Revisiting the Knowledge Recall and Selection in Chinese Spelling Correction

Anonymous ACL submission

Abstract

Chinese Spelling Correction (CSC) task is very challenging in the natural language processing area. However, the performance improvement is quite limited, primarily because the infusion of knowledge is limited. Previous work involved confusion sets as additional knowledge, but the size was too small and served only as a role of additional feature. To address this, we propose a knowledge recall and selection network (ReSC). First through four recall methods to achieve an average recall rate above 93%, with individual character recall of around 150 related characters/words. Subsequently, we proposed a Knowledge Selection Algorithm, choosing the appropriate characters or words from numerous recall sets. The knowledge selection network is highly efficient, as the F1 score nearly reached 100%. Extensive experiments have proven ReSC is able to inject substantial amount of entities with even a lower False Positive Rate. This novel network acheves the new SOTA results across three domain-specific datasets.

1 Introduction

The field of Chinese Spelling Correction (CSC) has always been a crucial foundational task in natural language processing (NLP) with applications across various areas. Such as web search (Martins and Silva, 2004), speech recognition (Chen et al., 2021), and machine translation (Zhou et al., 2019).

Historically, the SOTA approaches in CSC have favored rephrasing methods over tagging methods (Liu et al., 2023a; Wu et al., 2023). Research has sufficiently demonstrated the limitations of tagging-based methods, whereas models tend to memorize error correction patterns rather than understanding the sentence intrinsically to perform correction. In rephrasing, however, there is a limitation due to the lack of information supplementation, which has led to the restricted expressiveness of methods like ReLM (Liu et al., 2023a).

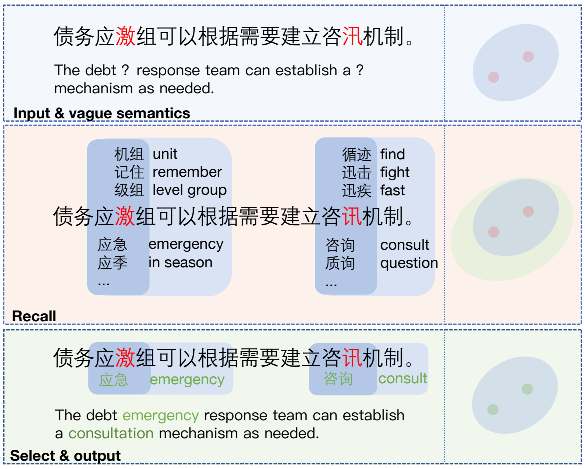


Figure 1: Example of human spelling correction. Misspelled characters are indicated in red, and the correct ones are in green. Ambiguous semantics refers to the interpretative process undertaken by humans, who then think of potential candidates before determining the correct term. The right part is an illustration of the human CSC process.

As indicated in Figure 1, the ReLM model merely simulates the process of human semantic understanding, but it does not include the capability for knowledge retrieval. Therefore, incorporating a knowledge recall and selection mechanism is crucial.

Another focal point is the confusion set (Liu and Cao, 2016), a collection of words or characters that are often mistakenly used interchangeably due to their similar appearances or pronunciations, this set can provide potential candidates for correction. Merely introducing confusion sets does not clarify which candidates are useful and which are not, these candidates could act as noise and have a detrimental effect. In human error correction in Figure 1, the process should understand first, search for knowledge and then filter it.

Additionally, since there is no filtering function after the introduction of the confusion set (Cheng et al., 2020; Guo et al., 2021), its size will not be large, which directly determines the upper limit of the recall rate. In other words, introducing more candidate sets will lead to a greater extent of knowledge recall.

To address the above issues, from a high-level perspective, CSC requires a recall and selection model. Given input sentence X, candidate sets C, output sentence Y, from the derivation of Appendix C, we have:

$$P(Y|X) \propto \underbrace{P(C|X)}_{\text{Recall}} \cdot \underbrace{P(Y|X,C)}_{\text{Selection}} \tag{1}$$

where the recall model decides the upper bound of knowledge injection, thus we utilize four recall methods to achieve this, including phonetic (pinyin) matching, four-corner code matching, radical matching, and similar shapes matching. Specifically, we perform a trie tree retrieval on character level one by one, searching for related characters/words.

Subsequently, the knowledge selection network performs granular filtering of the recall sets on a per-character level. To enhance the language model's ability to discern the relationship between potential candidates and erroneous words, we have developed a confidence mechanism. This approach entails training the network to acknowledge a candidate as correct only if its association with the candidate word surpasses its association with the original word. The selection network has demonstrated a significant learning effect, with F1 approaching nearly 100%.

Our contributions can be summarized as follows:

1. Broad Recall: To our knowledge, this is the first paper to utilize such an extensive recall set for domain-CSC tasks. It achieves a recall rate exceeding 93%, with single-character recall exceeding 150 characters/words.

2. Ease of Use: Despite employing a four-way recall, we significantly reduce recall time complexity using trie search plus a segmentation-free approach. The selection is lightweight, which facilitates its application to other networks.

3. SOTA results: Our model demonstrates impressive performance, achieving SOTA results across three datasets. There was an average improvement of 3.36% on domain-specific datasets.

2 Method

2.1 Problem Formulation

The Chinese Spelling Correction (CSC) task aims to identify and correct spelling errors in Chinese text. In the context of CSC, character alignment is essential, as it refers to mapping each character in the erroneous input sequence to the corrected character in the output sequence.

