scientific reports

Check for updates

OPEN

Intelligent recognition of counterfeit goods text based on BERT and multimodal feature fusion

Tinghao Wang[1,2], Yuheng Li[1,2,4], Weiping Li[3], Lijuan Zhou[1,2]✉ & Ning Luo[1,2]✉

Counterfeit goods are often imitated through the similarity of pronunciation or character shape of the trade name, for example, '蓝月亮' is altered to '蓝月壳', and this text-level imitation means brings great trouble to consumer identification. However, there is a scarcity of research on intelligent recognition techniques for this phenomenon. Although the Chinese Spelling Correction (CSC) technique provides some ideas for solving this problem, it still faces the challenges of scarce datasets, significant interference of erroneous characters with the contextual semantics, and high confusion between erroneous characters and correct characters in terms of pronunciation or glyphs in practical applications. In view of the above problems, this paper proposed a Corrector-Detector Auxiliary Network named CDANet. Specifically, (i) A lightweight Transformer Block is used to assist in locating erroneous characters to reduce their interference with contextual semantics; (ii) The multimodal information of erroneous characters is deeply exploited by integrating glyph, pinyin, and semantic features to enhance the correction accuracy; (iii) A counterfeit goods text dataset (CGT-Dataset) containing 289,851 samples was constructed to alleviate the problem of data scarcity. The experimental results show that CDANet achieves the current optimal performance on the self-built CGT-Dataset and exhibits excellent generalization ability on three public benchmark datasets, providing an efficient solution for counterfeit goods text recognition.

Keywords Counterfeit Goods, Intelligent Recognition of Counterfeit Goods, Chinese Spelling Correction, Counterfeit Goods Dataset

In the context of globalized markets, the proliferation of counterfeit goods has become a serious problem that needs to be solved. These counterfeit goods not only violate intellectual property rights, but also pose a significant threat to consumer health and safety[1–3]. Counterfeit goods usually mislead consumers by imitating the name or trademark of a well-known brand and creating the illusion of a strong auditory or visual similarity. For example, the well-known brand '王老吉' has been altered to '王老古', which is very similar to the word shape of the two, and this kind of counterfeiting technique is very confusing. As shown in Fig. 1, the red marking is the tampered character and the orange marking is the corresponding correct character.

In order to effectively identify and combat these counterfeiting behaviors, traditional manual review means are already difficult to effectively deal with them, and automation technology is urgently needed to improve detection efficiency and precision. However, there is a scarcity of research on intelligent identification of counterfeit goods text specifically targeting phonetic or glyph similarity. In view of the fact that the core of this task lies in the detection and correction of erroneous characters, the Chinese Spelling Correction (CSC) technology provides a new way of thinking for solving this problem due to its unique advantages in the detection and correction of erroneous characters. The CSC task focuses on recognizing and correcting a small number of erroneous characters by keeping the length of the input and output sequences consistent, and its characteristics are highly compatible with the counterfeiting of counterfeit goods texts through minor textual changes [4,5].

In recent years, with the successful application of large pre-trained language models, CSC tasks have made significant progress and have been widely applied to many downstream tasks, such as named entity recognition, Optical Character Recognition (OCR) and Automatic Speech Recognition (ASR) [1,6,7]. However, directly applying

[1]School of Cyberspace Security (School of Cryptology), Hainan University, Haikou 570228, Hainan, China. [2]Key Laboratory of Internet Information Retrieval of Hainan Province, Haikou 570228, Hainan, China. [3]School of Software and Microelectronics, Peking University, Beijing 100871, China. [4]Yuheng Li contributed equally to this work. ✉email: zhoulijuan@hainanu.edu.cn; luoning@hainanu.edu.cn

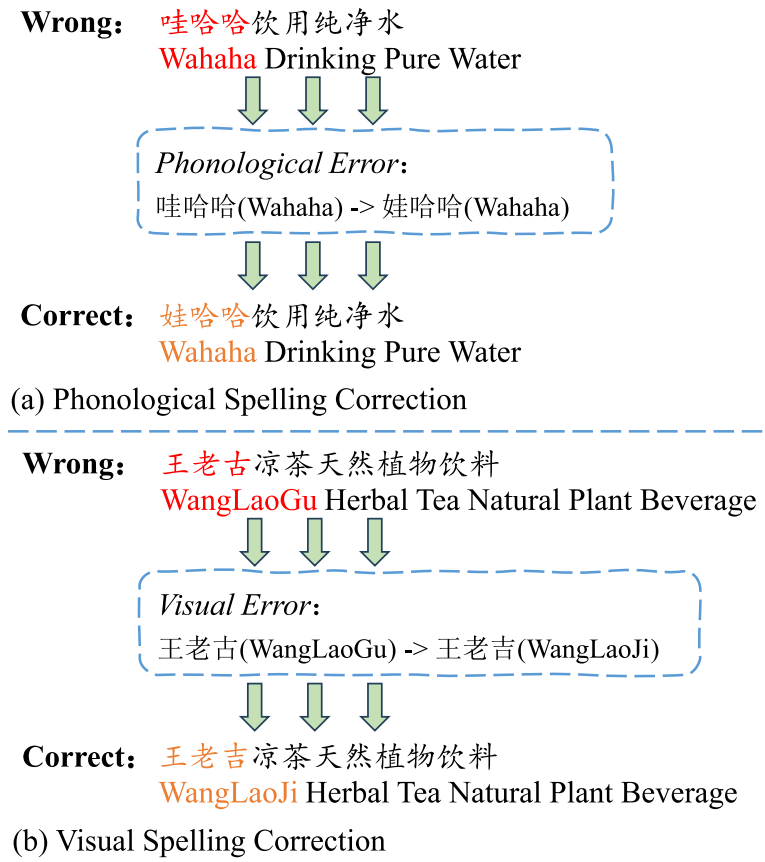

Fig. 1. Example of counterfeiting using harmonics and word similarity.

CSC technology to intelligently identify counterfeit product texts with similar pronunciation or character forms presents numerous challenges. First, the severe scarcity of product text datasets hinders effective progress in related research. Second, the presence of erroneous characters disrupts overall semantic information, leading to biased semantic understanding. As noted in relevant studies [8], approximately 83 % of textual errors relate to phonetic similarity, while 48 % relate to visual similarity. The morphological and phonetic diversity of these erroneous characters further complicates recognition. Most critically, the counterfeit product text recognition task possesses unique characteristics distinct from general CSC. General CSC addresses randomly distributed errors (e.g., spelling and grammatical mistakes) within sentences, whereas counterfeit text errors are typically deliberate, targeted, and concentrated on specific key phrases like brand names. Such errors exploit high phonetic and visual similarity to deceive consumers. This targeted nature means surrounding text often remains coherent, making detection difficult for standard language models reliant on overall sentence entropy values. Though only a single character, the erroneous character severely disrupts the most critical semantic entities within the text, creating a unique challenge that generic CSC methods struggle to address.

