Computer Modeling in Engineering & Sciences

Tech Science Press

https://doi.org/10.32604/cmes.2026.085408

ARTICLE

SE-CSC: A Novel Summarization-Enhanced Chinese Spelling Check with Phonetic and Glyph Embeddings

Wen-Chin Hsu, Yi-Cheng Chen* and Yi-Hsuan Kuo

Department of Information Management, National Central University, Taoyuan City, Taiwan

*Corresponding Author: Yi-Cheng Chen. Email: ycchen@mgt.ncu.edu.tw

Received: 11 May 2026; Accepted: 31 July 2026; Published: 28 August 2026

ABSTRACT: Due to the structural complexity of Chinese characters, the occurrence of homophones and visual similarity among glyphs directly increases the difficulties presented in Chinese spell checking (CSC). These factors also indicate the importance of the connection between CSC and context-dependency. In this study, a novel framework, the Summarization-Enhanced Chinese Spell Checking (abbreviated as SE-CSC) model, is proposed, which integrates phonetic and glyph embeddings to further enhance context awareness in error detection and correction. We utilize sentence-level summarization features to augment and generate an error-guided mask that can effectively detect errors and derive more precise corrections. Several comprehensive experiments conducted on real datasets demonstrated the superiority of the proposed SE-CSC compared to existing baselines, particularly in reducing miscorrections and improving accuracy. In addition, attention visualizations and case studies are provided to confirm the ability of SE-CSC for key contextual concentration, offering a structured and adaptable approach for spelling correction in linguistically complex environments.

KEYWORDS: Spelling check; Chinese spell checking; natural language processing; text summarization

1 Introduction

Over the past decade, Natural Language Processing (NLP) has emerged as a highly attractive topic in the field of artificial intelligence, due to its wide-ranging applications, e.g., smart assistants, text generation, and sentiment analysis, to name a few. With the surge in and the advancement of comprehensive digitalization, massive volumes of unstructured data have been generated. NLP is particularly effective in terms of processing and analyzing unstructured data with the extraction of valuable insights, which are crucial for businesses to make decisions or to understand their customers.

Spell checking (SC) is one essential NLP technique which plays an important role in widespread applications. It aims to ensure that the words and phrases in a text are spelled correctly, thereby enhancing communication clarity and overall readability. In modern digital services, SC is not merely an auxiliary tool for text editors but is also important in many other applications. For example, in e-commerce and web searching, users may struggle to find useful information because of misspelled merchandise. As a result, search engines are generally equipped with SC techniques to check the original query and to derive its corrected form for improving search accuracy and hit rates. Whether in daily communication, academic writing, or professional documentation, the correct use of language is undoubtedly a basic requirement. This commitment to linguistic accuracy forms the necessary foundation for building efficient and robust SC models.

Among many languages, Chinese spell checking (CSC) is considered a challenging task. Indeed, using the same approach as SC, CSC aims to detect and convert incorrect Chinese words into correct ones in sentences. However, compared with alphabetic spelling systems, the Chinese writing system has a unique characteristic. Chinese uses characters as the basic unit, and each word directly carries a specific meaning; in comparison, most other language systems use letters as the basic unit, and only combinations of letters can form meaningful words. This fundamental difference makes the CSC task more complex and difficult.

We give an example to explain in detail. Fig. 1 illustrates the differences in the spelling error characteristics of English and Chinese. As shown in Fig. 1a, in English, since the combination of English letters is relatively fixed, spelling errors can often be detected quickly. For instance, when there is a problem with the order of letters in a word, most people can easily identify and correct it based on the spelling rules and context. However, CSC presents totally different challenges from those of general SC. As shown in Fig. 1b,c, Chinese characters are not only complex in structure but also feature numerous homophones and homographs. The same pronunciation may correspond to dozens of different Chinese characters, and these characters may represent different meanings in different contexts; the similarity of glyphs directly increases the difficulty of identification and correction. Hence, CSC requires comprehensive analysis including pronunciation, glyph similarity, and semantic context.

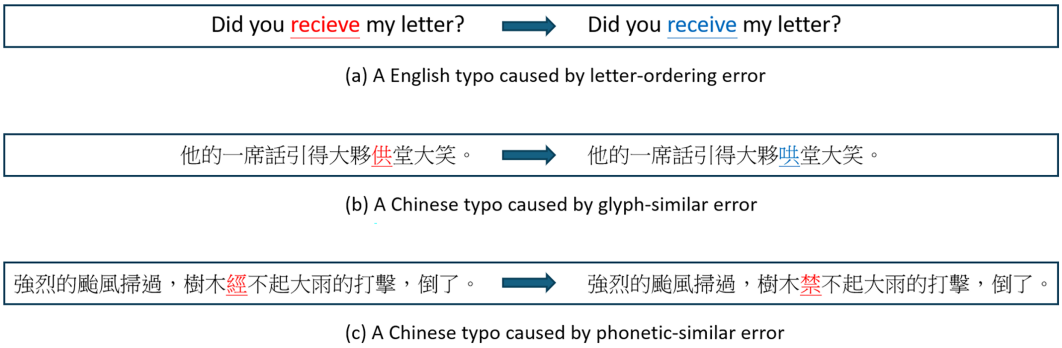


Figure 1: An example of English and Chinese typos with different types of errors.

As mentioned above, designing efficient error correction models and enhancing automatic error correction capabilities has emerged as a focal point of research. Despite recent advancements in NLP, CSC still faces some problems that may hinder model development. First, one challenge is overcorrection, where correct words are mistakenly identified as errors, or incorrect words are corrected into other common but equally wrong words. Obviously, the overcorrection problem may lead to unintended changes in the original meaning of the sentence. Second, without a doubt, the domain specification usually affects the precision of CSC. When a CSC model is applied across different domains, domain-specific terminology and writing styles vary significantly. For example, the phrases used in transportation differ considerably from those in telecommunications, leading to domain-specific spelling patterns and error types. If a CSC model is trained primarily on general corpora, it may fail to recognize rare or technical terms correctly. Therefore, adapting CSC models to diverse domains is essential for both improving correction accuracy and minimizing semantic distortion caused by incorrect edits.

Finally, although large language models (LLMs) have achieved tremendous success across a wide range of NLP tasks, CSC by LLMs still faces certain limitations when compared to stand-alone applications. A stand-alone application refers to a system that can operate entirely on local devices without requiring internet connectivity or access to external cloud-based services. Hence, the stand-alone ability is also an important issue for CSC models. A stand-alone CSC could offer greater flexibility in environments with limited or

no network access and provide enhanced data privacy, as all computations are performed locally without transmitting sensitive information to remote servers. Stand-alone CSC models are particularly suitable for scenarios involving highly confidential data, for example, corporate documents, medical records, or government communication, for secure deployment.

In this study, a stand-alone framework, Summarization-Enhanced Chinese Spell Checking (abbreviated as SE-CSC), was proposed to directly tackle the aforementioned limitations and challenges with several optimization strategies. To more effectively solve the problem of Chinese typo correction, we split the CSC task into three sub-tasks: summarization, detection, and correction. We first extract a summary of the input text to derive the vital information. Such a summary could highlight the most semantically relevant parts of a sentence to distinguish meaningful tokens from potential noise, especially in domain-specific or ambiguous contexts. Clearly, the summarization provides a global semantic representation of the input sentence. Summarization could capture the main intent or semantic focus of the sentence, which helps the SE-CSC model distinguish between candidates that are phonetically or visually similar but semantically different. This summarized sentence-level information directly highlights semantically important regions and guides the model to focus on suspicious characters that disrupt the sentence meaning. Furthermore, the summary integration also strengthens the connection between character-level features and sentence-level semantics. Phonetic and glyph embeddings are effective for modeling Chinese spelling error patterns, but they may be insufficient when correction requires deeper contextual reasoning. Summarization complements these features by providing a high-level semantic constraint, enabling the model to make corrections that are not only phonetically or visually reasonable but also contextually coherent. Then, by explicitly identifying the positions of potential errors before attempting any modifications, we utilize the detection module to induce the error likelihood and location. These valuable signals could effectively guide the correction module to revise only required tokens, thereby improving both precision and robustness. In the correction stage, we first compute the empirical correction probabilities between erroneous and corrected characters based on large-scale training data. SE-CSC integrates the correction probabilities with the signals from the summarization and detection module to generate an accurate correction. This strategy improves the domain adaptability of the SE-CSC model while also reducing the possibility of unlikely or uncommon substitutions.

The main contributions of the proposed SE-CSC framework are detailed as follows:

- We propose a tri-stage SE-CSC framework including summarization, detection and correction modules. This design enabled the CSC model to first localize likely errors and then refine the correction with more compact and focused information, which significantly improves both precision and interpretability.
- SE-CSC introduces a novel summarization-guided CSC strategy. The summarization of the input sentence is generated and incorporated through a cross-attention mechanism with the original input embeddings. This integration encourages the error detection to attend more strongly to semantically important tokens. Empirical results show that including summarization information leads to consistent improvements in detection accuracy, as demonstrated by the ablation experiments presented in the paper.
- We propose an error-guided mask equipped with SE-CSC to optimize error correction. The mask is constructed by analyzing error-correction patterns in the training corpus and computing the probability distribution of common corrections. These probabilities are transformed using a logarithmic function and applied as a bias to the model output, effectively increasing the likelihood of contextually probable corrections. Several experiments demonstrate that this masking mechanism significantly improves correction accuracy.
- CSC has been comprehensively adopted as a preprocessed checking component for system input in many applications. Compared to current LLM-based models, SE-CSC has a better response time and

lower computation resources, which is a critical issue for real-time scenarios. The proposed SE-CSC model could be implemented and encapsulated as an application programming interface (API) to easily integrate into a system framework.
– The proposed SE-CSC was applied to SC tasks on several real datasets to demonstrate its outstanding performance and practicability. In addition, a case study is given to discuss the CSC results and to analyze the correction behavior.

