applied sciences

MDPI

Article

Self-Distillation and Pinyin Character Prediction for Chinese Spelling Correction Based on Multimodality

Li He [1,2], Feng Liu [1,2,*], Jie Liu [1,2], Jianyong Duan [1,2] and Hao Wang [1,2]

[1] School of Information Science and Technology, North China University of Technology, Beijing 100144, China; heli@ncut.edu.cn (L.H.)
[2] CNONIX National Standard Application and Promotion Lab, Beijing 100144, China
* Correspondence: liufeng@mail.ncut.edu.cn

Abstract: Chinese spelling correction (CSC) constitutes a pivotal and enduring goal in natural language processing, serving as a foundational element for various language-related tasks by detecting and rectifying spelling errors in textual content. Numerous methods for Chinese spelling correction leverage multimodal information, including character, character sound, and character shape, to establish connections between incorrect and correct characters. Research indicates that a majority of spelling errors stem from pinyin similarity, with character similarity accounting for half of the errors. Consequently, effectively modeling character pinyin and character relationships emerges as a key challenge in the CSC task. In this study, we propose enhancing the CSC task by introducing the pinyin character prediction task. We employ an adaptive weighting method in the pinyin character prediction task to address predictions in a more granular manner, achieving a balance between the two prediction tasks. The proposed model, SPMSpell, utilizes ChineseBERT as an encoder to capture multimodal feature information simultaneously. It incorporates three parallel decoders for character prediction, pinyin prediction, and self-distillation modules. To mitigate potential overfitting concerning pinyin, a self-distillation method is introduced to prioritize character information in predictions. Extensive experiments conducted on three SIGHAN benchmark tests showcase that the model introduced in this paper attains a superior level of performance. This substantiates the correctness and superiority of the adaptive weighted pinyin character prediction task and underscores the effectiveness of the self-distillation module.

Keywords: Chinese spelling correction; multimodality; pinyin prediction

Citation: He, L.; Liu, F.; Liu, J.; Duan, J.; Wang, H. Self-Distillation and Pinyin Character Prediction for Chinese Spelling Correction Based on Multimodality. *Appl. Sci.* **2024**, *14*, 1375. https://doi.org/10.3390/app14041375

Academic Editor: Andrea Prati

Received: 20 January 2024
Revised: 4 February 2024
Accepted: 6 February 2024
Published: 7 February 2024

1. Introduction

Chinese spelling correction (CSC) is dedicated to identifying and rectifying spelling errors in text, playing a pivotal role in natural language processing (NLP) with far-reaching implications in downstream applications like search engines [1], OCR [2], and ASR. In contrast to languages like English, Chinese spelling correction poses a more formidable challenge owing to the intricacies of Chinese spelling rules. The Chinese language is characterized by a profusion of homophones, morphologically similar characters, and intricate phonetics, creating a fertile ground for spelling errors. Additionally, the complexity is amplified as Chinese spelling correction must consider contextual nuances, word-matching intricacies, and other linguistic features, further elevating the difficulty of error correction. Spelling errors in Chinese typically fall into two primary categories: pinyin errors and glyph errors, arising from the misuse of characters with similar pinyin pronunciations or visual resemblances, respectively. Figure 1 illustrates instances of these errors, such as the confusion between the pinyin-similar characters “稍” (meaning “little”) and “烧” (meaning “burn”), or the misinterpretation of visually similar shapes “人” (meaning “human”) and “入” (meaning “enter”). The origins of spelling errors predominantly stem from human writing mistakes and machine recognition errors, particularly those induced by Automatic Speech Recognition (ASR) and Optical Character Recognition (OCR) systems [3].

Type	Sentence	Correction
Phonological	炉子上正稍（shao）着水。 Water is little on the stove.	烧（shao） burn
Visual	必须持有门票才能人（ren）场。 You must have a ticker to human the venue.	入（ru） enter

Figure 1. Examples of pinyin error and glyph error, where red color indicates the wrong character.

In recent years, leveraging pre-trained language models (PLM) has emerged as a dominant approach for Chinese spelling correction tasks. Prominent examples include FASpell [4], Softmasked-BERT [5], SpellGCN [6], and PLOME [7]. Concurrently, certain researchers have directed their focus towards the phonological and glyph features of Chinese characters, aiming to enhance the model's error correction capabilities by amalgamating phonetic and visual information [7,8]. Previous studies underscore the significance of pinyin and glyph similarities, revealing that 83% of spelling errors are attributed to pinyin similarity and 48% to glyph similarity [9]. Effectively modeling Chinese pronunciation in the context of the CSC task remains a pivotal challenge, and nearly all contemporary state-of-the-art CSC methods explicitly or implicitly incorporate Chinese pronunciation.

Explicit methods delve into the Chinese pronunciation of entire characters, such as encoding the pinyin of a Chinese character and integrating it into the character representation using specific strategies [10]. Alternatively, Chinese pronunciation prediction models the relationship between similar characters in pinyin [7]. Implicit methods primarily focus on pinyin glyph similarity between Chinese character pairs, either by increasing the decoding probability of characters with akin pronunciations [6] or by incorporating pinyin glyph similarity into the encoding process via graph convolutional networks (GCN) [6]. Despite the notable performance gains achieved, these approaches confront two potential challenges: during training, pinyin information may either be overlooked or overshadowed by textual information. For instance, a specific BERT model that exclusively considers pinyin sequences, excluding Chinese characters, can still detect and rectify erroneous characters, while models like REALISE [8] encode and merge textual and pinyin information through a gating mechanism that overlooks an error. Second, the introduction of pinyin features may compromise the representation of normal text. Using Figure 2 as an illustration, a standard BERT model can correct the misspelled character "的" in the input, whereas REALISE fails to do so. This issue is attributed to REALISE's tendency to over-rely or over-fit on pinyin information.

Currently, only a fraction of the available multimodal information is harnessed in the existing Chinese spelling correction (CSC) methods, highlighting the underutilization of the inherent value in Chinese character multimodality. This underscores the unexplored potential for enhancement within the field of multimodality for the CSC task. Building upon this realization, we present a novel approach named SPMSpell (**S**elf-Distillation and Character **P**inyin Prediction Based on **M**ultimodality). SPMSpell introduces a finely tuned pinyin prediction task, an adaptive weighting mechanism, and a self-distillation module, aiming to refine the performance of the CSC task.