To tackle these specific challenges, we argue that a generic CSC model is insufficient. The targeted nature of counterfeit text errors requires a specialized architecture. Therefore, we propose a Corrector-Detector Auxiliary Network, CDANet, which is explicitly designed to first isolate the disruptive erroneous characters before attempting correction, a crucial step to preserve contextual integrity. The framework consists of two parts: the Auxiliary Position Detection Network and the Corrector. The Auxiliary Position Detection Network utilises two lightweight Transformer Blocks to pinpoint erroneous characters, thus effectively reducing the interference of erroneous characters with the contextual semantics. Its output final hidden states are not only used for binary classification loss optimisation, but also deeply fused with semantic features to generate fused semantic features, which provide more accurate contextual information for the subsequent correction process. The corrector further integrates fused semantic features, pinyin features and glyph features to learn multimodal features of erroneous characters. The pinyin features learn the phonetic information of the erroneous characters through 1D convolution and pooling operations. The glyph features use Tianzige-CNN to capture the visual information of erroneous characters to achieve multi-dimensional visual feature construction. The fused semantic features, on the other hand, combine the contextual information provided by Bert and the knowledge provided by the auxiliary position detection network. The end-to-end multimodal feature fusion training makes full use of the multimodal information of erroneous characters to achieve accurate character correction.

The main contributions of this study are summarized as follows: (i) We propose an innovative end-to-end framework, CDANet, which uniquely integrates an auxiliary position detection network with a multimodal corrector through a refined two-stage fusion mechanism. This approach addresses the problem by mitigating

semantic interference and improving recognition accuracy for highly similar characters. (ii) We introduce an auxiliary position detection network with dual functionality. This lightweight network not only precisely locates erroneous characters through explicit supervision, effectively reducing their interference in subsequent tasks, but also enhances semantic representations via hidden state fusion, providing richer contextual information for the corrector. (iii) Constructs the large-scale counterfeit goods text dataset CGT-Dataset, comprising 289,851 forged text samples involving phonetic and orthographic similarity errors. Comprehensive validation on this dataset and three public benchmarks demonstrates that CDANet achieves state-of-the-art performance on CGT-Dataset while exhibiting robust generalization capabilities, providing an effective solution for counterfeit goods text forgery.

Related work

Counterfeit goods detection

Combating counterfeit goods is a long and arduous task. At present, methods of identifying counterfeit goods fall into two main categories.

The first category is methods based on overt or covert technical means. Overt technical means include holograms, watermarks, color-changing inks, and product serial numbers[9–11]. Covert technical means, on the other hand, are similar to overt techniques and cover RFID tags, QR codes, biological, chemical or microscopic markers, digital watermarks or anti-counterfeit inks[9,12,13]. Although these methods have proven reliable in real-world applications, they still have significant limitations: overt identification methods usually rely on authentication details on the surface of the item, which can be easily duplicated or removed by imitators through reverse engineering; covert methods, although more secure, are difficult for many organizations to integrate effectively due to the need to be deeply embedded in the production process.

The second category is methods based on deep learning techniques. For example, Garcia-Cotte H et al.[14] developed a deep neural network-based image recognition system for smartphones, which is capable of detecting counterfeit products with high accuracy without the need for special security labels or any alterations to the products. Mishra et al.[15] utilized a variety of algorithms including support vector machines, convolutional neural networks, linear regression, and logistic regression, to achieve highly accurate detection of counterfeit medicines. Peng J et al.[1] proposed Hybrid Attention Network (HANet) for detecting counterfeit luxury handbags, which combines spatial and channel attentional units to learn the important information and is trained with an appraiser-guided loss function to be able to recognize the subtle differences between genuine and fake products. However, these deep learning-based methods mainly focus on the recognition of visual features, and it is difficult to properly deal with the problem of pronunciation or word similarity in the text of counterfeit goods information.

Given that the key to the intelligent recognition of counterfeit goods text lies in the detection and correction of erroneous text, the CSC technology is highly adaptable to this task. Therefore, this study tries to realize this goal with the help of CSC technology.

Chinese spelling correction

The CSC task has a long history of research. The task traditionally focuses on word substitution, with input and output sentences of the same length, and a relatively single form, and most of the research is based on the SIGHAN13/14/15 evaluation task dataset. Early studies[16–18] mainly used unsupervised learning methods to identify potential errors by constructing a confusion set and determining the correctness using the language model perplexity. Some subsequent studies modeled the Chinese Spelling Correction task as a sequence labeling problem and solved it with the help of Conditional Random Field (CRF) or Hidden Markov Model (HMM)[19,20].

CSC task has a long history of research. Traditionally, this task mainly focuses on word substitution, with the same length of input and output sentences, in a relatively single form, and the research is mostly based on the SIGHAN13/14/15 evaluation task dataset. Early studies[16–18] mainly used unsupervised learning methods to identify potential errors by constructing a confusion set and determining the correctness using the language model perplexity. Later approaches treated Chinese spelling correction as a sequence labeling problem, utilizing models such as CRF or HMM to address the task[19,20].

With the rapid advancement of large-scale pre-training techniques in the field of natural language processing, pre-trained models such as BERT have been extensively employed by numerous researchers to enhance the performance of the CSC task. These models are able to efficiently correct erroneous characters by virtue of their powerful ability to capture contextual semantic information. For example, Zhang et al.[21] proposed Soft-Masked BERT, a two-stage detection and correction method, which first detects erroneous characters in the text by error probability masking, and then feeds the masked input into the BERT model for error correction. The REALISE model[22], on the other hand, takes a different approach by combining the information from three modalities, namely text, sound and vision, to comprehensively capture the semantic, phonetic and graphical features of Chinese characters, and significantly improves the performance of spell checking with the help of a selective modal fusion mechanism.The bidirectional detector-corrector framework Bi-DCSpell proposed by Wu et al.[23], which includes independent detection and correction encoders as well as an innovative interactive learning module, optimizes the representation learning process and further enhances the performance of the CSC task by promoting the interaction of bidirectional features between detection and correction.