The organization of the remaining parts in this paper is as follows. Section 2 discusses the Related Work and Section 3 presents the proposed SE-CSC framework in detail. Section 4 provides the experimental results of a performance study. Finally, Section 5 concludes the paper.

2 Related Work

2.1 Chinese Spell Checking

Many prior studies have discussed effective solutions for CSC. CNM [1] is an early traditional method that integrates both bi-gram and tri-gram language models alongside Chinese word segmentation to handle textual input. To enhance its computational efficiency, CNM incorporates dynamic programming techniques to mitigate the issue of data sparsity commonly found in n-gram-based models. CCSE [2] includes phonological information in the model by utilizing a pre-training and fine-tuning framework. Rather than adopting the traditional approach of masking tokens with a special symbol, CCSE substitutes tokens with the corresponding phonetic representations and acoustically similar characters. Based on BERT [3], CP-BERT [4] was designed to tackle the prevalent issue of spelling mistakes in formal document writing. To enhance correction accuracy, CP-BERT also employs a Bi-LSTM network to identify potential error positions and integrates phonetic prior knowledge specifically. REALISE [5] addresses the CSC task by directly utilizing the multimodal characteristics of Chinese characters. By capturing semantic, phonological, and visual cues from the input, REALISE significantly improves the accuracy of error correction. PHMOSpell [6] is an end-to-end framework that extracts phonetic and visual features of Chinese characters from audio and image modalities. These representations are incorporated into a pre-trained language model through an adaptive gating mechanism specifically designed to fuse multimodal information. SpellGCN [7] integrates phonological and visual similarity information through a dedicated graph convolutional architecture. It constructs a character-level graph and transforms it into a collection of interrelated classifiers. Similarly, CLSpell [8] employs contrastive learning to integrate multiple sources of information, while leveraging multi-task joint training to simultaneously capture both local and global contextual features. PLOME [9] replaces selected tokens with visually or phonetically similar characters from a confusion set instead of using mask tokens. It incorporates a pronunciation prediction task to capture phonetic-level error patterns, and employs GRU networks to model character phonology and stroke information. ECSpell [10] employs an error-consistent masking strategy during pretraining to bridge the significant gap between real-world inputs and automatically generated corpora. FASPell [11] combines a denoising autoencoder (DAE) to enhance computation speed and structural simplicity while maintaining powerful error detection and correction. RMVSpell [12] introduce a post-fusion network to fuse features from different modalities at multiple levels.

Several framework-based solutions have also been adopted to address both error detection and correction. Soft-Masked BERT [13] links error detection and correction via a soft-masking mechanism, which improves the correction network with detection results in a differentiable manner. MDCSpell [14] adopts a detection-correction architecture to mitigate the influence of misspelled characters. The model applies a late fusion strategy by integrating the hidden representations from both the detector and corrector. DCSpell [15] proposes a transformer-based detection-correction framework with a confusion-set-based post-processing step to refine the final corrections. Tail-to-Tail [16] network utilizes a BERT-initialized

encoder for information modeling to capture bidirectional context for error correction. To tackle the local context issue for inaccurate corrections, GAD [17] leverages broad contextual cues to boost correction accuracy with global dependencies between correct characters and likely error variants. CRASpell [18] handles multiple typos with noisy contexts to retain correct characters for reducing unnecessary changes and overcorrections. CoSPA [19] incorporates an alterable copy mechanism to prevent incomplete detection of phonetic or morphological errors. The model enhances shape representation by mining character glyphs with ResNet and integrating stroke features via an adaptive gating unit. DORM [20] addresses several challenging issues of CSC by separating textual and phonetic features to learn phonetic representations. A self-distillation module in DORM effectively utilizes semantic information to enhance performance across multiple CSC benchmarks. MCRSpell [21] learns semantic knowledge adaptively from multiple intermediate features to construct separate models for the spelling error correction task without any data augmentation. LSTM-Transformer [22] uses a global attention mechanism to capture token dependencies and local context within the sequence for CSC tasks. DRMSpell [23] introduced a dynamic multimodality module to reweight various modalities for obtaining more multimodal information in CSC.

Post-processing techniques have also emerged as critical components of CSC pipelines. Think Twice [24] identifies the issue of mismatch between training corpora and real-world text, and enables a post-processing operation to filter incorrect outputs based on character and contextual features. PTCSpell [25] improves correction quality by integrating pronunciation and shape pre-training objectives. To ensure detector precision, the PTCSpell model balances the loss with incorrect and correct characters. ECOPO [26] enhances pre-trained language models by adjusting the internal representations to reduce the tendency of frequent incorrect characters. Since pretrained language models often favor semantically plausible or high-frequency substitutions, ECOPO could be integrated with various existing CSC frameworks to further improve performance. Bao et al. [27] expanded traditional confusion sets by including semantically relevant candidates to handle a wider range of error types. Several global optimization strategies have also been discussed to improve overall coherence and accuracy. DCN [28] uses an attention-based network to capture the dependencies between consecutive characters and to facilitate correction model training. SpellBERT [29] integrates stroke and phonetic features through a graph neural network to achieve competitive results despite having only half the size of the standard BERT model on CSC tasks. ChineseBERT [30] integrates both glyph and phonetic information into the pretraining process to effectively address the common issue of heteronyms in Chinese. Yang et al. [31] encoded the pronunciations of both context and target characters to leverage phonology, morphology, and semantics for comprehensive representation learning in Chinese word embeddings.

2.2 Text Summarization

Abstractive summarization mainly generates new sentences by modeling the semantic content of the source document. Several prior studies have made successful advancements in extractive summarization across many applications. HETERSUMGRAPH [32] strengthens cross-sentence connections using semantic nodes of different granularities and extends multi-document summarization by considering document nodes. Zhong et al. [33] aligned source documents with candidate summaries in a shared semantic space to address the gap between sentence-level and summary-level extraction of dataset properties. SummaRuN-Ner [34] is an RNN-based extractive summarization model that enables learning from human-written summaries without requiring sentence-level extractive labels. BanditSum [35] avoids reliance on heuristic labels by training models with a policy gradient reinforcement concept to optimize sentence evaluation scores. BERTSUM [36] leverages pre-trained BERT architecture with reinforcement learning models [37] to rank and generate concise summaries in many real datasets. GAN^2 [38] proposed a dual generative

adversarial network architecture to achieve precise text generation which could be directly applied to text summarization.

3 Proposed Model: SE-CSC

The goal of this study was to develop an efficient and effective model for CSC tasks. We proposed a novel framework, Summarization-Enhanced Chinese Spell Checking (abbreviated as SE-CSC), to detect typos in Chinese text and to further provide suggested corrections for improving the correctness and readability of the text. The system architecture of SE-CSC is given in Fig. 2; it consists of three core components: (1) sentence summarization, (2) error detection, and (3) error correction. First, the sentence summarization component extracts keywords from the input sentence to simplify the main content of the sentence. The extracted keywords are then fused with the original sentence to form semantically enriched input for subsequent processing. Then, the typo detection component embeds the input and focuses on locating potential typos in the sentence to provide a clear basis for subsequent corrections. Finally, the typo correction component also embeds the input and replaces the wrong characters detected in the previous stage with correct Chinese characters based on the context and language rules. Notice that SE-CSC adopts a pretrained BERT-based model [30], i.e., ChineseBERT, to encode input tokens in sequences. In order to improve the accuracy of the correction, we introduced an error-guided mask mechanism to adjust the output weights of the embedding model and to effectively find the precise correction.

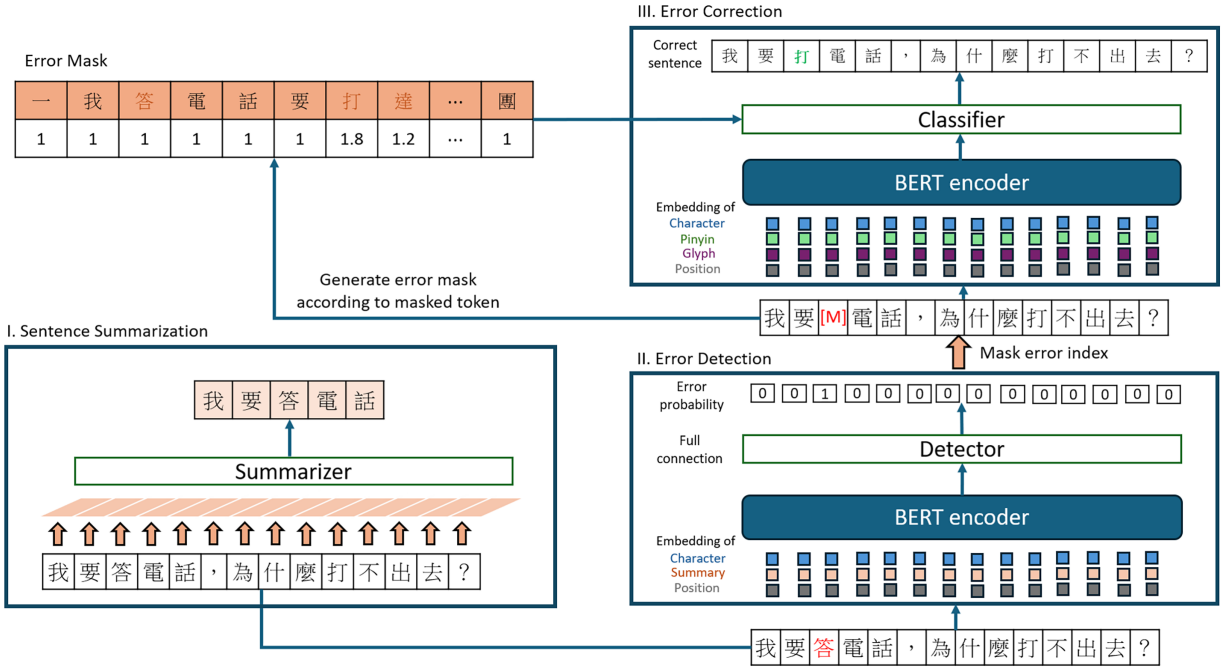


Figure 2: System architecture of the proposed SE-CSC framework.

