MCSSpell:Optimal Path Selection of Candidate Characters by Integrating Multimodal Information and Copy Mechanism for Chinese Spelling Correction

Qing Zhang[1,2], Cheng Yang[1,2], Jianyong Duan[1,2,*], Hao Wang[1,2], Li He[1,2], Jie Liu[1,3]
[1]*School of Information Science and Technology, North China University of Technology*
[2]*CNONIX National Standard Application and Promotion Lab*
[3]*China Language Intelligence Research Center, Capital Normal University*
Beijing, China
{zqicl, duanjy, heli}@ncut.edu.cn, yangcheng_1212@163.com, wanghaomails@gmail.com, liujxxxy@126.com

***Abstract*—Chinese spelling correction (CSC) is a traditional Natural Language Processing task that aims to detect and correct spelling errors in Chinese text. Many advanced studies have adopted BERT non-autoregressive language models, masking method performs well on the CSC task. However, the model still has certain limitations. Firstly, BERT predicts tokens based on the assumption of independence, meaning it does not learn the relationships between masked tokens. As a result, it fails to model character-level dependencies effectively. Additionally, BERT tends to correct characters to more common habitual expressions, leading to the problem of over correction. To address these issues, this paper proposes a novel Candidate Character Optimal Path Selector (CCOPS) that models the dependencies between adjacent Chinese characters using attention mechanisms to alleviate the issue of character coherence. We incorporate a particular copy mechanism into the selector, which guides the model to choose the original character when both the original input and the candidate output are plausible, mitigating the problem of over correction. Furthermore, we effectively integrate the multimodal information of characters to guide error correction in terms of semantics, phonetics, and visual similarities. The experiments show that our model, compared to the latest research, has improved Detection-Level F1 by 2.7%, 1.2%, and 1.2% on three datasets, and Correction-Level F1 by 2.6%, 0.8%, and 1.3% respectively.**

***Index Terms*—Multimodal, CSC, Transformer, Copy Mechanism, BERT**

I. Introduction

Chinese Spelling Correction (CSC) is a very important Natural Language Processing which aims to detect and correct erroneous characters in input text. This task has been widely applied, such as search query correction [1], and automated essay scoring [2], among others. CSC task has a certain research foundation. In recent years, sizeable pre-trained language models, have performed well in many NLP tasks. Some researchers have applied a non-autoregressive language model based on BERT(Bidirectional Encoder Representation from Transformers) to the CSC task and achieved good results.

* Corresponding author

sentence	Correction
请你发间(jian)信(xin)给我。	简(jian)讯(xun)
Please send me a between letter.	text message

Fig. 1. The example of Chinese spelling correction using CCOPS and multimodal information, with incorrect characters marked in blue.

Therefore, most researchers use the masked language model of BERT for fine-tuning on the CSC dataset.

However, the BERT method may also cause the model to output text that contains common sense knowledge learned during pre-training when performing correction tasks. In the masking pre-training, the prediction of masked characters is independent, while in Chinese sentences, non-independent errors often occur, where the preceding and succeeding Chinese characters have mutual error impact and dependence. For example, in Fig. 1, the input text “间信”(between letter) in the sentence might be incorrectly corrected by BERT as “短信”(text message) because this is a more commonly used expression. However, if the character’s phonetic and glyphic information is integrated into the model, the phonetic and glyphic similarity between “间”(between) and “简”(simple) is high, so the prediction of non-autoregressive models is more inclined to independent assumptions, and it is likely to correct it to the wrong correction “简信”(simple letter). This highlights the importance of modeling character dependencies in CSC. However, when using CCOPS with integrated multimodal information, it would first correct “间” to “简” based on glyphic information and then, through the path selection mechanism, determine that the combination “简讯”(message) is the most suitable. To address the issue of character dependency, Conditional Random Fields (CRF) [3] have also been applied to CSC. However, since the model is based on a vocabulary for classification, the transition matrix of CRF is too large, which can lead to low efficiency of the model.