Although the significant progress made in CSC research, its performance in the task of intelligent recognition of counterfeit goods text is unclear due to the highly scarce counterfeit goods text dataset. In view of this, this study aims to address this issue and achieve intelligent recognition of counterfeit merchandise text by weakening the negative impact of erroneous characters on contextual semantics as well as by leveraging the ability of characters' multimodal information for correction.

Methodology

Problem formulation

The complexity and diversity of counterfeit goods pose a serious challenge to market supervision, especially those cases of counterfeiting with the help of Chinese character pronunciation or character shape similarity, which are extremely covert and greatly increase the difficulty of identification and supervision. Given that the core of this task lies in the detection and correction of erroneous texts, CSC technology shows strong potential as a specialized solution for such tasks.

The CSC task is a sequence labeling problem that transforms an input sequence of characters $X = \{x_1, x_2, \ldots, x_n\}$ into an output sequence $Y = \{y_1, y_2, \ldots, y_n\}$, where the incorrect characters are corrected and the correct characters remain unchanged. Unlike other sequence-to-sequence tasks such as machine translation or text summarization, the input and output sequences of the CSC task are of the same length and most of the characters do not need to be changed, while only a few incorrect characters need to be identified and corrected.

Model

Our model consists of a corrector and an auxiliary position detection network. In particular, the corrector contains three feature extractors and a multimodal feature fusion operation. Figure 2 illustrates the structure of our model. Given a sentence, our model first performs semantic feature extraction. At the same time, two lightweight Transformer Blocks are used to capture sensitive positional information and deeply fuse it with the semantic features to generate fused semantic features that provide more accurate contextual information for the subsequent correction process. Immediately after that, pinyin features and glyph features are extracted for each character, and finally these three features are subjected to multimodal feature fusion operation. Here, in order to keep the dimension consistent after splicing, a linear projection layer is used for representation learning and dimension transformation. Subsequently, the composite representation of each character is fed into the fully

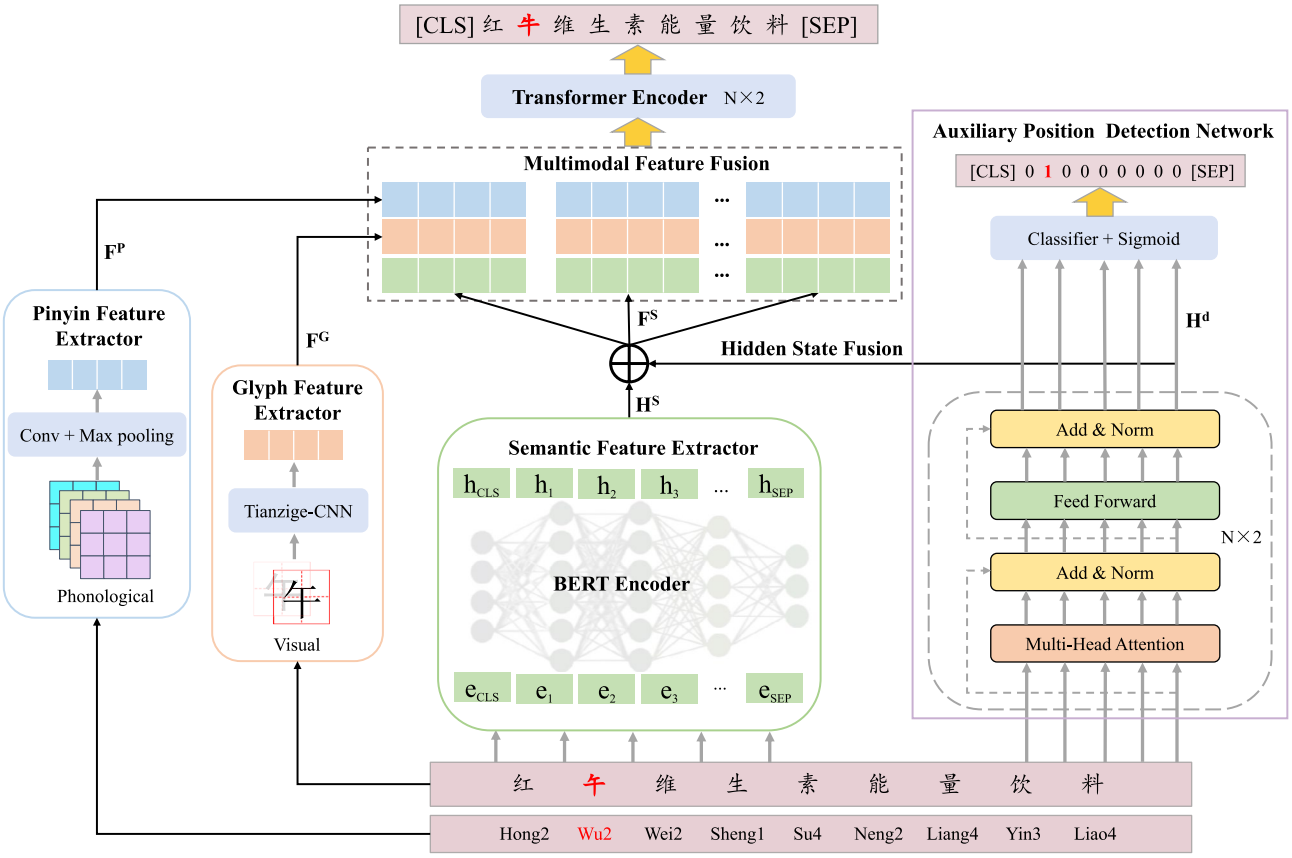

Fig. 2. Overview of the CDANet architecture. The auxiliary position detection network (right) reduces interference with downstream tasks by explicitly supervising erroneous characters. It further integrates these hidden states with semantic features from the BERT encoder (Equation 6). This fusion represents features from the pinyin extractor (Equation 8) and character shape extractor (Equation 9), consolidated through multimodal feature fusion (Equation 10) before being fed into the Transformer encoder for correction. In the example input, regarding the erroneous character '午'(Wu2, meaning noon), we need not only contextual information for assistance but also have to rely on the contextual information to help us to identify and locate the erroneous character. information for assistance but also have to rely on the visual or phonetic characteristics of the character itself to make a judgment.

connected layer, which is responsible for computing the probability distribution of each character in the entire vocabulary. Eventually, the character with the highest probability will be selected as the prediction result. In the next subsections, we will delve into the specific implementation details of each module.