Formally, the task can be described as follows: Given an erroneous input sequence $X = \{x_1, x_2, ..., x_n\}$ of n Chinese characters, the objective is to generate a corrected output sequence $Y = \{y_1, y_2, ..., y_n\}$, ensuring that each character x_i from the input is correctly aligned with the corresponding character y_i in the output. Unlike previous work utilizing only character-level candidates (Guo et al., 2021; Cheng et al., 2020), we amalgamated character and word information to augment the model's expressive capacity for the CSC task. The character candidates of x_i are defined as $[char_{i1}, char_{i2}, \ldots]$ and the word candidates of x_i are defined as $[word_{i1}, word_{i2}, \ldots]$. Then use $cand_i$ to represent the collection of character and word candidates about x_i.

2.2 Framework

To maximize recall, we employed multiple recall techniques. After this, we utilized a Knowledge Selection Network to assess the validity of the candidate. Furthermore, the training of the Knowledge Selection Network is necessary. And employing a cross-entropy to constrain the accuracy of the attention softmax. For a detailed description, refer to the Figure 2.

2.3 Knowledge Recall

This process can be expressed as $P(C|X)$, where C represents the recall set for the entire sentence. Unlike previous work (Song et al., 2023), our recall process excludes word segmentation because if there are errors in the sentence, the segmentation result is very likely to be incorrect as well.

To ensure a higher recall rate, we utilize similar pinyin, similar four-corner codes, similar radicals, and shape-similar for candidates' recall. First, we build a trie search tree based on these four features. When features key match, candidates are recalled. For example, in the case of Figure 2, based on the radical " 远", we first search the trie tree to find all characters with the radical " 辶". Then, if " 辶 辶" still exists in the trie tree, we retrieve all words

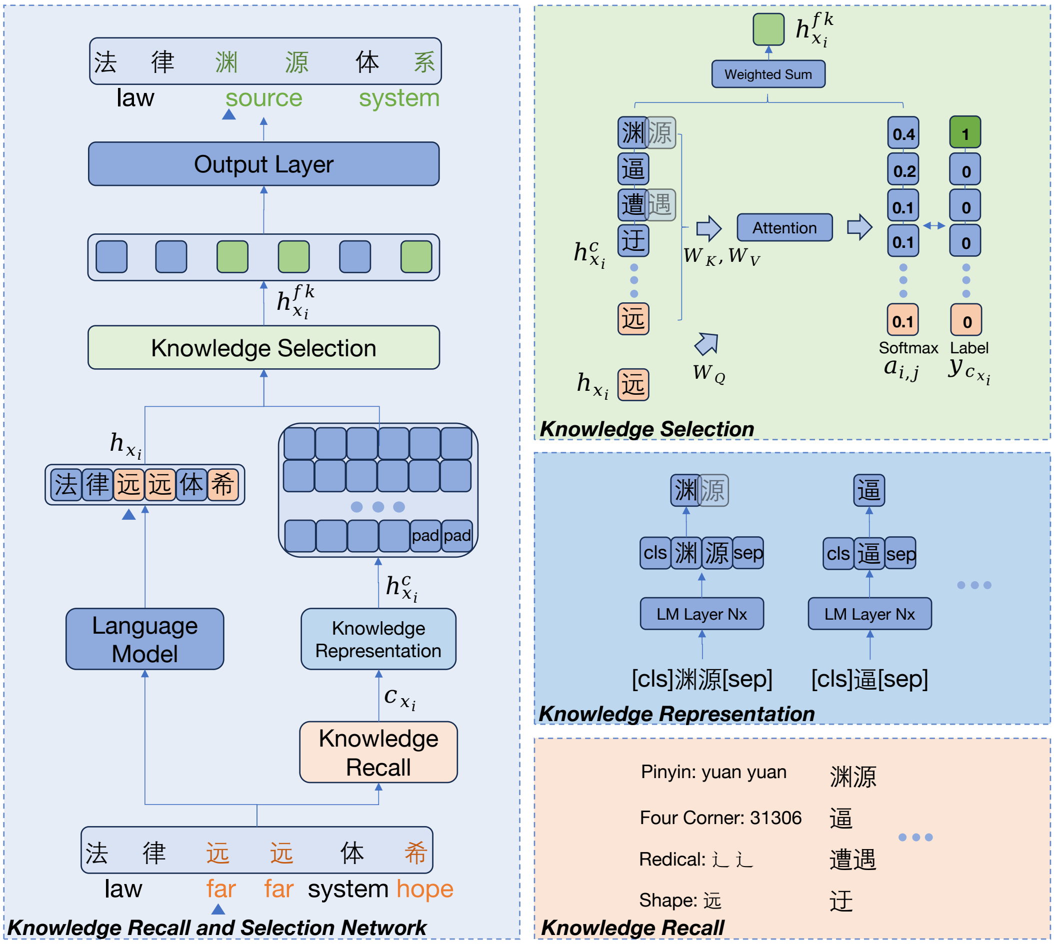


Figure 2: An overview of Knowledge Recall and Selection Network. The left side describes the overall error correction process. In contrast, the right side mainly elaborates on the character "远", which involves knowledge recall, then knowledge representation, followed by knowledge selection.

with the radical " 辶辶", and stop since there are no words with radical "辶辶亻". Detailed recall methods are in Appendix B.

2.4 Knowledge Selection Network

Knowledge Representation In previous work, character embeddings were often employed to form candidate set vectors, which encapsulate semantic information but lack correction-related insights. For instance, the embedding of "已"(meaning already) and "巳" (meaning fetus) have entirely different meanings, thus it is difficult to view the similarity in correction level for the pre-trained model, as shown in Appendix D Figure 5. Thus if we want the distance between "已" and "巳" to be small, embedding is not a good choice.

So our candidate representation is directly from the last layer from LM, as it contains more correction-level information compared to the first layer. Another reason is its capability to produce word vectors that project on individual characters, thanks to self-attention. For example, in Figure 2 Knowledge Representation part, we use the latent vector corresponding to "渊" to represent the intrinsic meaning of the word "渊源". The representation of c_{ij} are represented as $h_{c_{ij}}$.