In the candidate recall module, we use a candidate character

979-8-3503-3078-6/23/$31.00

optimal path selector with a fusion and copy mechanism to optimize and select suitable path combinations for the candidate character set. Specifically, we first predict the top k candidate characters for each position and append the original input character to form k+1 candidate characters. Next, we score all possible paths between adjacent candidates to evaluate the strength of the dependencies between characters. The score of a sub-path is calculated based on the attention score [4] between adjacent candidate characters. Finally, we find the highest-scoring path as the final correction result quickly. In this process, we consider the dependencies between Chinese characters and can generate candidate $(k+1)^n$ paths, where n is the length of the input text.

In summary, our contributions can be summarized as follows: (1) we effectively utilize the semantic, phonetic, and morphological information of characters and integrate multimodal information to enhance the correction ability of the model; (2) we propose a novel candidate character optimal path selector (CCOPS) to alleviate the non-coherent problem of non-autoregressive language models in CSC tasks; (3) we introduce a unique copy mechanism to mitigate the problem of over-correction; (4) experimental results show that our approach achieves good performance on three CSC datasets.

II. Related Work

Chinese Spelling Correction (CSC) task is of great importance in the field of NLP. Due to the characteristics of Chinese characters, such as unclear segmentation boundaries and rich semantic information, CSC is a highly challenging task. With the introduction of the confusion set [5] the success of large-scale pre-trained language models, BERT has been widely used in the field of NLP. At the same time, this mask-based pre-training method is also very suitable for CSC tasks. The introduction of FASpell [6] opened the door for BERT models in CSC tasks. This model uses BERT as a denoising autoencoder (DAE), using the acoustic and visual information of text to generate candidate characters. SpellGCN [7] proposed an earlier method of using convolutional neural networks to incorporate phonetic and visual information into the BERT pre-trained language model. However, the similarity graph constructed by this method relies on a predefined set of confusions, and is therefore limited by the fixed size and inflexibility of the confusion set, making it difficult to model all similarities. PHMOSpell [8] uses the phonetic information and morphological knowledge of characters to model the similarity of Chinese characters, demonstrating the effectiveness of this multimodal approach for CSC tasks.The CRASpell [9] model constructs a noisy sample for each training sample. The correction model outputs results that are more similar to the original training data and noisy samples. To address the problem of over-correction, the model combines a replication mechanism, which allows the model to choose the input character when the error correction and input character are both effective based on the given context.

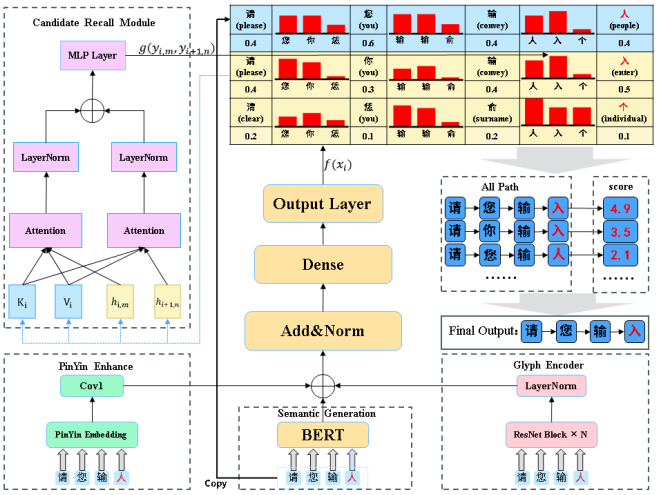

Fig. 2. MCSSpell model structure. We only explain the case when the number of candidates is 2. In the candidate recall module, we will only explain how to calculate the coherence score between the characters "输"(convey) and "人"(people).The meaning of the input sentence is "Please input someone". And the meaning of the output sentence is "Please input". "copy" represents the model using the copying mechanism to add the input sentence to the candidate characters.

III. Our Approach

A. Problem

The CSC aims to correct spelling errors in Chinese sentences, specifically targeting misspelled words or characters that occur sporadically. Typically, given a Chinese character sequence X=$\{x_1, x_2, x_3, \cdots, x_n\}$ of length n with spelling errors, the model needs to detect and correct the erroneous characters in the input sequence$Y$=$\{y_1, y_2, y_3, \cdots, y_n\}$, resulting in an output sequence that represents the corrected sentence. This task is often modeled as a sequence labeling problem, where the erroneous characters in the input sentence X are corrected in the output sentence Y while maintaining the same sentence length.