Semantic feature extractor

In line with previous studies[21,22], we employed BERT [24] as the core of our semantic encoder. BERT provides extensive contextual semantic insights through unsupervised pre-training on vast text collections. When presented with an input sequence $X = \{x_1, x_2, \ldots, x_n\}$, the extractor utilizes the hidden states from the last layer of BERT $H^s = \{h_1^s, h_2^s, \ldots, h_n^s\}$ as semantic feature outputs, where $h_i^s \in \mathbb{R}^{d_s}$ representing the dimension of the semantic features d_s. This process is illustrated in the 'Semantic Feature Extractor' block of Fig. 2.

Auxiliary position detection network

The Auxiliary Position Detection Network functions as a binary classification task, leveraging the Transformer architecture. For an input text $X = \{x_1, x_2, \ldots, x_n\}$, the input to the detection network is the embedding sequence $\mathrm{E} = (\mathrm{e}_1, \mathrm{e}_2, ..., \mathrm{e}_\mathrm{n})$, and this embedding sequence is is the sum of word embeddings and positional embeddings. Each layer of Transformer uses the same block structure, which is defined as follows:

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \mathrm{head}_2, \ldots, \mathrm{head}_n)W^O \tag{1}$$

$$\mathrm{head}_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{2}$$

$$FFN(X) = max(0, XW_1 + b_1)W_2 + b_2 \tag{3}$$

Finally, the projection layer is utilized to project the coding vectors into a two-dimensional space, representing the correct and incorrect probabilities of the characters at each position, respectively. Here, *Q*, *K*, and *V* denote the current input sequence, which can be an embedding of a character or the output of the previous Transformer block. FFN and MultiHead stand for Feedforward Network and MultiHead Self-Attention, respectively, which are the basic building blocks of the Transformer model. We denote the last layer of the hidden state sequence of the Transformer module as $H^d = (h_1^d, h_2^d, \ldots, h_n^d)$, the probability of error detection is defined as P^d, and the probability distribution of the error of the ith character is derived by softmax operation $\hat{y}_i^d$:

$$P^d = \sigma(WH^d + b) \tag{4}$$

$$\hat{y}_i^d = \mathrm{softmax}(Wh_i + b) \tag{5}$$

where P^d denotes the conditional probability and σ represents the sigmoid activation function, *W* and *b* are the parameters of the classifier.

It is worth noting that the hidden state H assumes a dual role. On the one hand, it is used to predict the location of the error character. On the other hand, it passes the contextual location information to the semantic feature extractor. This is because according to experiments[21], specifying the exact location of the error character can significantly improve the effectiveness of the model. Based on this, we deeply fuse the hidden state H with the final hidden state of the semantic feature extractor to fully utilize its role.

$$F^S = H^s + H^d,\ F^S \in \mathbb{R}^{L \times d_s} \tag{6}$$

where F^S denotes the final semantic feature representation, d_s is the dimension of the final semantic feature, and L denotes the length of the input sentence. This process is illustrated in the 'Auxiliary Position Detection Network' block of Fig. 2.

Pinyin feature extractor

The pinyin of a Chinese character consists of three parts: consonants, rhymes and tones. There are 21 consonants and 39 rhymes, and they are represented by English letters. The five tones, on the other hand, can be represented by numbers. Taking the ending 'a' as an example, $\bar{a}, \acute{a}, \check{a}, \grave{a}, a$ can be mapped to the numbers 1,2,3,4,0. Specifically, we obtained the pinyin representation based on PyPinyin (a Chinese character-to-pinyin library in Python). For a given input sentence $X = \{x_1, x_2, \ldots, x_n\}$, we processed the character x_i one by one. First, the character x_i is converted to the overall pinyin form. Then, we strictly follow the Hanyu Pinyin Scheme to process the consonants and rhymes, and obtain the consonant, rhyme, and tone, i.e., the three separated forms of pinyin, for each character x_i. Then, the overall pinyin of each character and its separated forms of pinyin are connected and converted into the corresponding id representation as the final pinyin representation.

$$p_i = \left[\ p_{x_i}^{All} \cdot p_{x_i}^{Initial} \cdot p_{x_i}^{Final} \cdot p_{x_i}^{Tone}\ \right] \tag{7}$$

where $[\cdot]$ denotes the concatenation operation between embeddings, $p_{x_i}^{All}$ is the overall pinyin string representation of x_i, $p_{x_i}^{Initial}$, $p_{x_i}^{Final}$, $p_{x_i}^{Tone}$ denotes the three separated forms of consonants, rhymes, and tones respectively, and p_i is the final pinyin representation. Finally, we apply a CNN network with a convolutional kernel of 2 and a maximum pooling function to extract the pinyin information.

$$F^P = pool\left(conv\left(p_i\right)\right),\ F^P \in \mathbb{R}^{L \times d_p} \tag{8}$$

where F^P denotes the final pinyin feature representation, d_p is the dimension of the pinyin embedding, and L denotes the length of the input sentence. This process is illustrated in the 'Pinyin Feature Extractor' block of Fig. 2.

Glyph feature extractor

Glyph features are extracted using a Glyce encoder called Tianzige-CNN structure[25]. Tianzige ('田字格') is a traditional form of Chinese calligraphy. It is a four-square format (similar to the Chinese character '田'), which is ideal for beginners to practice writing Chinese characters. In a way, it is more relevant to the origin of Chinese characters than other methods, such as object detection[5] or stroke sequences [26,27]. Considering the meaning of radicals, the frame structure and the current main usage of Chinese characters, the fonts '宋体（正文）' and '黑体' were finally chosen.

In practice, the input image is first passed through a convolutional layer with 5 convolutional kernels and 1024 output channels with the aim of capturing the low-level graphical features of the image. Then, a maximum pooling operation is performed on the generated feature map with a pooling kernel size of 4, aiming to reduce the resolution of the feature map from 8×8 to 2×2. This 2×2 field grid structure effectively demonstrates the arrangement of the internal radicals of Chinese characters and their writing order. Finally, we use group convolution instead of the traditional convolution operation to map the field grid features to the final output. In this study, two types of glyphs are used, namely, '宋体（正文）' and '黑体'. Therefore, the original two-dimensional $\mathrm{d_{font}} \times \mathrm{d_{font}}$ needs to be changed to three-dimensional $\mathrm{d_{font}} \times \mathrm{d_{font}} \times 2$. Where $\mathrm{d_{font}}$ indicates the font size and 2 indicates that there are two fonts.

For a given input sentence $X = \{x_1, x_2, \ldots, x_n\}$, define its glyph embedding as follows:

$$F^G = GlyphEncoder\,(x_i) \tag{9}$$

where $F^G \in \mathbb{R}^{L \times d_g}$, L is the length of the input sentence and d_g is the dimension of the glyph embedding. This process is illustrated in the 'Glyph Feature Extractor' block of Fig. 2.