Concretely, when provided with a sentence containing spelling errors, ChineseBERT [11] serves as the encoder backbone to fuse three pivotal feature dimensions: semantic, phonological, and graphemic. Three distinct decoders are then constructed based on this fusion: one for accurate character prediction, another for predicting the consonant-rhyme of each target character (i.e., pinyin prediction), and the last one devoted to the self-distillation module, ensuring the model's prediction consistency when presented with the original text input. Throughout the training phase, the pinyin prediction task serves as an auxiliary task, with its predictions discarded during inference.

Source	我真是户秃（hu tu）。 I am so household bald.	
Target	我真是糊涂（hu tu）。 I am so silly.	
BERT	我真是护突。(I am so protector.)	Wrong
PinyinBERT	我真是糊涂。(I am so silly.)	Right
REALISE	我真是户涂。(I am so household smear.)	Wrong
Our SPMSpell	我真是糊涂。(I am so silly.)	Right
Source	我什么事都不济的（ji de）。 I can't do anything.	
Target	我什么事都不记得（ji de）。 I can't remember anything.	
BERT	我什么事都不记得。(I can't remember anything.)	Right
PinyinBERT	我什么事都不记的。(I can't write anything.)	Wrong
REALISE	我什么事都不记的。(I can't write anything.)	Wrong
Our SPMSpell	我什么事都不记得。(I can't remember anything.)	Right

Figure 2. Two examples of Chinese spelling error correction and the predictions of different models.

The primary contributions of this paper can be succinctly summarized as follows: (1) Introduction of the pinyin prediction task to enhance the efficacy of the CSC task, adopting a fine-grained prediction approach for pinyin. (2) Proposition of a self-distillation module to mitigate the risk of the model overly fixating on pinyin features. (3) Introduction of an adaptive weighting method to harmonize subtasks within the pinyin prediction task, encompassing consonant, rhyme, and tone predictions while simultaneously balancing character and pinyin prediction tasks. (4) Achievement of an elevated performance benchmark across three prominent CSC datasets.

2. Related Work

2.1. Chinese Spelling Correction

Chinese spelling correction is a foundational task within the realm of natural language processing. Simultaneously, the inherent complexities of the Chinese language, such as polyphonic characters and morphologically similar characters, render this task highly challenging. The increasing interest among natural language processing researchers underscores the significance of Chinese spelling correction. Presently, neural network-based models, particularly pre-trained language models, dominate this domain and can be broadly classified into two research directions.

One line of research centers on enhancing the semantic modeling of text features [4,12,13]. This approach treats Chinese spelling correction as a sequence annotation task, utilizing pre-trained language models to derive contextual representations. For instance, Soft-Masked BERT [5] integrates a detection network to predict the correctness of individual characters. It subsequently generates soft-masked embeddings for the correction network to rectify errors. MDCSpell [14] introduces a multitasking detector-correction framework, merging the representations of detection and correction networks.

Another avenue of research involves incorporating phonological information into the correction process. This stems from the observation that homophones contribute significantly to usage errors [9]. Models like MLM-phonetics [15] and PLOME [7] adopt a word replacement strategy during pre-training, substituting randomly selected characters with phonetically or visually similar ones. REALISE [8] and PHMOSpell [3] employ multiple encoders to capture textual, phonological, and visual features, utilizing a selective gating mechanism for fusion. SCOPE [16] introduces an assisted pronunciation prediction task and devises an iterative inference strategy to enhance performance. Nonetheless, these methods often amalgamate textual and phonological features without facilitating direct

and deep interaction between them, potentially resulting in the underutilization of pinyin information.

2.2. Multimodal Learning

Numerous studies have explored the integration of information from diverse modalities to enhance overall performance. Notable advancements include multimodal sentiment analysis [17,18], visual quizzing [19,20], and multimodal machine translation [21,22]. Recent developments involve the introduction of multimodal pre-training models, such as VL-BERT [23], Unicoder-VL [24], and LXMERT [25]. To incorporate visual information from Chinese characters into language models, Meng et al. [26] innovatively designed a Tianzige-CNN to enhance various NLP tasks, including named entity recognition and sentence classification.

Subsequently, researchers have endeavored to embed multimodal character information into error correction models to facilitate more accurate corrections. Xu et al. [8] meticulously model the semantic, phonetic, and visual information of input characters, introducing a gating mechanism to selectively blend information from these modalities for the final correction prediction. Guo et al. [12] adopt a strategy of pre-training BERT with artificially constructed obfuscated sets containing phonological and graphemic features of similar characters. This approach enables BERT to fully leverage character speech and visual features for error correction.

Wang et al. [10] enhance error correction by utilizing a pinyin-enhanced candidate based on character pronunciation features. This method, combined with an attentional mechanism to model neighboring token dependencies, contributes to more accurate predictions. Zhang et al. [15] integrate a pre-trained detection module and an error correction module based on speech features, effectively predicting the final correction. Guo et al. [12] combine a pre-trained set of character speech features with an error correction module for final corrections. Zhang et al. [15] undertake joint fine-tuning error correction based on pre-trained detection and error correction modules with phonological features.

Liu et al. [7] employ a GRU to extract phonological and visual features of characters, predicting the pronunciation of the target character in a coarse-grained, non-adaptive manner. Li et al. [13] introduce a speech prediction assistance task, combining fine-grained phonological features of characters to achieve adaptive weighting. Wei et al. [27] establish two new pre-training targets for the error corrector, capturing the phonetic and shape information of characters. These features are later fused with semantic information to achieve effective error correction.

2.3. Self-Distillation

Knowledge distillation [28] functions as a methodology to extract concise student models from their larger teacher counterparts. As a specialized distillation strategy, deep mutual learning [29] facilitates collaborative knowledge acquisition among various student models, enabling them to guide and refine each other throughout the training process. Specifically, when these student models share identical parameters, this methodology is termed self-distillation [30].

The implementation of self-distillation in the context of Chinese spelling correction (CSC) has resulted in significant performance enhancements. SDCL [31] autonomously encodes both the original sentence and its correct counterpart, employing contrast loss to enhance contextual representations. On the other hand, CRASpell [32] generates a noisy sample for each input, applying KL dispersion to both outputs to augment the performance in handling multiple misspelled sentences. In this study, the self-distillation module is intricately designed to alleviate the risk of overfitting associated with pinyin information during the model training phase.