3.1 Sentence Summarization

The first stage of SE-CSC is sentence summarization, which extracts significant keywords and enhances the ability to identify and correct sentence errors. Without any doubt, pinyin and glyph features are critical for CSC because many Chinese spelling errors are caused by homophones or visually similar characters. However, these features alone cannot always determine the correct answer. For example, several candidate characters may have similar pronunciation: “在/再,” “的/得/地,” “做/作,” and “已/以.” In these cases,

summarization provides semantic-level evidence. It helps the model decide which candidate best fits the overall meaning of the sentence. Additionally, summarization plays an important role when the sentence has a clear semantic focus. If the input sentence contains a coherent topic or intention, the summary can provide useful guidance for CSC. Several prior studies focused on text summarization; however, most of them mainly discussed applications in English. In this study, we adopted and fine-tuned the multilingual T5 model (mT5) [39], which includes Traditional Chinese. The LCSTS [40] dataset was then utilized to train the mT5 model for fine-tuning. The goal of this stage was to facilitate the learning process of error detection and correction with summary information. The SE-CSC model could focus on keywords that have a greater impact on sentence meaning and reduce the probability of misjudgment of typos that do not affect the meaning.

For training mT5, the input is the original sentence containing typos, and the output is a summary of the sentence. During learning, we specifically selected short texts with a length of no more than 100 characters for model training for effective keyword concentration. The objective function is defined as Eq. (1):

$$L_{sum} = -\sum_{t=1}^{T} \log p\left(y_t | x, y_{<t}\right) \tag{1}$$

where T is the length of the output summary sequence, x is the input sequence, and $y = \langle y_1, y_2, \ldots, y_T \rangle$ is the ground-truth summary sequence. The term $p\left(y_t | x, y_{<t}\right)$ represents the predicted probability of the token y_t at position t, given the input x and all previously generated tokens $y_{<t}$.

3.2 Error Detection

The second stage is error detection. In the detection component, we adopt BERT [3] as the backbone model, as shown in the system framework in Fig. 3. The detection component converts both the input sentence and summary into word vector representations, and determines which tokens may be errors. For the training model, given a set of input sentences $\mathbb{X}$ and their corresponding summary $\mathbb{S}$, a sentence $X = \langle x_1, \ldots, x_i, \ldots, x_n \rangle \in \mathbb{X}$ and its summary $S = \langle s_1, \ldots, s_i, \ldots, s_m \rangle \in \mathbb{S}$ where x_i and s_i respectively represent the i-th token of the input sentence and summary, we first convert and embed both texts into embedding token sequences as Eq. (2):

$$E_X = f_{Emb}\left(Tokenizer\left(X\right)\right) \in \mathbb{R}^{n \times d}, \quad E_S = f_{Emb}\left(Tokenizer\left(S\right)\right) \in \mathbb{R}^{m \times d} \tag{2}$$

where d represents the word embedding dimension, and n and m are the numbers of sentence and summary tokens, respectively.

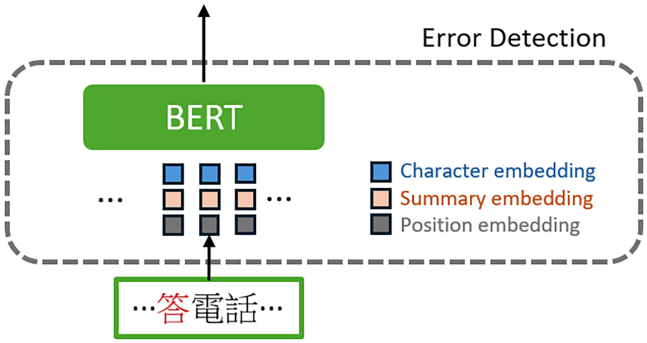


Figure 3: Framework of the error detection component.

To integrate input and summary information, we use the multi-head attention [41] mechanism to combine two sources properly. The Query (Q), Key (K), and Value (V) of each input are defined as follows in Eq. (3):

$$Q = W_Q E_S, \quad K = W_K E_X, \quad V = W_V E_X \tag{3}$$

where W_Q, W_K, W_V are learnable weight matrices. Notice that Q is derived from summary embeddings and is expressed as "query of overall summary information," while K and V are derived from sentence embeddings and are used to calculate the correlation with summary embeddings. Then, we utilize a scaled dot-product operation to compute the attention based on the Q, K and V matrices as Eq. (4):

$$Attention\left(Q, K, V\right) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

For the j-th attention head, the Q, K and V projections are computed as Eq. (5):

$$head_j = Attention\left(QW_j^Q, KW_j^K, VW_j^V\right) \tag{5}$$

Then, the outputs from all attention heads are concatenated and projected using an output weight matrix W^O to form the final multi-head attention output as Eq. (6):

$$H^d = MultiHead\left(Q, K, V\right) = Concat\left(head_1, \ldots, head_j, \ldots, head_h\right)W^O \tag{6}$$

where h is the number of attention heads.

Obviously, the error detection model could leverage both the contextual representation from the original sentence and the incorporated summary information to enhance robustness and mitigate the impact of typos in less frequent words. Finally, we use a classification layer, consisting of a fully connected network, to predict whether each word in the input sequence is a typo or not. This classification layer is utilized solely for computing detection loss, enabling the model to learn to identify erroneous words. When inferencing, the model will use the detection contextual hidden states for the subsequent correction output processing. Note that a dropout layer is employed to prevent overfitting. Hence, we enhance the model's ability to detect error tokens by Layer Normalization to stabilize training, GELU activation for smooth non-linearity, and Dropout regularization to mitigate overfitting, as shown in Eq. (7):

$$Y = \left(Dropout\left(GELU\left(LayerNorm\left(W_1H^d + b_1\right)\right)\right)W_2 + b_2\right) \tag{7}$$

Given an output prediction sequence $Y = \langle y_1, \ldots, y_i, \ldots, y_n \rangle$ and ground-truth sequence $\hat{Y} = \langle \hat{y}_1, \ldots, \hat{y}_i, \ldots, \hat{y}_n \rangle$, the training loss function is defined as Eq. (8):

$$L_{det} = -\sum_{i=1}^{n}\left[\hat{y}_i \log\left(y_i\right) + \left(1 - \hat{y}_i\right)\log\left(1 - y_i\right)\right] \tag{8}$$

where n is the total number of tokens in the input sequence. $\hat{y}_i$ is the ground-truth label for the i-th token, where $\hat{y}_i = 1$ indicates a typo and $\hat{y}_i = 0$ represents a correct word. y_i is the prediction probability that the i-th token belongs to the ground-truth label.

3.3 Error Correction

In the error correction stage, as shown in Fig. 4, we also utilize BERT [3] as the backbone model. Since Chinese spelling errors may be caused by homophones (e.g., "在" and "再") or similar characters (e.g., "問"

and "間") in several real-world applications, SE-CSC includes pinyin embedding and glyph embedding when correcting typos. This strategy could enhance the recognition ability when dealing with similar characters and pronunciations.

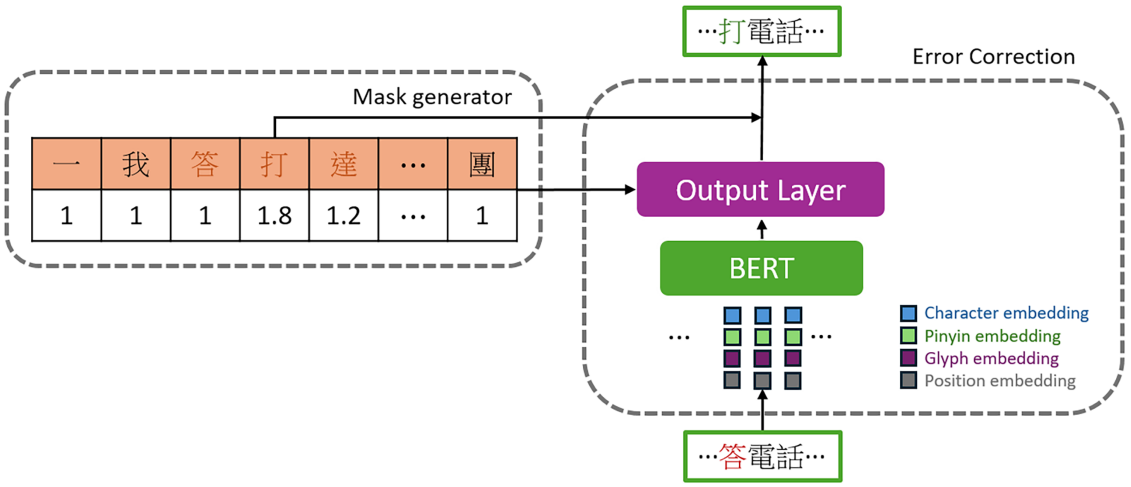


Figure 4: Framework of the error correction component.