Multimodal feature fusion

After the previous feature extraction, we obtain the fused semantic features F^S, pinyin features F^P and glyph features F^G. To fully exploit the multimodal information of the erroneous characters, we employ a Multi-Layer Perceptron (MLP) to consolidate these three types of features.

$$H = \mathrm{MLP}\left(\left[F^S \cdot F^P \cdot F^G\right]\right) \tag{10}$$

where $H \in \mathbb{R}^{d_f}$ denotes the fused features, d_f denotes the dimension of the output from the Transformer encoder, which is consistent with d_s, d_p, d_g, and $[\cdot]$ denotes the concat operation between the features.

Subsequently, we utilize the Transformer encoder to thoroughly grasp these multimodal features and ascertain the probability distribution of the i-th character $\hat{y}_i^c$ using the softmax function.

$$H_l = \mathrm{Transformer}(H_{l-1}),\ l \in [1, N] \tag{11}$$

$$\hat{y}_i^c = \mathrm{softmax}(W h_i + b),\ h_i \in H_N \tag{12}$$

where N denotes the number of Transformer layers, and W and b are trainable parameters learned during the training process.

Finally, referring to the experience of[25], we combined the loss from the token classification task, the glyph classification task, and the auxiliary detection binary classification task to form our final training goal.

$$\mathscr{L}_{glyph} = -\log p(z|x) = -\log \mathrm{softmax}(W \times h_{\mathrm{image}}) \tag{13}$$

where z denotes the label of the font image x, and h_{image} is the hidden state of the CNNs in Glyce.

$$\mathscr{L}_d = -\frac{1}{n}\sum_{i=1}^{n} y_i^d \log(\hat{y}_i^d) + (1 - y_i^d)\log(1 - \hat{y}_i^d) \tag{14}$$

$$L_c = -\sum_{i=1}^{n} \log P^c(\hat{y}_i^c = y_i^c \mid X) \tag{15}$$

where $\hat{y}_i^c$, y_i^c denotes the predicted value and label of X, respectively, and $\hat{y}_i^d$, y_i^d denotes the predicted value and predicted probability of the detector, respectively, and the final training objective is given as follows:

$$\mathscr{L} = \mathscr{L}_d + \mathscr{L}_c + \mathscr{L}_{glyph} \tag{16}$$

Experiments

Dataset

CGT-Dataset: The core data resources come from Jingdong Mall. Based on the product classification standards of this e-commerce platform, we manually selected raw product descriptions from 29 core product categories,

Class	Quantity	Percentages	Class	Quantity	Percentages
Food	13836	4.77%	Indoor	10135	3.50%
Tea	13152	4.54%	Mother and Baby	10109	3.49%
Furniture	12625	4.36%	Computer	10073	3.48%
Jewellery	11834	4.08%	Men's Clothes	9501	3.28%
House Pet	11665	4.02%	Fine Chemicals	9413	3.25%
Metal Hardware	11604	4.00%	Numerals	9291	3.21%
Household	11248	3.88%	Boots and Shoes	9233	3.19%
Womenswear	11071	3.82%	Work Out	9228	3.18%
Beverage Preparation	10875	3.75%	Luggage	9003	3.11%
Wine	10866	3.75%	Undergarments	8749	3.02%
Home Decoration	10803	3.73%	Musical Instruments	8497	2.93%
Beauty Products	10723	3.70%	Personal Care Cleaning	8236	2.84%
Domestic Electric Appliance	10718	3.70%	Adult Product	5679	1.96%
Children's Clothing	10474	3.61%	Mobile Phones	960	0.33%
Cookware	10250	3.54%			

Table 1. Statistics of the CGT-Dataset.

Training Data	# Line	Avg.Length	# Errors
SIGHAN 2013	350	49.2	350
SIGHAN 2014	6,526	49.7	10,087
SIGHAN 2015	3,174	30.0	4,237
Wang271K	271,329	44.4	382,704
Test Data	**# Line**	**Avg.Length**	**# Errors**
SIGHAN 2013	1,000	74.1	1,227
SIGHAN 2014	1,062	50.1	782
SIGHAN 2015	1,100	30.5	715

Table 2. Statistical data from generic domain datasets.

totaling 289,851 entries. This dataset encompasses multiple aspects of consumers' daily lives, forming a text corpus with broad representativeness and deep coverage. Since all raw texts are directly sourced from online product sales descriptions, no anomalous data is included. For specific data distribution, please refer to Table 1.

To construct more authentic and representative counterfeit product text samples, we first conducted an in-depth analysis of common error patterns in existing CSC benchmark datasets (e.g., SIGHAN2013[28], SIGHAN2014[29], SIGHAN2015[19], and Wang271K[4]). Our research revealed that these errors are highly concentrated in homographs and homophones. Inspired by this, we designed a semi-automated generation process to create the counterfeit product text dataset. Specifically, core entities such as brand names or key product attributes were identified from the original product description texts. Subsequently, for each entity, 1 to 4 characters were selected with a certain probability based on its name length for obfuscation replacement. The obfuscation strategies included visually similar characters, homophones, near-homophones, and random substitutions. This process references common error patterns observed in benchmark datasets and employs a priority-based multi-type substitution mechanism, with a 74.9 % probability of character replacement. For instance, 蓝月亮 (Lán Yuè Liàng) is replaced with 蓝月壳 (Lán Yuè Ké), simulating a near-homophone error. The final dataset comprises approximately 70 % homograph errors, 20 % homophone/near-homophone errors, and 10 % random substitution errors.

Finally, we developed and applied a suite of automated scripts to execute character substitution, structured output, statistical analysis, and preliminary text quality control. Additionally, all generated samples underwent manual review by two annotators to ensure authentic simulation of common forgery techniques while maintaining contextual coherence. Any illogical samples were discarded. This rigorous quality control process guarantees the high quality and relevance of the final dataset.

Public Benchmark Datasets: To evaluate the generalization ability of our model, we also test it on three widely-used public Chinese Spelling Correction datasets: SIGHAN 2013, SIGHAN 2014, and SIGHAN 2015. The statistics of these datasets are summarized in Table 2.

Baselines

We analyzed the CDANet model in comparison with the following classical models:

-BERT[24]: A language representation model pre-trained on the Transformer architecture. BERT employs a bidirectional training approach that takes into account both the left and right contexts of a word, thereby capturing more comprehensive semantic information.