B. Candidate Generation Module

1) Semantic Generation: The encoder in the semantic generation part aims to learn the semantic information and common knowledge [10] [11]. Rich semantic information can help the model generate candidate characters more accurately. This paper adopts BERT model as the base model and conduct unsupervised pretraining using a large amount of data. To ensure compatibility with the Chinese Chinese Correction task, excessively long or short sentences are filtered and processed. Each news article is segmented into individual sentences, with sentence lengths ranging from 4 to 256. The masking strategy used differs from the MASK masking strategy of the BERT model. In this strategy, 15% of randomly selected characters in the input text are replaced, where 10% of the characters remain unchanged, 20% are replaced with visually similar characters, 50% are replaced with phonetically similar characters, and 20% are replaced with MASK tokens. The pre-trained

BERT model is capable of generating hidden representations $X = \{x_1, x_2, \ldots, x_n\}$ containing semantic information for the input text sequence $H^s = \{h_1^s, h_2^s, \ldots, h_n^s\}$.

$$H^s = \{h_1^s, h_2^s, \ldots, h_n^s\} = BERT(x_1, x_2, \ldots, x_n) \quad (1)$$

where $H^s \in \mathbb{R}^{N\times 768}$,$N$ is the sentence length, $h_n^s \in \mathbb{R}^{768}$.

2) Pinyin Enhance: DORM [12] mentions that the phonetic similarity mainly depends on their initial consonants or final vowels, rather than the tones. Therefore, we focus only on the initial consonants and final vowels as the phonetic features of Chinese characters. In this paper, we only rely on a simple convolutional operation to model the pinyin information of the model, eliminating the need for additional pretraining to incorporate pinyin information. First, we construct a Chinese character pinyin vocabulary and establish a mapping between the Chinese characters and their corresponding pinyin using the BERT Chinese character vocabulary. Based on this mapping, the input Chinese character x_i is mapped to the pinyin vector space E^p. We initialize the pinyin embedding e_i^p for character x_i in the model using this mapping.

$$e_i^p = E^p(x_i) \quad (2)$$

where $E^p \in \mathbb{R}^{n_{voc}^p \times 768}$, n_{voc}^p represent the size of the pinyin vocabulary, and x_i represent the $i-th$ character.

Modeling the correction sentence solely based on simple pinyin embeddings e_i^p can introduce significant ambiguity, as one Chinese pinyin may correspond to multiple Chinese characters. To address this issue, we encode consecutive pinyin embeddings by passing them through convolutional layers, resulting in hidden layer representations h_i^p that incorporate pinyin information. This approach of using multiple consecutive pinyin embeddings helps greatly reduce the ambiguity in modeling the correction sentence.

$$h_i^p = Conv(e_{i-1}^p, e_i^p, e_{i+1}^p) \quad (3)$$

where $h_i^p \in \mathbb{R}^{768}$.

3) Glyph Encoder: During pretraining, we adopted a simple yet effective residual network, ResNet [13], as the backbone for extracting glyph features. To capture the image features of characters as much as possible, we select three fonts: Simplified Chinese, Regular Script (Kaiti), and Small Seal Script (Xiaozhuan). They correspond to the three channels of character images. Character images are obtained using image tools from predetermined font files, with a size of 32×32 pixels. Unlike the fine-tuning training phase, after obtaining the final hidden layer representation, a CLS layer is applied for character prediction and cross-entropy loss is calculated with the input characters. After pretraining, we encoded the character images during the fine-tuning phase using a ResNet5 model, which consists of 5 ResNet blocks, and then a normalization layer is applied to obtain the final representation h_i^g, which integrates the visual information. The formula is as follows:

$$\widetilde{h}_i^g = ResNet5(e_i^g) \quad (4)$$

$$h_i^g = LayerNorm(\widetilde{h}_i^g) \quad (5)$$

where e_i^g represents the image embedding of the $i-th$ character in the sentence, $LayerNorm$ represents the layer normalization operation, and $h_i^g \in \mathbb{R}^{768}$.

