PAPER

Chinese Spelling Correction Based on Knowledge Enhancement and Contrastive Learning

Hao WANG[†,††a)], Yao MA[†,††b)], *Nonmembers*, Jianyong DUAN[†,††c)], *Member*, Li HE[†,††], *and* Xin LI[†,††], *Nonmembers*

SUMMARY Chinese Spelling Correction (CSC) is an important natural language processing task. Existing methods for CSC mostly utilize BERT models, which select a character from a candidate list to correct errors in the sentence. World knowledge refers to structured information and relationships spanning a wide range of domains and subjects, while definition knowledge pertains to textual explanations or descriptions of specific words or concepts. Both forms of knowledge have the potential to enhance a model's ability to comprehend contextual nuances. As BERT lacks sufficient guidance from world knowledge for error correction and existing models overlook the rich definition knowledge in Chinese dictionaries, the performance of spelling correction models is somewhat compromised. To address these issues, within the world knowledge network, this study injects world knowledge from knowledge graphs into the model to assist in correcting spelling errors caused by a lack of world knowledge. Additionally, the definition knowledge network in this model improves the error correction capability by utilizing the definitions from the Chinese dictionary through a comparative learning approach. Experimental results on the SIGHAN benchmark dataset validate the effectiveness of our approach.

key words: *Chinese spelling correction, contrastive learning, knowledge graph, world knowledge, definition knowledge*

1. Introduction

Chinese spelling correction (CSC) aims to detect and correct spelling errors in texts [1]. CSC has many practical applications in daily life, such as in online searches, where search engines automatically correct input text errors, and when users receive error suggestions while using Chinese input methods.

Many existing research works have employed BERT [2] to tackle the task of CSC, yielding significant achievements. World knowledge encompasses structured information and relationships across various fields and subjects. According to the research conducted by Zhang et al. [11] and our observations, it has been found that in Chinese spelling correction methods based on BERT, approximately 39% of uncorrected results are attributed to a lack of world knowledge. Despite BERT's ability to learn some common sense and background knowledge through statistical patterns, the absence of world knowledge can hinder its effectiveness in correcting errors that require reasoning and comprehension abilities. As shown in Table 1, in the sentence "埃及的著名景点金子塔" (The famous landmark of Egypt is the golden tower.), lacking guidance from world knowledge, the model might not correct "金子塔" (the golden tower) to "金字塔" (the Pyramids).

Table 1 Examples of Chinese spelling errors. The words in red are incorrect, and the words in blue are correct.

Wrong	埃及的著名景点金子塔。 The famous landmark of Egypt is the golden tower.
Correct	埃及的著名景点金字塔。 The famous landmark of Egypt is the Pyramids.
Wrong	他是办公室主人。 He is the owner of the office.
Correct	他是办公室主任。 He is the office director.

Many researchers have recognized that Chinese pronunciation and character forms contain abundant knowledge that can guide spelling correction models. These efforts introduce multimodal information into the task of CSC, addressing errors arising from similar pronunciation and character forms. However, these approaches overlook the significance of definition knowledge within Chinese dictionaries. As a result, they may fail to correct errors that necessitate a deeper understanding of the context for accurate correction. Definition knowledge refers to the explanatory statements about words or characters found in dictionaries. These explanatory statements provide descriptions of a word's meaning, usage, category, and attributes. Introducing the definition knowledge from dictionaries can assist models in better understanding the meanings of words and characters. For instance, in the sentences "他是办公室主人" (He is the owner of the office) and "他是办公室主任" (He is the office director), the characters "主人 (owner)" and "主任 (director)" have similar pronunciations, and both "人" and "任" can be associated with "主" A pre-trained language model might struggle to determine the correct correction due to these similarities. However, based on the definitions in the dictionary, "主人 (owner)" is defined as "the person who receives guests (in contrast to 'guest')," while "主任 (director)" is defined as "a job title, the principal person in charge of a department or organization." Guided by definition knowledge, the model is more likely to associate "主任 (director)" with "办公室 (office)" in this context.

Manuscript received August 17, 2023.
Manuscript revised November 25, 2023.
Manuscript publicized May 17, 2024.

[†]School of Information Science and Technology, North China University of Technology, Beijing, China.
[††]CNONIX National Standard Application and Promotion Lab, Beijing, 100144 China.
a) E-mail: wanghaomails@gmail.com
b) E-mail: mayaoworkmail@163.com
c) E-mail: duanjy@ncut.edu.cn

DOI: 10.1587/transinf.2023EDP7166

In response to the above question, we have developed a CSC method based on knowledge graphs and contrastive learning. In essence, to enable the model to acquire world knowledge, we have designed a world knowledge network that incorporates world knowledge from the knowledge graph into sentences through the construction of a knowledge tree. To prevent injecting excessive knowledge that could introduce knowledge noise and lead to a shift in the original sentence's meaning, we introduce relative positioning and visible matrices to constrain the influence of external knowledge. Next, we have devised a correction network that integrates information from Chinese characters, glyph, and pinyin using various embedding and masking strategies. Furthermore, to better utilize the definitions from the Chinese dictionary, we have designed a definition knowledge network. This network employs a comparative learning approach to create positive and negative example pairs that include the definition knowledge. By training the model on these example pairs, we enable it to address errors that are difficult to correct solely based on phonetic and morphological information.

