Chinese Spelling Corrector Is Just a Language Learner

Lai Jiang[◇1,2,3], **Hongqiu Wu**[◇1,2,3], **Hai Zhao**[†1,2,3] and **Min Zhang**[4]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University
[3]Shanghai Key Laboratory of Trusted Data Circulation and Governance in Web3
[4]Harbin Institute of Technology, Shenzhen, China
{jianglai0023-sjth,wuhongqiu}@sjtu.edu.cn

Abstract

This paper emphasizes Chinese spelling correction by means of self-supervised learning, which means there are no annotated errors within the training data. Our intuition is that humans are naturally good correctors with exposure to error-free sentences, which contrasts with current unsupervised methods that strongly rely on the usage of confusion sets to produce parallel sentences. In this paper, we demonstrate that learning a spelling correction model is identical to learning a language model from error-free data alone, with decoding it in a greater search space. We propose *Denoising Decoding Correction (D^2C)*, which selectively imposes noise upon the source sentence to determine the underlying correct characters. Our method is largely inspired by the ability of language models to perform correction, including both BERT-based models and large language models (LLMs). We show that the self-supervised learning manner generally outperforms the confusion set in specific domains because it bypasses the need to introduce error characters to the training data which can impair the error patterns not included in the introduced error characters.

1 Introduction

Chinese spelling correction (CSC) is a fundamental natural language processing task for a series of AI applications (Martins and Silva, 2004; Gao et al., 2010; Yang et al., 2024; Afli et al., 2016; Gupta et al., 2021). Recent studies (Wu et al., 2023b; Liu et al., 2024) show that simply using the supervised signals within parallel sentences to fine-tune pre-trained language models (PLMs) achieves notable results across a series of benchmarks.

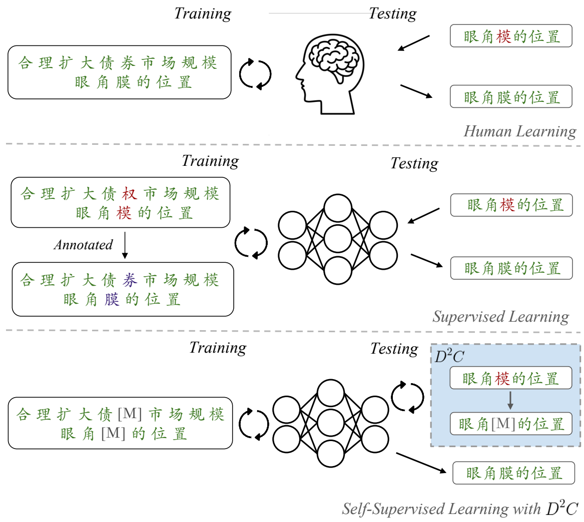

Figure 1: Comparison of human learning, supervised learning, and proposed self-supervised learning process for spelling correction. [M] refers to the mask token.

However, the high cost of annotation is blamed for the low accessibility of parallel sentences. Therefore, these models remain mediocre in handling massive domains in real applications, which makes the application of powerful self-supervised learning to CSC a pivotal issue that has received broad attention in the community. This paper emphasizes the value of self-supervised learning, where only error-free data is used to adapt models to specific target domains, which has still achieved marginal progress in recent years.

Previous unsupervised methods (Zhao and Wang, 2020; Liu et al., 2021; Li, 2022) focus on synthesizing pseudo parallel sentences, while the supervised signals do not derive from the real distribution but from the confusion set (an empirically constructed word set of common misspelled cases). By replacing certain characters in the original sentences with those in the confusion set, parallel sentences are obtained for fine-tuning the models. However, the gap between the confusion set and the real error patterns in the target domain can induce a high

[◇] Equal contribution.
[†] Corresponding author; This research was supported by the Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400).
Source code: `https://github.com/Jianglai-0023/self-supervised-csc`

Findings of the Association for Computational Linguistics ACL 2024, pages 6933–6943
August 11-16, 2024

false positive rate (Wu et al., 2023b). This paper raises a bold idea: *Can machine spelling correction learn from error-free data alone?*

Intriguingly, humans naturally learn to rectify mistakes in a sentence with minimal exposure to parallel data. We illustrate in Figure 1 that humans only learn to use correct sentences (error-free data) in daily life. When encountering a sentence with an error character "模" (*mold*), they can correct it to "膜" (*cornea*) with ease based on their knowledge. In contrast, machine spelling correction models cannot do this if they are not exposed to annotated edit pairs like "模" → "膜" in the training process.