4) Multi-modal Fusion and Candidate Generation: In this stage, the three components have already generated hidden layer representations, which incorporate the fused multi-modal information of the input x_i. The structure is shown in Fig. 2, where the hidden layer representations of the three modalities are stacked together and then passed through a normalization layer. Finally, the predictions are generated by a linear layer, as shown in the following equation:

$$h_i = LayerNorm(h_i^s + h_i^p + h_i^g) \quad (6)$$

$$y_{i,m} = h_i \omega_i \quad (7)$$

where h_i^s,h_i^p and h_i^g represent the hidden layer representations of the semantic, pinyin, and glyph modalities, respectively, for the input character x_i. h_i represents the final hidden layer representation obtained by combining the hidden layer representations of the three modalities. $y_{i,m}$ represents the $m-th$ candidate generation result for the $i-th$ word in the input sentence, and ω_i is a trainable weight vector.

Considering all the parts described above, the task of candidate generation in the candidate generation module can be modeled by the following equation:

$$y_i = f(x_i) \quad (8)$$

where $f(x_i)$ represents the mapping relationship between the $i-th$ character and its corresponding candidate characters at the respective position, and $y_{i,m}$ represents the candidate character at the $i-th$ position.

C. Candidate Recall Module

As shown in Fig. 2, the scores for candidate characters at each position are obtained in Candidate Generation Module. The Candidate Recall Module calculates the coherence scores between characters.

1) Copy Mechanism: Firstly, in the Candidate Generation Module introduced in the previous section, the model integrates the semantic, pinyin, and glyph modal information of characters to generate a series of final candidate characters. At each position in the sentence, the number of candidate characters generated is k. In the case of a sentence length of N, the generated candidate character matrix is denoted as $C \in \mathbb{R}^{N\times k\times 768}$. To accommodate the copying mechanism, we directly incorporate the original embeddings of the input characters $e = \{e_1, e_2, \ldots, e_N\}$, where $e \in \mathbb{R}^{N\times 768}$, into the candidate character matrix C, creating a special form of copying mechanism. In this case, the number of candidate characters at each position becomes k+1, resulting in the final candidate character matrix $C' \in \mathbb{R}^{N\times(k+1)\times 768}$.

2) Candidate Selection: The coherence score between characters needs to take into account contextual information, specifically the dependency relationship between two adjacent characters. We observe that the attention is effective in learning

the strength of dependency between characters. Because attention mechanisms are highly sensitive to positional information and the positional relationship between adjacent characters needs to be modeled, our Attention mechanism not only includes the final hidden layer representations of two adjacent characters but also incorporates their positional encoding. Therefore, we incorporate the corresponding positional embeddings into the hidden layer representations of the two adjacent characters as Query in the attention mechanism. The specific formula is as follows:

$$h_{i,m}^{'} = h_{i,m} + h_{i,m}^{pos} \tag{9}$$

$$h_{i+1,n}^{'} = h_{i+1,n} + h_{i+1,n}^{pos} \tag{10}$$

$$K_i = V_i = [h_{i,m}^{'}, h_{i+1,n}^{'}] \tag{11}$$

$$Q_{i,m} = h_{i,m}^{'} \tag{12}$$

$$Q_{i+1,n} = h_{i+1,n}^{'} \tag{13}$$

$$p_{i,m} = Attention(Q_{i,m}W^Q, K_iW^k, V_iW^v) \tag{14}$$

$$q_{i+1,n} = Attention(Q_{i+1,n}W^Q, K_iW^k, V_iW^v) \tag{15}$$

where i represents the character position in the sentence, i.e., the $i-th$ character in the sentence. m and n are the indices of the candidate characters, i.e., the $m-th$ and $n-th$ candidate characters at position i. Q, K and V represent the Query, Key, and Value in the attention mechanism, respectively. W^Q, W^k and W^v are learnable parameter matrices in the attention mechanism. $h_{i,m}$ and $h_{i+1,n}$ represent the hidden layer representations outputted by the candidate generation module. $h_{i,m}^{pos}$ and $h_{i+1,n}^{pos}$ represent the positional embeddings generated based on the BERT model for the characters. $h_{i,m}^{pos} \in \mathbb{R}^{768}$.$h_{i+1,n}^{pos} \in \mathbb{R}^{768}$.

