSL-Spell model is shown in Figure 2.



Figure 2. The overall framework structure.

MLSL-Spell's spelling detection module is shown in Figure 3. Suppose the sentence to be corrected is formulated as

$$X = \{x_1, x_2, \ldots, x_n\}$$

and the label sequence output after spelling detection as $Y = \{y_1, y_2, ..., y_n\}$, where x_i represents a Chinese character, y_i denotes "T", "P", "S", "B" or "O", and *i* refers to the position of the Chinese character in the sentence. We fuse the context vector e_i and the Pinyin vector p_i (each pair of e_i and p_i corresponds to one character in the sentence) into the vector e_i^f , and input it to the bidirectional GRU model. The calculation formula of the fusion vector is as follows:

$$e_i^f = [e_i; p_i] \tag{1}$$



Figure 3. Architecture of MLSL-Spell spelling detection module.

The hidden layer vector $\overline{h_i}$ of a certain direction at the current time step depends on the fusion vector e_i^f at the current time step and the hidden layer vector $\overline{h_{i\pm 1}}$ at the previous time step. We concatenate the hidden layer vectors in each direction as the output vector of the BiGRU model. The calculation formulas are shown below.

$$\overrightarrow{h_{i-1}} = BiGRU\left(\overrightarrow{h_{i-1}}, e_i^f\right)$$
(2)

$$\overleftarrow{h_{i+1}} = BiGRU\left(\overleftarrow{h_{i+1}}, e_i^f\right) \tag{3}$$

$$h_i^f = \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right] \tag{4}$$

$$y_{pred} = argmax \left(s(X, \tilde{y})_{\tilde{y} \in y_X} \right)$$
(5)

In the bidirectional GRU model, the hidden layer vectors fully learn the contextual information under the current sentence meaning. Next, MLSL-Spell sends the trained hidden layer vectors to CRF model to label the text in sequence. We use a linear function to normalize the hidden layer vector at each time step and obtain the respective scores of the five labels. With these scores attained, we can calculate the emission matrix of CRF,

 $E \in R^{n \times k}$, where *n* is the length of the input sentence, and *k* represents the number of the labels. In this paper, we define 5 subtypes of the sequence label: "T", "P", "S", "B" and "O", representing the correct character, the misspelled character of similar pronunciation, misspelled character of similar shape, misspelled character of both similar pronunciation and similar shape, and the misspelled character caused by other reasons, respectively. Therefore, *k* equals 5. The calculation formula of CRF model for constructing the emission matrix is as follows:

$$P_{i,y_i}(y_i = j, 0 \le j \le 4|x_i) = Softmax(Wh_i + b)[j]$$

$$\tag{6}$$

CRF model calculates the scores of sequence labels through the emission matrix and the label transfer matrix. The calculation formulas are as follows:

$$s(X,y) = \sum_{i=0}^{n} T_{y_i,y_{i+1}} + \sum_{i=1}^{n} P_{i,y_i}$$
(7)

$$P(y|X) = \frac{e^{s(X,y)}}{\sum_{\tilde{y} \in y_{all}} e^{s(X,\tilde{y})}}$$

$$\tag{8}$$

We adopt the negative logarithmic maximum likelihood function as the loss function of the model, as shown below:

$$loss = -\log p(y|X) \tag{9}$$

y represents the real label corresponding to *X*. In the end, we adopt the label sequence with the highest probability as the spelling detection result of MLSL-Spell, which is formulated as follows:

$$y_{pred} = argmax \left(s(X, \tilde{y})_{\tilde{y} \in y_x} \right)$$
(10)

In addition, we use the adamW optimizer to optimize it, which has the L2 regularization and higher computational efficiency than the adam optimizer.

3.2. Spelling Correction of MLSL-Spell

For the misspelled characters detected by MLSL-Spell model, we replace them with "[MASK]", and then use MLM model to infer candidate characters at the masking positions, from which we select Top 100 characters as candidate characters. If the misspelled Chinese character at the current position is a type of similar pronunciation, candidates will be screened by a similar-pronunciation confusion set. If it is a type of similar shape, candidates will be screened by a similar-shape confusion set. If it is a type of similar pronunciation and shape, candidates will be screened by the confusion set of both similar pronunciation and shape.

The number of candidate character should not be too large or too small. If it is too small, there will be fewer candidates available for the model, and the error correction ability of the model will decrease theoretically and intuitively. At the same time, if it is too large, the larger value of k makes the model get more possible candidates. But the possibility of these candidates is so very low that they can provide little help to the model. And it will increase the computational of the model to a certain extent. Based on this experience, we decided to set the number of candidate character to 100.

Additionally, We use pre-trained MLM model and it is not trained separately. But we don't need to solve the problem of domain adaptation. Our error correction model has obtained a priori knowledge of the error location. In this case, we just need to mark the error characters as "[MASK]" and send the sentence to the MLM model. When using the MLM model, our tasks and goals are consistent with the original MLM model.

We extract the following features from the final candidate characters: (1) the change in the number of word segmentation before and after replacing the misspelled character with the candidate character; (2) the change in the perplexity of the sentence before and after replacing the misspelled character with the candidate character; (3) the Pinyin edit distance between the candidate character and the misspelled character; (4) whether or not the misspelled character is a stop word. After extracting the features of the candidate characters, we use the trained XGBoost classifier to screen out the correct characters. The architecture of MLSL-Spell's spelling correction module is shown in Figure 4.



Figure 4. Architecture of MLSL-Spell spelling correction module.