In this paper, we demonstrate that a machine spelling corrector can also be learned from solely error-free data as illustrated at the bottom of Figure 1. The key is to have the model learn semantics rather than character-to-character editing, where the source sentence will first be encoded into the semantic space, and then rephrased to the correct sentence, demonstrate this ability. We call this manner self-supervised spelling correction. However, the resultant models still exhibit a low recall.

To address this problem, we propose a novel decoding algorithm *Denoising Decoding Correction (D^2C)*, which selectively imposes noise upon the source sentence to solve the underlying correct characters. We apply D^2C to two architectures: bidirectional models (represented by ReLM (Liu et al., 2024), the state-of-the-art model in Chinese spelling correction) and auto-regressive models (represented by a series of LLMs (OpenAI, 2023; Touvron et al., 2023; Yang et al., 2023; Wu et al., 2024)). D^2C achieves a significant performance boost over raw language models trained with error-free data.

To evaluate our method across different domains, we created a synthesized training set for LEMON (Wu et al., 2023b) using GPT-3.5 as a sentence generator, which contains only error-free sentences. This dataset permits the fine-tuning and evaluation of self-supervised models in various domains.

We summarize the contributions of this paper.

• We demonstrate that spelling correction can be directly transferred from language modeling on error-free data.

• We propose a novel decoding algorithm, creating an effective self-supervised learning procedure that allows spelling correction models to adapt to target domains with minimal expense.

• We build synthetic error-free training data from LEMON to benchmark unsupervised domain adaption in the community.

2 Related Works

Correcting spelling errors poses a challenging yet crucial task in natural language processing. Early endeavors primarily relied on unsupervised techniques, assessing sentence perplexity as a key metric (Yeh et al., 2013; Yu and Li, 2014; Xie et al., 2015). Recent methods model spelling correction as a sequence tagging problem that maps each character in a given sentence to its accurate counterpart (Wang et al., 2018, 2019). On top of pre-trained language models (PLMs), some BERT-based models with the sequence tagging training objective are proposed. Zhang et al. (2020) identify the potential error characters by a detection network and then leverage the soft masking strategy to enhance the eventual correction decision. Zhu et al. (2022a) use a multi-task network to minimize the misleading impact of the misspelled characters (Cheng et al., 2020). There is also a line of work that incorporates phonological and morphological knowledge through data augmentation and enhances the BERT-based encoder to assist mapping the error to the correct one (Guo et al., 2021; Li et al., 2021; Liu et al., 2021; Cheng et al., 2020; Huang et al., 2021; Zhang et al., 2021). Recent studies (Liu et al., 2024) focus on the rephrasing training objective, which achieves notable results.

While in the unsupervised spelling correction domain, previous works focus on generating pseudo annotated data or detecting error characters with confusion dataset (Zhao and Wang, 2020; Liu et al., 2021; Li, 2022). While these methods are based on heuristics, our method is based on self-supervised learning (Devlin et al., 2019; Gao et al., 2021; Wu et al., 2022, 2023a) which seeks to perturb the language representation of PLMs.

3 From Language Modeling to Spelling Correction

This section provides the motivation for our work. The basic goal is to learn spelling correction from error-free data, which we term self-supervised spelling correction. First, we discuss the transferability between language modeling and spelling correction. Second, we highlight that rephrasing is the primary training objective for self-supervised spelling correction.

We discuss the transferability from two perspectives: (1) The coherence of training objectives between rephrasing spelling correction and language modeling, and (2) The inclusion of knowledge about spelling correction into the pre-training process.

3.1 Language Modeling

First, we introduce the training objectives of language modeling.

Given an input sentence $Y = \{y_1, y_2, \cdots, y_n\}$ of n characters, auto-regressive language modeling seeks to predict the character y_i based on its left context, namely $P(y_i|y_1, y_2, \cdots, y_{i-1})$.