We pass the output of the attention mechanism through a normalization layer to obtain two representations, $p_{i,m}^{'}$ and $q_{i+1,n}^{'}$. Then, we concatenate these two representations and feed them into an MLP layer. Finally, we use a linear layer to calculate the final coherence score between the characters.

$$p_{i,m}^{'} = LayerNorm(p_{i,m}) \tag{16}$$

$$q_{i+1,n}^{'} = LayerNorm(q_{i+1,n}) \tag{17}$$

$$s = MLP(p_{i,m}^{'}, q_{i+1,n}^{'}) \tag{18}$$

$$s^{'} = g(y_{i,m}, y_{i+1,n}) = s\omega^{'} \tag{19}$$

where $\omega^{'}$ is a learnable parameter matrix, and $g(y_{i,m}, y_{i+1,n})$ represents the dependency score between the $m-th$ candidate character at position i and the $n-th$ candidate character at position i+1 in the sentence. $\omega^{'} \in \mathbb{R}^{768\times n_{voc}}$, n_{voc}represent the size of the vocabulary.

According to the above description, the candidate recall module combines the top-k candidate characters generated by the candidate generation module with the embeddings of the original input characters to form a candidate character matrix. The matrix has a sentence length of N and the number of candidate characters at each position is (top-k)+1. The model ultimately generates N×(k+1) candidate words, N×(k+1) scores for each candidate word, and $(k+1)^N$ candidate paths. The final score of each path can be modeled using the following formula:

$$S(X,Y) = \sum_{i=1}^{N} f(x_i) + \sum_{i=1}^{N-1} g(y_{i,m}, y_{i+1,n}) \tag{20}$$

where x_i represents the original input character at position i, $y_{i,m}$ and $y_{i+1,n}$ represent the $m-th$ and $n-th$ candidate characters at position i and i+1, and $f(x_i)$ represents the candidate score outputted by the candidate generation module after integrating the three modalities.

TABLE I
THE PERFORMANCE OF NON-PRETRAINED MCSSPELL AND PRETRAINED MCSSPELL(MCSSPELL*) WAS COMPARED WITH BASELINE MODELS ON THE SENTENCE-LEVEL VALIDATION SETS OF SIGHAN13(CSC_{13}), SIGHAN14(CSC_{14}), AND SIGHAN15(CSC_{15}). D REPRESENTS THE DETECTION TASK, C REPRESENTS THE CORRECTION TASK, AND P, R, AND F REPRESENT PRECISION, RECALL, AND F1 SCORES, RESPECTIVELY. THE BASELINE MODEL'S EXPERIMENTAL RESULTS ARE DIRECTLY TAKEN FROM THE PAPER. FOR CONVENIENCE, WE HAVE REMOVED THE PERCENTAGE (%) SYMBOLS FROM THE DATA IN THE TABLE.

Dataset	Model	Detection-level			Correction-level		
		D-P	D-R	D-F	C-P	C-R	C-F
CSC_{13}	GAD	85.7	79.5	82.5	74.9	78.7	81.6
	DCN	86.8	79.6	83.0	84.7	77.7	81.0
	REALISE	88.6	82.5	85.4	87.2	**81.2**	84.1
	MDCSpell	89.1	78.3	83.4	87.5	76.8	81.8
	MCSSpell	89.2	82.3	85.6	87.4	80.5	83.8
	MCSSpell*	**89.7**	**82.9**	**86.1**	**87.9**	81.1	**84.4**
CSC_{14}	GAD	66.6	**71.8**	69.1	65.0	70.1	67.5
	DCN	67.4	70.4	68.9	65.8	68.7	67.2
	REALISE	67.8	71.5	69.6	66.3	70.0	68.1
	MDCSpell	70.2	68.8	69.5	**69.0**	67.7	68.3
	MCSSpell	69.8	70.8	70.3	67.1	69.8	68.4
	MCSSpell*	**70.3**	71.1	**70.7**	67.9	**70.4**	**69.1**
CSC_{15}	GAD	75.6	80.4	77.9	73.2	77.8	75.4
	DCN	77.1	80.9	79.0	74.5	78.2	76.3
	REALISE	77.3	81.3	79.3	75.9	79.9	77.8
	MDCSpell	**80.8**	80.6	80.7	78.4	78.2	78.3
	MCSSpell	80.1	82.2	81.1	78.2	79.6	78.9
	MCSSpell*	80.7	**83.1**	**81.9**	**78.8**	**80.4**	**79.6**