Knowledge Selection Model The selection of knowledge directly determines the model's error correction capability. Our approach employs attention mechanisms to facilitate this. However, we must account for scenarios where the model fails to successfully retrieve candidates. To address this, we include the original input h_{x_i} in the composition of keys and values, allowing the model to learn a stronger correlation with itself in the absence of viable candidates. Conversely, if the recall set contains appropriate candidates, the model is trained to prioritize the correction of characters, potentially even over the score of the original x_i.

Formally, we construct the candidate set c_{x_i} from Section 2.3, candidate representations $h^c_{x_i} \in$

$\mathbb{R}^{N\times d}$ with fixed length N using the knowledge representation.

$$c_{x_i} := \{x_i; cand_{i1}, cand_{i2}, \dots\}. \quad (2)$$

$$h^c_{x_i} := \{h_{x_i}; h_{cand_{i1}}, h_{cand_{i2}}, \dots\}. \quad (3)$$

Then, through an attention network, it is calculated to determine whether one of the current candidates can serve as a correct error correction

$$a_{i,j} = \frac{\exp(W_Q h_{x_i} \cdot W_K h^{c_j}_{x_i})}{\sum_j \exp(W_Q h_{x_i} \cdot W_K h^{c_j}_{x_i})} \quad (4)$$

where $W_K, W_Q \in \mathbb{R}^{d\times d}$ are learnable projection matrics, it is noteworthy that the attention weights $\{a_{i,j}\}_{j=1}^N$ induces a knowledge selection model $P_{KS}(c^j_{x_i}|h_{x_i}, h^c_{x_i})$. Thus we can learn the parameters W_K and W_Q via the following knowledge selection loss:

$$\mathcal{L}_{KS} := \frac{1}{N}\sum_i \sum_j y_{c_{x_i}} \log P_{KS}(c^j_{x_i}|h_{x_i}, h^c_{x_i}) \quad (5)$$

note that $c^j_{x_i}$ belongs to c_{x_i}, and $y_{c_{x_i}}$ is a one-hot label for a true candidate with length N. Besides if the candidate set does not include the ground truth label, we take the original word x_i as the true label to calculate the cross entropy in (5).

Spelling Correction Model. Our spelling correction model is on top of the knowledge selection model. Specifically, we construct the fused knowledge representation through a weighted sum between the knowledge representations in (3) and attention weights in (4)

$$h^{fk}_{x_i} := \lambda_{fk} \sum_j a_{i,j} W_V h^{c_j}_{x_i} + (1-\lambda_{fk}) h_{x_i} \quad (6)$$

$W_V \in \mathbb{R}^{d\times d}$ is a learnable parameter. And λ_{fk} is the parameter for fusing knowledge. Finally, the spelling correction model $P_{SC}(y_i|x_i)$ is defined as the following softmax probability:

$$P_{SC}(y_i|x_i) := \text{softmax}(W_O h^{fk}_{x_i}) \quad (7)$$

Where $W_O \in \mathbb{R}^{|\mathcal{V}|\times d}$ is the output layer, and $\mathcal{V}$ means vocabulary size. We train the parameters W_V and W_O through the following spelling correction loss:

$$\mathcal{L}_{SC} := \sum_i y_i \log P_{SC}(y_i|h^{fk}_{x_i}) \quad (8)$$

In practice, our final loss function is defined as

$$\mathcal{L} = (1-\lambda_{KS})\mathcal{L}_{SC} + \lambda_{KS}\mathcal{L}_{KS} \quad (9)$$

During the inference process, it is possible to apply either the knowledge selection model P_{KS} or the spelling correction model P_{SC} to do Chinese spelling correction. In practice, we observe that P_{SC} has better performance. To this end, we mainly report our results using P_{SC} and leave the study of P_{KS} in Section 4.2.

2.5 Special Cases

Nested character and word For example, if “渊源” and “渊” are both in the recall set of “远”. During training, we prefer “渊源” better than “渊” since it captures more information. Thus the training objection of this selection should be “渊源”.

Nested words For instance, say we retrieve “渊源” for the first “远”, and “源体” for the second “远”. Despite the apparent overlap, it doesn't affect our knowledge selection since it's based on individual characters. We just need to update the network to correct the first “远” to “渊源” and the second to “源体”. Our model is based on the encoder structure, where such overlaps are manageable, unlike in the decoder, which would cause series issues.

3 Experiment

3.1 Dataset

ECSPell Introduced by (Lv et al., 2023) in 2022, it stands as a domain-specific benchmark for Chinese Spelling Correction (CSC), featuring three distinct sectors: legal (LAW); medical (MED); official document composition (ODW). The statistics are in Table 1. Each domain is meticulously curated to reflect the unique linguistic challenges and terminologies inherent to their respective fields. For a fair comparison, the domain dictionary stays the same as Rspell (Song et al., 2023).

SIGHAN Follow previous work (Guo et al., 2021; Lv et al., 2023; Cheng et al., 2020; Wu et al.,

	data	# Train	# Test
SIGHAN	SIGHAN13	350	1000
	SIGHAN14	3437	1062
	SIGHAN15	2338	1100
	Wang27k	271,329	0
ECSpell	LAW	1960	500
	MED	2500	500
	ODW	1728	500

Table 1: The statistics of the ECSpell and Sighan dataset, # Train and # Test represent the number of train sentences and test sentences. Wang27k represents a large generated CSC dataset from (Wang et al., 2018).

2023), we also compare result on SIGHAN13, SIGHAN14, and SIGHAN 15. The statistics are in Table 1. For a fair comparison, the confusion set is the same as (Cheng et al., 2020). Since its set is character level, so we only have character level result ReSC$_{char}$.