3.2 Spelling Correction

Second, we introduce the training objectives of spelling correction. A spelling correction model can be learned by two dominant objectives, sequence tagging and rephrasing.

Spelling correction aims to rectify the underlying misspelled characters in the source sentence. Denote the source sentence as $X = \{x_1, x_2, \cdots, x_n\}$ and the target sentence as $Y = \{y_1, y_2, \cdots, y_n\}$ and suppose x_i is one of the typos in X, the model learns to correct x_i to y_i based on the entire source sentence, namely $P(y_i|x_1, x_2, \cdots, x_n)$.

Tagging The above modeling process can also be viewed as sequence tagging from X to Y. While this has been widely adopted in previous work, a recent study (Liu et al., 2024) shows that tagging-based spelling correction models will lean towards point-to-point editing, thus ignoring the specific context. The final training objective degenerates into $P(y_i|x_i)$.

Rephrasing In comparison, rephrasing (Liu et al., 2024) is shown to be a more effective training objective for spelling correction. It specifically seeks to rewrite the entire sentence, namely $P(y_i|x_1, x_2, \cdots, x_n, y_1, y_2, \cdots, y_{i-1})$. To ensure that the rephrasing process is based on semantics instead of copying, a ratio of noise (e.g., masking with an unused token) is introduced to the source sentence, written as $P(y_i|\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n, y_1, y_2, \cdots, y_{i-1})$.

3.3 Self-supervised Spelling Correction

The unsupervised learning setting is naturally akin to language modeling, where the model is trained on error-free data. Comparing the above two training objectives with language modeling, we find that rephrasing and language modeling are formally the same. In rephrasing, the input sentence is the concatenation of the source and target. This implies that the spelling correction model can better utilize the knowledge in a pre-trained language model and be transferred from it.

	LAW	MED	ODW
Top-20	93.8	88.8	93.8
Top-10	90.8	86.0	90.6
Top-5	86.9	82.0	88.7
Top-1	69.5	66.3	76.8

Table 1: Accuracy of the top-k predictions of MLM from the vanilla BERT model. LAW, MED (medical treatment), and ODW (official document writing) represent three domain datasets in ECSpell (Lv et al., 2023). Top-k means the top-k candidates in the mask token's position.

3.4 Knowledge in Vanilla PLM

We hypothesize that, after large-scale pre-training, the language model already contains the knowledge needed for spelling correction.

To verify this hypothesis, we mask the error characters in the source sentence and have the vanilla model (non-fine-tuned one) output the corrected sentence. We then check the predicted characters at the positions of the mask tokens of the output sentence and compare them with the right characters.

As shown in Table 1, we see that the vanilla model can already recall the correct characters in its top-k candidates without any fine-tuning on spelling correction. For example, in about 90% of the cases, the model's top 10 predictions have covered the correct answer. This indicates that pre-trained language models already possess the necessary knowledge for spelling correction through mask-infilling.

3.5 Tagging Model vs. Rephrasing Model

In this section, we evaluate the tagging model and the rephrasing model (Liu et al., 2024) through two small-scale experiments, uncovering their different emphases during the spelling correction task. The tagging model excels at remembering the characters' mapping relations while the rephrasing model performs better in understanding the meaning of the sentences.

Error-free Data Table 2 shows that the tagging model trained on error-free data is ineffective. We

	Method	LAW	MED	ODW
EF.	Tagging	0.5	0.6	0.5
	Tagging-MFT	10.1	5.3	10.5
	Rephrasing	**71.3**	**68.6**	**71.9**
Shuf.	Tagging	29.5	15.3	16.7
	Tagging-MFT	**34.0**	**17.3**	**18.9**
	Rephrasing	27.6	12.3	13.3

Table 2: Comparison (F1) of tagging and rephrasing (Liu et al., 2024) on error-free (self-supervised) / shuffled characters. The details of the models and dataset are in Sec. 6. EF. means error-free and Shuf. means shuffled.

conjecture that the model only learns point-to-point copying since the source is always the same as its target, thus losing the ability to make modifications to the source sentence. In contrast, the rephrasing model can learn well even with error-free data. This confirms that pre-trained language models can learn spelling correction from error-free data alone.