-Soft-Masked BERT[21]: This model employs a two-stage approach. It first utilizes an error detection network to predict the probability of each character being incorrect. These probabilities are then used to apply a 'soft mask' to the input sequence, which is subsequently fed into a BERT-based correction network to generate the corrected text.

-REALISE[22]: An end-to-end Chinese spelling correction model. Unlike traditional methods that use special tokens (e.g., [MASK]) to mask words during language model training, REALISE replaces target words with their phonetic characteristics and similar words. Additionally, it incorporates an adaptive weighting mechanism to simultaneously train error detection and correction tasks within a cohesive framework.

-MDCSpell[30]: A multi-task framework for detectors and correctors. The corrector captures the visual and phonetic features of each character in a sentence based on BERT, and integrates the hidden states of the detector and the corrector through a later fusion strategy to reduce the interference of spelling errors on the corrector and improve the error correction effect.

-ReLM[31]: A Chinese spelling correction method based on sentence reconstruction instead of traditional character-to-character annotation. The method corrects errors by reconstructing the whole sentence, simulating human error correction thinking and improving the generalisation ability and migration performance of the model.

-PTCSpell[32]: A pre-trained model designed for the Chinese spelling correction task. It combines visual and phonetic features of Chinese characters to improve correction accuracy, while mitigating the negative impact of detection errors on the correction process by balancing the loss function.

Implementation details

The experimental hardware consists of an Intel(R) Xeon(R) Platinum 8474C @ 3.00 GHz CPU and two NVIDIA GeForce RTX A6000 GPUs (48 GB of graphics memory each). The model implementation is based on the PyTorch framework[33] and the Transformer library provided by Huggingface[34]. The semantic feature extractor uses a pre-trained bert-base-chinese model (pre-trained on a large Chinese corpus)[1]. The first two layers of the model are used to initialise the weights of the Transformer in the auxiliary location detector.

For training the CDANet model, we utilized the Adam optimizer[35] with a batch size of 256, a learning rate of 2e-5, and trained for 10 epochs. The CGT-Dataset was divided into training, validation, and test sets in a ratio of 7:2:1. We used the SIGHAN 2013, SIGHAN 2014, and SIGHAN 2015 benchmark datasets Consistent with previous studies[22,36], we employed the OpenCC 1.0 tool[2] to convert traditional Chinese characters in our datasets. convert traditional Chinese characters in our dataset to simplified characters. Our evaluation metrics, including sentence-level Precision, Recall Our evaluation metrics, including sentence-level Precision, Recision, Recall, and F1-score, are in line with those used in prior research. It's important to note that the correction task is significantly more complex than the detection task, as the success of correction is contingent upon the accuracy of the initial detection.

Main results

The main experimental results on our self-constructed CGT-Dataset are presented in Table 3. On the CGT-Dataset, our proposed CDANet model achieves state-of-the-art performance across all metrics for both detection and correction tasks. Specifically, CDANet's F1 scores for detection (96.3%) and correction (89.3%) surpass the strongest baseline models byby 2.7 percentage points (vs. MDCSpell) and 0.6% (vs. PTCSpell), respectively. This demonstrates the superiority of our approach on the target task of counterfeit goods text recognition.

To further analyze the source of these improvements, we compare our model with two closely related works, REALISE and MDCSpell. CDANet exhibits overall superior performance to REALISE, particularly in terms of robustness to multimodal interference. Although REALISE also integrates semantic, pinyin, and grapheme features, its direct integration of the erroneous characters' multimodal information can easily interfere with the contextual semantics. In contrast, our proposed Auxiliary Position Detection Network is designed to first identify the error's position, which significantly enhances the model's robustness to such interference. The results on the CGT-Dataset validate this, showing a 0.3 percentage point improvement in correction-level precision over REALISE. It is also worth noting that MDCSpell adopts a similar detection-assisted correction design. However, its utilization of visual and phonetic features is still implicitly limited to the pre-training capability of BERT, leaving room for optimization. Our experiments fill this gap by explicitly modeling these features. As can be seen from the experimental results, this leads to a meaningful improvement of 1.3% in the F1 score over MDCSpell, which further demonstrates the effectiveness of incorporating phonological and morphological knowledge into the semantic space.

Finally, to verify the generalization ability of CDANet, we conducted further experiments on the SIGHAN2013, SIGHAN2014 and SIGHAN2015 datasets. As can be seen from the results in Table 3, CDANet performs well on both detection and correction tasks. Compared with the BERT baseline that utilizes only contextual semantic information, CDANet improves the F1 scores of the correction task by 10.4%, 12.8% and 11%, respectively. These significant improvements further validate the effectiveness and robustness of our approach. In addition, compared to PTCSpell, although PTCSpell has higher precision in recognizing and correcting spelling errors by specifically learning the visual and phonetic features of Chinese characters through a pre-training phase, its

[1] https://huggingface.co/google-bert/bert-base-chinese
[2] https://github.com/BYVoid/OpenCC

Dataset	Model	Detection			Correction		
		Prec.	Rec.	F1.	Prec.	Rec.	F1.
CGT-Dataset	Soft-Masked BERT[21]	89.2	87.6	88.4	88.6	85.2	86.9
	REALISE[22]	91.8	90.0	90.9	90.7	86.3	88.4
	MDCSpell[30]	95.2	92.1	93.6	88.2	87.9	88.0
	PTCSpell[32]	96.1	90.3	93.1	90.8	86.6	88.7
	Bi-DCSpell*[23]	95.3	91.3	93.3	90.0	85.4	87.6
	ReLM[31]	-	-	-	90.4	85.9	88.1
	BERT[24]	85.6	83.4	84.5	82.9	80.4	81.6
	CDANet(ours)	**98.2**	**94.5**	**96.3**	**91.0**	**87.9**	**89.3**
SIGHAN 2013	Soft-Masked BERT*[37]	81.1	75.7	78.3	75.1	70.1	72.5
	REALISE[22]	88.6	82.5	85.4	87.2	**81.2**	84.1
	PTCSpell[32]	**99.7**	80.6	**89.1**	**99.7**	79.2	**88.3**
	Bi-DCSpell*[23]	88.2	80.6	84.2	86.8	78.7	82.6
	BERT[24]	79.0	72.8	75.8	77.7	71.6	74.6
	CDANet (ours)	89.3	**82.6**	85.8	89.2	**81.2**	85.0
SIGHAN 2014	Soft-Masked BERT *[37]	65.2	70.4	67.7	63.7	68.7	66.1
	REALISE[22]	67.8	71.5	69.6	66.3	70.0	68.1
	PTCSpell[32]	**84.1**	71.2	77.1	**83.8**	69.4	75.9
	Bi-DCSpell*[23]	69.9	70.9	70.4	68.5	68.7	68.6
	BERT[24]	65.6	68.1	66.8	63.1	65.5	64.3
	CDANet (ours)	79.5	**78.2**	**78.8**	77.7	**76.5**	**77.1**
SIGHAN 2015	Soft-Masked BERT*[37]	67.7	78.7	72.7	63.4	73.9	68.3
	REALISE[22]	77.3	81.3	79.3	75.9	79.9	77.8
	PTCSpell[32]	**89.6**	81.2	**85.2**	**89.4**	79.0	83.8
	Bi-DCSpell*[23]	79.6	82.4	81	77.5	80.2	78.8
	BERT[24]	73.7	78.2	75.9	70.9	75.2	73.0
	CDANet (ours)	83.4	**85.1**	84.2	83.0	**84.9**	**84.0**