4. Experiments

4.1. Dataset

Train data The train data used by MLSL-Spell model for spelling detection are from the misspelling dataset constructed by Wang et al. [4] and the training sets of SIGHAN 2013, SIGHAN 2014 and SIGHAN 2015 provided in the SIGHAN competition. MLSL-Spell model uses the SIGHAN training sets as the training data for the XGBoost classifier. SIGHAN datasets are written in Traditional Chinese, so we use OpenCC (https://github.com/BYVoid/OpenCC (accessed on 1 March 2023)) to convert them into Simplified Chinese. The statistics of train data are shown in Table 2.

Table 2. Statistics of CSC datasets. #Sent represents the total number of sentences in the corresponding dataset.

 #Errors represents the total number of misspelled characters in the corresponding dataset.

| Training Data | #Sent | Avg.Length | #Errors |
|-------------------------|---------|------------|---------|
| (Wang et al., 2018 [4]) | 271,329 | 44.4 | 382,704 |
| SIGHAN 2013 | 350 | 49.2 | 350 |
| SIGHAN 2014 | 6526 | 49.7 | 10,087 |
| SIGHAN 2015 | 3174 | 30.0 | 4237 |
| Total | 281,379 | 44.4 | 397,378 |
| Test Data | #Sent | Avg.Length | #Errors |
| SIGHAN 2013 | 1000 | 74.1 | 1227 |
| SIGHAN 2014 | 1062 | 50.1 | 782 |
| SIGHAN 2015 | 1100 | 30.5 | 715 |

Test data MLSL-Spell model uses the test sets of SIGHAN 2013, SIGHAN 2014 and SIGHAN 2015 as the test data for spelling detection and correction. The statistics of test data are shown in Table 2.

Corpus MLSL-Spell model uses People's Daily Segmented Corpus (2014 edition) to build a 4-gram language model based on word granularity. We adopt the Modified Kneser-Ney [38] smoothing algorithm in the language model.

Confusion Set MLSL-Spell model uses the confusion set provided by Chen et al. [18].

Evaluation Metrics We adopt the evaluation criteria proposed by Liu et al. [2], and use the evaluation metrics of precision, recall rate, and F1 score to evaluate the performance of the spelling detection and correction of the model. The calculation formulas are as follows:

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN}$$
(12)

$$F_1 = \frac{2 * Precision * Recall}{Precisio + Recall}$$
(13)

where TP means that the model correctly corrects errors in the sentence, FP means that the model incorrectly changes the correct part of the sentence, and FN represents that the model does not correct errors in the sentence.

Benchmark Models We compare the following 11 models in terms of spelling correction or spelling detection. The first 3 are benchmark models, followed by 7 ablation experiments of MLSL-Spell model, and the last model is used to explore the impact of the data set on the model.

- LMC Model Xie et al. [8] used the confusion set provided by the SIGHAN competition to replace the suspicious characters, adopted the traditional N-gram language model to score the perplexity of the replaced sentences, and selected the sentence with the highest score as the CSC result. We call this method LMC.
- **SL Model** Wang et al. [4] used sequence labeling to detect positions of misspelled characters. The correct position is labeled as 1, and the wrong position is labeled as 0. Sequence labeling is abbreviated as SL.
- **PN Model** Wang et al. [7] proposed an end-to-end pointer network model, through which the correct character is copied from the input character list or generated from the confusion set instead of from the entire vocabulary. Pointer network is abbreviated as PN.
- **Bert** Devlin et al. [5] proposes a framework from pre-training to fine-tuning. The Model is pre-trained on a large corpus and then fine-tuned when it is used for specific tasks.
- **FASpell** Hong et al. [16] utilizes a denoising autoencoder (DAE) to generate candidate characters.

4.2. Hyper-Parameters

MLSL-Spell model uses BERT [5] pre-trained context vectors and Pinyin vectors, which are 768 dimensional and 128-dimensional vectors respectively, and it uses AdamW [39] to optimize the objective function. We set the Batch size and learning rate as 32 and 3×10^{-5} , respectively, and train MLSL-Spell's spelling detection model for 12 epochs. The learning rate of XGBoost classifier and LightGBM classifier is set as 0.01, and the remaining parameters are set as default. Logistic Regression, K-Nearest Neighbor, and Support Vector Machines classifiers all use default parameters.

4.3. Main Results

Table 3 shows that LMC model and SL model perform similarly in CSC, but far worse than PN model and MLSL-Spell model. On the SIGHAN 2014 and SIGHAN2015 test sets, SL model achieves better performance in spelling detection than LMC model. On the SIGHAN 2013 test set, SL model and LMC model gain comparable performance. On the SIGHAN 2014 test set, the spelling detection F1 score of SL model is 15.2% higher than that of LMC model; on the SIGHAN 2015 test set, the spelling detection F1 score of SL model is 22.3% higher than that of LMC model.

method of conducting sequence labeling using neural network and CRF model is effective, contributing to a significant improvement over the traditional method based on word segmentation. Although both SL model and MLSL-Spell model use sequence labeling-based method to detect spelling errors, MLSL-Spell model outperforms than SL model. Specifically, on the SIGHAN 2013 test set, the spelling detection F1 score of MLSL-Spell model is 27.7% higher than that of SL model; on the SIGHAN 2014 test set, the spelling detection F1 score of MLSL-Spell model is 27.7% higher than that of SL model is 19.6% higher than that of SL model; on the SIGHAN 2015 test set, the spelling detection F1 score of MLSL-Spell model is 23.2% higher than that of SL model. MLSL-Spell model uses the fusion vectors of Pinyin vectors and character-based pre-trained context vectors (trained via BERT model on a large corpus). It is on the basis of the fusion vectors that MLSL-Spell model can efficiently capture contextual information and Pinyin information. However, the character-based vectors in SL model are initialized randomly. This shows that the adoption of fusion vectors is the most essential part in MLSL-Spell spelling detection.