Shuffling of Characters Specifically, we shuffle the characters in the source and target sentences pairwise to spoil their semantics. We use these highly noisy samples to fine-tune the rephrasing and tagging models. From Table 2 (Shuf.), we find that the tagging model outperforms the rephrasing model on samples that do not convey semantic information.

Conversely, it verifies that the tagging model focuses more on point-to-point editing at the expense of semantics. As mentioned before, it is the semantics that are key to learning spelling correction from error-free data. Therefore, in this paper, we choose to rephrase as the primary training objective for self-supervised spelling correction.

4 Synthetic LEMON Training Set

To evaluate self-supervised models' performance across multiple domains, we release a GPT-3.5-generated synthetic LEMON training set.

LEMON (Wu et al., 2023b) is a multi-domain benchmark that allows us to evaluate the multi-domain generalization of CSC models. However, it only includes a test set without a training set. The synthetic data is generated in two steps: (1) Extract the words in each domain. (2) Randomly select words and request GPT-3.5 to generate error-free sentences mimicking the style of specific domains. See our prompts in Appendix A.

Statistical information about the size of the different training sets is provided in Table 3.

GAM	ENC	COT	MEC	CAR	NOV	NEW
2389	2489	1707	2222	2381	3669	4273

Table 3: Number of sentences in each training set. The domains include game (GAM), encyclopedia (ENC), contract (COT), medical care (MEC), car (CAR), novel (NOV), and news (NEW).

5 Method

In this section, we first introduce two rephrasing architectures. Then, we propose an enhanced decoding method to unleash the potential of pre-trained language models. Additionally, we suggest using a confusion dataset to improve the recall score.

5.1 Two Rephrasing Architectures

Our method can be implemented using two architectures: non-auto-regressive rephrasing and auto-regressive rephrasing.

Auto-regressive Model Auto-regressive models, such as GPT-like models (Brown et al., 2020), are the primary choice for generating rephrasing.

To improve the quality of rephrasing, it is an easy yet effective way to mask a ratio of characters in the source sentence with an unused token (Wu et al., 2023b). In this paper, we denote the masked source sentence as $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n\}$.

ReLM Rephrasing Language Model (ReLM) (Liu et al., 2024) is the current state-of-the-art spelling correction model based on BERT (Devlin et al., 2019). It rephrases the source sentence by filling the masked slots. Specifically, the model is fed with the concatenation of the source sentence and a sequence of mask tokens. Due to the bidirectional nature of BERT, the rephrasing process can be expressed as $P(y_i|\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n, m_1, m_2, \cdots, m_n)$, where m_i refers to the mask token. Unlike auto-regressive models, ReLM predicts all characters simultaneously.

5.2 Denoising Decoding Correction

The model trained with rephrasing still suffers from low recall when tested on real sentences because there are no mask tokens present. The situation becomes even more challenging when multiple errors occur in a single sentence. The cascade effect of these errors makes it increasingly difficult to correct the erroneous characters.

To address these problems, we propose a novel decoding algorithm, where we actively introduce noise to the source sentence to encourage the model to recall more candidates. Since the mask operation in the inference stage is consistent with that in the training stage of rephrasing, the model's correction capability can be boosted. We call this method *Denoising Decoding Correction (D²C)*.

Specifically, we mask the leftmost character in the source sentence if its confidence level falls below β (0.995). Such a character is considered a potential error. Then we send this masked sentence to the model and figure out whether the original character appears in the prediction's **top-k** candidates. If it does, we keep the original character; otherwise, we note the new character and its confidence if this confidence is bigger than a **threshold** ϵ. We then mask the second character from the left and repeat the procedure. We refer to the process of masking all characters in the sentence as an iteration. After each iteration, we select the character with the highest recorded confidence and update the original sentence with it. We continue iterations until no further updates are needed. As the number of errors decreases, the challenges associated with correcting multiple typos also diminish. Thus, this iterative decoding method is robust against multiple errors.