3.2 Baseline Approaches

Masked-Fine-Tuning (MFT) It utilizes a simple mask technique for characters during CSC task training, which brought a good result for BERT based model (Liu et al., 2023b).

BERT We directly fine-tune the BERT model with the MFT trick.

Baichuan2 We finetune Baichuan2, one of the famous Chinese Large Language Model (LLM). We use the MFT technique to get better results.

ChatGPT We implement ChatGPT to do CSC tasks using OpenAI API.

MDCSpell It is an enhanced BERT-based model proposed by (Zhu et al., 2022a). Based on a detector-corrector approach, this model tries to retain the crucial visual and phonological cues of misspelled characters.

ReLM The Rephrasing Language Model (ReLM) (Liu et al., 2023a) takes a rephrasing approach to Chinese Spelling Correction by rephrasing whole sentences for error correction, rather than the basic tagging method. During pre-training, there is another auxiliary task where it randomly substitutes tokens with incorrect characters and then corrects these artificial errors.

RSpell It is a retrieval-augmented framework for CSC tasks that enhances domain-specific error correction by integrating relevant domain terms through a pinyin fuzzy confusion set. It features an adaptive control mechanism to tailor the influence of this external knowledge and an iterative strategy that boosts correction capabilities (Song et al., 2023).

ECSpell^{UD} Introduced by (Lv et al., 2023), it is an Error-consistent masking strategy for data generation during pretraining. This strategy ensures that the types of errors found in the automatically generated sentences are representative of those encountered in actual usage. ECSpell^{UD} features a User Dictionary guided inference module (UD), which is affixed to a general token classification-based speller.

SpellGCN It is a graph convolutional network designed for CSC that leverages the relational information between Chinese characters to enhance error detection and correction capabilities (Cheng et al., 2020).

GAD The Global Attention Decoder, known as GAD, is introduced by (Guo et al., 2021). This model captures global contextual relationships between characters and candidates to enhance correction accuracy.

3.3 Evaluation Metrics

To maintain a focus on the core aspects, consistent with previous work (Wu et al., 2023; Liu et al., 2023a), we concentrate on sentence-level error correction results and employ commonly used classification metrics to evaluate the quality of the model.

3.4 Main Results

ECSpell The results of ECSpell are in Table 2. In this dataset, we have implemented two approaches: one at both character and word level ReSC$_{word}$, the other only at character level ReSC$_{char}$, to highlight the fact that our word-level information integration is more substantial.

Compared to Rspell, it is clear that the recall results are significantly better than the retrieval results. This is fundamentally due to the inadequate number of items retrieved, and Rspell's approach of segmenting words before retrieval, which leads to the inability to correctly identify certain words. In the law domain, our method's F1 score is 11% higher than Rspell's, representing a substantial difference.

When compared to ReLM, our method stands out because it incorporates a greater amount of word and character information. As a result, the performance is more pronounced, with an average improvement of 3.36% across the three domains. Compared to the ECSpell method, even though it

Domain	Method	Prec.	Rec.	F1
LAW	ChatGPT	46.7	50.1	48.3
	BERT-MFT	73.2	79.2	76.1
	MDCSpell	77.5	83.9	80.6
	ECSpell^{UD}	78.3	74.9	76.6
	Rspell	85.3	81.6	83.4
	Baichuan2	85.1	87.1	86.0
	ReLM	89.9	94.5	92.2
	ReSC$_{char}$	92.0	94.5	93.2
	ReSC$_{word}$	**93.1**	**95.7**	**94.4**
MED	ChatGPT	21.9	31.9	26.0
	BERT-MFT	74.4	77.0	75.7
	MDCSpell	69.9	69.3	69.6
	ECSpell^{UD}	75.9	71.2	73.5
	Rspell	86.1	77.0	81.3
	Baichuan2	72.6	73.9	73.2
	ReLM	85.5	85.3	85.4
	ReSC$_{char}$	86.7	90.7	88.6
	ReSC$_{word}$	**88.3**	**91.6**	**90.0**
ODW	ChatGPT	56.5	57.1	56.8
	BERT-MFT	77.5	78.7	78.1
	MDCSpell	65.7	68.2	66.9
	ECSpell^{UD}	82.3	74.5	78.2
	Rspell	89.0	79.9	84.2
	Baichuan2	86.1	79.3	82.6
	ReLM	85.7	87.8	86.7
	ReSC$_{char}$	88.9	86.9	87.9
	ReSC$_{word}$	**90.3**	**89.6**	**89.9**

Table 2: The sentence-level performance on the correction level. For a fair comparison, the results of Rspell and ECSpell$_{UD}$ are from (Song et al., 2023), and ReLM are from (Liu et al., 2023a).

Methods	Pre	Rec	F1
SIGHAN13			
SpellGCN	78.3	72.7	75.4
GAD	84.9	78.7	81.6
BERT	86.3	78.0	81.9
ReLM	84.1	**80.4**	82.2
ReSC$_{char}$	**84.6**	80.1	**82.3**
SIGHAN14			
SpellGCN	63.1	67.2	65.3
GAD	65.0	70.1	67.5
BERT	**65.5**	67.2	66.3
ReLM	64.7	70.5	67.5
ReSC$_{char}$	64.8	**73.1**	**68.7**
SIGHAN15			
SpellGCN	72.1	77.7	75.9
GAD	73.2	77.8	75.4
BERT	75.5	75.6	75.6
ReLM	73.8	80.7	77.1
ReSC$_{char}$	**76.0**	**81.1**	**78.5**

Table 3: The sentence-level performance on the correction level. For a fair comparison, the results of SpellGCN and GAD (Guo et al., 2021) are directly from the original paper (Guo et al., 2021).

utilizes a vast dictionary, its results are relatively poor due to the inadequate exploitation of the dictionary's contents.