3. Methodology

In this section, we present the SPMSpell model designed for the Chinese spelling correction task. We begin by providing a clear problem definition and subsequently delve into the intricate details of the proposed model's implementation.

3.1. Problem Definition

The Chinese Spell-Checking (CSC) task is to detect and correct spelling errors in Chinese text. Given a source sentence $X = \{x_1, x_2, \ldots, x_n\}$, with n characters containing spelling errors, the CSC model takes X as input, detects the errors and corrects them at the character level, and outputs the correct target sentence $Y = \{y_1, y_2, \ldots, y_n\}$. The lengths of X and Y are equal, and thus, the task can be regarded as a sequence labeling task, i.e., modeling the $p(Y|X)$ probability. We further give the fine-grained pinyin of each character y_i in the correct sentence Y, denoted as a ternary of the form $(\alpha_i, \beta_i, \gamma_i)$, where α_i, β_i and γ_i denote the vowel, rhyme, and tone, respectively. Please note that this transformation of output utterances is required and is only provided during training. Usually, there are no or only a small number of misspelled characters in the sentence. Most of the characters should be copied, and the misspelled character $x_i \in X$ bears some similarity to its correct character $y_i \in Y$.

3.1.1. Architecture

The fundamental concept driving SPMSpell is to employ self-distillation, fine-grained pinyin character prediction tasks, and an adaptive task weighting mechanism to augment the effectiveness of the Chinese spelling correction (CSC) task. To realize this objective, SPMSpell is constructed around a shared encoder featuring three parallel decoders. The first is primarily dedicated to character prediction, addressing the core CSC task. The second decoder focuses on the fine-grained pinyin prediction task, while the third serves the self-distillation module. The equilibrium between the two prediction tasks, as well as within the pinyin prediction task itself, is dynamically determined based on the pinyin character and vowel-rhyme-tone similarities discerned between the input and target sentences. Figure 3 provides a comprehensive overview of the overall architecture of SPMSpell.

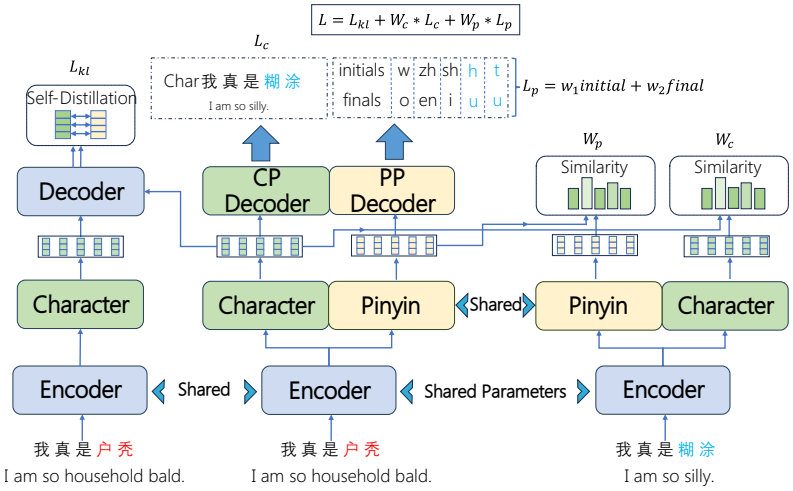

Figure 3. SPMSpell overall framework.

3.1.2. Encoder

Chinese is hieroglyphic, and character shapes and character sound features contain important information. Similar to the recent CSC methods utilizing multimodal information [7,8], we use ChineseBERT [11] as an encoder, as shown in Figure 4 ChineseBERT is a pre-trained language model that incorporates both Chinese pinyin and glyph information in a pre-trained language model to extract semantic, phonological and morphological

features for the CSC task. Since the multimodal information of characters is captured and fused by the same model without adding additional networks, the problem of ambiguity between different feature information can be largely mitigated; furthermore, using only ChineseBERT as a multimodal encoder further simplifies the architectural complexity of the overall model of the CSC; on the other hand, in terms of the degree of pinyin feature extraction, the model uses a fine-grained level, i.e., the consonants, rhymes and tones of characters are extracted as graphemic features instead of capturing the whole pinyin as in previous work. Fine-grained pronunciation feature capture allows for better modeling of the phonetic similarity between the target character and the error character. Specifically, for each character X_i in the input sentence X, the encoder first generates all its character embeddings, pinyin embeddings, and grapheme embeddings with embedding size D. These three embeddings are then concatenated in series and mapped to the D-dimensional fusion embeddings via a fully connected layer. Afterward, as in most other pre-trained language models, the fusion embeddings are added to the positional embeddings and fed into the Transformer's stack to generate a contextual representation $h_i \in R^D$ of the input character x_i. We denote the representation of the character after this encoding process as $H = \{h_1, h_2 \ldots, h_n\}$.

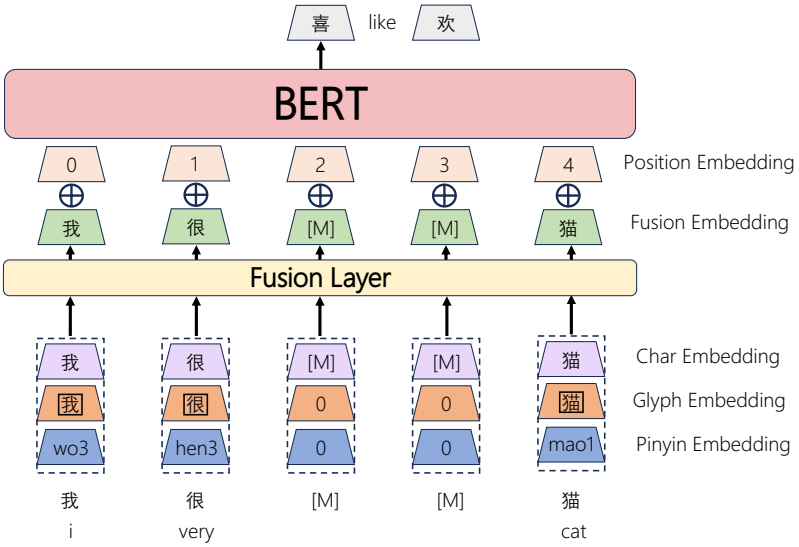

Figure 4. ChineseBERT framework.