2. Related Work

2.1 Chinese Spelling Correction

Early research methods for CSC often followed a process of error detection, candidate generation, and candidate selection [3]. These methods primarily employed unsupervised language models and rule-based approaches for error detection and correction [4]. Some approaches treated Chinese text correction as a text labeling problem and incorporated Conditional Random Fields (CRF) and Hidden Markov Models (HMM) into the correction models [5]. Certain methods utilized n-gram statistical language models [6], [7], where a character was considered a spelling error if its probability of appearing in an n-gram language model was below a predefined threshold.

Recently, deep learning has been applied to CSC tasks. Wang et al. [8] utilized a bidirectional Long Short-Term Memory (LSTM) as the framework for their correction model. Hong et al. [9] introduced FASpell, which employs a Seq2Seq model with BERT as the encoder. In this approach, the language model is used as a candidate word generator, and a confidence-similarity curve is used to select the best candidate words. Guo et al. [10] proposed GAD, which involves a global attention decoder method and a confusion set-guided replacement strategy for pretraining BERT. Zhang et al. [11] introduced Soft-Masked-Bert, which uses a GRU-based error detection network to calculate spelling error probabilities. Based on these error probabilities, BERT is used to correct errors. However, pre-trained language models like BERT only consider the semantic features of characters, ignoring their visual and phonetic features. Cheng et al. [12] introduced the SpellGCN model using Graph Convolutional Networks (GCN) [13]. This model combines embeddings of characters with similar pronunciation and shape and leverages BERT to model dependencies between characters. Liu et al. [14] proposed PLOME, which incorporates a confusion set-based masking strategy and introduces phonetic and stroke information. MDCSpell [24] is a multi-task detector-corrector framework that employs BERT to capture the visual and phonetic features of each character in the original sentence. These works acknowledge the guiding role of phonetic and visual information for CSC models, but they overlook the instructive role of world knowledge in spelling correction. Additionally, they do not take into account the potential utilization of rich definition knowledge from the Chinese dictionary for the correction task.

2.2 Contrastive Learning

Contrastive learning is an unsupervised learning method whose objective is to establish a representation learning model by capturing the similarities and differences between samples. This is achieved by pulling semantically similar samples closer together in the embedding space, while pushing semantically dissimilar samples farther apart, thereby acquiring meaningful representations. Contrastive Learning has found wide applications in the field of computer vision, and in recent years, it has gradually garnered substantial attention in the domain of natural language processing. For instance, prior research in natural language processing has employed contrastive learning to generate improved word embeddings [15] and sentence embeddings [16]. Recently, with the dominance of Transformer-based models in natural language processing tasks, contrastive learning has also been utilized to train Transformer models [17]. This paper adopts the principles of contrastive learning to enable the CSC model to better absorb definition knowledge from dictionaries.

3. Methodology

3.1 Problem Formulation

The CSC task can be formally represented as follows: Given a text sequence $X = \{x_1, x_2, \ldots, x_n\}$, which may potentially contain spelling errors, where n represents the total number of characters in the input text, the model aims to replace incorrect words or characters within the sequence X with their correct counterparts. The resulting output is a corrected text sequence $Y = \{y_1, y_2, \ldots, y_n\}$.

3.2 Model

As illustrated in Fig. 1, the model primarily consists of three components. The dashed box depicted in the lower left corner represents the world knowledge network, the dashed box in the upper left corner outlines the correction network, and the dashed box in the upper right corner represents the definition knowledge network. The input sequence at the bottom is passed through the world knowledge network, incorporating knowledge from the knowledge graph. The resulting tensor from the correction network is then fed into the definition

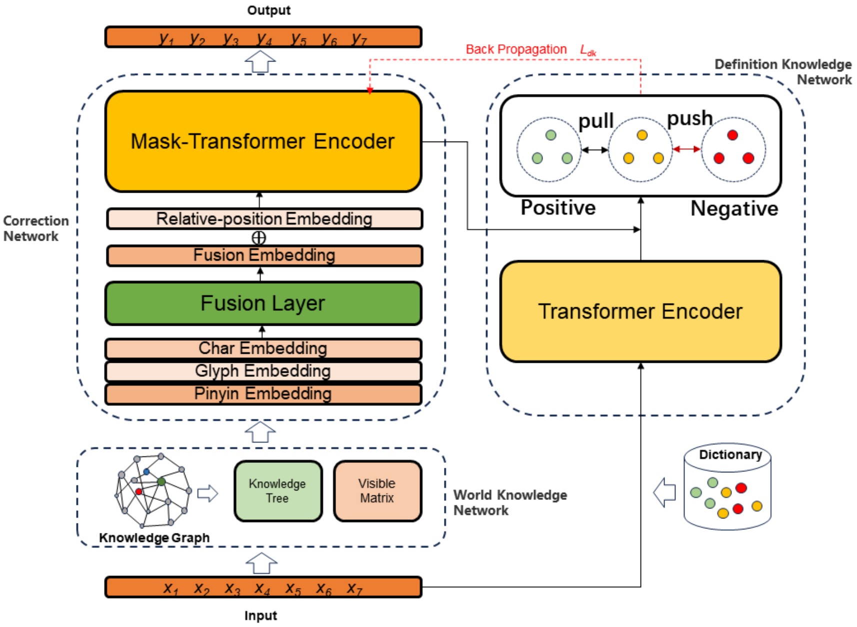

Fig. 1 Overview of our model.

knowledge network. The green circles denote positive examples, while the red circles represent negative examples. Using the contrastive learning approach, the loss L_{dk} is propagated back to the correction network, ultimately yielding the output sequence at the top. ⊕ stands for vector addition.