Significantly, it is worth noting that large language models (LLM), such as ChatGPT and Baichuan2, do not perform well for the CSC task. This underperformance can be attributed to their inability to ensure character alignment. Such as Appendix D Table 8 case 1. When ChatGPT rewrites an answer, it cannot guarantee that the characters are aligned, writing about “冰冷饮料” instead of correcting it to “槟榔”. When considering CSC tasks with aligned characters, the weakness of LLM becomes evident. Also, we have listed ten candidate prompts in the Appendix D Table 9.

SIGHAN The ReSC method does not perform well on this dataset since there is no comprehensive domain dictionary, hence our confusion set is at the character granularity, consistent with (Cheng et al., 2020). Therefore, the purpose of setting up this experiment is merely to verify the efficiency of the selection network.

Our method shows a significant improvement over SpellGCN, shown in Table 3, particularly on the SIGHAN13 dataset with an approximate 6% increase in performance. The enhancement is also evident when compared to ReLM, with notable gains on both the SIGHAN14 and SIGHAN15 datasets. The similar results with ReLM on SIGHAN13 can be attributed to its smaller training set, which limits learning and increases the model's susceptibility to overfitting. However, our method's advantages become especially clear in this dataset when compared to both SpellGCN and GAD, illustrating that our use of a confusion set allows our network to more effectively discern which candidates are necessary and which are not.

3.5 Experimental Details

To ensure the validity of our experimental results, we did not utilize tagging-based models such as BERT for this study. Instead, we opted for ReLM as our language model, given its superior capabil-

	LAW		MED		ODW	
	Rec.	#words/char	Rec.	#words/char	Rec.	#words/char
Rspell	45.1	0.3	59.0	0.3	65.8	0.3
ReSC						
with Seg	77.5	67.1	84.9	62.0	80.1	69.4
w/o Seg	**93.7**	**147.6**	**96.1**	**139.5**	**93.8**	**157.8**
w/o Seg & w/o Four-Coner	93.3	144.8	96.1	137.0	93.7	154.7
w/o Seg & w/o Radical	92.3	106	94.1	104.1	92.1	110.3
w/o Seg & w/o ShapeSim	82.1	115.8	90.8	108.5	84.5	127.8
w/o Seg & w/o pinyin	37.6	76.0	38.4	68.8	31.4	80.8

Table 4: The ablation study of the recall of Rspell and ReSC$_{word}$, whereas w/o represents without and Seg represents word segmentation. #words/char represents the total number of words and characters that can be recalled on average for each character.

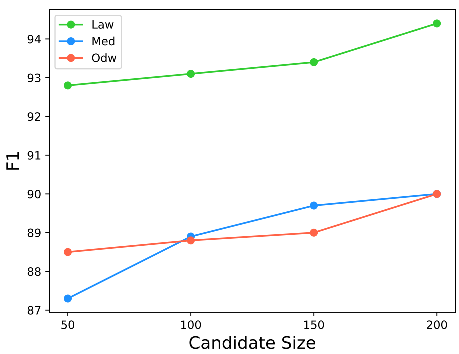


Figure 3: The effects of different recall set sizes on F1 scores for three domain-related datasets. Detailed statistics are in Appendix D Table 7.

	Pre	Rec	F1	Utilize By LM
Law	98.2	98.1	98.1	97.8
Med	99.1	99.2	99.1	97.7
Odw	98.3	98.4	98.3	97.5

Table 5: The statistics of Knowledge Selection. Utilized by LM indicates the percentage of selected items that have been accepted by the language model.

ity in capturing semantic information. For this experiment, we employed one NVIDIA V100 GPU and trained for 2 hours for ECSpell and half an hour for SIGHAN. Besides the λ_{fs} and λ_{KS} are both 0.2 during training and inference.

When training on the ECSpell dataset, our parameters were consistent with those of ReLM. We set the batch size to 64 and the learning rate to 2e-5, with training steps hovering around 5,000. For the SIGHAN dataset, we followed the approach established by (Wu et al., 2023; Guo et al., 2021), initially training the ReLM model on the Wang27K dataset (Wang et al., 2018). Subsequently, we conducted separate training and fine-tuning on the SIGHAN13-15. Given the relative simplicity of the SIGHAN, the number of training steps was limited to approximately 500.

Method	Law	Med	Odw	Avg
BERT-MFT	14.7	11.2	15.5	13.8
MDCSpell	14.3	10.5	16.4	13.7
ReLM	8.4	5.0	6.9	6.8
ReSC$_{word_{200}}$	**4.5**	**4.6**	**3.3**	**4.1**

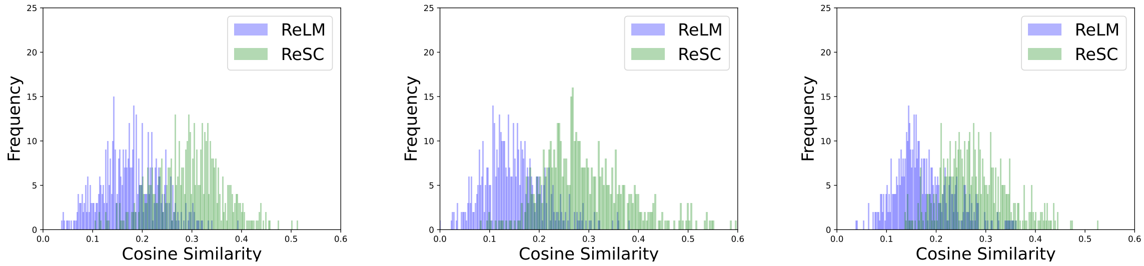
Table 6: Results of False Positive Rate (FPR) on ECSpell. The lower the score, the better the CSC system. The score of ReLM is directly from (Liu et al., 2023a).