D. Learning

The probability distribution of the output sequence Y can be obtained using the following formula:

$$p(Y|X) = \frac{e^{S(X,Y)}}{\sum_{Y_i^p} e^{S(X,Y_i^p)}} \tag{21}$$

where Y_i^p are the path generated by the candidate characters. The loss function is the maximum likelihood function of the probability distribution, given by the following formula:

$$\mathcal{L} = -\log(p(Y|X)) \tag{22}$$

IV. EXPERIMENT AND RESULTS

A. Dataests and Parameter Settings

According to previous work, we also utilized the open-source Hybird [14], SIGHAN13 [15], SIGHAN14 [17], and

TABLE II
ABLATION RESULTS OF THE MCSSPELL MODEL AVERAGED ON SIGHAN TEST SETS. WE APPLY THE FOLLOWING CHANGES TO MCSSPELL: REMOVING -GLYPH ENCODER(-GLYPH), REMOVING PINYIN ENHANCE(-PINYIN), REMOVING PINYIN ENHANCE AND GLYPH ENCODER(-PINYIN& GLYPH), REMOVING ONLY THE COPY MECHANISM IN CCOPS (-COPY) AND REMOVING ALL CCOPS (-CCOPS). FOR CONVENIENCE, WE HAVE REMOVED THE PERCENTAGE (%) SYMBOLS FROM THE DATA IN THE TABLE.

Model	Detection-level			Correction-level		
	D-P	D-R	D-F	C-P	C-R	C-F
MCSSpell-pretrain(ours)	**80.7**	**83.1**	**81.9**	**78.8**	**80.4**	**79.6**
-Glyph	80.1	82.4	81.2	78.1	80.1	79.1
-Pinyin	79.8	82.2	81.0	77.9	79.8	78.8
-Pinyin&Glyph	79.5	81.7	80.6	77.1	79.4	78.2
-Copy	80.3	82.7	81.5	78.5	79.6	79.0
-CCOPS	77.7	81.8	80.8	77.8	78.7	78.2
REALISE (baseline)	77.3	81.3	79.3	75.9	79.9	77.8

SIGHAN15 [5] as training data for the fine-tuning stage. The test data from SIGHAN13, SIGHAN14, and SIGHAN15 were used as validation data. We used the pre-trained large-scale corpora wiki2019zh and news2016zh, which consist of 1 million Wikipedia articles and 2.5 million news articles.

We used the AdamW optimizer with learning rates of {2e-5, 3e-5, 5e-5} and conducted a series of experiments using 8 RTX 3090 GPUs. The training batch size is set to {32, 64} and evaluation batch size is set to {16, 32}. During the semantic pre-training phase, we set the batch size to 64, trained for 6 epochs, and used a learning rate of 5e-5. The number (k) of candidate characters generated in the candidate generation module is set to 6.

B. Main results

The experimental results, as shown in Table 1, demonstrate that our proposed model achieves good performance on the traditional SIGHAN datasets. After pretraining on Wikipedia and news data, MCSSpell shows further improvement. This demonstrates that pretraining the model using a special masking approach indeed enhances its effectiveness in the correction task. It suggests that having more data or employing more appropriate pretraining can be beneficial for further improving the model's performance.GAD [16] proposed a global attention mechanism. Our approach outperformed GAD, achieving improvements of 4% and 4.2% in detection and correction, respectively. This further confirms the effectiveness of our method in CSC. When DCN [18] fuses input information, it may overlook the information of glyph shapes. MCSSpell achieved improvements of 2.9% and 3.3% in detection and correction, respectively, demonstrating the effectiveness of incorporating multimodal information in MCSSpell. REALISE [19] considered three modalities of information and performed fusion but did not optimize the dependency relationship among candidate characters. MCSSpell achieved improvements of 2.6% and 1.8% respectively, demonstrating the effectiveness of our model in addressing the coherence of candidate characters. MDCSpell [10] proposed a new Detector-Corrector framework, as using BERT alone as a corrector is not sufficient for extracting multimodal information from text. MCSSpell achieved improvements of 1.2% and 1.3% in detection and correction tasks, respectively, demonstrating the effectiveness of our approach.