3.2.1 World Knowledge Network

World knowledge involves various facts, relationships, concepts, and rules, which can provide the model with a richer understanding of semantics and background information. In this module, we concatenate entity-relation triples from the knowledge graph with entities in the input sentence to construct a knowledge tree that contains the input sequence and relationship triples. This enables the model to learn world knowledge and enhance its ability to correct errors. To elaborate, given a sentence $S = \{w_1, w_2, \ldots, w_n\}$ and a knowledge graph KG, where each token in sentence S is from the vocabulary set V (i.e., $w_i \in V$), KG includes numerous relation triples $R = (w_i, r_j, w_k)$, with r_j representing the relation in the triple. Through knowledge injection, the sentence tree T is derived as follows: $T = \{w_1, \ldots, w_i\{(r_{i1}, w_{i1}), \ldots, (r_{ij}, w_{ij})\}, \ldots, w_n\}$.

Traditional BERT models can only handle sequential sentence inputs and cannot directly process sentence trees. Converting a sentence tree directly into a sequence would result in the loss of structural information inherent in the sentence tree. We ingeniously retain the structural information of the sentence tree during the conversion process using two different forms of position indices. As illustrated in Fig. 2, circles represent nodes in the sentence tree, with entities inside. The black numbering represents absolute position indices, assigned to nodes in the sentence tree following a preorder traversal. These absolute position indices guide the generation of the visible matrix. The red numbering corresponds to relative position indices, where different branches of the same node in the sentence tree share the same index. This approach helps recover the structural information lost in the absolute position indices.

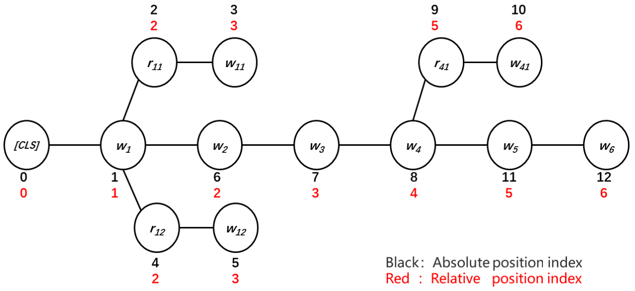

Fig. 2 Illustration of sentence tree structure.

Due to the potential risk of altering the original sentence's meaning through excessive knowledge injection, we have constrained the depth of the knowledge tree to 1. This is mainly because when the depth of the sentence tree is not restricted to 1, entities in the sentence tree are likely to iteratively derive branches based on the triplets in the knowledge graph. As a result, the sentence tree becomes exceptionally complex, and redundant knowledge may even alter the original meaning of the sentence, leading to a decrease in the correction performance. Simultaneously, we employ a visible matrix to control the visibility between main tokens and branch tokens, effectively mitigating the issues caused by knowledge noise. The visible matrix is defined as follows:

$$M_{ij} = \begin{cases} 0, & w_i \Leftrightarrow w_j \\ -\infty, & w_i \not\Leftrightarrow w_j \end{cases} \tag{1}$$

Where $w_i \Leftrightarrow w_j$ indicates that tokens w_i and w_j are

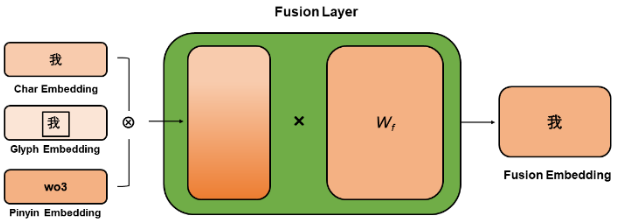

Fig. 3 Illustration of fusion layer. ⊗ denotes vector concatenation, × stands for vector matrix multiplication, W_f denotes the learnable matrix.

in the same branch, and $w_i \nLeftrightarrow w_j$ indicates that w_i and w_j are not in the same branch. Here, i and j represent absolute position indices. Specifically, tokens on the main trunk of the sentence tree are mutually visible only with other tokens on the main trunk. Tokens at cross-nodes are visible to both the main trunk and branches, while tokens on branches are visible only to tokens at cross-nodes and other tokens on the same branch. This approach prevents tokens on different branches from accessing each other's information, thus preserving the original sentence's semantics.

3.2.2 Correction Network

This paper employs a pre-trained Mask-Transformer Encoder as the backbone of the correction network. To enable the pre-trained model to learn the similarities between Chinese characters, pinyin (pronunciation), and glyph (visual shape), we fuse the char embeddings, glyph embeddings, and pinyin embeddings of the input text.

Specifically, the execution of char embedding is similar to token embedding used in BERT but operates at the character granularity. We utilize three types of Chinese fonts: *XingKai*, *FangSong*, and *LiShu*. Each font is instantiated as a 24 × 24 image with floating-point pixels ranging from 0 to 255. The resulting 24 × 24 × 3 vector is flattened and fed into an FC layer to obtain the output glyph embedding. To obtain the pinyin sequence, we utilize an open-source phonetics package. Tonal symbols representing the character's tone are added to the end of each character's pinyin sequence. The pinyin sequence is then processed using a CNN model with a width of 2, followed by max-pooling, to derive the final pinyin embedding.