Given a sentence $X = \langle x_1, \ldots, x_i, \ldots, x_n \rangle \in \mathbb{X}$, each token x_i is transformed into three distinct feature representations, word embedding, glyph embedding, and pinyin embedding, to capture different aspects of character expression. For word embedding, each token x_i is mapped to a word embedding vector e_i^w which encodes the semantic relationships between characters in the language model. Then, to incorporate character shape information, we implement a glyph embedding layer that projects the visual structure of each character into a vector e_i^g. Finally, for pinyin embedding, since many spelling errors come from phonetic similarities, we encode the pinyin representation of each token by a pinyin embedding layer. The extracted pinyin features are processed through a 1D convolutional neural network and max pooling to capture the most relevant phonetic characteristics. Three embeddings are defined in Eq. (9) where d_w, d_g, and d_p are corresponding embedding dimensions:

$$e_i^w = E_w(x_i) \in \mathbb{R}^{d_w}, \quad e_i^g = E_g(x_i) \in \mathbb{R}^{d_g}, \quad e_i^p = \max\left(CNN\left(E_p(x_i)\right)\right) \in \mathbb{R}^{d_p} \tag{9}$$

Next, we concatenate and project three feature representations into BERT's hidden layer with linear transformation as Eq. (10):

$$h_i^c = W_f E[e_i^w; e_i^g; e_i^p] + b_f \tag{10}$$

where $W_f \in \mathbb{R}^{d_h \times (d_w + d_g + d_p)}$ is the transformation matrix. Note that d_h represents BERT's hidden layer dimension, and b_f is the bias term. This transformation ensures that the combined embeddings are aligned with BERT's input requirements. Finally, we feed the derived representations into the BERT encoder for further contextual learning. To incorporate the information provided by the detection network, we add the hidden embedding states from both the detection and correction networks. The combined representation can be expressed as follows:

$$h_i^f = h_i^c + h_i^d \tag{11}$$

Then, we project the fused vector H^f onto the vocabulary space to predict the most probable word for each position. Assuming the vocabulary size is $|\mathbb{V}|$, the output of this layer is a logit vector z_i of length $|\mathbb{V}|$ for each token, as shown in Eq. (12):

$$z_i = W \cdot h_i^f + b \tag{12}$$

where W is a learned weight matrix.

To further improve the ability of model learning, we adopted an error-guided mask mechanism to increase the correction accuracy. We first derive a prior bias based on error–correction statistics in training data. For each erroneous character x_j and its possible corrections c_k, we compute the conditional probability $P\left(c_k|x_j\right)$. The logits are adjusted by a bias term defined as Eq. (13):

$$B_{j,k} = \begin{cases} \lambda \cdot \log\left(P\left(c_k|x_j\right) + \epsilon\right), & if\ x_j \in \mathbb{V}_{error} \\ 0, & othereise \end{cases} \tag{13}$$

where λ is a tunable hyperparameter and ϵ is a small smoothing constant to prevent numerical instability.

The probability of error can be expressed as follows:

$$P\left(c_k|x_j\right) = \frac{N_e\left(c_k\right)}{T_e} \tag{14}$$

where $N_e\left(c_k\right)$ is the number of times that error word x_j will be corrected to the correct word c_k, and T_e is the number of times that error word x_j appears in the data.

The final logits are computed as $\tilde{z}_{j,k} = z_{j,k} + B_{j,k}$, and the corrected character is predicted by applying softmax, as shown in Eq. (15):

$$y_j = softmax(\tilde{z}_j) \tag{15}$$

Hence, the objective function for the training error correction model is defined as Eq. (16):

$$L_{cor} = -\sum_{i=1}^{n} \hat{y}_i log p_c(y_i|X) \tag{16}$$

where X represents the input sequence and n is the number of tokens in X. $\hat{y}_i$ is the ground-truth label for the i-th token, and $p_c(y_i|X)$ is the predicted probability of the word c.

4 Experiments

In this section, we assess the performance of the proposed SE-CSC on CSC task with several distinct datasets as evaluation benchmarks. The SIGHAN13 [42] dataset is introduced as part of the SIGHAN 2013 Bake-off shared task on CSC. It consists of sentences collected from essays written by middle school students. A total of 700 compositions were selected for the training set, while 1000 were chosen for the test set. The SIGHAN14 [43] dataset originates from the essay portion of the Chinese Proficiency Test conducted in Taiwan. It aims to promote the development of more advanced techniques for detecting and correcting spelling errors in Chinese. The dataset comprises manually annotated essays to identify spelling errors including the corresponding correction. Among these, 1301 essays were selected to form the training set, which contains a total of 5284 spelling mistakes. The test set includes 1062 paragraphs and features 792 annotated spelling errors. The SIGHAN15 [44] dataset was also derived from the essay section of the Chinese Proficiency Test conducted in Taiwan. It was manually annotated by trained native Chinese

speakers, with each spelling error accompanied by the correct form. The training set comprises 970 selected essays, containing a total of 3143 spelling mistakes. The test set includes 1100 paragraphs, half of which are error-free, while the other half contains at least one spelling error per paragraph. The Wang271K [45] corpus was automatically constructed to address the scarcity of annotated data in CSC tasks. By simulating realistic spelling errors through two complementary methods, Optical Character Recognition (OCR)-based and Automatic Speech Recognition (ASR)-based techniques, the authors generated a large-scale dataset containing 271,329 sentences with a total of 381,962 annotated errors. The real-world datasets are listed and summarized in Table 1.

Table 1: A summary of four real-world datasets [42–45].

Dataset	# of Train	# of Test
SIGHAN13	700	1000
SIGHAN14	1301	1062
SIGHAN15	970	1100
Wang271K	271,329	

To assess the performance of the proposed SE-CSC with baseline approaches, we employed Recall, Precision, and F1-score as evaluation metrics. All metrics were derived from the confusion matrix to help determine the correctness of the predictions. The confusion matrix includes four components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). TP is the number of instances where the model correctly predicts a positive case. TN happens when the model correctly identifies a negative case. FP refers to a case where the model predicts a positive outcome, but the actual label is negative. FP is also called "Type I error." Finally, FN describes a situation where the model predicts a negative outcome, while the actual label is positive. FN is also known as a "Type II error."

The Recall metric calculates the proportion of correctly identified positive instances among all instances which are truly positive, as defined in Eq. (17):

$$Recall = \frac{TP}{(TP + FN)} \tag{17}$$

The Precision metric calculates the proportion of correctly identified positive instances among all instances predicted as positive, as expressed in Eq. (18):

$$Precision = \frac{TP}{(TP + FP)} \tag{18}$$

The F1-score metric simultaneously considers Precision and Recall, aiming to maximize both while minimizing their difference for a balanced performance measure, as defined in Eq. (19):

$$F1\,score = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{19}$$

Without any doubt, the evaluation of correctness involves two aspects: error detection and error correction. For detection to be considered successful, the identified positions of all incorrect characters in a paragraph must match the ground truth exactly. Similarly, successful correction requires that both the

locations of all incorrect characters and their proposed corrections precisely align with the ground truth. As a result, we utilized precision, recall, and the F1-score as standard metrics for evaluation of CSC tasks.

4.1 Baseline Models

In this study, to demonstrate the effectiveness of the SE-CSC framework, several state-of-the-art models were implemented as baselines for performance comparison, including SpellGCN [7], PLOME [9], FASPell [11], MDCSpell [14], MCRSpell [21] and LSTM-TRM [22]. Brief descriptions of each model are provided as follows:

- SpellGCN [7] integrates knowledge about how characters sound and look similar directly into language models. SpellGCN uses a specialized graph convolutional network to create a graph based on characters, and learns interconnected character classifiers.
- PLOME [9] adopts a transformer architecture by replacing selected tokens with similar characters from a confusion set instead of using the standard "[MASK]" token. In addition, a GRU backbone is utilized to learn the misspelling knowledge for phonetics and stroke patterns of characters, including the pronunciation related to how words sound.
- FASPell [11] utilizes a denoising autoencoder (DAE) to train an adaptable model for different types of Chinese text, which could keep the system simple while still being efficient in terms of fixing errors. The decoder of DAE could eliminate the use of a confusion set to increase the flexibility and sufficiency of utilizing the salient feature of Chinese character similarity.
- MDCSpell [14] utilizes a BERT-based corrector to capture the visual and phonological features from each character in the input sentence. Furthermore, MDCSpell introduces a late fusion strategy to combine the internal representations to reduce the negative impact of any misspelled characters.
- MCRSpell [21] learns semantic knowledge adaptively from multiple intermediate features to construct separate models for the spelling error correction task without any data augmentation.
- LSTM-TRM [22] uses a global attention mechanism to capture token dependencies and local context within the sequence for CSC tasks.

Furthermore, two masking-based models, ECSpell [10] and Soft-Masked BERT [13], were also implemented for performance comparison:

- ECSpell [10] addresses data scarcity in CSC by using an error-consistent masking strategy to generate realistic training data. With the proposed user dictionary structure, the enhanced ECSpellUD model could improve domain adaptation and achieve strong performance across both general and domain-specific benchmarks.
- Soft-masked BERT [13] overcomes the limitation of consistent error identification with error detection and correction networks based on BERT. Both networks use a soft-masking strategy to ensure flexible detection and to correct typos in CSC tasks.

All models for experiments were implemented using the PyTorch framework in Python and were conducted on a workstation equipped with an Intel i9-14900KF CPU, 128 GB RAM, and two NVIDIA RTX 5090 GPUs, running on Ubuntu 22.04.4 LTS. Note that all baseline models were implemented with their original hyperparameter settings to ensure correctness and fairness.

4.2 Performance Comparison

We evaluated the effectiveness of our proposed SE-CSC framework by comparing the performance to the state-of-the-art baselines with three evaluation metrics, precision rate (P), recall rate (R), and F1-score (F1) on three real benchmark datasets. To train the summarization mT5 model, the input is the original

sentence containing typos, and the output is a summary of the sentence. During learning, we specifically selected short texts with a length of no more than 100 characters for effective keyword concentration. The batch size, learning rate, and training epochs are set as 32, 3e−5, and 5, respectively. The training process of SE-CSC consists of two stages: pre-training and fine-tuning. In the pre-training stage, we utilized the Wang271k dataset to initialize the model parameters with a batch size of 32 and a learning rate of 2e−5. Subsequently, in the fine-tuning stage, we trained on all training data from the SIGHAN13, SIGHAN14, and SIGHAN15 benchmark datasets. The fine-tuning stage also employed a batch size of 32, but with a reduced learning rate of 1e−5 to ensure more stable convergence and better generalization. To ensure fairness, all baseline results were generated by reproducing the original code, and all training procedures followed the descriptions provided in the respective studies [7,9,11,14,21,22]. For baselines where the original pre-training data were unavailable, we used the Wang271k dataset to ensure consistency with our experimental setup.