3.1.3. Character Prediction Decoder

This decoder is used to predict the character in the correct sentence Y based on the coded output H. Specifically, given each input character x_i, we first project its coded output h_i into the character-specific feature space (*GeLu* is a Gaussian error linear unit activation function, and its derivative is continuous, which makes it easier to propagate the gradient when training a deep neural network, avoiding the problem of discontinuity of the derivative at zero, thus reducing the problem of gradient vanishing during training):

$$h_i^c = GeLu(W^c h_i + b^c) \tag{1}$$

The corresponding correct character $\hat{y}_i$ is then predicted based on the projected output *Softmax* is an activation function that normalizes a numerical vector into a probability distribution vector, where the sum of all probabilities is equal to 1. It is commonly employed as the final layer in neural networks, particularly for the output in multi-class classification problems. The Softmax layer is often used in conjunction with the cross-entropy loss function:

$$p(\hat{y}_i|X) = softmax(W^y h_i^c + b^y) \tag{2}$$

where $W^c \in R^{D\times D}, b^c \in R^D$ is the learnable parameter of the character-specific feature projection layer; $W^y \in R^{V\times D}, b^y \in R^V$ is the learnable parameter of the character prediction layer; and V is the vocabulary size.

3.1.4. Pinyin Prediction Decoder

This decoder is used to predict fine-grained predictions, i.e., to determine the vowel, rhyme, and tone of each character in the correct sentence Y, based on the coded output H. Similarly, given each input character x_i and its encoded output h_i, we project h_i into the feature space specific to pronunciation:

$$h_i^p = GeLu(W^p h_i + b^p) \tag{3}$$

Based on the projected output, we predict the consonant $\hat{\alpha}_i$, rhyme $\hat{\beta}_i$, and tone $\hat{\gamma}_i$of the corresponding correct character:

$$p(\hat{\alpha}_i \mid X) = softmax(W^\alpha h_i^p + b^\alpha) \tag{4}$$

$$p(\hat{\beta}_i \mid X) = softmax(W^\beta h_i^p + b^\beta) \tag{5}$$

$$p(\hat{\gamma}_i \mid X) = softmax(W^\gamma h_i^p + b^\gamma) \tag{6}$$

where $W^p \in R^{D\times D}, b^p \in R^D$ is the learnable parameter of the articulation-specific feature projection layer; $W^\tau \in R^{U\times D}, b^\tau \in R^U$, where $\tau \in \{\alpha, \beta, \gamma\}$ is the learnable parameter of the articulation prediction layer; and U is the total number of articulatory units (consonants, rhymes, and tones).

3.1.5. Self-Distillation Decoder

The decoder is used for the self-distillation module, i.e., based on the encoded output H, which is generated by taking the last layer of hidden states of the encoder. Then, h_i is projected into a pronunciation-specific feature space, and the probability distribution s_i for the i character is computed based on h_i:

$$p(s_i \mid X) = softmax(W^s h_i^c + b^s) \tag{7}$$

After obtaining the output distribution for each character, the model performs another forward pass with the original sequence X as input, generating for each character x_i another output distribution $q_i \in R^V$. The two sets of distributions are then forced to be close together by applying a bidirectional KL divergence:

$$L_{kl} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}(D_{kl}(s_i||q_i) + D_{kl}(q_i||s_i)) \tag{8}$$

3.1.6. Adaptive Weighting

We design an adaptive weighting method to balance the character prediction and pinyin prediction tasks during training, as well as the balance of consonant, rhyme, and tone prediction within pinyin prediction. Given an input sentence X, the character prediction task aims to match the predicted character $\{\hat{y}_i\}_{i=1}^n$ with the truth value $\{y_i\}_{i=1}^n$, while the pinyin prediction task aims to match the predicted fine-grained character $\{(\hat{\alpha}_i, \hat{\beta}_i, \hat{\gamma}_i)\}_{i=1}^n$ with the ground truth value $\{(\alpha_i, \beta_i, \gamma_i)\}_{i=1}^n$ match. Where the character prediction task loss function is defined as:

$$L_i^c = -logp(\hat{y}_i = y_i \mid X) \tag{9}$$

Then, as we discussed in the previous introduction, the phonetic prediction task may provide different levels of benefit with different input characters. The more similar the pronunciation of the input and target characters, the more likely it is that spelling errors caused by phonetic similarity will occur. In this case, the pinyin prediction task may

provide greater benefits and should be assigned greater weights. To compute such adaptive weights, we feed the target correct sentence Y to the encoder and subsequent pronunciation-specific projection layers. Then, we compute the cosine similarity $cos(h^p_{x_i}, h^p_{y_i})$ for each input character x_i and its target character y_i based on their pronunciation-specific feature representations $h^p_{x_i}, h^p_{y_i}$, and accordingly define the adaptive weights at the i position as:

$$w^p_i = sin((\frac{cos(h^p_{x_i}, h^p_{y_i}) + 1}{2})^2) \tag{10}$$

The higher the cosine similarity, $cos(h^p_{x_i}, h^p_{y_i})$, the greater the weight w_i will be. Similarly, for the pinyin prediction task internally, consonants, rhymes, and tones provide different levels of benefit for different input characters. However, empirically, tones should be weaker than the previous two in comparison to consonants and rhymes, so their weight values should be smaller than the previous two when weighting. Therefore, the fine-grained pinyin prediction task loss function is defined as follows:

$$L^p_i = -k_1 w_\alpha logp(\hat{\alpha}_i = \alpha_i|X) - k_2 w_\beta logp(\hat{\beta}_i = \beta_i|X) - k_3 w_\gamma logp(\hat{\gamma}_i = \gamma_i|X) \tag{11}$$

where, w_α, w_β, w_γ re the weights of consonants, rhymes, and tones computed based on cosine similarity, and k_1, k_2, k_3 hyperparameters. Finally, the total loss function is defined as the sum of the character prediction loss and the pinyin prediction loss, as well as the self-distillation loss:

$$L = \frac{1}{n}\sum_{i=1}^{n}(w^c_i L^c_i + w^p_i L^p_i + L_{kl}) \tag{12}$$

where w^c_i is the character prediction, and the target correct sentence Y is computed based on the cosine similarity. Two points are worth noting here: (1) The branch that encodes and maps the target sentence Y is used to compute the adaptive weights only in the forward passes, while it will be separated in the backward passes. (2) The pinyin prediction, the self-distillation module, and the adaptive weighting scheme are introduced only during training. During inference, we use the character prediction decoder alone for prediction.