As illustrated in Fig. 3, we concatenate the char embedding, glyph embedding, and pinyin embedding to form a three-dimensional vector. The fusion layer maps this vector to a one-dimensional representation using a FC layer. The resulting fusion embedding is then added to the relative-position embedding mentioned in the previous section and input into the Encoder.

We utilize a masking method different from that of BERT. Specifically, BERT uses the [MASK] token for random masking, wherein approximately 15% of the tokens are replaced with the special [MASK] token. During pre-training, the model endeavors to predict these masked tokens, aiming to learn contextual information and relationships among vocabulary items. According to the research by Liu et al. [22], approximately 83% of Chinese spelling errors are caused by the misuse of phonetically similar characters, while 48% are due to the misuse of visually similar characters. Therefore, in most cases, we mask characters that are phonetically similar based on their findings. We mask a total of 15% of tokens in the corpus. Among these, 60% are masked using characters from the confusion set that are phonetically similar. This trains the model to predict the original character based on the pinyin of easily confused Chinese characters. Another 15% are masked using visually similar characters from the confusion set, allowing the model to recover the original character based on visually similar glyphs. An additional 15% are masked using the [MASK] token, training the model to restore the masked character based on contextual information. Finally, 10% of characters are randomly selected and masked, training the model to correct characters from randomly occurring errors.

In order to incorporate the structured information of the sentence tree into BERT while preventing alterations in the original semantic meaning, we employ a mask-self-attention mechanism combined with a visible matrix to restrict the self-attention region. The mask-self-attention can be formally described as follows:

$$Q^t, K^t, V^t = h^{t-1}W_q, h^{t-1}W_k, h^{t-1}W_v \tag{2}$$

$$S^t = softmax\left(\frac{Q^t\, K^{t^{\mathrm{T}}} + M}{\sqrt{d_k}}\right) \tag{3}$$

$$h^t = S^t V^t \tag{4}$$

Where W_q, W_k, W_v are trainable model parameters. h^trepresents the hidden state of the t-th mask-self-attention. M is the visible matrix. d_k is a scaling factor. In summary, if w_j and w_k are not in the same branch of the sentence tree, they are considered invisible to each other. The value at the corresponding position in the visible matrix becomes negative infinity, causing M_{jk} to set the attention score S^t_{jk} to 0. This implies that the hidden state of w_j does not contribute to w_k.

Ultimately, after passing through the correction network, the probability of the i-th token predicting a character in the given sentence X is defined as:

$$P_c\,(y_i = j|X) = Softmax\,(Wh_i + b)\,[j] \tag{5}$$

Where $P_c(y_i = j|X)$ represents the conditional probability that the i-th character x_i in the input sentence X is predicted as the j-th character in the vocabulary $\boldsymbol{V}$. h_i signifies the output tensor of the last hidden layer of x_i after passing through the correction network. $W \in \mathbb{R}^{n\times 768}$ and $b \in \mathbb{R}^n$ are the parameters trained by the model, where n is the size of the vocabulary.

3.2.3 Definition Knowledge Network

Definition knowledge provides relevant information about word meanings, usages, and more, enabling the model to better infer within the context. Previous research has predominantly focused on enhancing CSC by utilizing phonetic and visual characteristics. However, the rich conceptual information within the Chinese dictionary has been overlooked. In this aspect, we apply the abundant definition knowledge from the Chinese dictionary to the CSC model using a contrastive learning approach. Specifically, leveraging the idea of contrastive learning, we construct a positive sample (d^o, d^p) and N negative samples $\left(d^o, d_i^n\right)_{i=1}^N$ for the original sentence. Here, d^o represents the erroneous word or character in the input sentence, and negative samples are acquired from the structured Chinese dictionary D based on d^o.

For positive sample d^p and negative sample d^n of length l, we use the encoder E_d to map them into a sequence of representations, obtaining e^p and $\{e_i^n\}$. The encoder employed is BERT. For the original sentence d^o, we utilize the encoder E_c from the correction network for encoding, resulting in the sentence representation e^o. Formally expressed as:

$$e^o = E_c(d^o) \tag{6}$$

$$e^p = E_d(d^p) \tag{7}$$

$$e_i^n = E_d\left(d_i{}^n\right) \tag{8}$$

When fine-tuning the CSC model using definition knowledge from the dictionary, we start by considering an original sentence x^o = “今天田七不错” and its corresponding target sentence x^g = “今天天气不错”. We tokenize the target sentence using a segmentation tool, resulting in “今天/天气/不错”. Consequently, we determine the error position s where “天气” occurs incorrectly in the original sentence. Next, we locate the concept explanation for “天气” in the dictionary, which serves as the positive sample d^p. Simultaneously, we select N different concept explanations from the dictionary as negative samples $\{d_i^n\}$. In cases where a single concept has multiple explanations, we devise a simple retriever to calculate the vector similarity between the target sentence and the vectors representing the definitions in the dictionary. This allows us to retrieve the most contextually relevant concept explanation for use as positive and negative samples in the contrastive learning process.