We notice that picking a character with the biggest confidence in each iteration results in a large decoding overhead. Given that there is always a small number of errors in a sentence, we rank the characters in the sentence by their confidence from the lowest to highest, mask the top α of them respectively, and send the sentence to the model to figure out whether the original character appears in its top-k candidates. If it does, we remain the original character (same as original D²C strategy), else we update it with a new character that has the highest confidence if this confidence is bigger than a threshold ϵ.

Pseudo Code The overall procedure of D²C is described in Algorithm 1.

5.3 Fine-tune with Confusion Set

Given the low recall rate, we can improve the model by replacing some tokens with a confusion set instead of mask tokens during the fine-tuning process. The confusion set is constructed based on Chinese pronunciations and fonts. Using this confusion set, we can create a parallel dataset for training.

Algorithm 1: D^2C

Input: Input sentence Y; Threshold ϵ; Top-k; Language Model LM; Set S.
Output: Predict Result Z

```
for t ∈ [0,length(Y)] do
    Clear S;
    for i ∈ [0,length(Y)] do
        Mask y_i;
        Get top-k predictions {y_i^1, y_i^2, ..., y_i^k} and confidences {p_i^1, p_i^2, ..., p_i^k} from LM;
        if y_i ∉ {y_i^1, ..., y_i^k} and p_i^1 > ε then
            Store y_i^1 and p_i^1 to S;
        else
            Continue;
        end
    end
    if S is empty then
        Break;
    else
        Choose y_i^1 with biggest p_i^1 from S;
        Replace y_i with y_i^1;
    end
end
Z = Y;
```

For an error-free sentence, we randomly select one character and replace it with a character from the confusion set.

Specifically, our method for using the confusion set is as follows: we initially train our self-supervised model using entirely error-free data. Then, we generate parallel error-annotated data by introducing a rate of error-free data into the confusion set. Finally, we continue to fine-tune the model using this parallel data.

6 Experiments

In this section, we report the empirical results of a series of spelling correction benchmarks.

We concentrate on two benchmarks:

- *ECSpell* (Lv et al., 2023): a small-scale multi-domain Chinese spelling correction dataset of Law (LAW), medical treatment (MED), and official document writing (ODW), which is particular due to its large number of errors in the test set that do not appear in the training set;
- *Syn-LEMON*: it is generated from LEMON

	Method	EC-LAW (%)				EC-MED (%)				EC-ODW (%)			
		F1	P	R	FPR	F1	P	R	FPR	F1	P	R	FPR
Supervised	BERT	38.6	42.1	35.7	12.2	24.2	27.1	21.9	10.5	24.9	29.9	21.3	13.9
	BERT-MFT	74.6	73.2	76.1	14.3	61.7	62.4	60.9	10.5	60.8	59.7	62.0	18.9
	MDCSpell-MFT	81.5	77.2	86.3	15.9	65.1	62.3	68.1	16.8	64.1	61.3	67.2	21.4
	Baichuan2	86.0	85.1	87.1	4.5	73.2	72.6	79.3	5.5	82.6	86.1	79.3	4.0
	ReLM	**95.8**	**93.6**	**98.0**	5.7	**89.9**	**86.6**	**93.5**	7.4	**92.2**	**93.3**	**91.1**	2.5
Self-supervised	BERT	0.5	0.7	0.4	9.0	0.6	0.9	0.4	8.0	0.5	0.8	0.4	12.4
	BERT-MFT	10.1	14.1	7.8	9.4	5.3	7.7	4.0	9.1	10.5	15.1	8.0	12.8
	MDCSpell-MFT	36.2	45.3	30.2	9.4	20.9	28.7	16.4	8.8	25.9	33.7	21.7	13.7
	Baichuan2	23.5	25.5	21.6	26.5	17.4	25.2	13.3	13.5	24.4	27.2	22.2	20.9
	Baichuan2-UD	26.9	30.8	23.9	20.4	18.3	27.4	13.7	11.7	28.0	32.7	24.4	14.5
	Baichuan2-D^2C	27.6	30.6	25.1	22.4	20.2	26.2	16.4	12.4	30.5	33.8	27.8	17.5
	GPT-4 (5-shot)	67.9	67.7	68.3	6.5	56.4	50.4	64.2	24.1	72.5	73.6	71.4	1.7
	ReLM	71.3	78.1	75.7	0.4	68.6	70.8	66.5	7.02	71.9	79.7	65.5	0.8
	ReLM-UD	89.5	**89.2**	89.9	4.7	**79.3**	**74.1**	85.4	18.5	84.6	**88.5**	81.0	2.3
	ReLM-Conf.(10%)	83.8	79.1	89.0	15.6	70.8	67.5	74.4	14.7	75.5	71.5	79.8	18.5
	ReLM-Conf.(100%)	84.1	77.7	91.8	19.7	69.7	57.6	88.4	41.1	73.4	68.5	79.1	19.3
	ReLM-D^2C	**90.2**	87.7	**92.9**	8.6	75.7	66.8	**87.4**	25.5	**85.9**	85.7	**86.1**	7.3