Table 3. Results of MLSL-Spell model and baseline models (%). D, C represent the detection, correction, respectively. The best results for each test set are in bold.

| Methods | I | Detection-Leve | : 1 | C | orrection-Lev | el |
|----------------|------|----------------|------------|------|---------------|------|
| SIGHAN 2013 | D-P | D-R | D-F | C-P | C-R | C-F |
| LMC [8] | 79.8 | 50.0 | 61.5 | 77.6 | 22.7 | 35.1 |
| SL [4] | 54.0 | 69.3 | 60.7 | (-) | (-) | 52.1 |
| PN [7] | 56.8 | 91.4 | 70.1 | 79.7 | 59.4 | 68.1 |
| FASpell [16] | 76.2 | 63.2 | 69.1 | 73.1 | 60.5 | 66.2 |
| BERT [5] | 57.7 | 56.5 | 57.1 | 57.3 | 56.1 | 56.7 |
| MLSL-Spell | 89.2 | 87.7 | 88.4 | 79.7 | 78.3 | 79.0 |
| SIGHAN 2014 | D-P | D-R | D-F | C-P | C-R | C-F |
| LMC [8] | 56.4 | 34.8 | 43.0 | 71.1 | 50.2 | 58.8 |
| SL [4] | 51.9 | 66.2 | 58.2 | (-) | (-) | 56.1 |
| PN [7] | 63.2 | 82.5 | 71.6 | 79.3 | 68.9 | 73.7 |
| FASpell [16] | 61.0 | 53.5 | 57.0 | 59.4 | 52.0 | 55.4 |
| BERT [5] | 55.6 | 63.6 | 59.3 | 54.9 | 62.8 | 58.6 |
| MLSL-Spell | 80.7 | 75.1 | 77.8 | 66.4 | 61.8 | 64.0 |
| SIGHAN 2015 | D-P | D-R | D-F | C-P | C-R | C-F |
| LMC [8] | 83.8 | 26.2 | 40.0 | 67.6 | 31.8 | 43.2 |
| SL [4] | 56.6 | 69.4 | 62.3 | (-) | (-) | 57.1 |
| PN [7] | 66.8 | 73.1 | 69.8 | 71.5 | 59.5 | 64.9 |
| FASpell [16] | 67.6 | 60.0 | 63.5 | 66.6 | 59.1 | 62.6 |
| BERT [5] | 71.1 | 78.3 | 74.5 | 69.2 | 76.3 | 72.6 |
| MLSL-Spell | 88.6 | 82.6 | 85.5 | 74.2 | 69.3 | 71.7 |

Next, we will check the performance of MLSL-Spell model by comparing its F1 scores (as shown in Table 3) with the F1 scores of PN model on various datasets. On the SIGHAN 2013 test set, MLSL-Spell model's spelling detection F1 score (88.4%) is 18.3% higher than PN model's, and its spelling correction F1 score (79%) is 10.9% higher than PN model's. On the SIGHAN 2014 test set, MLSL-Spell model's spelling detection F1 score (64%) is 9.7% lower than PN model's. On the SIGHAN 2015 test set, MLSL-Spell model's spelling detection F1 score (64%) is 9.7% lower than PN model's. On the SIGHAN 2015 test set, MLSL-Spell model's spelling detection F1 score (71.7%) is 6.8% higher than PN model's. Despite the 9.7% poorer performance on SIGHAN 2014 test set, MLSL-Spell model, on the whole, achieves better performance than PN model in terms of spelling detection and correction, particularly on SIGHAN 2013 and 2015 test sets.

The subtask of spelling detection is extremely crucial in the CSC task. Since detecting the position of the misspelled character is a prerequisite for spelling correction, the performance of the spelling detection module directly affects the spelling correction system. In this connection, we use the fusion vectors and adopt the sequence labeling-based method for spelling detection. The experimental results prove that MLSL-Spell model outperforms the other three benchmark models in terms of spelling detection. To be specific, the maximum differences of F1 on the SIGHAN2013, 2014 and 2015 test sets are 27.7%, 34.8%, and 45.5%, respectively. For the subtask of spelling correction, we use MLM model to generate Top 100 candidates based on the context and then perform corresponding screenings based on the error type to generate the final candidate set. Following that, we screen out the candidate characters with XGBoost and take them as the spelling correction result. Experiments show that MLSL-Spell model achieves a good performance in spelling correction, and on the SIGHAN 2013, 2014 and 2015 test sets, the maximum differences of F1 are 43.9%, 7.9%, and 28.5%, respectively.

4.4. Ablation Studies

Pruning of MLSL-Spell We prune the MLSL-Spell, the pruned model is named and explained in this section.

- MLSL-Spell(-Pinyin) Model We remove the Pinyin vectors and only use characterbased pre-trained context vectors in the spelling detection module of MLSL-Spell.
- MLSL-Spell(Top 1) Model In the spelling correction system of MLSL-Spell, we adopt the candidate character with the maximum probability (generated by the MLM model) as the spelling correction result, instead of using XGBoost classifier to screen out candidate characters.
- MLSL-Spell(lg) Model We use Logistic Regression [40] classifier instead of XGBoost classifier to screen out candidates.
- MLSL-Spell(knn) Model We use K-Nearest Neighbor [41] classifier instead of XG-Boost classifier to screen out candidates.
- MLSL-Spell(svm) Model We use the Support Vector Machines [3] classifier instead of XGBoost classifier to screen out candidates.
- MLSL-Spell(lgb) Model We use LightGBM [42] classifier instead of XGBoost classifier to screen out candidates.
- MLSL-Spell(-multi label) Model We screen out candidate characters by directly using XGBoost classifier instead of considering the label types.
- MLSL-Spell(sighan) Model We only use SIGHAN2013, SIGHAN2014 and SIGHAN 2015 training sets to train MLSL-Spell model.