The comparison of error detection performance with different baseline models is shown in Table 2. The experimental results indicate that SE-CSC consistently outperformed the state-of-the-art models on the SIGHAN [42–44] benchmarks, which demonstrates the successful integration of summarization for robustness and generalization capability. Among all baselines, PLOME [9] could not perform as well as our proposed SE-CSC model. This is mainly due to its single architecture with a simple Transformer encoder. To achieve better performance, systems based on PLOME may require a substantially larger amount of training data. Although FASpell [11] incorporates phonological and visual information, it still struggles to capture long-range dependencies and complex semantic relationships, which limits its ability to detect errors in linguistically complex contexts. The reason is partly due to the simple modeling strategy for contextual semantics. SpellGCN [7] connects phonologically and visually similar characters through graph construction; however, the adopted static graph structure may not capture dynamic contextual variations, which limits the generalizability across different scenarios. In addition, the multi-task learning framework of MDCSpell [14] may fail to fully integrate external phonological and visual knowledge, which restricts the overall correction capability. We could find that the transformer-based models also achieved elegant performance. With the multi-augmented information, MCRSpell [21] has good performance with the transformer-based architecture. This also indicates that richer information could directly enhance the prediction precision. Note that LSTM-TRM [22] also achieved a qualified result, indicating the significance of the memory combination with the transformer.

Table 2: The performance comparison of error detection with different models on three SIGHAN [42–44] datasets.

Model	SIGHAN13			SIGHAN14			SIGHAN15		
	P	R	F1	P	R	F1	P	R	F1
FASPell	0.762	0.632	0.691	0.612	0.535	0.571	0.676	0.601	0.636
PLOME	0.448	0.347	0.391	0.624	0.432	0.510	0.760	0.560	0.644
SpellGCN	0.665	0.621	0.642	0.641	0.608	0.624	0.754	0.775	0.764
MDCSpell	0.740	0.656	0.695	0.730	0.608	0.663	0.835	0.739	0.784
MCRSpell	0.718	0.623	0.667	0.683	0.605	0.642	0.811	0.722	0.764
LSTM-TRM	0.747	0.665	0.703	0.721	**0.619**	0.666	0.796	0.765	0.780
SE-CSC	**0.765**	**0.688**	**0.724**	**0.743**	0.609	**0.669**	**0.837**	**0.793**	**0.814**

Note: Bold indicates the best value.

We now discuss the experimental results of error correction. The comparison of error correction performance with different baseline models is shown in Table 3. Clearly, SE-CSC shows significant performance

gains. On the SIGHAN13 dataset, SE-CSC could achieve the best F1-score among all models. This indicates that the proposed framework not only precisely identifies typos, but also effectively corrects erroneous characters at the sentence level. Compared to FASPell, which achieved F1 = 0.662, and MDCSpell, with F1 = 0.676, SE-CSC could demonstrate a clear advantage, particularly in recall, to improve coverage of errors. Additionally, on SIGHAN14 and SIGHAN15, SE-CSC also had elegant performance of both detection and correction, outperforming all other baseline models. We can observe that SpellGCN exhibited relatively high recall but lower precision, while SE-CSC maintained a better balance between precision and recall. In real applications, this balance is crucial in sentence-level evaluation, where even a single false positive or false negative affects the overall metric. In summary, the experimental results on several real datasets demonstrate the practicability of the summarization integration for CSC tasks. The proposed SE-CSC framework successfully showed excellent performance, robustness, and generalization capability.

Table 3: The performance comparison of error correction with different models on three SIGHAN [42–44] datasets.

Model	SIGHAN13			SIGHAN14			SIGHAN15		
	P	R	F1	P	R	F1	P	R	F1
FASPell	0.731	0.605	0.662	0.594	0.520	0.554	0.666	0.591	0.626
PLOME	0.350	0.271	0.305	0.547	0.378	0.447	0.661	0.487	0.561
SpellGCN	0.654	0.611	0.632	0.627	0.641	0.634	0.738	0.761	0.749
MDCSpell	0.719	0.637	0.676	0.708	0.590	0.644	0.816	0.722	0.766
MCRSpell	0.700	0.642	0.669	0.693	0.586	0.635	0.772	0.735	0.753
LSTM-TRM	0.731	0.650	0.688	0.709	**0.665**	**0.686**	0.815	0.763	0.788
SE-CSC	**0.740**	**0.675**	**0.706**	**0.712**	0.659	0.684	**0.819**	**0.771**	**0.794**

Note: Bold indicates the best value.

4.3 Effectiveness of Summarization Inclusion in SE-CSC

In this section, to evaluate the impact of summary embedding on the error detection and correction performance, we compare SE-CSC models with or without a summarization component. The summary embedding was generated from a pre-trained mT5 summarization model, and was incorporated into the detection network via a cross-attention mechanism, as shown in Eqs. (3) and (4). By introducing condensed semantic information, summary information could help the model focus on the most significant parts of the sentence. We discuss the effectiveness of the model on the SIGHAN15 dataset in terms of precision, recall, and F1-score metrics, with the results presented in Table 4.

Table 4: The effectiveness of summarization for error detection and correction on the SIGHAN15 [44] dataset.

Model	Error Detection			Error Correction		
	P	R	F1	P	R	F1
SE-CSC w/o summarization	0.829 (↓0.9%)	0.724 (↓8.7%)	0.773 (↓5.1%)	0.811 (↓0.9%)	0.709 (↓8.0%)	0.756 (↓4.7%)
SE-CSC	**0.837**	**0.793**	**0.814**	**0.819**	**0.771**	**0.794**

Note: Bold indicates the best value.

When the summary embeddings were removed from the detection network, the model performance was downgraded by 0.9% for precision, 8.7% for recall, and 5.1% for F1-score. Meanwhile, in the correction

task, precision, recall, and F1-score decreased by 0.9%, 8.0%, and 4.7%, respectively. These results demonstrate that summary embeddings effectively help the model to capture key semantic content and further enhance the ability of error detection and the subsequent CSC correction task. To actually illustrate the effect, we provide qualitative analysis and attention visualizations using case studies in the following section, which demonstrate how summary information enhances the model's sensitivity to potential errors.

4.4 Influence of Mask Generation in SE-CSC

The mask generation and integration are also important in the SE-CSC framework. To show the influence of masking for CSC, we compare SE-CSC with two prior models, ECSpell [10] and Soft-masked BERT [13] equipped with the mask component. We discuss the influence of the model on the SIGHAN15 dataset with precision, recall, and F1-score metrics, with the results presented in Table 5. SE-CSC achieved strong performance with F1 = 0.814 for detection and F1 = 0.794 for correction, which outperformed all baselines. From Table 5, we find that the SE-CSC framework surpasses two of the prior mask-based models and shows a more balanced precision-recall tradeoff. These consistent results indicate that SE-CSC is not overfitted to a specific dataset and can effectively generalize to diverse types of error distributions. Moreover, the narrow gap between detection and correction F1-scores implies that SE-CSC not only locates errors accurately but also provides high-quality corrections.

Table 5: The performance of error detection and correction with different masked-based models on the SIGHAN15 [44] dataset.

Model	Error Detection			Error Correction		
	P	R	F1	P	R	F1
ECSpell	0.764 (↓8.7%)	0.789 (↓0.5%)	0.781 (↓4.0%)	0.744 (↓9.1%)	0.768 (↓0.3%)	0.761 (↓4.1%)
Soft-masked BERT	0.737 (↓11.9%)	0.732 (↓7.6%)	0.735 (↓9.7%)	0.667 (↓18.5%)	0.662 (↓14.1%)	0.664 (↓16.3%)
SE-CSC	**0.837**	**0.793**	**0.814**	**0.819**	**0.771**	**0.794**

Note: Bold indicates the best value.

We observe that Soft-Masked BERT may struggle with over-correction due to limited semantic modeling. Since Soft-masked BERT relies primarily on token-level representations from BERT, it may lack higher-level semantic understanding and perform poorly when handling errors that require deeper sentence-level comprehension. The error-consistent masking of ECSpell requires static error patterns, early-stage fusion, and rule-based post-processing, which directly hampers the adaptability to real-world datasets and unseen domain terms.

4.5 Error-Guided Mask Integration

In this section, we discuss the weight setting of the λ coefficient for integrating an error-guided mask into the SE-CSC framework to achieve optimal performance. We adopted the SIGHAN15 dataset for the experiments with different λ coefficient settings. As for the definitions in Eqs. (13) and (14), the weight parameter λ balances the contribution of prior statistical information against model prediction. Different $\lambda = \{0, 0.25, 0.5, 0.75, 1\}$ settings were adopted for the experiments. As presented in Table 6, the coefficient $\lambda = 0.75$ could yield the best performance. This is explainable, clearly, since weight λ is utilized to control the significance of model prediction. The large λ setting will impair the model generalization (e.g., $\lambda = 1$) while the small λ setting (e.g., $\lambda = 0$ to $\lambda = 0.5$) will restrict the model ability. It was shown that $\lambda = 0.75$ is the most coordinated setting to achieve the optimized model performance. This experiment significantly depicts the optimal intensity coefficient setting when constructing the proposed SE-CSC model.