Table 4: Results on ECSpell, where F1, P, R, FPR refers to the F1 score, precision, recall, and false positive rate. Conf. (10%) means continually fine-tuning the self-supervised model with 10% confusion data. Conf. (100%) means continually fine-tuning the self-supervised model with 100% confusion data.

(Wu et al., 2023b) which spans 7 different domains with a total of 19,130 synthetic train samples.

We consider the following methods:

• *BERT* (Devlin et al., 2019): the fine-tuned tagging model based on BERT;

• *MDCSpell* (Zhu et al., 2022b): the strongest tagging model with a multi-task network of error detection and correction;

• *Masked-FT (MFT)* (Wu et al., 2023b): a simple yet effective fine-tuning technique on tagging models to uniformly mask the non-error characters in the source sentence;

• *ReLM* (Liu et al., 2024): the newly released state-of-the-art models on spelling correction, which rephrases the sentence in a non-autoregressive manner;

• *Baichuan2-7b* (Yang et al., 2023): one of the strongest Chinese LLMs following the autoregressive architecture;

• *User Dictionary (UD)* (Lv et al., 2023): an enhanced decoding method that leverages an expertise dictionary (law, medical treatment, and official document writing) to bias the beam search.

6.1 Training Settings

For BERT-based models, we set the batch size to 128 and the learning rate to 5e-5, swept from grid search. For Baichuan2, we set the batch size to 32 and the learning rate to 3e-4, and use LoRA (Hu et al., 2022) to reduce the training budget. For supervised spelling correction, the masking ratio is chosen from {0.2, 0.3}, while for self-supervised spelling correction, it is set to 0.5.

When fine-tuning with the confusion set, we set the batch size to 64 and the learning rate to 5e-5.

6.2 Results on ECSpell

Table 4 highlights the effectiveness of rephrasing models when using error-free data and demonstrates the robust performance of D^2C.

We first find that ReLM outperforms MDCSpell-MFT by 35.1, 47.7, and 46.0 absolute points of F1 respectively on LAW, MED, and ODW,

When empowered with D^2C, it further significantly produces the increase of 18.9, 7.1, and 14.0 absolute points. The biggest increase is in the recall rate, which is consistent with the design of D^2C. Furthermore, we find that D^2C is competitive against using a user dictionary (UD), or even more powerful. It suggests that some of the domain knowledge in the user dictionary has already been stored in the pre-trained language models, and D^2C plays a key role in unlocking their great power.

When utilizing the confusion set, the increase is weaker. The confusion set method increases the FPR score, reaching 41.1 on MED. A higher confusion rate is related to a higher FPR score.