Effectiveness of Pinyin Vectors Pinyin information is an essential feature in CSC task and Pinyin vectors contain similar Pinyin information. We compare the spelling detection performance of MLSL-Spell model and MLSL-Spell(-Pinyin) model. The latter solely adopts character-based pre-trained context vectors. The experimental results in Table 4 indicate that on SIGHAN 2013, 2014 and 2015 test sets, MLSL-Spell's spelling detection model outperforms MLSL-Spell(-Pinyin) model on all three indicators (Precision, Recall, and F1), proving the effectiveness of the Pinyin vectors. It is worth noticing that the spelling detection performance of MLSL-Spell model in Table 4 seems poorer than that of MLSL-Spell model in Table 3. This is because MLSL-Spell detection module might detect the correct character as the misspelled one, but MLSL-Spell correction module can identify the mis-detected character. In the experiment to verify the effectiveness of Pinyin vectors, we don't need to consider the case where the spelling correction module recognizes the mis-detected characters.

| Methods | | Chinse Spelling Detection | n |
|------------------------------------|---------------------|---------------------------|---------------------|
| SIGHAN 2013 | Precision | Recall | F1 |
| MLSL-Spell (-Pinyin) MLSL-Spell | 82.1 82.4 | 87.2 87.7 | 84.6 85.0 |
| SIGHAN 2014 | Precision | Recall | F1 |
| MLSL-Spell (-Pinyin) MLSL-Spell | 60.4 63.8 | 76.1 75.1 | 67.4 69.0 |
| SIGHAN 2015 | Precision | Recall | F1 |
| MLSL-Spell (-Pinyin) MLSL-Spell | 69.1 70.9 | 81.8 82.6 | 74.9 76.3 |

Table 4. Comparison of spelling detection performance between MLSL-Spell model and MLSL-Spell(-Pinyin) model. The best results for each test set are in bold.

Effectiveness of Spelling Correction Module We compare the correction performance of MLSL-Spell model and MLSL-Spell (Top 1) model, the latter of which adopts candidate characters with the maximum probability (generated by MLM model) as the result of spelling correction. The experimental results shown in Table 5 present that three indicators (Precision, Recall and F1) of MLSL-Spell's spelling correction module are better than those of MLSL-Spell (Top 1) model on the test sets of SIGHAN 2013, 2014 and 2015. With these results, the effectiveness of MLSL-Spelling's spelling correction module is validated.

Table 5. Comparison of spelling correction performance between MLSL-Spell model and MLSL-Spell(Top 1) model. The best results for each test set are in bold.

| Methods | C | hinses Spelling Correction | on |
|----------------------------------|---------------------|----------------------------|---------------------|
| SIGHAN 2013 | Precision | Recall | F1 |
| MLSL-Spell (Top 1) MLSL-Spell | 63.6 79.7 | 63.9 78.3 | 63.8 79.0 |
| SIGHAN 2014 | Precision | Recall | F1 |
| MLSL-Spell (Top 1) MLSL-Spell | 37.3 66.4 | 42.0 61.8 | 39.5 64.0 |
| SIGHAN 2015 | Precision | Recall | F1 |
| MLSL-Spell (Top 1) MLSL-Spell | 40.7 74.2 | 45.8 69.3 | 43.1 71.7 |

Effectiveness of XGBoost Classifier In terms of using the classifier to screen out correct characters, we compare MLSL-Spell model, MLSL-Spell(lg) model, MLSL-Spell(knn) model, MLSL-Spell(svm) model and MLSL-Spell(lgb) model. The experimental results shown in Table 6 indicate that MLSL-Spell(knn) model and MLSL-Spell(svm) model achieve comparable performance in spelling correction, while MLSL-Spell(lg) model, MLSL-Spell(lgb) model and MLSL-Spell model gain comparable performance. Moreover, the spelling correction module of MLSL-Spell has achieved the best performance on the test sets of SIGHAN 2013, SIGHAN 2014 and SIGHAN 2015, which validates the effectiveness of using XGBoost classifier to screen out correct characters.

| Methods | Cł | ninese Spelling Correction | on |
|-----------------|-----------|----------------------------|------|
| SIGHAN 2013 | Precision | Recall | F1 |
| MLSL-Spell(lg) | 79.7 | 77.3 | 78.5 |
| MLSL-Spell(knn) | 69.3 | 65.9 | 67.6 |
| MLSL-Spell(svm) | 65.1 | 62.7 | 63.9 |
| MLSL-Spell(lgb) | 77.8 | 75.4 | 76.6 |
| MLSL-Spell | 79.7 | 78.3 | 79.0 |
| SIGHAN 2014 | Precision | Recall | F1 |
| MLSL-Spell(lg) | 60.5 | 61.0 | 60.8 |
| MLSL-Spell(knn) | 52.6 | 48.8 | 50.6 |
| MLSL-Spell(svm) | 59.0 | 54.8 | 56.8 |
| MLSL-Spell(lgb) | 64.2 | 59.7 | 62.0 |
| MLSL-Spell | 66.4 | 61.8 | 64.0 |
| SIGHAN 2015 | Precision | Recall | F1 |
| MLSL-Spell(lg) | 68.3 | 68.0 | 68.1 |
| MLSL-Spell(knn) | 57.0 | 53.5 | 55.2 |
| MLSL-Spell(svm) | 68.9 | 64.7 | 66.8 |
| MLSL-Spell(lgb) | 72.3 | 68.0 | 70.1 |
| MLSL-Spell | 74.2 | 69.3 | 71.7 |

Table 6. Comparison of spelling correction performance between MLSL-Spell model and MLSL-Spell(lg) model, MLSL-Spell(knn) model, MLSL-Spell(svm) model, and MLSL-Spell(lgb) model. The best results for each test set are in bold.