3.2. Inference

Similar to the models proposed by Liu et al. [32] and Devlin et al. [33], addressing the challenges of error correction in multi-error texts remains a substantial hurdle, leading to suboptimal effects. In an effort to overcome these challenges and mitigate the tendency toward excessive error correction, this paper introduces a novel polling error correction strategy during the inference phase.

Specifically, during inference, each input sentence undergoes an iterative process of character detection and correction, deviating from the conventional all-at-once correction approach. In each iteration, only erroneous characters within a specified window size around each correction position from the preceding iteration are eligible for correction. If a position undergoes modification in every iteration, it is reverted to its original character without any correction. Drawing from experimental insights, the paper opts for two iterations, with a window size set to 5.

This iterative correction process involves counting a specified number of low-probability words as the targets for the subsequent error correction after each prediction. The corrected information from the previous round is then incorporated into the next round, enriching contextual information and facilitating more nuanced error correction. This iterative strategy proves conducive to elevating the overall quality of error correction.

4. Experiments

In this section, we provide a comprehensive overview of the experiments conducted to assess the efficacy of the proposed models.

4.1. Datasets and Metrics

As with the previous work [6–8], our training data consists of two parts. One part is manually annotated training samples from SIGHAN 13 [34], SIGHAN 14 [35], and SIGHAN 15 [36]. The other part comprises 271k training samples generated by Wang et al. [37] using methods based on Automatic Speech Recognition (ASR) and Optical Character Recognition (OCR). We use the test sets of SIGHAN 13, SIGHAN 14, and SIGHAN 15 for evaluation. We follow previous work [6,8] to convert them to Simplified Chinese using OpenCC. We further used pypinyin (pypinyin is a very popular Python library for converting Chinese characters to pinyin. It can output the corresponding pinyin according to the Chinese characters and supports polyphonic words and tone selection. Pinyin link address: https://pypi.org/project/pypinyin, 15 November 2023) to obtain the pronunciation of each character and segment them into consonants, rhymes, and tones using a predefined vocabulary of consonants and rhymes provided by Xu et al. [8].

We use the metrics of sentence-level precision, recall, and F1 to evaluate our model for detection and correction. Sentence-level metrics are more stringent than character-level metrics because a sentence is considered correct when and only when all errors in the sentence are successfully detected and corrected. Errors were reported on the Detect and Correct subtask. In addition to the game-level evaluation, we also considered the character-level evaluation and the official SIGHAN evaluation.

4.2. Baselines

We compare SPMSpell to the following baseline methods. All these methods use character speech information in some way and represent the current state of the art on the SIGHAN baseline.

- **GAD** [12]: This method learns the global relationship between potentially correct input characters and candidates for potentially incorrect characters.
- **DCN** [10]: Through a unique dynamic connection network, K^n paths (K denotes the number of candidate words, n denotes the length of the sentence) are generated in the output stage of the model, and then an optimal path is selected by scoring through the dynamic connection network; an attentional mechanism is introduced to model the dependency relationship between neighboring characters.
- **REALISE** [8]: The method predicts the output by encoding phonological and graphemic information based on semantic information and finally introduces a gating mechanism to selectively fuse semantic, phonological, and graphemic information.
- **uChecker** [38]: A masked pre-trained language model is proposed as an unsupervised Chinese spell checker. The method uses the pre-trained model to learn contextual information and uses a masked language modeling task to predict misspelled words for spell-checking.
- **MDCSpell** [14]: The method designs a multitasking framework with BERT as a corrector that captures visual and phonetic features of characters and integrates the hidden state of the detector to minimize the impact of errors.
- **ECOPE** [13]: This thesis proposes a method for Chinese spell correction using error-driven comparison probability optimization to improve the accuracy of future spell-checking by learning the comparison between past spelling errors and correct spellings.
- **UMRSpell** [39]: It aims to unify the detection and correction parts of the pre-trained model to achieve Chinese missing, redundancy, and spelling correction. The method utilizes the contextual information of the pre-trained model and the Transformer structure to achieve spelling detection and correction by means of joint learning.

4.3. Experimental Parameter Setting

We use the pre-trained ChineseBERT as the encoder. In the specific training process, the dimension of the hidden layer feature vectors was set to 768 based on the model's representational capacity and computational efficiency. The learning rate, determining the magnitude of weight adjustments during each update, was set to 5×10^{-5} with linear

decay, as determined by experimental considerations. Dropout was generally set to 0.1. The Batch Size, determining the number of samples used for each weight update, was set to 32 based on experimental conditions and device capabilities. The number of epochs set to 30 based on fitting results during the experimental process represented the training iterations. The AdamW optimizer was employed and widely recognized for its effectiveness in various deep-learning tasks. All experiments are conducted on one GeForce RTX 3090. (The GeForce RTX 3090 is manufactured by NVIDIA from Santa Clara, CA, USA).

4.4. Overall Results

The comprehensive results presented in Table 1 illustrate the evaluation outcomes of the proposed SPMSpell model alongside seven baseline models. The F1 scores for sentence-level detection and correction, as well as character-level detection and correction, are reported on the SIGHAN 13/14/15 test datasets, with bold font highlighting the best-performing results. The results showcase the superiority of the SPMSpell model across all test sets, both at the sentence and character levels. Particularly noteworthy is the exceptional performance on the SIGHAN 15 test set, where SPMSpell outperforms all baseline models, affirming the model's efficacy. When compared to models incorporating speech and visual features, such as REALISE, SPMSpell achieves a notable improvement in detection/correction F1 by 0.9%, 0.9%, 4.3%, and 2.9% on SIGHAN 14/15, respectively. This underscores the effectiveness of ChineseBERT-based encoders in mitigating the mismatch phenomenon arising from the fusion of multiple feature information.

It is noteworthy that the observed improvement in recall and F1, while substantial, is accompanied by a disparity in accuracy compared to other baseline models. This discrepancy may stem from the limitation of the training data volume. To address this, the introduction of a new public dataset could augment the training volume and potentially bridge the existing gap. Nevertheless, it is essential to highlight that the model's performance remains competitive, particularly in comparison to other methods incorporating speech information for this task.

Table 1. Experimental results of each model on test sets.