Table 3. Experimental results on the CGT-Dataset, SIGHAN13, SIGHAN14 and SIGHAN15 test sets. Each model includes sentence-level precision, recall and F1 score for detection and correction. * indicates that the results are derived from[37], and ‡ indicates that the results are taken from Chinese-BERT-wwm because the ChineseBERT pre-training process enhances the utilization of glyph (Glyph) and pinyin (Pinyin) information.

recall is not significantly improved. On the other hand, CDANet does not require additional pre-training tasks, reducing the dependence on resources and data labeling, while maintaining a better balance on all metrics. This shows that our approach not only has a lower implementation cost but also performs consistently across diverse task scenarios.

We also conducted an in-depth analysis of why CDANet exhibits inconsistent performance across different datasets. Our research revealed that the model demonstrates significantly greater improvement on the CGT-Dataset compared to its performance on the SIGHAN benchmark dataset. This discrepancy primarily stems from fundamental differences in the nature of errors across the two datasets: the CGT-Dataset is characterized by high-density, intentional phonetic and orthographic substitution errors within its named entities. CDANet's model architecture enables exceptional performance on this task by explicitly modeling pinyin and character features, while its auxiliary detection module effectively focuses on such high-impact errors. In contrast, the SIGHAN dataset exhibits a richer error distribution encompassing syntactic and cognitive-level errors. In these scenarios, purely semantic models like BERT have demonstrated strong error correction capabilities, thereby narrowing the performance gap between CDANet and baseline models.

Ablation study

To systematically validate the contribution of each component within CDANet, we conducted a comprehensive series of ablation studies on both the CGT-Dataset and the SIGHAN2015 test set. The results, as delineated in Table 4, unequivocally demonstrate the effectiveness of our proposed auxiliary position detection network and multimodal feature integration strategy. On the CGT-Dataset, every component proved to be crucial for the model's performance. The most significant performance degradation was observed upon removing the auxiliary position detection network (`- Auxiliary Position`), which caused the F1 score to plummet by 8.0 points from 89.3 to 81.3. This result underscores the pivotal role of this network in accurately locating errors and mitigating the semantic interference caused by incorrect characters, which is a core challenge in our dataset. The value of multimodal information was also prominently highlighted; removing both Pinyin and glyph features simultaneously (`- Pinyin & Glyph`) resulted in the second-largest performance drop, a 7.1-point decrease in the F1 score. Furthermore, ablating either the Pinyin (`- Pinyin`) or glyph (`- Glyph`) features individually

	CGT-Dataset			SIGHAN2015		
	Prec.	Rec.	F1.	Prec.	Rec.	F1.
BERT[24]	77.7	71.6	74.6	70.9	75.2	73.0
CDANet (ours)	**91.0**	**87.9**	**89.3**	**83.0**	**84.9**	**84.0**
- Pinyin	84.2	83.5	83.8	80.5	81.0	80.7
- Glyph	84.8	85.1	85.0	81.2	81.9	81.5
- Pinyin & Glyph	82.0	82.5	82.2	79.1	80.2	79.6
- Auxiliary Position	81.0	81.6	81.3	78.5	79.5	79.0

Table 4. Correction-level average ablation results for the CDANet model on the CGT-Dataset and SIGHAN2015 test set. We made the following modifications to CDANet: remove the pinyin feature extractor (- Pinyin), remove the glyph feature extractor (- Glyph), remove both the pinyin and glyph feature extractors (- Pinyin & Glyph), remove the auxiliary position detection network (- Auxiliary Position).

also led to notable performance declines of 5.5 and 4.3 F1 points respectively, confirming that they provide critical and complementary information for the correction task.

To further evaluate the generalization ability of our model architecture, we replicated the same ablation experiments on the public SIGHAN2015 benchmark. The results exhibited a consistent trend with our findings on the CGT-Dataset, re-validating the importance of each component in a broader context. Once again, the auxiliary position detection network was the most critical module, as its removal led to the largest F1 score drop of 5.0 points. Similarly, the multimodal features continued to provide a significant advantage over a purely semantic approach, with their removal causing a 4.4-point F1 decrease. Notably, on both datasets, all ablated model variants still comprehensively outperformed the baseline BERT model. This strongly proves that the fundamental architecture of CDANet is robust and that every integrated component makes an indispensable and positive contribution to the model's overall effectiveness in both specialized and general Chinese spelling correction scenarios.

Conclusion

In this paper, we proposed a Corrector-Detector Auxiliary Network, CDANet, designed to intelligently recognize counterfeit goods text that relies on phonetic or glyph similarity. Addressing the core challenge of high visual and auditory similarity in erroneous characters, CDANet effectively incorporates glyph, pinyin, and contextual semantic features. The complementarity of this multimodal information significantly enhances the model's recognition capabilities by providing discriminative signals beyond standard semantics. Furthermore, to mitigate the issue of misleading contextual semantics caused by incorrect characters, we introduced an Auxiliary Position Detection Network, which improves correction accuracy by precisely locating errors.

A key contribution of this work is also the construction of the CGT-Dataset, a large-scale textual dataset of counterfeit goods containing 289,851 samples, which facilitates research in this specific domain. Extensive experiments show that CDANet achieves state-of-the-art performance on our proposed CGT-Dataset and demonstrates strong, competitive performance on the public SIGHAN benchmark datasets. These results validate the model's effectiveness for the target task and its robust generalization ability.

Limitations and future work

Despite the promising results, this study has several limitations. First, our CGT-Dataset, while large, is constructed from a single e-commerce platform, which may not capture the full diversity of counterfeit text styles across different sources. Second, the counterfeit texts are generated via simulation, which may not fully encompass the complexity and subtlety of authentic, "in-the-wild" examples. Lastly, our model's primary focus is on phonetic and glyph-based errors, and its performance on more complex, semantic-level counterfeiting has yet to be explored.