Effectiveness of Multi-label Sequence Annotation The misspelled characters in MLSL-Spell's spelling detection module are labeled according to error kinds using a multi-label sequence labeling method. We compare the spelling correction performance of MLSL-Spell model and MLSL-Spell(-multi label) model. The experimental results in Table 7 indicate that MLSL-Spell attains the relatively best performance in terms of three indicators (Precision, Recall, F1). To be specific, on the SIGHAN 2013 test set, the spelling correction F1 score of MLSL-Spell model is 79.0%, which is 11.1% higher than that of MLSL-Spell(-multi label) model; on the SIGHAN 2014 test set, the spelling correction F1 score of MLSL-Spell model is 64.0%, which is 2.9% higher than that of MLSL-Spell(-multi label) model; on the SIGHAN 2015 test set, the spelling correction F1 score of MLSL-Spell model is 71.7%, which is 3.4% higher than that of MLSL-Spell(-multi label) model. These results validate the effectiveness of multi-label sequence annotation.

Table 7. Comparison of spelling correction performance between MLSL-Spell model and MLSL-Spell(-multi label) model. The best results for each test set are in bold.

| Methods | C | hinese Spelling Correction | on |
|------------------------------|-----------|----------------------------|------|
| SIGHAN 2013 | Precision | Recall | F1 |
| MLSL- Spell(-multi label) | 68.4 | 67.4 | 67.9 |
| MLSL-Spell | 79.7 | 78.3 | 79.0 |
| SIGHAN 2014 | Precision | Recall | F1 |
| MLSL- Spell(-multi label) | 63.1 | 59.2 | 61.1 |
| MLSL-Spell | 66.4 | 61.8 | 64.0 |
| SIGHAN 2015 | Precision | Recall | F1 |
| MLSL- Spell(-multi label) | 70.2 | 66.4 | 68.3 |
| MLSL-Spell | 74.2 | 69.3 | 71.7 |

4.5. Impact of the Traing Set on the Model

In this subsection, we compare the spelling detection performance of MLSL-Spell model and MLSL-Spell(sighan) model on different training sets, so as to explore the influence of training sets on spelling detection. The experimental results shown in Table 8 indicate that MLSL-Spell obtains the relatively better performance in terms of three indicators of spelling detection (Precision, Recall, F1). On the SIGHAN 2013 test set, the spelling detection F1 score of MLSL-Spell model is 85.0%, which is 37.6% higher than that of MLSL-Spell(sighan) model; on the SIGHAN 2014 test set, the spelling detection F1 score of MLSL-Spell model is 20.9% higher than that of MLSL-Spell(sighan) model; on the SIGHAN 2015 test set, the spelling detection F1 score of MLSL-Spell model is 76.3%, which is 12.6% higher than that of MLSL-Spell(sighan) model. The experimental results show that the data has a great influence on the model's performance. The larger the amount of data is, the more sufficient the model learns the Pinyin and shape features of characters, as well as the contextual features, and the better the performance of spelling detection is. To sum up, using the training set proposed by Wang et al. [4] has considerably improved the spelling detection performance of MLSL-Spell model.

Methods **Chinese Spelling Detection** SIGHAN 2013 Precision Recall F1 47.3 47.7 47.4 MLSL-Spell(sighan) MLSL-Spell 82.4 87.7 85.0 SIGHAN 2014 Precision Recall F1 MLSL-Spell(sighan) 44.4 52.4 48.1 MLSL-Spell 63.8 75.1 69.0 SIGHAN 2015 Precision Recall F1 MLSL-Spell(sighan) 56.7 60.2 63.7 MLSL-Spell 70.9 82.6 76.3

Table 8. Comparison of spelling detection performance between MLSL-Spell model and MLSL-Spell(sighan) model. The best results for each test set are in bold.

4.6. Different Labels' Results of MLSL-Spell

Tables 9–11 show the spelling detection and correction performance of MLSL-Spell model with different labels on SIGHAN 2013-2015 test sets. The results reveal that MLSL-Spell model has good spelling detection and correction performance on labels "P", "S" and "B", but has poor performance on label "O". On SIGHAN 2013, 2014 and 2015 test sets, there are a large amount of labels "P" and "S", a moderate amount of label "O" and a small amount of label "S". Since label "S" counts for the smallest amount among the test sets, it is not surprising that MLSL-Spell model performs well in terms of detecting label "S", achieving 100% precision on the SIGHAN 2015 test set. The label "O" means that misspelled character is neither a similar pronunciation error nor a similar shape error, and it appears primarily because: (1) the confusion set is limited and cannot contain all the confused characters; (2) errors occur when converting Traditional Chinese into Simplified Chinese, and the converted character is not included in the confusion set; (3) the misspelled character is neither a similar pronunciation error nor a similar shape error. In this case, the model can only rely on the knowledge learned from the corpus and training set to correct the incorrect characters. Although it is feasible, improving the model's error correction performance to the label "O" can be achieved by expanding the confusion set or adding grammatical information, such as part of speech. Therefore, one of our future tasks is to expand the confusion set and integrate grammatical information, such as part of speech, into the model.