Table 6: The performance of error detection and correction with different weight parameter λ settings in SE-CSC on the SIGHAN15 [44] dataset.

λ Coefficient	Error Detection			Error Correction		
	P	R	F1	P	R	F1
$\lambda = 0$	0.836	0.752	0.792	0.808	0.727	0.765
$\lambda = 0.25$	0.836	0.764	0.799	0.809	0.742	0.774
$\lambda = 0.5$	0.837	0.775	0.805	0.810	0.756	0.782
$\lambda = 0.75$	**0.837**	**0.793**	**0.814**	**0.819**	**0.771**	**0.794**
$\lambda = 1$	0.837	0.787	0.811	0.810	0.767	0.788

Note: Bold indicates the best value.

4.6 Ablation Study

To evaluate the contribution of each component in the SE-CSC framework, we conducted a series of ablation tests by systematically removing specific modules and observing the impact on performance. The ablation study was performed on the SIGHAN15 dataset with precision, recall, and F1-score for each variant SE-CSC model as follows:

- SE-CSC$^{-}_{summary}$: SE-CSC framework without summarization network.
- SE-CSC$^{-}_{detect}$: SE-CSC framework without entire detection network.
- SE-CSC$^{-}_{Pinyin}$: SE-CSC framework without Pinyin embedding in the correction network.
- SE-CSC$^{-}_{glyph}$: SE-CSC framework without glyph embedding in the correction network.
- SE-CSC$^{-}_{mask}$: SE-CSC framework without the error-guided mask in the correction network

Notice that all variant models for the experiments were implemented in the same environment as SE-CSC. The experimental results are given in Table 7.

Table 7: The performance of error detection and correction by excluding each component of the SE-CSC framework on the SIGHAN15 [44] dataset.

Model	Error Detection			Error Correction		
	P	R	F1	P	R	F1
SE-CSC$^{-}_{summary}$	0.829 (↓1%)	0.724 (↓8.7%)	0.773 (↓5.0%)	0.811 (↓1%)	0.709 (↓8.0%)	0.756 (↓4.8%)
SE-CSC$^{-}_{detect}$	0.826 (↓1.3%)	0.732 (↓7.7%)	0.776 (↓4.7%)	0.804 (↓1.8%)	0.712 (↓7.7%)	0.755 (↓4.9%)
SE-CSC$^{-}_{Pinyin}$	0.745 (↓11%)	0.703 (↓11.3%)	0.723 (↓11.2%)	0.671 (↓18.0%)	0.630 (↓18.3%)	0.649 (↓18.3%)
SE-CSC$^{-}_{glyph}$	0.813 (↓2.9%)	0.765 (↓3.5%)	0.788 (↓3.2%)	0.740 (↓9.6%)	0.665 (↓13.7%)	0.701 (↓11.7%)
SE-CSC$^{-}_{mask}$	0.836 (↓0.1%)	0.752 (↓5.2%)	0.792 (↓2.7%)	0.808 (↓1.3%)	0.727 (↓5.7%)	0.765 (↓3.7%)
SE-CSC	**0.837**	**0.793**	**0.814**	**0.819**	**0.771**	**0.794**

Note: Bold indicates the best value.

The experimental results indicate that both summarization and detection networks play a critical role in error correction performance, especially when focusing on the recall rate of prediction. As shown in Table 7, without summary information, the detection model performance was downgraded by 1% for precision, 8.7% for recall, and 5% for F1-score. For the correction task, the precision, recall, and F1-score decreased by 1%, 8%, and 4.8%, respectively. Furthermore, by excluding the detection network, the detection performance dropped by 1.3%, 7.7%, and 4.7% for precision, recall, and F1-score, respectively. For the correction performance, the model was also downgraded by 1.8% for precision, 7.7% for recall, and 4.9% for F1-score. These

findings confirm the effectiveness of the proposed tri-stage summarization-detection-correction system architecture. The summarization and detection stages directly affect the overall performance.

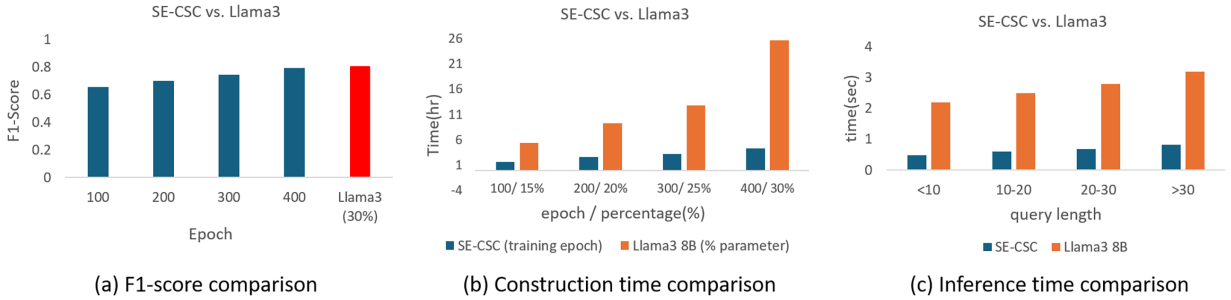
Additionally, we could observe that Pinyin embedding is the most important component of the SE-CSC framework. When pinyin embedding is removed, the detection task directly collapses by 11%, 11.3%, and 11.2% for precision, recall, and F1-score performance, respectively. For the correction task, the precision, recall, and F1-score metrics also decreased by 18%, 18.3%, and 18.3%, respectively. This substantial performance degradation could highlight the essential importance of phonetic information in identifying homophone-related errors and maintaining correction accuracy. The significant impact on both error detection and correction demonstrates the heavy reliance on pinyin features to resolve phonological ambiguities for the CSC task. We now discuss the contribution of glyph embedding in SE-CSC. As shown in Table 7, without glyph embedding, the performance of the SE-CSC detection task is downgraded by about 3%. The detection precision, recall, and F1-score decrease by 2,9%, 3.5%, and 3.2%, respectively. The correction precision, recall, and F1-score dramatically declined by 9.6%, 13.7%, and 11.7%, respectively. Although the influence is less pronounced compared to pinyin exclusion, the consistent drops across all evaluation metrics still indicate that glyph features provide valuable visual cues for error identification, particularly in distinguishing visually similar characters. The experimental results also confirm that phonetic information is the most critical among the auxiliary features, while glyph information offers meaningful support in enhancing model robustness.

Finally, we focus on the significance of the error-guided mask. As shown in Table 7, when the error-guided mask was removed from the correction network, the detection performance decreased by 0.1% in precision, 5.2% in recall, and 2.7% in F1-score. For the correction task, precision, recall, and F1-score dropped by 1.3%, 5.7%, and 3.7%, respectively, for the SE-CSC framework. While this performance decline might appear minor, the consistent impact across all key metrics, both for detection and correction, still strongly supports the critical role of the error-guided mask in guiding the CSC tasks. This positive contribution highlights the effectiveness of focusing attention on high-risk positions, ultimately leading to more accurate model predictions.

4.7 LLM-Based Model Comparison

In this section, we discuss the advantage of the proposed SE-CSC model compared to the LLM-based models. As mentioned previously, the stand-alone CSC model is usually smaller and more lightweight than conventional LLM models. The proposed SE-CSC requires less training for model construction and inference time for user queries. Several experiments were conducted with the SIGHAN15 dataset to show the performance and computation costs of the SE-CSC and LLM models. We adopted Llama3 8B [46] as the baseline LLM model for comparison due to its superiority and multi-language support. The Llama3 8B model is fine-tuned with different percentages (i.e., from 15% to 30%) of model parameters. As shown in Fig. 5a, the proposed SE-CSC could achieve almost the same performance as the Llama3 8B (30% parameter fine-tuning) on 400 training epochs. This also demonstrates the successful contribution of summarization inclusion in the SE-CSC model. However, the computation cost was extremely different for the two models. As shown in Fig. 5b, the model fine-tuning time of Llama3 8B was about 5 times greater on different percentages of parameter tuning. The model inference time was also different. We can observe that SE-CSC achieved more real-time responses for correcting user queries with different lengths. This is very important for real-world applications. Clearly, the CSC module is usually applied as a component for input preprocessing. Without any doubt, for one system, the time requirement for input checking directly affects its practicability. The experimental results also indicate the applicability of the proposed SE-CSC model, which is more suitable, compared to LLM-based models, for real-time applications such as customer service chatbots,

frequently asked questions (FAQ) matching systems, or enterprise search systems where fast responses are necessarily required.



(a) F1-score comparison (b) Construction time comparison (c) Inference time comparison

Figure 5: The performance and computation time of the SE-CSC and Llama3 8B [46] models.