The formula for calculating the similarity between positive and negative samples and the original sample in this paper can be formalized as follows:

$$f_d\left(e^o, e^p, s\right) = cos(avg\left(e^o[s, s+w]\right), avg\left(e^p\right)) \tag{9}$$

$$f_d\left(e^o, e_i^n, s\right) = cos\left(avg\left(e^o[s, s+w]\right), avg\left(e_i^n\right)\right) \tag{10}$$

Where $cos(a, b)$ calculates the cosine distance between vectors a and b, $e^o[s, s+w]$ represents the vector of the phrase within the index range from s to $s+w$ of the erroneous character, where $s + w \leq l$, and $avg()$ denotes the mean pooling operation.

3.2.4 Loss

We define the loss function of the correction network as:

$$L_c = -\sum_{i=1}^{N} logP_c\left(y_i|X\right) \tag{11}$$

We define the loss function of the definition knowledge network based on contrastive learning as:

$$L_{dk} = -log\frac{f_d\left(e^o, e^p, s\right)}{f_d\left(e^o, e^p, s\right) + \sum_{i=1}^{N} f_d\left(e^o, e_i^n, s\right)} \tag{12}$$

Where L_c is the training objective of the correction network, L_{dk} is the training objective of the definition knowledge network, and f_d is the representation measurement function for definition knowledge in the dictionary. In a batch, all sentences have a length of l, and the s-th character represents the one with the spelling error.

The learning process of our model involves optimizing the correction network and the contrastive learning module. The final loss function is represented as:

$$L = \lambda_1 L_c + \lambda_2 L_{dk} \tag{13}$$

We believe that the loss L_c of the correction network and the loss L_{dk} of the definition knowledge network contribute equally to the overall performance of the model. Therefore, we set the weights λ_1 and λ_2 for the two losses to be 1.

4. Experiments

4.1 Datasets

In this study, we utilized the official training data from SIGHAN and pseudo data generated by Wang et al. [8] as the training set. Extensive testing was conducted on SIGHAN13 [19], SIGHAN14 [20], and SIGHAN15 [4]. Table 2 presents detailed information regarding the training and testing data, where sent_num, avg_len, and errors_num respectively indicate the number of sentences, average length, and the count of errors in the dataset. We incorporated knowledge into the model through the triplet relationships from the Chinese knowledge graph CN-DBpedia [21]. CN-DBpedia, curated and maintained by Fudan University, is

Table 2 Experimental training and testing data statistics.

Train Set	**sent_num**	**avg_len**	**errors_num**
SIGHAN13	700	41.8	343
SIGHAN14	3,437	49.6	5,122
SIGHAN15	2,338	31.3	3,037
Wang271K	271,329	42.6	381,962
Total	277,804	42.6	390,464
Test Set	**sent_num**	**avg_len**	**errors_num**
SIGHAN13	1,000	74.3	1,224
SIGHAN14	1,062	50.0	771
SIGHAN15	1,100	30.6	703
Total	3,162	50.9	2,698

Table 3 Experimental results of the model on SIGHAN13, SIGHAN14, and SIGHAN15 test datasets.

Dataset	Method	Detection Level			Correction Level		
		P	R	F1	P	R	F1
SIGHAN13	BERT	85.0%	77.0%	80.8%	83.0%	75.2%	78.9%
	SpellGCN	80. 1%	74. 4%	77. 2%	78. 3%	72. 7%	75. 4%
	DCN	86. 8%	79. 6%	83. 0%	84. 7%	77. 7%	81. 0%
	GAD	85.7%	79.5%	82.5%	84.9%	78.7%	81.6%
	MLM-phonetics	82.0 %	78.3%	80.1 %	79.5 %	77.0 %	78.2 %
	Ours	**87.6%**	**80.5%**	**83.9%**	**85.2%**	**79.5%**	**82.3%**
SIGHAN14	BERT	64.5%	68.6%	66.5%	62.4%	66.3%	64.3%
	SpellGCN	65. 1%	69. 5%	67. 2%	63. 1%	67. 2%	65. 3%
	DCN	67. 4%	70. 4%	68. 9%	65. 8%	68. 7%	67. 2%
	GAD	66.6%	71.8%	69.1%	65.0%	70.1%	67.5%
	MLM-phonetics	66.2 %	**73.8 %**	69.8%	64.2 %	**73.8 %**	68.7%
	Ours	**71.7%**	70.1%	**70.9%**	**69.5%**	69.7%	**69.6%**
SIGHAN15	BERT	74.2%	78.0%	76.1%	71.6%	75.3%	73.4%
	SpellGCN	74. 8%	80. 7%	77. 7%	72. 1%	77. 7%	75. 9%
	DCN	77. 1%	80. 9%	79. 0%	74. 5%	78. 2%	76. 3%
	GAD	75.6%	80.4%	77.9%	73.2%	77.8%	75.4%
	MLM-phonetics	77.5 %	**83.1 %**	80.2%	74.9 %	80.2%	77.5%
	Ours	**79.1%**	82.7%	**80.9%**	**77.9%**	**81.2%**	**79.5%**

a comprehensive domain-agnostic structured encyclopedia knowledge graph, encompassing millions of entities and relationships. To enhance the guidance provided by the original knowledge graph, we filtered out triplet pairs where entity names were less than 2 characters in length or contained special characters. After this processing, approximately 5.17 million entity relationship triplets were retained.

4.2 Experiment Setup

In this experiment, the correction network consists of 12 Transformer layers with 12 attention heads, each having a vector dimension of 768. The training batch size is set to 32, and the number of epochs is set to 10. The learning rate is set to 5e-5. We utilized the PyTorch training framework and employed the NVIDIA GeForce RTX 3090 GPU equipment.