6.3 Results on Syn-LEMON

Table 5 summarizes the results of self-supervised methods on Syn-LEMON. It indicates that, except for GAM (game) and NOV (novel), using the con-

Method	GAM	ENC	COT	MEC	CAR	NOV	NEW
Previous SoTA (Wu et al., 2023b)	33.8	48.6	67.2	54.3	53.1	38.6	58.7
ReLM	63.7	51.5	69.3	57.6	55.3	43.9	58.6
ReLM-D^2C	**65.5**	53.7	69.6	58.4	58.6	**50.0**	63
ReLM-Conf.(100%)	46.5	**58.6**	**75.5**	**65.8**	**63.3**	49.7	**70.0**
ReLM-Conf.(10%)	52.2	51.7	71.1	55.1	55.0	40.4	59.2

Table 5: Results on LEMON. Conf. (100%) means 100% data is trained as a confusion set and Conf. (10%) means we use 90% data as self-supervised training data and 10% data as continued fine-tuning confusion data.

fusion set outperformed D^2C's F1 score. These variances reveal that different domains possess distinct data properties, significantly influencing performance outcomes when employing the confusion set versus D^2C.

7 Discussion

7.1 D^2C vs. Using Confusion Set

We compare D^2C and the data augmentation method using the confusion set, a widely used technique in previous work. In Table 4, we find that D^2C outperforms using the confusion set on two of the chosen datasets. Table 5 indicates that D^2C surpasses using confusion set on GAM and NOV.

First, the results indicate that both D^2C and using the confusion set can increase the recall rate. The common phenomenon is caused by different reasons. The confusion set introduces character-to-character corrections during the training process that are similar to test examples. While D^2C introduces mask tokens to the test examples, which is inherited from the fine-tuning process. However, using the confusion set has a disadvantage compared with D^2C. The non-matching segments in the confusion set can cause gaps in the real error patterns in the testing time. Therefore, using the confusion set always has lower P scores and higher FPR scores. D^2C is a more suitable choice when it comes to domains that contain professional knowledge.

Second, compared to D^2C, using the confusion set is relatively straightforward and efficient. Employing the confusion set presents an alternative approach in various application scenarios, offering efficiency but potentially posing a risk to performance.

7.2 Seen and Unseen Errors

To take a closer look at the correction ability, we divide the test set into two subsets, exclusive (E) and inclusive (I) sets, which refer to the test errors that occur or do not occur in the training set.

	Models	F1(%) LAW	MED	ODW
Supervised	MDCSpell (I)	71.8	51.3	54.9
	MDCSpell (E)	7.5	4.0	0.8
	MDCSpell-MFT (I)	94.3	78.4	81.7
	MDCSpell-MFT (E)	76.0	60.7	57.8
Self-supervised	MDCSpell-MFT (I)	52.6	32.9	32.1
	MDCSpell-MFT (E)	48.0	26.0	33.7
	ReLM (I)	93.2	73.5	82.2
	ReLM (E)	92.5	74.7	73.1
	ReLM-D^2C (I)	98.2	79.2	88.3
	ReLM-D^2C (E)	97.0	81.5	82.7

Table 6: Performances on seen (I) and unseen (E) errors, measured by F1 scores.

From Table 6, it is clear that supervised models fit the internal error set well but the performances drop sharply on the external error set. While models trained with error-free data have a high degree of similarity between the performance on the external error set and the internal error set. Besides, D^2C boosts the performance on the external and internal sets simultaneously.

Surprisingly, MDCSpell-MFT performs even better on self-supervised learning than supervised on the exclusive set. This suggests that the tagging objective degenerates the learned representation in the pre-trained language model, leading to a drop in generalizability.

7.3 Effect of Mask Rate

We also investigate the impact of the mask rate. From Figure 2, it is apparent that the F1 scores for ECSpell's Law improve consistently as the mask rate increases from 0% to approximately 30%, after which they experience a slight decline. A closer examination reveals that increasing the mask rate significantly enhances recall (R) scores more than precision (P) scores, while P scores tend to remain unchanged or even decline. Since the error-free fine-tuning process introduces noise solely through

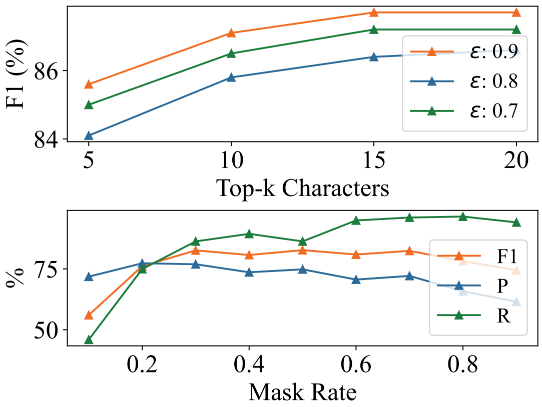

Figure 2: Effect of hyperparameters on LAW of Ecspell. In the top table, we show the F1 score related to different thresholds ϵ and top-k characters. In the bottom table, we show the F1, P (precision), and R (recall) scores with different mask rates.