4 Further Analysis

4.1 Knowledge Recall Analysis

Knowledge Recall Ablation Study The result is in table 4. Firstly, the number of candidates recalled by our method significantly surpasses that of the Rspell approach, yielding an average recall rate above 94%. Secondly, after segmenting is eliminated, there is a notable increase in recall. Lastly, In the other four recall streams, the most apparent reduction can be attributed to the omission of phonetically similar recall and the discarding of candidates based on character shape similarity.

Number of candidates As shown in Figure 3, it can be clearly seen that as the number of candidates increases, the F1 score continues to rise. This graph indicates our recall network has not yet reached an upper bound.



(a) Law (b) Med (c) Odw

Figure 4: The knowledge injection representation for fine-tuned ReSC$_{word_{200}}$ and ReLM model. Cosine similarity represents the closeness between a domain entity and a sentence, with a higher value indicating a higher likelihood that the sentence contains information related to the entity.

4.2 Knowledge Selection Analysis

Classification Statistics To better assess the efficiency of our selection network, we analyzed the confusion matrix results in Table 5. The analysis demonstrated a significantly impressive result since F1 scores are near 100%. Notably, the Utilized by LM metric has also surpassed 97%, suggesting that the majority of the knowledge post-selection is assimilated by the pre-trained model. This serves as a strong testament to the high efficiency of the selection network.

False Positive Rate can measure the overcorrection behavior of CSC models. As shown in Table 6, although we recall a great number of candidates, including many even for potential correct characters, our network still does not overcorrect. This also indirectly demonstrates the reliability of the knowledge selection network.

Knowledge Injection As shown in Figure 4. We conducted this experiment through three domain test datasets. We compute the cosine similarity of latent vectors for every entity in a sentence and the vector of the sentence itself, then measure the mean distance between the entities and the sentence. The data indicates the average deviation for ReSC is 0.12 for law, 0.14 for med, and 0.10 for odw compared with ReLM. This suggests that ReSC better incorporates entity information to correct errors in characters. This process is similar to human error correction as shown in Figure 1, where our method mimics the steps of understanding, integrating entity information, and then correcting errors.

4.3 Case Study

To better analyze the effectiveness of our model, we utilized the ECSpell dataset. As demonstrated in the Appendix D Table 8, our results appear superior due to integrating more character and word information and the selective use of knowledge. However, the ReLM model, despite its strength in semantic understanding, falls short due to the lack of knowledge input, as seen in Case 2. The closeness in meaning between "制约" and "掣肘" suggests that ReLM has learned much about semantic information. Rspell, on the other hand, underperforms mainly because its mechanism of segmenting first and then retrieving leads to errors, as in Case 2. "制肘" is not recognized as a word, and during segmentation, it is incorrectly split into [融资困难, 制, 肘, 发展], which hinders the correct retrieval of candidate words due to the segmentation error. In contrast, for the ReSC$_{word}$ model, as in Case 3, the recalled terms include " 经济相关" (from Pinyin Recall), making it easier to learn information at the word level.

5 Conclusion

In this study, we mimic the process of human CSC tasks. Specifically, our network comprises two parts: knowledge recall and knowledge selection. Detailed experiments have demonstrated the reliability of our method's recall capability, as well as the accuracy of the selection network. Moreover, our approach achieved SOTA results on three datasets from ECSpell.

Limitations

The issue of an excessively high number of recalls is one of the present challenges. Additionally, there is an inability to better integrate lexical information from perspectives of temporal and syntactic ordering.

References

Yi-Chang Chen, Chun-Yen Cheng, Chien-An Chen, Ming-Chieh Sung, and Yi-Ren Yeh. 2021. Integrated semantic and phonetic post-correction for chinese speech recognition. *arXiv preprint arXiv:2111.08400*.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. 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*, pages 871–881, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhao Guo, Yuan Ni, Keqiang Wang, Wei Zhu, and Guotong Xie. 2021. Global attention decoder for chinese spelling error correction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1419–1428.

Yuzhong Hong, Xianguo Yu, Neng He, Nan Liu, and Junhui Liu. 2019. 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)*, pages 160–169, Hong Kong, China. Association for Computational Linguistics.

Chuan-Jie Lin and Wei-Cheng Chu. 2015. A study on chinese spelling check using confusion sets and? n-gram statistics. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language*.

Liangliang Liu and Cungen Cao. 2016. A seed-based method for generating chinese confusion sets. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(1):1–16.

Linfeng Liu, Hongqiu Wu, and Hai Zhao. 2023a. Chinese spelling correction as rephrasing language model. *arXiv preprint arXiv:2308.08796*.

Linfeng Liu, Hongqiu Wu, and Hai Zhao. 2023b. Chinese spelling correction as rephrasing language model. *ArXiv*, abs/2308.08796.

Qi Lv, Ziqiang Cao, Lei Geng, Chunhui Ai, Xu Yan, and Guohong Fu. 2023. General and domain-adaptive chinese spelling check with error-consistent pretraining. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(5).

Bruno Martins and Mário J Silva. 2004. Spelling correction for search engine queries. In *Advances in Natural Language Processing: 4th International Conference, EsTAL 2004, Alicante, Spain, October 20-22, 2004. Proceedings 4*, pages 372–383. Springer.

Siqi Song, Qi Lv, Lei Geng, Ziqiang Cao, and Guohong Fu. 2023. Rspell: Retrieval-augmented framework fornbsp;domain adaptive chinese spelling check. In *Natural Language Processing and Chinese Computing: 12th National CCF Conference, NLPCC 2023, Foshan, China, October 12–15, 2023, Proceedings, Part I*, page 551–562, Berlin, Heidelberg. Springer-Verlag.

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2517–2527.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided pointer networks for Chinese spelling check. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785, Florence, Italy. Association for Computational Linguistics.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023. Rethinking masked language modeling for chinese spelling correction.