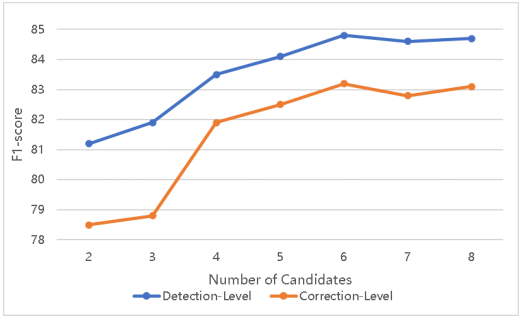

Fig. 3. The impact of the number of top-K candidate characters on the model performance.

C. Abaltion Study

In this section, we mainly analyzed the effectiveness of multi-modal information fusion and candidate recall module, and validated our conclusions on the SIGHAN15 dataset. As shown in Table 2, when glyph information is removed, the F1 scores of the detection and correction tasks decrease by 0.7% and 0.5%, respectively. When phonetic information is removed, the F1 scores of the detection and correction tasks decrease by 0.9% and 0.8%, respectively. When both phonetic and glyph information are removed simultaneously, the F1 scores of the detection and correction tasks decrease by 1.3% and 1.4%, respectively. Regardless of which modality of information is removed, the model's performance decreases. This demonstrates the effectiveness of phonetic and glyph information in Chinese error correction tasks.

Additionally, as shown in Table 2, after removing all CCOPS and predicting candidate characters only through multi-modal fusion, MCSSpell achieves higher F1 scores than the baseline model in both the detection and correction tasks. This demonstrates the effectiveness of our multi-modal fusion mechanism, which may be attributed to the richer corpus and more effective masking strategy in our pre-training component.

To address the character dependency issue among candidate characters, we propose CCOPS in the candidate recall module, aiming to allow the model to select better neighboring characters for pairing. As shown in Table 2, under the condition of removing all CCOPS, the model's F1 scores in the detection and correction tasks decrease by 1.1% and 1.4%, respectively. This demonstrates the importance of character dependency in the CSC task, and CCOPS can alleviate this problem. We also observe that CCOPS has a more significant impact on the correction task, possibly because after detecting the erroneous character, the coherence score between characters helps the model choose a more correct character combination. Additionally, under the condition of removing only the copy mechanism of CCOPS, the model's F1 scores decrease by 0.4% and 0.6%, respectively. This confirms the effectiveness of the copy mechanism in mitigating the issue of over-correction

by the BERT model.In the Candidate Generation Module, at each position in the sentence, K candidate characters are generated in a top-k manner. We conducted experiments to investigate the impact of the number of generated candidate characters (k) on the model's performance. As shown in Fig. 3, as the number of candidate characters increases, the model's performance improves significantly. However, once the number of candidate characters reaches 6, the performance becomes stable without significant improvement. Considering that a high number of candidate characters can lead to increased computational complexity, in other experiments, we set K to 6 to strike a balance between performance and computational efficiency.

V. Conclusion

This paper addresses the task of CSC and proposes a novel candidate character optimal path selector to solve the issue of incoherent dependencies between characters. To model the semantic, phonetic, and visual information of characters, we introduce a candidate generation module that integrates the multi-modal information of characters. To alleviate the problem of excessive correction by the BERT model, we incorporate a unique copy mechanism into the candidate character optimal path selector, forming a novel candidate recall module. Experimental results demonstrate that MCSSpell achieves state-of-the-art performance on three authoritative datasets for CSC, confirming the effectiveness of our approach. In future work, we will further investigate the applicability of MCSSpell in other domains, such as Chinese Grammar Correction.

Acknowledgment

This work is supported by R&D Program of Beijing Municipal Education Commission (KM202210009002), the Beijing Urban Governance Research Base of North China University of Technology(2023CSZL16), the National Natural Science Foundation of China (61972003), National Key Research and Development Program of China (2020AAA0109700) and National Natural Science Foundation of China (62076167). We would also like to thank the anonymous reviewers for their helpful comments. We would like to thank the referees for their comments, which helped improve this paper considerably.