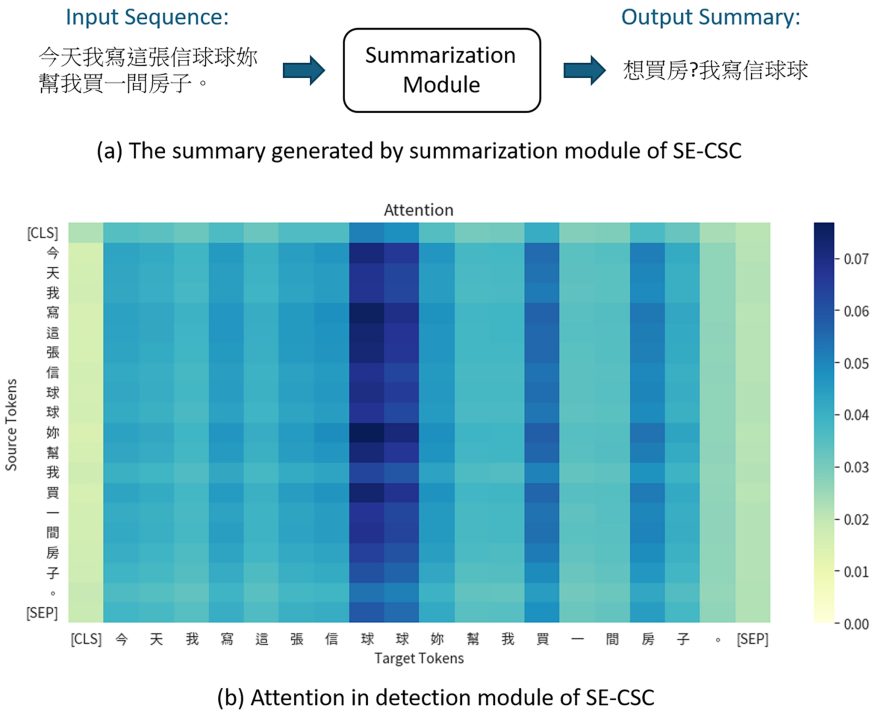
4.8 Case Study

First, a case study is given to observe the attention weights after incorporating summary information. We utilize an attention map to show that the SE-CSC model could focus on tokens highlighted by the summary, including the erroneous segments. The case study could indicate the enhanced sensitivity toward potential errors by the proposed summarization inclusion concept. We use a real sentence in the SIGHAN13 [42] dataset as an example. As shown in Fig. 6a, the original input sentence is "今天我寫這張信球球妳幫我買一間房子," which is summarized as "想買房?我寫信球球" from our proposed SE-CSC. Clearly, the summary effectively extracts key semantic elements "買" and "房," while excluding less important elements like "今天" and "一間." Then, as illustrated in Fig. 6b, the attention of the detection component of SE-CSC precisely focuses on key lexical positions highlighted by the summary. Particularly, these positions also contain potential errors in the input sentence. In this context, retaining the phrase "買房" is sufficient for the model to recognize the typo "球球," which is semantically unrelated to the core meaning of the sentence, and thus identifies it as an incorrect word. The SE-CSC model leverages glyph and phonetic similarities to correct the typo "球球" to the intended word "求求," which is not only visually and phonetically similar but also contextually relevant to "買房." The case study also indicates the functionality of summary information for leading SE-CSC to concentrate on the most semantically significant parts of the input and further improves the model's ability to identify potential typos. By providing a high-level overview of sentence meaning, summaries help the SE-CSC framework disambiguate intent and highlight linguistically important segments, which in turn strengthens the model's sensitivity to errors.

Then, to show its practicability, we use a case study to analyze typo correction on real datasets. We present the quality comparison with the powerful model, ECSpell [10], which has the best performance among prior state-of-the-art models, and the traditional model, BERT [3], which performs efficiently in real-time scenarios, so as to illustrate the effectiveness of the proposed SE-CSC. As shown in Table 8, we analyzed both phonologically and visually similar spelling errors and demonstrated how the model correctly identified and rectified these challenging cases on the SIGHAN13 dataset.

In the first example, we use a visually similar case where the character "悔" is miswritten as a typo "侮." While the ECSpell incorrectly replaces it with "辜", a common but contextually inappropriate character, the proposed SE-CSC successfully corrects "侮" to "悔," which preserves both the semantic integrity and the grammatical correctness of the sentence. This comparison highlights the tendency of ECSpell to overcorrect, where the model favors frequent characters even when they are not contextually suitable. The second example further emphasizes this issue. The ground-truth sentence contains the word "虔心," but the incorrect word "潛心" is mistakenly used. The ECSpell baseline replaces the error character with "辛心," another plausible

but incorrect substitution, reflecting an inability to distinguish subtle phonetic differences. However, SE-CSC could precisely correct the phrase with the correct word, demonstrating its ability to leverage contextual and phonological information effectively.



(a) The summary generated by summarization module of SE-CSC

(b) Attention in detection module of SE-CSC

Figure 6: An example of the summary and attention generated by SE-CSC for real sequences in the SIGHAN13 [42] dataset.

Table 8: The correction example of SE-CSC and ECSpell [10] models for glyph- and phonetic-similar errors on the SIGHAN13 [42] dataset.

Input: Sentence with glyph-similar error	**Model**	**Output:** Corrected Sentence
“我將無侮, 因為就算我地決定有錯誤.”	ECSpell	“我將無辜, 因為就算我的決定有錯誤.”
	SE-CSC	“我將無悔, 因為就算我的決定有錯誤.”
Input: Sentence with phonetic-similar error	**Model**	**Output:** Corrected Sentence
“一陣滂沱大雨, 把大家潛心栽培來的成果吹落後.”	ECSpell	“一陣滂沱大雨, 把大家辛心栽培來的成果吹落後.”
	SE-CSC	“一陣滂沱大雨, 把大家虔心栽培來的成果吹落後.”

Furthermore, we conducted the same experiment with the commonly adopted BERT [3] model to show the typo correction for the CSC task, as the results presented in Table 9 indicate. We used a case where the character “待” is mistakenly written as “侍” due to their visual similarity. The BERT model corrected “侍” to “法,” a character that is common but contextually incorrect. In contrast, the proposed SE-CSC

accurately restored the original character to maintain both the meaning and grammatical structure of the sentence. In the second case, the correct phrase "成就" was miswritten as "成究," and the BERT model incorrectly changed it to "成績." This word may appear plausible but is phonetically and semantically off. However, SE-CSC correctly identified and restored the intended phrase, showing its strength in capturing both context and subtle phonetic distinctions. All the aforementioned examples confirm that the SE-CSC model is not only capable of detecting errors, but also effectively avoids overly aggressive corrections to degrade sentence quality.

Table 9: The correction example of SE-CSC and BERT [3] models for glyph- and phonetic-similar errors on the SIGHAN13 [42] dataset.

Input: Sentence with glyph-similar error	**Model**	**Output:** Corrected Sentence
"只是每個人如何去看侍的."	BERT	"只是每個人如何去看法的."
	SE-CSC	"只是每個人如何去看待的."
Input: Sentence with phonetic-similar error	**Model**	**Output:** Corrected Sentence
"也得到比風雨前的他們幾十倍得成究."	BERT	"也得到比風雨前的他們幾十倍的成績."
	SE-CSC	"也得到比風雨前的他們幾十倍的成就."

Finally, from the case study, we found that pinyin and glyph features are critical for CSC because many Chinese spelling errors are caused by homophones or visually similar characters. However, these features alone cannot always determine the correct answer. For example, several candidate characters may have similar pronunciation: "在/再," "的/得/地," "做/作," and "已/以." In these cases, summarization provides semantic-level evidence. It helps the model decide which candidate best fits the overall meaning of the sentence. As the examples in Tables 8 and 9, the proposed SE-CSC could correct such errors precisely.

5 Conclusion

In this paper, we have proposed a novel tri-stage framework, SE-CSC, which includes summarization, detection, and correction components with phonetic and glyph embeddings to improve both the precision and interpretability of CSC tasks. We introduced a summarization integration strategy to effectively enhance both error detection and correction. This strategy leverages domain-relevant contextual cues through text summarization to better identify semantically important positions, thereby improving the model's ability to locate and correct errors. In addition, an error-guided mask mechanism was included to enable SE-CSC to precisely lead the correction by analyzing error-correction patterns in the training corpus and computing the probability distribution of common corrections. Finally, the proposed SE-CSC was applied to CSC tasks on several real datasets to demonstrate its performance and practicability. A case study was given to discuss the CSC results and to analyze the correction behavior.

Currently, the research limitation of this study is that it was restricted to traditional Chinese. The input only focused on a single language source without considering multiple language domains. For the proposed SE-CSC, summarization may be less helpful for very short sentences, fragmented queries, or sentences without enough semantic context, since there is not enough information to summarize meaningfully. Additionally, as observed from experimental results, the error-guided mask plays an important role in the SE-CSC framework. However, since the mask directly relies on statistical priors estimated from the training set, applications with out-of-distribution or domain-transfer scenarios may not have qualified performance. This is one critical limitation of this study.

Several future research directions have emerged from this study. First, SE-CSC adopts a static error-guided mask that is derived from the distribution of the corpus to enhance error correction. Future research could explore a knowledge graph that adjusts the association for distribution. The proposed framework could have a deeper understanding of word meanings, relations, and contextual constraints. Second, the integration with LLM-based models is also a potential research direction. The proposed SE-CSC framework could tackle the problems of expensive inference time and computation cost, especially for real-time applications. Finally, joint optimization for both detection and correction is also a promising direction to integrate multiple training stages for complexity reduction. In addition, for performance measurement, more real datasets and evaluation metrics could be adopted to analyze and compare the models from different points of view.

Acknowledgement: Not applicable.

Funding Statement: The work of Yi-Cheng Chen was supported by the National Science and Technology Council under Grant 111-2628-H-008-005-MY4, Grant 113-2410-H-008-065-MY3, and Grant 115-2410-H-008-064-MY3.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Yi-Cheng Chen, Wen-Chin Hsu; data collection: Yi-Cheng Chen, Yi-Hsuan Kuo; analysis and interpretation of results: Wen-Chin Hsu, Yi-Cheng Chen; model implementation: Yi-Cheng Chen, Yi-Hsuan Kuo; manuscript preparation: Wen-Chin Hsu, Yi-Cheng Chen. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: SIGHAN13 [42], SIGHAN14 [43], SIGHAN15 [44], Wang271K [45].

Ethics Approval: Not applicable.