4.3 Baseline Models

BERT [25]: Fine-tuning BERT directly using spelling correction training data to adapt it for the Chinese spelling correction task.
SpellGCN [12]: By combining GCN and BERT, the information from the confusion set is incorporated into the model to model the relationships between characters within the confusion set.
DCN [26]: A phonetic-enhanced candidate generator is proposed, which introduces a dynamic linking network to establish dependencies and utilizes this network to score and search for the optimal path.
GAD [10]: The global attention decoder is utilized to learn the overall relationships between potential correct input characters and candidate incorrect character candidates, acquiring rich global contextual information, and effectively alleviating the impact of local erroneous context information.
MLM-phonetics [18]: During pre-training, mask words with speech features and phonetically similar pronunciations, integrating speech features into the language model.

4.4 Evaluation Metrics

In this study, to evaluate the spelling correction performance of the CSC model, recall, precision, and F1 score are employed as evaluation metrics to assess the experimental results. The F1 score is used to comprehensively consider both accuracy and recall, serving as a benchmark to gauge the effectiveness of the model.

4.5 Main Results

Table 3 presents the experimental results of our approach on the SIGHAN13, SIGHAN14, and SIGHAN15 test sets, with the best results highlighted in bold. By comparing with multiple models, it can be observed that our model achieves the best F1 scores in both detection and correction on all three datasets.

Specifically, when using BERT alone for correction, the model relies solely on contextual semantics and overlooks other supportive information inherent to the Chinese language. On the SIGHAN15 dataset, our method outperforms the pure BERT-based approach with improvements

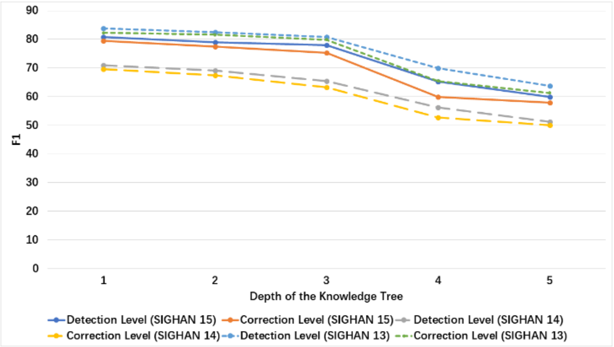

Fig. 4 Effect of knowledge tree with different depth on error correction performance.

of 4.8% and 6.1% in F1 scores for sentence-level detection and correction, respectively. Furthermore, compared to the GAD model utilizing global attention decoding, our approach shows improvements of 3% and 4.1% in detection and correction F1 scores, respectively. This is attributed to the fact that the world knowledge network not only provides more guiding information to the model but also the utilization of mask-self-attention mitigates the issue of excessive knowledge injection, thus affirming the effectiveness of incorporating world knowledge and employing mask-self-attention.

MLM-phonetics only considered incorporating phonetic features into the model. In comparison, our approach yielded improvements of 0.7% and 2% in detection and correction F1 scores on the SIGHAN15 dataset, surpassing the results of using MLM-phonetics directly for error correction. While SpellGCN focused solely on the visual and phonetic similarities within Chinese characters, neglecting the definition knowledge from dictionaries, our approach achieved gains of 3.2% and 3.6% over SpellGCN in detection and correction F1 scores, respectively. This is attributed to the fact that definition knowledge dissects the underlying meanings of words, enabling the model to better align with the context for error correction. This demonstrates that incorporating not only multi-modal knowledge such as phonetics and character shapes but also adding definitional knowledge from dictionaries can indeed guide the improvement of error correction model performance.

To demonstrate the rationality of setting the depth of the tree to 1, we explored the F1 scores of the model on three datasets under different tree depths. As shown in Fig. 4, it is evident that as the tree depth increases, the F1 scores of the model decrease on all three datasets. We believe that when the tree depth is greater, the external knowledge carried by the knowledge tree becomes more complex and excessive injection of knowledge can alter the original meaning of the sentence, thus leading to a decrease in the correction performance.

4.6 Ablation Study

We conducted ablation experiments on the SIGHAN15 dataset to investigate the roles of the two crucial modules in our approach. As shown in Table 4, when world knowledge guidance is not used, we observe a decrease of 0.6% and 0.4% in the F1 values for both detection and correction performance, respectively. This indicates that incorporating world knowledge indeed enhances the performance of the correction model. When the definition knowledge network based on contrastive learning is removed, the model's detection and correction performance show a larger decrease of 1.1% and 0.7%, respectively, compared to the removal of world knowledge. This demonstrates the significant impact of the definition knowledge from the Chinese dictionary on the correction model. Moreover, when both world knowledge and definition knowledge are simultaneously removed, the model's detection and correction capabilities decrease by 1.6% and 1.5%, respectively. Nevertheless, the performance is still superior to directly using the BERT model for CSC. This confirms the substantial contributions of the two types of knowledge introduced in our approach to the improvement of experimental results.

Table 4 Presents the results of ablation experiments on the world knowledge network and the definition knowledge network on the SIGHAN15 test dataset. WK represents the world knowledge module, and DK represents the definition knowledge module.