References

[1] Jianfeng Gao, Xiaolong Li, Daniel Micol, Chris Quirk,and Xu Sun, "A large scale ranker-based system for search query spelling correction," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China, pp. 358–366,2010.

[2] Fei Dong and Yue Zhang, "Automatic features for essay scoring–an empirical study," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1072–1077, 2016.

[3] John D. Lafferty, Andrew McCallum, and FernandoC. N. Pereira, "Conditional random fields:Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, Morgan Kaufmann, Williams College, Williamstown, MA, USA, pp. 282-289, June 28-July 1,2001.

[4] Yang, Shoujian, and Lian Yu, "CoSPA: An improved masked language model with copy mechanism for Chinese spelling correction," in *Uncertainty in Artificial Intelligence*, PMLR, pp. 2225-2234,2022.

[5] Yuen-Hsien Tseng, Lung-Hao Lee, Li-Ping Chang, and Hsin-Hsi Chen, "Introduction to SIGHAN2015 bake-off for Chinese spelling check," in *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, Association for Computational Linguistics, Beijing,China, pp. 32–37,2015.

[6] Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu, "FASPell: A fast, adaptable, simple, powerful Chinese spell checker based on DAEdecoder paradigm." in *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, Association for Computational Linguistics, Hong Kong, China. pp. 160–169, 2019.

[7] Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, andYuan Qi, "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*, pp. 871–881, 2020.

[8] Li Huang, Junjie Li, Weiwei Jiang, Zhiyu Zhang, Minchuan Chen, Shaojun Wang, and Jing Xiao, "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*, Online, Association for Computational Linguistics, vol. 1, pp. 5958–5967, 2021.

[9] Shulin Liu, Shengkang Song, Tianchi Yue, Tao Yang, Huihui Cai, TingHao Yu, and Shengli Sun, "CRASpell: A Contextual Typo Robust Approach to Improve Chinese Spelling Correction," in *Findings of the Association for Computational Linguistics: ACL*, pp. 3008-3018, May, 2022.

[10] Li Jing, Gaosheng Wu, Dafei Yin, HaoZhao Wang and Yonggong Wang, "MDCSpell: A Detector-Corrector Framework for Chinese Spelling Error Correction," DOI: 10.18653/v1/2022.findings-acl.98, 2021.

[11] Yinghui Li, Qingyu Zhou, Yangning Li, Zhongli Li, Ruiyang Liu, Rongyi Sun, Zizhen Wang, Chao Li, Yunbo Cao and Hai-Tao Zheng, "The past mistake is the future wisdom: Error-driven contrastive probability optimization for Chinese spell checking," [Online], arXiv:2203.00991, 2022.

[12] Liang, Zihong, Xiaojun Quan, and Qifan Wang, "Disentangled Phonetic Representation for Chinese Spelling Correction," [Online], arXiv:2305.14783, 2023.

[13] Chao-Huang Chang, "A new approach for automatic chinese spelling correction," in *Proceedings of Natural Language Processing Pacific Rim Symposium*, Citeseer, vol. 95, pp. 278–283,1995.

[14] Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang, "A hybrid approach to automatic corpus generation for chinese spelling check," in *Proceedings ofthe 2018 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Brussels, Belgium, pp. 2517–2527, October 31 - November 4, 2018.

[15] Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee, "Chinese spelling check evaluation at SIGHAN bake-off 2013," in *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, SIGHAN@IJCNLP 2013, Asian Federation of Natural Language Processing, Nagoya, Japan, pp. 35–42, October 14-18, 2013.

[16] Zhao Guo, Yuan Ni, KeqiangWang, Wei Zhu, and Guotong Xie, "Global attention decoder for Chinese spelling error correction," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1419–1428, 2021.

[17] Junjie Yu and Zhenghua Li, "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*, pp.220–223,2014.

[18] Baoxin Wang, Wanxiang Che, Dayong Wu, Shijin Wang, Guoping Hu, and Ting Liu, "Dynamic connected networks for Chinese spelling check," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2437–2446,2021.

[19] Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, Heyan Huang, and XianLing Mao, "Read, listen, and see: Leveraging multimodal information helps Chinese spell checking," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 716–728,2021