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

References

1. Xie W, Huang P, Zhang X, Hong K, Huang Q, Chen B, et al. Chinese spelling check system based on N-gram model. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing; 2015 Jul 30–31; Beijing, China. p. 128–36. doi:10.18653/v1/w15-3120.
2. Zhang R, Pang C, Zhang C, Wang S, He Z, Sun Y, et al. Correcting Chinese spelling errors with phonetic pre-training. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; Online. p. 2250–61.
3. Devlin J, Chang MW, 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; 2019 Jun 2–7; Minneapolis, MN, USA. p. 171–86.
4. Tan M, Chen D, Li Z, Wang P. Spelling error correction with BERT based on character-phonetic. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC); 2020 Dec 11–14; Chengdu, China. p. 1146–50. doi:10.1109/iccc51575.2020.9345276.
5. Xu HD, Li Z, Zhou Q, Li C, Wang Z, Cao Y, et al. Read, listen, and see: leveraging multimodal information helps Chinese spell checking. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; Online. p. 716–28. doi:10.18653/v1/2021.findings-acl.64.
6. Huang L, Li J, Jiang W, Zhang Z, Chen M, Wang S, 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; 2021 Aug 1–6; Online. p. 5958–67. doi:10.18653/v1/2021.acl-long.464.

7. Cheng X, Xu W, Chen K, Jiang S, Wang F, Wang T, et al. 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; 2020 Jul 5–10; Online. p. 871–81. doi:10.18653/v1/2020.acl-main.81.
8. Mao X, Shan Y, Li F, Chen X, Zhang S. CLSpell: contrastive learning with phonological and visual knowledge for Chinese spelling check. Neurocomputing. 2023;554:126468. doi:10.1016/j.neucom.2023.126468.
9. 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; 2021 Aug 1–6; Online. p. 2991–3000. doi:10.18653/v1/2021.acl-long.233.
10. Lv Q, Cao Z, Geng L, Ai C, Yan X, Fu G. General and domain-adaptive Chinese spelling check with error-consistent pretraining. ACM Trans Asian Low-Resour Lang Inf Process. 2023;22(5):1–18. doi:10.1145/3564271.
11. Hong Y, Yu X, He N, Liu N, Liu J. FASPell: a fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm. In: Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019); 2019 Nov 4; Hong Kong, China. p. 160–9. doi:10.18653/v1/d19-5522.
12. He Z, Tang B, Tao B. RMVSpell: Chinese spelling check based on multimodality and image input enhancement. In: 2025 7th International Conference on Natural Language Processing (ICNLP); 2025 Mar 21–23; Guangzhou, China. p. 287–91. doi:10.1109/ICNLP65360.2025.11108631.
13. 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; 2020 Jul 5–10; Online. p. 882–90. doi:10.18653/v1/2020.acl-main.82.
14. Zhu C, Ying Z, Zhang B, Mao F. MDCSpell: a multi-task detector-corrector framework for Chinese spelling correction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022); 2022 May 22–27; Dublin, Ireland. p. 1244–53. doi:10.18653/v1/2022.findings-acl.98.
15. Li J, Wu G, Yin D, Wang H, Wang Y. DCSpell: a detector-corrector framework for Chinese spelling error correction. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021 Jul 11–15; Virtual Event. p. 1870–4. doi:10.1145/3404835.3463050.
16. Li P, Shi S. Tail-to-tail non-autoregressive sequence prediction for Chinese grammatical error correction. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing; 2021 Aug 1–6; Online. p. 4973–84. doi:10.18653/v1/2021.acl-long.385.
17. Guo Z, Ni Y, Wang K, Zhu W, Xie G. Global attention decoder for Chinese spelling error correction. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing; 2021 Aug 1–6; Online. p. 1419–28. doi:10.18653/v1/2021.findings-acl.122.
18. Liu S, Song S, Yue T, Yang T, Cai H, Yu T, et al. CRASpell: a contextual typo robust approach to improve Chinese spelling correction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022); 2022 May 22–27; Dublin, Ireland. p. 3008–18. doi:10.18653/v1/2022.findings-acl.237.
19. Yang S, Yu L. CoSPA: an improved masked language model with copy mechanism for Chinese spelling correction. In: Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence; 2022 Aug 1–5; Eindhoven, The Netherlands. p. 2225–34.
20. Liang Z, Quan X, Wang Q. Disentangled phonetic representation for Chinese spelling correction. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Aug 9–14; Toronto, ON, Canada. p. 13509–21. doi:10.18653/v1/2023.acl-long.755.
21. Li C, Zhang M, Zhang X, Yan Y. MCRSpell: a metric learning of correct representation for Chinese spelling correction. Expert Syst Appl. 2024;237(8):121513. doi:10.1016/j.eswa.2023.121513.
22. Xu M, Liu J, Peng K, Li Z. Chinese spelling correction based on long short-term memory network-enhanced Transformer and dynamic adaptive weighted multi-task learning. Nat Lang Process. 2025;31(5):1265–84. doi:10.1017/nlp.2024.61.

23. Li Y, Huang H, Wang B, Gao Y. DRMSpell: dynamically reweighting multimodality for Chinese spelling correction. Front Inform Technol Electron Eng. 2025;26(3):354–66. doi:10.1631/fitee.2300816.
24. Gou W, Chen Z. Think twice: a post-processing approach for the Chinese spelling error correction. Appl Sci. 2021;11(13):5832. doi:10.3390/app11135832.
25. Wei X, Huang J, Yu H, Liu Q. PTCSpell: pre-trained corrector based on character shape and pinyin for Chinese spelling correction. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics; 2023 Aug 9–14; Toronto, ON, Canada. p. 6330–43. doi:10.18653/v1/2023.findings-acl.394.
26. Li Y, Zhou Q, Li Y, Li Z, Liu R, Sun R, et al. The past mistake is the future wisdom: error-driven contrastive probability optimization for Chinese spell checking. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022); 2022 May 22–27; Dublin, Ireland. p. 3202–13. doi:10.18653/v1/2022.findings-acl.252.
27. Bao Z, Li C, Wang R. Chunk-based Chinese spelling check with global optimization. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020); 2020 Nov 16–20; Punta Cana, Dominican Republic. p. 2031–40. doi:10.18653/v1/2020.findings-emnlp.184.
28. Wang B, Che W, Wu D, Wang S, Hu G, Liu T. Dynamic connected networks for Chinese spelling check. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021); 2021 Aug 1–6; Online. p. 2437–46.
29. Ji T, Yan H, Qiu X. SpellBERT: a lightweight pretrained model for Chinese spelling check. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021); 2021 Nov 7–11; Punta Cana, Dominican Republic. p. 3544–51. doi:10.18653/v1/2021.emnlp-main.287.
30. Sun Z, Li X, Sun X, Meng Y, Ao X, He Q, et al. 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 (ACL-IJCNLP 2021); 2021 Aug 1–6; Online. p. 2065–75. doi:10.18653/v1/2021.acl-long.161.
31. Yang Q, Xie H, Cheng G, Wang FL, Rao Y. Pronunciation-enhanced Chinese word embedding. Cogn Comput. 2021;13(3):688–97. doi:10.1007/s12559-021-09850-9.
32. Wang D, Liu P, Zheng Y, Qiu X, Huang X. Heterogeneous graph neural networks for extractive document summarization. In: Proceedings of The 58th Annual Meeting of the Association For Computational Linguistics; 2020 Jul 5–10; Online. p. 6209–19. doi:10.18653/v1/2020.acl-main.553.
33. Zhong M, Liu P, Chen Y, Wang D, Qiu X, Huang X. Extractive summarization as text matching. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul 5–10; Online. p. 6197–208. doi:10.18653/v1/2020.acl-main.552.
34. Nallapati R, Zhai F, Zhou B. SummaRuNNer: a recurrent neural network based sequence model for extractive summarization of documents. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017); 2017 Feb 4–9; San Francisco, CA, USA. p. 3075–81. doi:10.1609/aaai.v31i1.10958.
35. Dong Y, Shen Y, Crawford E, van Hoof H, Cheung JCK. BanditSum: extractive summarization as a contextual bandit. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing; 2018 Oct 31–Nov 4; Brussels, Belgium. p. 3739–48. doi:10.18653/v1/d18-1409.
36. Liu Y. Fine-tune BERT for extractive summarization. arXiv:1903.10318. 2019.
37. Narayan S, Cohen SB, Lapata M. Ranking sentences for extractive summarization with reinforcement learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2018 Jun 1–6; New Orleans, LA, USA. p. 1747–59. doi:10.18653/v1/n18-1158.
38. Sue KL, Chen YC. A dual adversarial structure of generative adversarial network for nature language generation. Ind Manag Data Syst. 2025;125(4):1279–305. doi:10.1108/imds-05-2024-0435.
39. Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, et al. mT5: a massively multilingual pre-trained text-to-text transformer. In: Proceedings of the 2021 Conference of the North American Chapter of the Association

for Computational Linguistics: Human Language Technologies; 2021 Jun 6–11; Online. p. 483–98. doi:10.18653/v1/2021.naacl-main.41.
40. Hu B, Chen Q, Zhu F. LCSTS: a large scale Chinese short text summarization dataset. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing; 2015 Sep 17–21; Lisbon, Portugal. p. 1967–72. doi:10.18653/v1/d15-1229.
41. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17); 2017 Dec 4–9; Long Beach, CA, USA. p. 5998–6008.
42. Wu SH, Liu CL, Lee LH. Chinese spelling check evaluation at SIGHAN bake-off 2013. In: Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing; 2013 Oct 14; Nagoya, Japan. p. 35–42.
43. 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; 2014 Oct 20–21; Wuhan, China. p. 220–3. doi:10.3115/v1/w14-6835.
44. Tseng YH, Lee LH, Chang LP, Chen HH. Introduction to SIGHAN 2015 bake-off for Chinese spelling check. In: Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing; 2015 Jul 30–31; Beijing, China. p. 32–7. doi:10.18653/v1/w15-3106.
45. 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; 2018 Oct 31–Nov 1; Brussels, Belgium. p. 2517–27. doi:10.18653/v1/d18-1273.
46. Grattafiori A, Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, et al. The llama 3 herd of models. arXiv:2407.21783. 2024.