Method	Detection Level			Correction Level		
	P	R	F1	P	R	F1
Ours	79.1%	82.7%	80.9%	77.9%	81.2%	79.5%
Ours-WK	78.9%	81.8%	80.3%	77.4%	80.8%	79.1%
Ours-DK	78.8%	80.9%	79.8%	77.1%	80.6%	78.8%
Ours-WK-DK	78.1%	80.5%	79.3%	76.0%	80.1%	78.0%
BERT	74.2%	78.0%	76.1%	71.6%	75.3%	73.4%

To investigate the impact of our hybrid masking approach and mask-self-attention method on our correction network, we conducted experiments while keeping other settings unchanged, except for changing the masking strategy to match that of BERT. From the results in Table 5, we observe that the model's performance in both detection and correction aspects decreases by 1.4% and 1.7%, respectively. This is because our hybrid masking approach forces the model to learn more information about Chinese characters' phonetics and shapes compared to using only the [MASK] masking strategy. When we remove mask-self-attention, the structural information of the sentence tree becomes disordered, causing a change in the original semantics of the sentence. The model struggles to capture contextual information effectively, leading to a significant performance drop. Thus, both of the aforementioned strategies play a crucial role in the effectiveness of our approach.

To validate the impact of glyph embedding and pinyin embedding on the overall performance of our model, we removed glyph embedding and pinyin embedding separately and observed the changes in F1 scores on the SIGHAN15

Table 5 Shows the results of ablation experiments on the correction network on the SIGHAN15 test dataset.

Method	Detection Level			Correction Level		
	P	R	F1	P	R	F1
Ours	79.1%	82.7%	80.9%	77.9%	81.2%	79.5%
Using only [mask] for masking	77.9%	81.2%	79.5%	76.4%	79.3%	77.8%
Without using mask-self-attention.	75.3%	79.9%	77.5%	74.1%	76.6%	75.3%
BERT	74.2%	78.0%	76.1%	71.6%	75.3%	73.4%

Table 6 Results of ablation experiments on glyph embedding and pinyin embedding on SIGHAN15 test dataset. GE represents glyph embedding, and PE represents pinyin embedding.

Method	Detection Level			Correction Level		
	P	R	F1	P	R	F1
Ours	79.1%	82.7%	80.9%	77.9%	81.2%	79.5%
Ours-GE	77.1%	79.3%	78.2%	73.8%	76.2%	75.0%
Ours-PE	76.8%	79.4%	78.1%	73.3%	77.7%	75.4%
Ours-GE-PE	75.2%	79.0%	77.1%	72.4%	76.1%	74.2%

dataset. As shown in Table 6, We found that the absence of either embedding led to a varying degree of decline in both error detection and error correction performance of the model. This indicates that the combined use of these embeddings can enhance the error correction ability of the model more effectively.

4.7 Case Study

In order to visually demonstrate the impact of incorporating world knowledge and definition knowledge into the CSC model, we present the correction results of four cases in Table 7. During pre-training, ChineseBERT [23] incorporates the phonetic and character shape information of Chinese characters. Compared to directly using ChineseBERT for CSC, our model integrates world knowledge and definition knowledge. From the cases in the table, it can be observed that due to the identical pronunciation of “恒山 (Mount Heng)” and “衡山 (Mount Heng)” both of which are well-known mountains in China, but “恒山 (Mount Heng)” is located in Datong City, Shanxi Province, while “衡山 (Mount Heng)” is situated in Hengyang City, Hunan Province. ChineseBERT tends to correct errors into visually similar characters, but it does not take actual background knowledge into account. However, when guided by world knowledge, our model easily identifies that the correct term in the example should be “恒山 (Mount Heng)” instead of “衡山 (Mount Heng)”.

After incorporating definition knowledge, by learning the definitions of “眼睛 (eye)” and “眼镜 (glasses)” from the dictionary, the model can understand that “an eye is typically composed of the cornea, pupil, iris, lens, retina, vitreous body, etc.,” while “眼镜 (glasses)” are “lenses worn on the eyes to correct vision or protect the eyes.” Based on the key information “瞳孔 (pupil)” in the erroneous sentence, it can be inferred that “眼镜 (glasses)” should be corrected to “眼睛 (eye)”.

Table 7 Here are examples of our model’s inputs/outputs, with red indicating spelling errors and blue indicating correct ones.

Error correction guided by world knowledge
Wrong：兵马车(carriage)位于秦始皇陵以东约两公里处。 Correct：兵马俑(Terracotta Warriors and Horses)位于秦始皇陵以东约两公里处。 ChineseBERT: 兵马车(carriage)位于秦始皇陵以东约两公里处。 Ours：兵马俑(Terracotta Warriors and Horses)位于秦始皇陵以东约两公里处。
Wrong：我们去山西大同游览了蘅(reed)山悬空寺。 Correct：我们去山西大同游览了恒山(Mount Heng)悬空寺。 ChineseBERT: 我们去山西大同游览了衡山(Mount Heng)悬空寺。 Ours：我们去山西大同游览了恒山(Mount Heng)悬空寺。
Error correction guided by definition knowledge
Wrong：铁轨上开来一辆或(or)车。 Correct：铁轨上开来一辆火车(train)。 ChineseBERT：铁轨上开来一辆货车(truck)。 Ours：铁轨上开来一辆火车(train)。
Wrong：他的眼竟(unexpectedly)很好看，因为他瞳孔周围是蓝色的。 Correct：他的眼睛(eye)很好看，因为他瞳孔周围是蓝色的。 ChineseBERT：他的眼镜(glasses)很好看，因为他瞳孔周围是蓝色的。 Ours：他的眼睛(eye)很好看，因为他瞳孔周围是蓝色的。

In summary, the method we propose can effectively utilize world knowledge to correct errors that do not align with the actual context. Adding world knowledge and definition knowledge can assist the model in better understanding the true intended meaning of the original sentence.