Dataset	Model	Detection-Level			Correction-Level			Char-Level	
		D-P	D-R	D-F	C-P	C-R	C-F	D-F	C-F
SIGHAN13	GAD	85.7	79.5	82.5	84.9	78.7	81.5	87.6	93.5
	DCN	86.8	79.6	83.0	84.7	77.7	81.0	85.2	86.4
	REALISE	88.6	82.5	85.4	87.2	81.2	84.1	86.1	88.4
	uChecker	75.4	73.4	74.4	72.6	70.8	71.7	-	-
	MDCSpell	**89.1**	78.3	83.4	**87.5**	76.8	81.8	-	-
	ECOPE	87.2	81.7	84.4	86.1	80.6	83.3	-	-
	UMRSpell	83.0	73.6	78.0	80.0	71.0	75.2	84.9	**96.4**
	SPMSpell (our)	87.7	**83.7**	**85.6**	86.9	**82.8**	**84.6**	**92.1**	95.3
SIGHAN14	GAD	66.6	71.8	69.1	65.0	70.1	67.5	**82.9**	87.6
	DCN	67.4	70.4	68.9	65.9	68.7	67.2	78.9	86.2
	REALISE	67.8	71.5	69.6	66.3	70.0	68.1	78.5	80.1
	uChecker	61.7	61.5	61.6	57.6	57.5	57.6	-	-
	MDCSpell	**70.2**	68.8	69.5	**69.0**	67.7	68.3	-	-
	ECOPE	65.8	69.0	67.4	63.7	66.9	65.3	-	-
	UMRSpell	69.0	56.6	62.2	63.9	57.2	60.4	73.2	**93.3**
	SPMSpell (our)	68.6	**73.5**	**70.5**	67.0	**71.2**	**69.0**	81.5	89.0

Table 1. *Cont.*

Dataset	Model	Detection-Level			Correction-Level			Char-Level	
		D-P	D-R	D-F	C-P	C-R	C-F	D-F	C-F
SIGHAN15	GAD	75.6	80.4	77.9	73.2	77.8	75.4	88.2	90.1
	DCN	77.1	80.9	79.0	74.5	78.2	76.3	85.0	84.9
	REALISE	77.3	81.3	79.3	75.9	79.9	77.8	87.4	86.2
	uChecker	75.4	72.0	73.7	70.6	67.3	68.9	-	-
	MDCSpell	80.8	80.6	80.7	78.4	78.2	78.3	-	-
	ECOPE	78.2	82.3	80.2	76.6	80.4	78.4	-	-
	UMRSpell	77.2	72.2	75.0	69.3	64.8	67.0	83.0	91.5
	SPMSpell (our)	**81.7**	**85.6**	**83.6**	**79.4**	**83.4**	**81.3**	**88.3**	**92.8**

Note: The test sets are SIGHAN13, SIGHAN14, and SIGHAN15. In the table results, Detection-level and Correction-level represent sentence-level metrics, Char-level represents character-level metrics, D-P means detection module precision, D-R means detection module recall, D-F means detection module F1 value, C-P, C-R, and C-F are correction module corresponding metrics, and black bold in the table is the optimal result.

4.5. Ablation Study

In this subsection, we explore the impact of the hyperparameters k_1, k_2, k_3 in the loss function on the model performance as well as the contribution of the pinyin prediction task, the self-distillation module, and the adaptive weighting method to the SPMSpell model. For this ablation experiment, we evaluate the model using the SIGHAN 2015 test set.

According to what was mentioned earlier, in the pinyin prediction task, the gap between the roles of tones and rhymes is small and theoretically larger than the roles provided by tones, so for exploring the effects of hyperparameters k_1, k_2, k_3 can be transformed into exploring the effects of hyperparameter k. The transformation is as follows:

$$k_1, k_2, k_3 \rightarrow \frac{(1-k)}{2}, \frac{(1-k)}{2}, k \tag{13}$$

The effect of hyperparameter k taking value on F1 value of model performance is shown in Figure 5. Based on the experimental results, we can find that the F1 scores are basically increasing with increasing values of k. However, after the value exceeds 0.3, the F1 scores begin to decrease. Therefore, setting k to 0.2 achieves the overall best corrected F1 score. We further explored the reasons for this and found that in Mandarin, pinyin is the official system used for phonetic transcription. It uses the three components of consonants, rhymes, and tones to express the pronunciation and spelling of Chinese characters. Since the similarity of pronunciation of Chinese characters is mainly determined by their consonants or rhymes rather than their tones, its proportion of tones should be relatively low, and its proportion of consonants and rhymes high, i.e., the value of k should be taken to be relatively low.

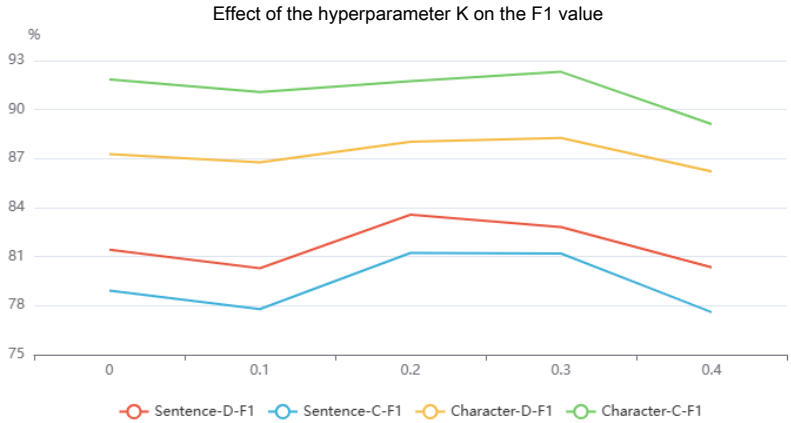

Figure 5. Effect of hyperparameter k on F1 value.

To further validate the effect of analyzing several constituent modules, this paper also performs ablation experiments on the SIGHAN 2015 test set in the following settings to explore the contribution of each module: (a) w/o PP refers to the removal of the pinyin prediction task from the SPMSpell model; (b) w/o SD refers to the removal of the self-distillation module from the SPMSpell model; and (c) w/o AW refers to the removal of the adaptive weighting method from the SPMSpell model.