Jinhua Xiong, Qiao Zhang, Shuiyuan Zhang, Jianpeng Hou, and Xueqi Cheng. 2015. HANSpeller: A unified framework for Chinese spelling correction. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 20, Number 1, June 2015-Special Issue on Chinese as a Foreign Language*.

Wangchunshu Zhou, Tao Ge, Chang Mu, Ke Xu, Furu Wei, and Ming Zhou. 2019. Improving grammatical error correction with machine translation pairs. *arXiv preprint arXiv:1911.02825*.

Chenxi Zhu, Ziqiang Ying, Boyu Zhang, and Feng Mao. 2022a. MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1244–1253, Dublin, Ireland. Association for Computational Linguistics.

Chenxi Zhu, Ziqiang Ying, Boyu Zhang, and Feng Mao. 2022b. MDCSpell: A multi-task detector-corrector framework for Chinese spelling correction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1244–1253, Dublin, Ireland. Association for Computational Linguistics.

A Related Work

A.1 Chinese Spelling Correction (CSC)

Some early works employ traditional machine learning such as (Xiong et al., 2015) consisting of a pipeline of error detection, candidate generation, and final candidate selection. Recently proposed works mainly focus on the deep learning paradigm, especially after the boosting application of BERT(Devlin et al., 2019).

One Stage vs. Two Stage Some works turn the CSC into a one-stage pipeline. Such as SpellGCN (Cheng et al., 2020), a specialized graph convolutional network designed to incorporate phonological and visual similarity knowledge into language models for CSC. It constructs a graph over Chinese characters, transforming it into interdependent character classifiers that enhance language models' error detection and correction capabilities. Some works turn the CSC into a two-stage pipeline: error detection and correction. (Hong et al., 2019; Zhu et al., 2022b) propose to use a detection module and correction module to train together and use the hidden states output by the detection module in the correction module.

Tagging vs. Rephrasing Different from the Grammar Error Correction task (GEC), the input and output of CSC have the same length, thus some works regard it as a sequence tagging task (Zhu et al., 2022b; Cheng et al., 2020), and others consider it as rephrasing such as decoder-based text generation model. However, just as (Wu et al., 2023) pointed out, fine-tuning a tagging-based model tends to over-fit the error pattern while underfitting out-of-distribution error patterns. Thus (Liu et al., 2023a) further implements a Rephrasing Language Model (ReLM). This method better mimics how humans think about language and leads to improved performance in both standard and unseen situations.

A.2 CSC with Knowledge

In the CSC task, the incorrectly spelled tokens often bear phonetic or visual resemblance to the correct ones, which allows for the incorporation of external knowledge, to boost the correction performance.

Word Level The granularity of word-level semantic knowledge enables a heightened precision in the rectification of text errors, thereby enhancing the efficacy of automated text correction systems. (Lv et al., 2023) suggests incorporating a User Dictionary (UD) into a token classification-based speller significantly improves performance on domain-specific datasets with uncommon terms. To precisely match related words, (Song et al., 2023) first introduces a retrieval augmented framework (Rspell) for CSC that enhances cross-domain error correction by incorporating domain-specific terms via pinyin fuzzy matching and employing an adaptive control mechanism and iterative strategy.

Character Level Most common in character level is confusion set, a collection of characters that are often mistaken for one another due to their similar shape or pronunciation. To help in accurately correcting spelling errors by focusing on characters that are commonly confused, (Wang et al., 2019) designed their model to use a confusion set to narrow down the character generation choices. This method improves efficiency and accuracy over traditional models that consider the entire vocabulary. To better capture the relation in confusion sets with potential wrong characters, (Cheng et al., 2020) introduce SpellGCN, a specialized graph convolutional network that integrates phonological and visual similarity knowledge directly into language models, outperforming previous methods through its ability to create inter-dependent character classifiers that enhance BERT's representations. Furthermore, (Guo et al., 2021) propose related techniques primarily rely on local context, disregarding the broader sentence context. To tackle this, they introduce the Global Attention Decoder (GAD) methodology that focuses on the global interplay between potentially correct input and likely erroneous character candidates.

B Recall Methods

Pinyin Recall Pinyin recall is the most important one, as (Song et al., 2023; Lin and Chu, 2015) proposed, the most common wrong spelling case is from pinyin. Our recall only used the expression form of $[initials, finals]$ and did not use tones, as most of the incorrect characters from the CSC task are wrong in tone. Such as "癫痫" (dian3xian2, meaning neurological disorder) and its wrong version "点线" (dian3xian4, meaning dot line).

Four Coner Recall To strengthen the recall ability of visual and character structure, we also use Four Coner as a recall method. The four-corner method [1] is a system for encoding Chinese characters. The system breaks down characters into parts and assigns a digit code to each char-

[1]https://en.wikipedia.org/wiki/Four-Corner_Method

acter based on its structural components, where each digit represents a specific feature of the character's top-left, top-right, bottom-left, and bottom-right corners respectively. For example, these characters share the same four-corner code 27620 but different shapes: 訇匐句旬甸.

Redical Recall Radicals are essential components that often hint at a character's meaning or pronunciation. For example, the character "椅" (meaning chair) closely resembles "桌" (meaning table), and both have the radical "木" (meaning wood'). These two characters share a similar structure and the same radical, indicating their relation to furniture.

Shape Recall Recalling visually similar characters, known as "形似字" (xíng sì zì), is a critical aspect of the recalling system as it leverages the shared structural features of characters to enhance the accuracy of corrections. Such as "句" (means sentence) and "甸"(means a suburb or field). Both have the "勹" component but are used differently.