| Label | D-P | D-S | D-B | D-0 | C-P | C-S | C-B | C-0 |
|-----------|------|------|------|------|------|------|------|------|
| Precision | 87.3 | 95.5 | 94.6 | 79.2 | 83.8 | 72.7 | 79.1 | 50.6 |
| Recall | 89.2 | 86.3 | 88.1 | 74.4 | 85.7 | 65.8 | 73.6 | 47.6 |
| F1 | 88.2 | 90.6 | 91.2 | 76.7 | 84.7 | 69.1 | 76.3 | 49.1 |

Table 9. Results of MLSL-Spell model with different labels in SIGHAN 2013 (%). D, C represent the detection and correction, respectively.

Table 10. Results of MLSL-Spell model with different labels in SIGHAN 2014 (%). D, C represent the detection and correction, respectively.

| Label | D-P | D-S | D-B | D-O | C-P | C-S | C-B | C-0 |
|-----------|------|------|------|------|------|------|------|------|
| Precision | 80.3 | 77.8 | 87.8 | 70.3 | 67.9 | 50.0 | 68.3 | 60.3 |
| Recall | 75.5 | 82.4 | 79.4 | 65.0 | 63.9 | 52.9 | 61.7 | 55.8 |
| F1 | 77 8 | 80.0 | 83.4 | 67.5 | 65.9 | 51.4 | 64.8 | 58.0 |

Table 11. Results of MLSL-Spell model with different labels in SIGHAN 2015 (%). D, C represent the detection and correction, respectively.

| Label | D-P | D-S | D-B | D-0 | C-P | C-S | C-B | C-0 |
|-----------|--------------|---------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Precision | 90.4 85.8 | 100.0 85.7 | 91.9 82 5 | 75.0 70 1 | 74.2 69 3 | 91.7 78.6 | 74.5 66 9 | 65.0 60.7 |
| F1 | 88.0 | 92.3 | 86.9 | 71.1 | 71.7 | 84.6 | 70.5 | 62.8 |

4.7. Case Study

In Table 12, we give four examples, each presenting an error type to demonstrate the CSC performance of MLSL-Spell model. In order to more intuitively understand the relationship between Chinese errors and English errors, the corresponding Chinese of the sample sentences in Table 12 is shown in Figure 5. Example 1 shows an error of similar pronunciation, in which the Chinese word "middle" is corrected to "clock". As for the CSC process, MLSL-Spell model labels the misspelled Chinese character "middle" as "P", and screens out the candidate characters generated by MLM according to the pronunciation feature. Finally, it uses XGBoost classifier to correct "middle" to "clock". Example 2 represents an error of similar shape, in which "go throwing the movies" is corrected to "go to the movies". As shown in (a) and (b) in Figure 6, "throw" and "go" have similar shapes in Chinese. For the CSC process, MLSL-Spell model labels the misspelled Chinese character "throw" as "S", and corrects "throw" to "go" according to the shape feature. Example 3 displays an error produced by both similar pronunciation and similar shape, in which "seat" is corrected to "sit". As shown in (c) and (d) in Figure 6, "sit" and "seat" are visually similar and share the same pronunciation "zuo" in Chinese. However, the former is a verb, while the latter is a noun. Under this circumstance, MLSL-Spell model corrects the misspelled Chinese character "seat" to "sit" according to the pronunciation feature, shape feature and contextual information. Example 4 shows another possible type of error, which is neither of pronunciation nor of shape. As for the CSC task, MLSL-Spell model corrects the wrong Chinese word "arrived" to "walk" according to contextual information. **Table 12.** Case analysis of CSC in MLSL-Spell model. Words marked in red are incorrect and should be corrected to the words marked in green.

| Wrong | Correct | | | | |
|--------------------------------------------------------------------------------|-----------------------------------------------------------------------|--|--|--|--|
| Example 1. "P" Similar Pron | unciation in Chinese: <i>zh</i> ōng | | | | |
| I'm going to the airport to see my parents off at 8 middle tomorrow. | I'm going to the airport to see my parents off at 8 o'clock tomorrow. | | | | |
| Example 2. "S" Similar Shape | in Chinese: "throw" and "go" | | | | |
| I'm really sorry, I can't go throwing the movies with you tomorrow. | I'm really sorry, I can't go to the movies with you tomorrow. | | | | |
| Example 3. "B" Similar Pronunciatio | n and Similar Shape in Chinese: zuò | | | | |
| There are a lot of people on the bus, so we don't have a seat to seat . | There are a lot of people on the bus, so we don't have a seat to sit. | | | | |
| Example 4. "O" Other: their pronunciation and shape are not similar in Chinese | | | | | |
| We arrived for almost ten minutes and we arrived. | We walked for almost ten minutes and we arrived. | | | | |

| Example 1 | Wrong:明天早上八点 <mark>中</mark> 我要去飞机场送我爸爸妈妈。 |
|-----------|--------------------------------------------|
| | Coreect:明天早上八点钟我要去飞机场送我爸爸妈妈。 |
| Example 2 | Wrong:实在不好意思,我明天不能跟你一起 <mark>丢</mark> 看电影。 |
| | Coreect:实在不好意思,我明天不能跟你一起去看电影。 |
| Example 3 | Wrong:在公车上有很多人,所以我们没有位子可以 <mark>座</mark> 。 |
| Example 5 | Coreect:在公车上有很多人,所以我们没有位子可以坐。 |
| Example 4 | Wrong: <mark>坐</mark> 路差不多十分钟,我们到了。 |
| | Coreect:走路差不多十分钟,我们到了。 |

Figure 5. Chinese sentences corresponding to the examples in Table 12. Words marked in red are incorrect and should be corrected to the words marked in green.