The experimental results, as presented in Table 2, showcase precision (P), recall (R), and F1 scores for sentence-level detection/correction, along with the F1 scores for character-level detection/correction. The primary aim of this study is to leverage a multimodal language model to unveil the intricate relationships between characters, therefore addressing the challenge of uneven fusion of multimodal feature information. Notably, upon excluding the pinyin prediction task, there is a discernible decrease in detection/correction F1 scores by 2.6% and 1.9%, respectively. This underscores the effectiveness of the pinyin prediction task in capturing the nuances of Chinese character pronunciation. Similarly, upon removal of the self-distillation module, there is a reduction in detection/correction F1 scores by 2.4% and 2.2%, respectively, affirming the efficacy of the introduced self-distillation module in mitigating the risk of overfitting to pinyin features.

Table 2. Results of model ablation experiments.

Dataset	Model	Detection-Level			Correction-Level			Char-Level	
		D-P	D-R	D-F	C-P	C-R	C-F	D-F	C-F
SIGHAN15	SPMSpell	81.7	85.6	83.6	79.4	83.4	81.3	88.3	92.8
	w/o PP	79.0	82.5	81.0	77.4	80.6	79.4	87.0	90.9
	w/o PP	80.1	82.3	81.2	78.0	80.2	79.1	87.1	91.2
	w/o PP	80.4	84.3	82.3	77.5	80.5	79.5	87.1	90.3

Note: The dataset used in the ablation experiment was SIGHAN15. In the table results, Detection-level and Correction-level represent sentence-level metrics, Char-level represents character-level metrics, D-P means detection module precision, D-R means detection module recall, D-F means detection module F1 value, C-P, C-R, and C-F are correction module corresponding metrics.

To further verify the effectiveness of different encoders, this study conducted ablation experiments with the following configurations on the SIGHAN13 test set to explore encoder selection: (a) SPMSpell (BERT) refers to using BERT as the encoder in the SPMSpell model; (b) SPMSpell (REALISE) refers to using the REALISE model as the encoder in the SPMSpell model; (c) SPMSpell (ChineseBERT) refers to using ChineseBERT as the encoder in the SPMSpell model, which is the proposed SPMSpell model in this paper.

The experimental results, as shown in Table 3, display the precision (P), recall (R), and F1 score of sentence-level detection/correction. Overall, compared to using BERT as the encoder, the REALISE and ChineseBERT models, possibly due to incorporating character shape and pinyin information into language model pre-training, achieve better performance. With the assistance of character shape and pinyin information, better performance is attained. When comparing REALISE with the ChineseBERT model, using ChineseBERT as the encoder yields superior results. This may be because the REALISE model is multimodal, emphasizing the integration of text, sound, and visual information, while ChineseBERT focuses on the features of the Chinese language, particularly the fusion of character shape and pinyin information. It can simultaneously capture semantic, character shape, and pronunciation features without encountering the issue of re-fusing different features obtained by different models, effectively addressing the problem of uneven feature information. On the other hand, using ChineseBERT alone as the multimodal encoder backbone further simplifies the overall architecture complexity of the CSC model.

Table 3. Results of encoder ablation experiments.

Dataset	Model	Detection-Level			Correction-Level		
		D-P	D-R	D-F	C-P	C-R	C-F
SIGHAN13	SPMSpell (BERT)	85.0	77.0	80.8	83.0	75.2	78.9
	SPMSpell (REALISE)	87.6	82.5	85.4	**87.2**	81.2	84.1
	SPMSpell (ChineseBERT)	**87.7**	**83.7**	**85.6**	86.9	**82.8**	**84.6**

Note: The dataset used in the ablation experiment was SIGHAN13. In the table results, Detection-level and Correction-level represent sentence-level metrics, D-P means detection module precision, D-R means detection module recall, D-F means detection module F1 value, C-P, C-R, and C-F are correction module corresponding metrics, and black bold in the table is the optimal result.

In essence, the performance of SPMSpell experiences a decline when any individual component is removed. This provides compelling evidence for the effectiveness of each component within our approach, emphasizing the integral role played by each element in contributing to the overall efficacy of SPMSpell.

5. Conclusions

The paper presents SPMSpell, a Chinese spelling error correction model grounded in multimodal features. Although prior studies have emphasized the importance of character, character sound, and character shape information, our approach innovatively employs the multimodal language model ChineseBERT as an encoder. This enables the simultaneous capture of character, character sound, and character shape features. Feature fusion is then applied to model character and character sound effectively, addressing issues related to multimodal feature information mismatch. This not only reduces model complexity but also mitigates training costs. For the character prediction task, fine-grained pinyin prediction task, and the self-distillation module, we utilize three parallel decoders. To tackle potential overfitting in pinyin prediction, we introduce the self-distillation method, ensuring a predominant role for character information in predictions. Adaptive weighting methods are incorporated to balance the model between character prediction and pinyin prediction tasks, as well as among their respective subtasks. Experimental results on the SIGHAN public dataset demonstrate SPMSpell's superior performance compared to other comparative models. Detailed analyses and studies reveal SPMSpell's excellence in leveraging pinyin features within multimodal information, showcasing robust generalization capabilities in the realm of Chinese spelling error correction. Looking ahead, our contributions lie in advancing the state of the art by effectively integrating multimodal features for improved spelling error correction. Future work will explore further optimizations, new multimodal architectures, and incorporating large language models to enhance the overall performance and applicability of the model.

Author Contributions: Conceptualization, L.H., F.L., J.L., J.D. and H.W.; writing—original draft preparation, L.H. and F.L.; writing—review and editing, L.H.; data curation, F.L.; validation, J.L.; supervision, L.H., J.L., J.D. and H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key Research and Development Program of China (2020AAA0109700), the 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).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We would 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.