5. Conclusion

We have introduced a Chinese Spelling Correction model that effectively leverages world knowledge from a knowledge graph and definition knowledge from a dictionary. To better incorporate these two heterogeneous forms of knowledge into the model, we have constructed a knowledge tree network to inject world knowledge into sentences. Additionally, to efficiently utilize the definition knowledge from the dictionary, we have employed a contrastive learning approach, creating positive and negative example pairs for model fine-tuning. Results on the SIGHAN dataset demonstrate the positive guiding significance of our method for Chinese Spelling Correction tasks. In the future, we will explore the role of this knowledge in models for Chinese grammar correction.

Acknowledgments

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

References

[1] J. Yu and Z. Li, "Chinese spelling error detection and correction based on language model, pronunciation, and shape," Proc. Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, pp.220–223, 2014.

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

[3] L. Zhang, M. Zhou, and C. Huang, "Automatic detecting/correcting errors in Chinese text by an approximate word-matching algorithm," Proc. 38th Annual Meeting of the Association for Computational Linguistics, pp.248–254, 2000.

[4] Y.-H. Tseng, L.-H. Lee, L.-P. Chang, and H.-H. Chen, "Introduction to sighan 2015 bake-off for Chinese spelling check," Proc. Eighth SIGHAN Workshop on Chinese Language Processing, pp.32–37, 2015.

[5] J. Xiong, Q. Zhang, and S. Zhang, "HANSpeller: a unified framework for Chinese spelling correction," International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language, 2015.

[6] W. Xie, P. Huang, X. Zhang, K. Hong, Q. Huang, B. Chen, and L. Huang, "Chinese spelling check system based on n-gram model," Proc. Eighth SIGHAN Workshop on Chinese Language Processing, pp.128–136, 2015.

[7] J.F. Yeh, S.F. Li, and M.R. Wu, "Chinese word spelling correction based on n-gram ranked inverted index list," Proc. Seventh SIGHAN Workshop on Chinese Language Processing, pp.43–48, 2013.

[8] D. Wang, Y. Song, J. Li, J. Han, and H. Zhang, "A hybrid approach to automatic corpus generation for Chinese spelling check," Proc. 2018 Conference on Empirical Methods in Natural Language Processing, pp.2517–2527, 2018.

[9] Y. Hong, X. Yu, N. He, N. Liu, and J. Liu, "FASPell: A fast, adaptable, simple, powerful Chinese spell checker based on DAE-decoder paradigm," Proc. 5th Workshop on Noisy User-generated Text (W-NUT 2019), pp.160–169, 2019.

[10] Z. Guo, Y. Ni, K. Wang, W. Zhu, and G. Xie, "Global attention decoder for Chinese spelling error correction," Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp.1419–1428, 2021.

[11] S. Zhang, H. Huang, J. Liu, and H. Li "Spelling error correction with soft-masked BERT," Proc. 58th Annual Meeting of the Association for Computational Linguistics., Seatle, Washington DC, USA, pp.882–890, 2020.

[12] X. Cheng, W. Xu, K. Chen, S. Jiang, F. Wang, T. Wang, W. Chu, and Y. Qi, "SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check," Proc. 58th Annual Meeting of the Association for Computational Linguistics, pp.871–881, 2020.

[13] T.N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.

[14] L. Huang, J. Li, W. Jiang, Z. Zhang, M. Chen, S. Wang, and J. Xiao, "PHMOSpell: Phonological and morphological knowledge guided Chinese spelling check," Proc. 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp.5958–5967, 2021.

[15] T. Mikolov, I. Sutskever, and K. Chen, "Distributed representations of words and phrases and their compositionality," Advances in Neural Information Processing Systems, p.26, 2013.

[16] L. Logeswaran and H. Lee, "An efficient framework for learning sentence representations," arXiv preprint arXiv:1803.02893, 2018.

[17] Z. Wu, S. Wang, and J. Gu, "Clear: Contrastive learning for sentence representation," arXiv preprint arXiv:2012.15466, 2020.

[18] R. Zhang, C. Pang, and C. Zhang, "Correcting Chinese spelling errors with phonetic pre-training," Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp.2250–2261, 2021.

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

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

[21] B. Xu, Y. Xu, J. Liang, C. Xie, B. Liang, W. Cui, and Y. Xiao, "CN-DBpedia: A never-ending Chinese knowledge extraction system[C]," International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems. Cham: Springer International Publishing, vol.10351, pp.428–438, 2017.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," arXiv preprint arXiv:1907.11692, 2019.

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

[24] C. Zhu, Z. Ying, B. Zhang, and F. Mao, "MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction," Findings of the Association for Computational Linguistics, Dublin, Ireland, pp.1244–1253, 2022.

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

[26] B. Wang, W. Che, D. Wu, S. Wang, G. Hu, and T. Liu, "Dynamic connected networks for Chinese spelling check," Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp.2437–2446, 2021.

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

Yao Ma is a master in School of Information Science and Technology, North China University of Technology. His major research field is Natural Language Processing.

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

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

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