Figure 6. (a–d) are the Chinese for "throw", "go", "sit" and "seat".

5. Discussion

In this study, we propose a two-step "detection and correction" model named MLSL-Spell for CSC and validate it experimentally. An important aspect that deserves further research is the simplification of MLSL-Spell. Although its multi-label approach can be helpful in the correction stage and limit overcorrection, there is still room for improvement in its computational efficiency and applicability. Our goal is to integrate the steps of error detection and correction into a sequence-to-sequence model. In addition, errors in Chinese characters are often caused by glyph or pronunciation similarities. Integrating glyph information has the potential to improve the accuracy of the model in responding to errors with similar glyphs and enhance its generalizability in areas such as OCR. Therefore, integrating glyph information into the model using a scientific approach is a promising endeavor.

6. Conclusions

We propose a novel CSC framework — MLSL-Spell. MLSL-Spell's spelling detection module can fully extract both the contextual information and Pinyin information, and employ a multi-label annotation method to label the misspelled characters in a more explicit and definite way. Based on MLM model, MLSL-Spell's spelling correction module generates the candidate characters, and then performs corresponding screenings according to error types, and finally selects candidate characters. Finally, MLSL-Spell model extracts the features of the candidate characters, and screens out the correct characters with XGBoost classifier. The experimental results reveal that MLSL-Spell model performs exceptionally well in terms of CSC's indicators. Specifically, on the SIGHAN 2013 dataset, the spelling detection F1 score of MLSL-Spell model is 18.3% higher than that of PN model, and the spelling correction F1 score is 10.9% higher; on the SIGHAN 2015 dataset, its spelling detection F1 value reaches 11% higher than that of Bert and 15.7% higher than that of PN model.

Author Contributions: Conceptualization, X.S.; methodology, L.J.; validation, L.J.; resources, J.Y.; writing—original draft preparation, Q.Z.; writing—review and editing, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant No. 2023C01143, the National Natural Science Foundation of China under Grant No. 62072121 and the Zhejiang Provincial Natural Science Foundation under Grant No. LLSSZ24F020001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- 1. Liu, C.L.; Lai, M.H.; Chuang, Y.H.; Lee, C.Y. Visually and phonologically similar characters in incorrect simplified chinese words. In *Proceedings 2010: Posters*; Coling 2010 Organizing Committee: Beijing, China, 2010; pp. 739–747.
- Liu, X.; Cheng, K.; Luo, Y.; Duh, K.; Matsumoto, Y. A hybrid Chinese spelling correction using language model and statistical machine translation with reranking. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, Nagoya, Japan, 14–18 October 2013; pp. 54–58.
- Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support vector machines. *IEEE Intell. Syst. Their Appl.* 1998, 13, 18–28. [CrossRef]
- Wang, D.; Song, Y.; Li, J.; Han, J.; Zhang, H. 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, 31 October–4 November 2018; pp. 2517–2527.
- Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Burstein, J., Doran, C., Solorio, T., Eds.; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [CrossRef]
- Chen, T.; He, T.; Benesty, M.; Khotilovich, V.; Tang, Y.; Cho, H.; Chen, K.; Mitchell, R.; Cano, I.; Zhou, T. Xgboost: Extreme Gradient Boosting; R Package Version 0.4-2; 2015; Volume 1, pp. 1–4.
- Wang, D.; Tay, Y.; Zhong, L. Confusionset-guided pointer networks for Chinese spelling check. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5780–5785.
- Xie, W.; Huang, P.; Zhang, X.; Hong, K.; Huang, Q.; Chen, B.; Huang, L. Chinese spelling check system based on n-gram model. In Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, Association for Computational Linguistics, Beijing, China, 30–31 July 2015; pp. 128–136.