C Derivation of Equation 1

Given $X = \{x_1, x_2, ..., x_n\}$ as input sentence and $Y = \{y_1, y_2, ..., y_n\}$ as output sentence. Also, C represents the whole recall set for this sentence. Then use $P(C|X)$ and $P(Y|X,C)$ as knowledge recall and knowledge selection model. We have

$$\sum_C P(C|X) \cdot P(Y|X,C) = \sum_C P(Y,C|X) = P(Y|X) \tag{10}$$

which gives (1).

D Experimental Details

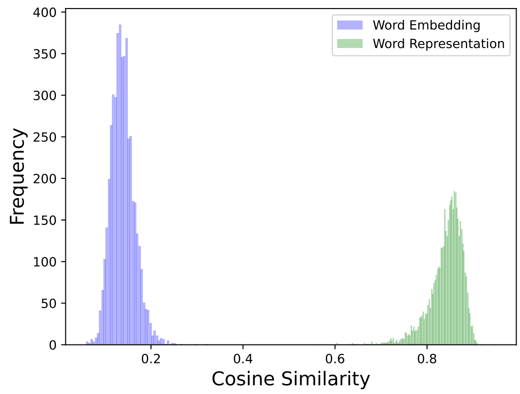


Figure 5: Cosine Similarity score from the character confusion set. The embedding vector is from the ReLM embedding layer and the representation vector is from the last layer of ReLM. Get one character, then compute the cosine similarity with its confusion set and take the average, it can be observed that the confusion set of representations is closer, compared to the embeddings. There is a 0.70 average shift between embedding and representation.

Method	LAW			MED			ODW		
	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
ReSC$_{word_{50}}$	91.9	93.7	92.8	86.7	87.9	87.3	89.7	87.3	88.5
ReSC$_{word_{100}}$	91.3	94.9	93.1	88.5	89.3	88.9	88.8	88.8	88.8
ReSC$_{word_{150}}$	93.0	93.7	93.4	89.7	89.7	89.7	88.9	89.2	89.0
ReSC$_{word_{200}}$	93.1	95.7	94.4	88.3	91.6	90.0	90.3	89.6	90.0

Table 7: The experiment of different candidate size, whereas ReSC$_{word_{50}}$ represents that the maximum recall set size for a single character is 50.

Case1	
Input	冰蓝容易引起口腔疾病 Ice blue can easily cause oral diseases.
Target	槟榔容易引起口腔疾病 Betel nut can easily cause oral diseases.
ReLM	冰榔容易引起口腔疾病 Ice lang can easily cause oral diseases.
ChatGPT	冰冷饮料容易引起口腔疾病 Ice beverage can easily bring oral diseases.
Rspell	槟蓝容易引起口腔疾病 Penta blue can easily cause oral diseases.
Resc$_{word}$	槟榔容易引起口腔疾病 Betel nut can easily cause oral diseases.
Case2	
Input	融资困难制肘发展 Financing difficulties create complications for development.
Target	融资困难掣肘发展 Financial constraints are impeding development.
ReLM	融资困难制肘发展 Financing difficulties create complications for development.
ChatGPT	融资困难制约发展 Financing difficulties restrict development.
Rspell	融资困难制约发展 Financing difficulties restrict development.
ReSC$_{word}$	融资困难掣肘发展 Financial constraints are impeding development.
Case3	
Input	推进平台进击相关市场 Advancing the platform to penetrate related markets.
Target	推进平台经济相关市场 Promote platform economy-related markets.
ReLM	推进平台进济相关市场 Promote platforms to enter relevant markets.
ChatGPT	推进平台进攻相关市场 Promote platforms to fight relevant markets.
Rspell	推进平台进积相关市场 Promote the platform to enter relevant markets.
ReSC$_{word}$	推进平台经济相关市场 Promote platform economy-related markets.

Table 8: Case Study of different models, where the red sections indicate the mistakes, and the green sections represent the correct character.

Prompt1	请检查以下中文句子，并纠正任何错误的字符，确保每个字都是正确的。 Please review the following Chinese sentence and correct any incorrect characters to ensure that each word is accurate.
Prompt2	逐字阅读这段中文文本，并更正其中的任何错词或者打字错误，确保对齐不变。 Read this Chinese text word by word and correct any word errors or typing mistakes , ensuring the alignment remains unchanged.
Prompt3	进行字符级别的纠正，确保输入和输出的长度一致，修改错别的字。 Perform character-level correction to ensure consistent length between input and output , modifying incorrectly substituted characters.
Prompt4	核对下面的文本，并修复所有拼写和错别字，保持正确的字序对齐 Check the text below and fix all spelling errors and typos while maintaining proper character alignment to ensure that each word is accurate.
Prompt5	在不改变原意的基础上，识别并修正所有中文字符错误，实现字对字的精确对齐。 Identify and correct all Chinese character errors in the text without changing the original meaning, achieving precise word-to-word alignment.
Prompt6	仔细查阅提供的文段，指出并修正所有字符层面的错误，以实现优质的纠错效果。 Carefully examine the provided text passage, and point out, and correct all character-level mistakes for quality error correction.
Prompt7	保持输入文本的长度和意思不变，找出并更正所有字符级的错误。 Maintain the length and meaning of the input text unchanged, identify and correct all character-level mistakes.
Prompt8	发现并改正每处不恰当或错误的中文字词，并且需要输入和输出长度一致。 Discover and correct every inappropriate Chinese character while maintaining good character order consistency, and the length of input and output needs to be consistent.
Prompt9	修改给定句子中的错别字，不可以进行删除或者增加操作。 Revise the typographical errors in the given sentence; if there are any mistakes, deletion or addition operations are not permitted.
Prompt10	依次比对文本中的中文字，纠正所有不适当的用词或笔误。 Compare the Chinese characters in the text in sequence, correct all inappropriate wording or slips of the pen.

Table 9: Different prompts on ChatGPT and Baichaun2. In the end, the results brought by prompt9 were the most ideal one.