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

References

1. Martins, B.; Silva, M.J. Spelling Correction for Search Engine Queries. In *Advances in Natural Language Processing, Proceedings of the International Conference on Natural Language Processing, EsTAL 2004, Alicante, Spain, 20–22 October 2004*; Springer: Berlin/Heidelberg, Germany, 2004.
2. Afli, H.; Qiu, Z.; Way, A.; Sheridan, P. *Using SMT for OCR Error Correction of Historical Texts*; Language Resources and Evaluation; European Language Resources Association (ELRA): Paris, France, 2016.
3. 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), Online, 1–6 August 2021; pp. 5958–5967.
4. 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 *Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Toronto, ON, Canada, 2019.
5. Zhang, S.; Huang, H.; Liu, J.; Li, H. Spelling Error Correction with Soft-Masked BERT. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
6. 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. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020.
7. 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 (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 2991–3000.
8. 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. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*; Association for Computational Linguistics: Toronto, ON, Canada, 2021.
9. Liu, C.-L.; Lai, M.-H.; Chuang, Y.-H.; Lee, C.-Y. Visually and Phonologically Similar Characters in Incorrect Simplified Chinese Words. In *Coling 2010: Posters*; Coling 2010 Organizing Committee: Beijing, China, 2010; pp. 739–747.
10. Wang, B.; Che, W.; Wu, D.; Wang, S.; Hu, G.; Liu, T. Dynamic Connected Networks for Chinese Spelling Check. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*; Association for Computational Linguistics: Toronto, ON, Canada, 2021.
11. Sun, Z.; Li, X.; Sun, X.; Meng, Y.; Ao, X.; He, Q.; Wu, F.; Li, J. ChineseBERT: Chinese Pretraining Enhanced by Glyph and Pinyin Information. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021.
12. Guo, Z.; Ni, Y.; Wang, K.; Zhu, W.; Xie, G. Global Attention Decoder for Chinese Spelling Error Correction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*; Association for Computational Linguistics: Toronto, ON, Canada, 2021.
13. 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. In *Findings of the Association for Computational Linguistics: ACL 2022*; Association for Computational Linguistics: Toronto, ON, Canada, 2022.
14. Zhu, C.; Ying, Z.; Zhang, B.; Mao, F. MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2022*; Association for Computational Linguistics: Toronto, ON, Canada, 2022; pp. 1244–1253.
15. 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 *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*; Association for Computational Linguistics: Toronto, ON, Canada, 2021; pp. 2250–2261.
16. Li, J.; Wang, Q.; Mao, Z.; Guo, J.; Yang, Y.; Zhang, Y. Improving Chinese Spelling Check by Character Pronunciation Prediction: The Effects of Adaptivity and Granularity. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022; pp. 4275–4286.
17. Zadeh, A.; Zellers, R.; Pincus, E.; Morency, L.P. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intell. Syst.* **2016**, *31*, 82–88. [CrossRef]
18. Zhang, D.; Li, S.; Zhu, Q.; Zhou, G. Effective Sentiment-relevant Word Selection for Multi-modal Sentiment Analysis in Spoken Language. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019.
19. Agrawal, A.; Lu, J.; Antol, S.; Mitchell, M.; Zitnick, L, C.; Batra, D.;Parikh, D. VQA: Visual Question Answering. *Int. J. Comput. Vision* **2016**, *123*, 2425–2433. [CrossRef]
20. Chao, W.L.; Hu, H.; Sha, F. Being Negative but Constructively: Lessons Learnt from Creating Better Visual Question Answering Datasets. In *North American Chapter of the Association for Computational Linguistics*; Association for Computational Linguistics: Toronto, ON, Canada, 2018.

21. Hitschler, J.; Riezler, S. Multimodal Pivots for Image Caption Translation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016.
22. Barrault, L.; Bougares, F.; Specia, L.; Lala, C.; Elliott, D.; Frank, S. Findings of the Third Shared Task on Multimodal Machine Translation. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, Belgium, Brussels, 27 July 2018.
23. Su, W.; Zhu, X.; Cao, Y.; Li, B.; Lu, L.; Wei, F.; Dai, J. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
24. Li, G.; Duan, N.; Fang, Y.; Gong, M.; Jiang, D.; Zhou, M. Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training. *arXiv* **2019**, arXiv:1908.06066.
25. Tan, H.; Bansal, M. LXMERT: Learning Cross-Modality Encoder Representations from Transformers. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019.
26. Meng, Y.; Wu, W.; Wang, F.; Li, X.; Nie, P.; Yin, F.; Li, M.; Han, Q.; Sun, X.; Li, J. Glyce: Glyph-vectors for Chinese Character Representations. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019.
27. Wei, X.; Huang, J.; Yu, H.; Liu, Q. PTCSpell: Pre-trained Corrector Based on Character Shape and Pinyin for Chinese Spelling Correction. In *Findings of the Association for Computational Linguistics: ACL 2023*; Association for Computational Linguistics: Toronto, ON, Canada, 2023; pp. 6330–6343.
28. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *Comput. Sci.* **2015**, *14*, 38–39.
29. Zhang, Y.; Xiang, T.; Hospedales, T.M.; Lu, H. Deep mutual learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018.
30. Mobahi, H.; Farajtabar, M.; Bartlett, P.L. Self-Distillation Amplifies Regularization in Hilbert Space. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020.
31. Zhang, X.; Yan, H.; Yu, S.; Qiu, X. Sdcl: Self-distillation contrastive learning for Chinese spell checking. *arXiv* **2022**, arXiv:2210.17168.
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 *Findings of the Association for Computational Linguistics: ACL 2022*; Association for Computational Linguistics: Toronto, ON, Canada, 2022; pp. 3008–3018.
33. 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), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
34. Wu, S.H.; Liu, C.L.; Lee, L.H. Chinese Spelling Check Evaluation at SIGHAN Bake-off 2013. In Proceedings of the International Joint Conference on Natural Language Processing. Asian Federation of Natural Language Processing, Nagoya, Japan, 14–19 October 2013.
35. Yu, L.-C.; Lee, L.-H.; Tseng, Y.-H.; Chen, H.H. Overview of SIGHAN 2014 Bake-off for Chinese Spelling Check. In Proceedings of the Cips-sighan Joint Conference on Chinese Language Processing, Wuhan, China, 20–21 October 2014. [CrossRef]
36. Tseng, Y.-H.; Lee, L.-H.; Chang, L.-P.; Chen, H.H. Introduction to SIGHAN 2015 Bake-off for Chinese Spelling Check. In Proceedings of the 8th SIGHAN Workshop on Chinese Language Processing (SIGHAN'15), Beijing, China, 30–31 July 2015.
37. 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.
38. Li, P. uChecker: Masked Pretrained Language Models as Unsupervised Chinese Spelling Checkers. In Proceedings of the 29th International Conference on Computational Linguistics, Gyeongju, Republic of Korea, 12–17 October 2022; pp. 2812–2822.
39. He, Z.; Zhu, Y.; Wang, L.; Xu, L. UMRSpell: Unifying the Detection and Correction Parts of Pre-trained Models towards Chinese Missing, Redundant, and Spelling Correction. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 10238–10250.