- Yu, J.; Li, Z. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Association for Computational Linguistics, Wuhan, China, 20–21 October 2014; pp. 220–223.
- Chiu, H.W.; Wu, J.C.; Chang, J.S. Chinese spell checking based on noisy channel model. In Proceedings of the Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Association for Computational Linguistics, Wuhan, China, 20–21 October 2014; pp. 202–209.
- 11. Jia, Z.; Wang, P.; Zhao, H. Graph model for Chinese spell checking. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, Asian Federation of Natural Language Processing, Nagoya, Japan, 14–18 October 2013; pp. 88–92.
- Han, D.; Chang, B. A maximum entropy approach to Chinese spelling check. In Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, Asian Federation of Natural Language Processing, Nagoya, Japan, 14–18 October 2013; pp. 74–78.
- Xiong, J.; Zhang, Q.; Zhang, S.; Hou, J.; Cheng, X. HANSpeller: A unified framework for Chinese spelling correction. In International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language; The Association for Computational Linguistics and Chinese Language Processing: Taipei City, Taiwan, 2015.
- Duan, J.; Wang, B.; Tan, Z.; Wei, X.; Wang, H. Chinese spelling check via bidirectional LSTM-CRF. In Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 24–26 May 2019; Institute of Electrical and Electronics Engineers: Piscataway, NJ, USA, 2019; pp. 1333–1336.
- 15. Wang, H.; Wang, B.; Duan, J.; Zhang, J. Chinese Spelling Error Detection Using a Fusion Lattice LSTM. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2021**, *20*, 1–11. [CrossRef]
- Hong, Y.; Yu, X.; He, N.; Liu, N.; Liu, J. Faspell: A fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm. In Proceedings of the 5th Workshop on Noisy User-Generated Text (W-NUT 2019), Association for Computational Linguistics, Hong Kong, China, 4 November 2019; pp. 160–169.
- 17. Zhang, S.; Huang, H.; Liu, J.; Li, H. Spelling error correction with soft-masked BERT. arXiv 2020, arXiv:2005.07421.
- 18. Cheng, X.; Xu, W.; Chen, K.; Jiang, S.; Wang, F.; Wang, T.; Chu, W.; Qi, Y. Spellgcn: Incorporating phonological and visual similarities into language models for chinese spelling check. *arXiv* 2020, arXiv:2004.14166.
- Huang, L.; Li, J.; Jiang, W.; Zhang, Z.; Chen, M.; Wang, S.; Xiao, J. PHMOSpell: Phonological and Morphological Knowledge Guided Chinese Spelling Check. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); Association for Computational Linguistics: Hong Kong, China 2021; pp. 5958–5967.
- Bao, Z.; Li, C.; Wang, R. Chunk-based chinese spelling check with global optimization. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, Association for Computational Linguistics, Online Event, 16–20 November 2020; pp. 2031–2040.
- Gou, W.; Chen, Z. Think Twice: A Post-Processing Approach for the Chinese Spelling Error Correction. *Appl. Sci.* 2021, 11, 5832. [CrossRef]
- 22. Nguyen, M.; Ngo, H.G.; Chen, N.F. Domain-shift Conditioning using Adaptable Filtering via Hierarchical Embeddings for Robust Chinese Spell Check. *arXiv* 2021, arXiv:2008.12281.
- Zhang, R.; Pang, C.; Zhang, C.; Wang, S.; He, Z.; Sun, Y.; Wu, H.; Wang, H. Correcting Chinese Spelling Errors with Phonetic Pre-training. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online Event, 1–6 August 2021; pp. 2250–2261.
- Wang, B.; Che, W.; Wu, D.; Wang, S.; Hu, G.; Liu, T. Dynamic Connected Networks for Chinese Spelling Check. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Association for Computational Linguistics, Online Event, 1–6 August 2021; pp. 2437–2446.
- 25. Liu, S.; Yang, T.; Yue, T.; Zhang, F.; Wang, D. 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, Virtual Event, 1–6 August 2021; Volume 1: Long Papers; pp. 2991–3000.
- 26. Xu, H.D.; Li, Z.; Zhou, Q.; Li, C.; Wang, Z.; Cao, Y.; Huang, H.; Mao, X.L. Read, listen, and see: Leveraging multimodal information helps Chinese spell checking. *arXiv* 2021, arXiv:2105.12306.
- Ji, T.; Yan, H.; Qiu, X. SpellBERT: A lightweight pretrained model for Chinese spelling check. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Virtual Event, 7–11 November 2021; pp. 3544–3551.
- 28. Li, Y.; Ma, S.; Zhou, Q.; Li, Z.; Yangning, L.; Huang, S.; Liu, R.; Li, C.; Cao, Y.; Zheng, H. Learning from the dictionary: Heterogeneous knowledge guided fine-tuning for chinese spell checking. *arXiv* 2022, arXiv:2210.10320.
- 29. Liang, Z.; Quan, X.; Wang, Q. Disentangled Phonetic Representation for Chinese Spelling Correction. arXiv 2023, arXiv:2305.14783.
- Wei, X.; Huang, J.; Yu, H.; Liu, Q. PTCSpell: Pre-trained Corrector Based on Character Shape and Pinyin for Chinese Spelling Correction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023; pp. 6330–6343.
- 31. Li, Y.; Zhou, Q.; Li, Y.; Li, Z.; Liu, R.; Sun, R.; Wang, Z.; Li, C.; Cao, Y.; Zheng, H.T. The past mistake is the future wisdom: Error-driven contrastive probability optimization for Chinese spell checking. *arXiv* **2022**, arXiv:2203.00991.

- 32. Liu, S.; Song, S.; Yue, T.; Yang, T.; Cai, H.; Yu, T.; Sun, S. CRASpell: A contextual typo robust approach to improve Chinese spelling correction. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 3008–3018.
- Yang, S.; Yu, L. CoSPA: An improved masked language model with copy mechanism for Chinese spelling correction. In Proceedings of the Conference on Uncertainty in Artificial Intelligence, Eindhoven, The Netherlands, 1–5 August 2022; pp. 2225–2234.
- 34. Sheng, L.; Xu, Z.; Li, X.; Jiang, Z. EDMSpell: Incorporating the error discriminator mechanism into chinese spelling correction for the overcorrection problem. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 101573. [CrossRef]
- 35. Wu, H.; Zhang, S.; Zhang, Y.; Zhao, H. Rethinking Masked Language Modeling for Chinese Spelling Correction. *arXiv* 2023, arXiv:2305.17721.
- Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Institute of Electrical and Electronics Engineers, Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.
- Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), San Francisco, CA, USA, 28 June–1 July 2001.
- 38. Chen, S.F.; Goodman, J. An empirical study of smoothing techniques for language modeling. *Comput. Speech Lang.* **1999**, 13, 359–394. [CrossRef]
- 39. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
- Tolles, J.; Meurer, W.J. Logistic regression: Relating patient characteristics to outcomes. JAMA 2016, 316, 533–534. [CrossRef] [PubMed]
- 41. Peterson, L.E. K-nearest neighbor. Scholarpedia 2009, 4, 1883. [CrossRef]
- 42. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 3146–3154.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.