Future work could proceed in several exciting directions. We plan to enrich our dataset by incorporating data from multiple platforms and including more sophisticated examples of counterfeiting. Another valuable step would be to investigate the adaptation and deployment of CDANet in a real-world, large-scale detection system to assess its practical utility and efficiency.

Data availability

The CGT-Dataset introduced in this study is available upon reasonable request from the corresponding author.

Code availability

The code supporting the findings of this study is available upon reasonable request from the corresponding author.

Materials availability

Not applicable. This study does not involve any specific materials requiring access.

Received: 1 August 2025; Accepted: 18 November 2025

Published online: 27 November 2025

References

1. Peng, J., Zou, B. & Zhu, C. A two-stage deep learning framework for counterfeit luxury handbag detection in logo images. *Signal Image Video Process.* **17**, 1439–1448 (2023).
2. Şerban, A., Ilaş, G. & Porușniuc, G.-C. Spotthefake: an initial report on a new cnn-enhanced platform for counterfeit goods detection. arXiv preprint arXiv:2002.06735 (2020).
3. Kumar, S. N., Singal, G., Sirikonda, S. & Nethravathi, R. A novel approach for detection of counterfeit indian currency notes using deep convolutional neural network. In *IOP conference series: materials science and engineering*, vol. 981, 022018 (IOP Publishing, 2020).
4. 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*, 2517–2527 (2018).
5. Huang, L. et al. 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)*, 5958–5967 (2021).
6. Yeh, J.-F., Li, S.-F., Wu, M.-R., Chen, W.-Y. & Su, M.-C. Chinese word spelling correction based on n-gram ranked inverted index list. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, 43–48 (2013).
7. Afli, H., Qui, Z., Way, A. & Sheridan, P. Using smt for ocr error correction of historical texts. In *Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 23–28 (2016).
8. 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*, 739–747 (2010).
9. Baldini, G., Fovino, I. N., Satta, R., Tsois, A. & Checchi, E. Survey of techniques for the fight against counterfeit goods and intellectual property rights (ipr) infringement. *Publ Off Eur Union* 1–130 (2015).
10. Xie, R., Hong, C., Zhu, S. & Tao, D. Anti-counterfeiting digital watermarking algorithm for printed qr barcode. *Neurocomputing* **167**, 625–635 (2015).
11. Bansal, D., Malla, S., Gudala, K. & Tiwari, P. Anti-counterfeit technologies: a pharmaceutical industry perspective. *Sci. Pharm.* **81**, 1 (2012).
12. Ol'ga, F. C., Tat'yana, V. P. & Gorbacheva, M. V. Biological analysis for counterfeit detection of orenburg downy shawls. *Theory and Practice of Forensic Science* **13**, 88–96 (2018).
13. Taylor, D. Rfid in the pharmaceutical industry: addressing counterfeits with technology. *J. Med. Syst.* **38**, 141 (2014).
14. Garcia-Cotte, H., Mellouli, D., Rehman, A., Wang, L. & Stork, D. G. Deep neural network-based detection of counterfeit products from smartphone images. arXiv preprint arXiv:2410.05969 (2024).
15. Mishra, A. K. & Essop, M. H. Low-cost spectrogram based counterfeit medicine detection. arXiv preprint arXiv:1904.07152 (2019).
16. Chen, K.-Y., Lee, H.-S., Lee, C.-H., Wang, H.-M. & Chen, H.-H. A study of language modeling for chinese spelling check. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, 79–83 (2013).
17. 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*, 220–223 (2014).
18. Xie, W. et al. Chinese spelling check system based on n-gram model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, 128–136 (2015).
19. 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 Eighth SIGHAN Workshop on Chinese Language Processing*, 32–37 (2015).
20. 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* (2015).
21. Zhang, S., Huang, H., Liu, J. & Li, H. Spelling error correction with soft-masked bert. arXiv preprint arXiv:2005.07421 (2020).
22. Xu, H.-D. et al. Read, listen, and see: Leveraging multimodal information helps chinese spell checking. arXiv preprint arXiv:2105.12306 (2021).
23. Wu, H., Zhang, H., Xuan, R. & Song, D. Bi-dcspell: A bi-directional detector-corrector interactive framework for chinese spelling check. arXiv preprint arXiv:2406.01879 (2024).
24. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).
25. Meng, Y. et al. Glyce: Glyph-vectors for chinese character representations. *Adv. Neural Inf. Process. Syst.* **32** (2019).
26. Han, Z., Lv, C., Wang, Q. & Fu, G. Chinese spelling check based on sequence labeling. In *2019 International Conference on Asian Language Processing (IALP)*, 373–378 (IEEE, 2019).
27. 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)*, 2991–3000 (2021).
28. Wu, S.-H., Liu, C.-L. & Lee, L.-H. Chinese spelling check evaluation at sighan bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, 35–42 (2013).
29. 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 Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, 126–132 (2014).
30. 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**, 1244–1253 (2022).
31. Liu, L., Wu, H. & Zhao, H. Chinese spelling correction as rephrasing language model. *Proc. AAAI Conf. Artif. Intell.* **38**, 18662–18670 (2024).
32. 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**, 6330–6343 (2023).
33. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **32** (2019).
34. Wolf, T. et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45 (2020).
35. Kingma, D. P. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
36. Cheng, X. et al. Spellgcn: Incorporating phonological and visual similarities into language models for chinese spelling check. arXiv preprint arXiv:2004.14166 (2020).
37. Zhang, R. et al. Correcting chinese spelling errors with phonetic pre-training. *In Findings of the Association for Computational Linguistics: ACL-IJCNLP* **2021**, 2250–2261 (2021).

Author contributions

T.W. was responsible for the investigation, software, data analysis, original draft writing, and visualization. Y.L. contributed to the conceptualization, methodology, and original draft writing. L.W. contributed to data collec-

tion, methodology validation, and manuscript review & editing. L.Z. handled funding acquisition, methodology, validation, and manuscript review & editing. N.L. managed data curation, resources, and manuscript review & editing. All authors reviewed the manuscript.

Funding
This research was supported by National Key R&D Program of China (2023YFC3304903).

Declarations

Competing interests
The authors declare no competing interests.

Ethics approval and consent to participate
Not applicable. This study did not involve human participants or animals.

Consent for publication
All authors consent to the publication of this manuscript.

Additional information
Correspondence and requests for materials should be addressed to L.Z. or N.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.