mask tokens, the models are more inclined to preserve the source sentences without modification, resulting in lower R scores. During the evaluation stage, error characters serve as noise for the model; therefore, a higher mask rate improves the models' performance on R scores.

7.4 Effect of Hyperparameters

We assess the effect of hyperparameters in D^{2}C. As a representative, we depict the curves on ReLM in Figure 2.

Threshold Figure 2 shows that a higher threshold (ϵ) leads to improved performance. For instance, D^{2}C with a higher ϵ (0.9) achieves better results on LAW.

Top-k There is a common phenomenon in Figure 2 that a higher top-k character count uplifts the F1 score under different thresholds ϵ.

7.5 Efficiency

We compare the decoding efficiency of D^{2}C and normal decoding in Table 7. Our observations indicate that, compared to directly decoding each sentence, D^{2}C requires approximately twice the time on ReLM and three times the time on Baichuan.

8 Case Study

We further showcase some examples to illustrate how D^{2}C improves the decoding process.

Multi-typo In this case, (*What are the innovations in meniscal* (半月板) *calcification* (钙化)), error characters are (钙→ 改) and (半→ 伴), which are very similar in pronunciation but meaningless as words in the sentence. We noticed in the experiment that ReLM without D^{2}C failed to correct this sentence with two error characters while successful with a single error character if one of the two errors has been corrected before. Therefore, with D^{2}C we introduce noise into the source sentence to correct "伴" and "改" step by step.

	Dataset	Normal (s)	D^{2}C (s)
ReLM	MED	0.024	0.048
	LAW	0.022	0.038
	ODW	0.022	0.044
Baichuan	MED	1.0	3.2
	LAW	0.6	1.6
	ODW	0.7	2.2

Table 7: Comparison between D^{2}C and normal decoding on ReLM and Baichuan, by second per sample.

SRC	伴月板改化的病因有哪些
Trans.	What are the innovations in meniscαl change?
TRG	半月板钙化的病因有哪些
Trans.	What are the innovations in meniscal calcification?

Table 8: Multi-typo case can be better corrected by D^{2}C. Blue characters are right and red are wrong.

Can't Recall Considering sentences in spelling correction sometimes have short lengths, models receive limited semantic information and tend to under-correct error characters just like the case in Table 9. This case (*How to calculate child's weight* (体重)) has the error pattern of (体→ 休), which are similar in terms of their visual appearance. In the presence of semantics limitations, D^{2}C directs models to reword specified positions to incorporate more suitable characters and effectively mitigate the issue of under-correction.

9 Conclusion

This paper studies self-supervised spelling correction based on rephrasing-based models. We demonstrate that machine spelling correction does not necessitate parallel data and can be learned from error-free data alone. We propose a novel decoding algorithm named *D^{2}C* to effectively enhance the recall ability of the self-supervised model. We also compare the D^{2}C method with the confusion set method. Results on Chinese spelling correction showcase the significant improvement brought by our method. We hope this paper can bring new in-

SRC	小孩休重怎么计算	
Trans.	How to calculate child's weght?	
TRG	小孩体重怎么计算	
Trans.	How to calculate child's weight?	

Table 9: D^{2}C improves the recall rate.

sight and vigor to future research on self-supervised spelling correction.

Limitations

Our work focuses on Chinese. Other languages, such as Korean have not been studied in this work. Additionally, D^{2}C leads to a decline in the decoding speed.

References

Haithem Afli, Zhengwei Qiu, Andy Way, and P'araic Sheridan. 2016. Using smt for ocr error correction of historical texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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, ACL 2020, Online, July 5-10, 2020*, pages 871–881. 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.

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

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 6894–6910. 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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1419–1428. Association for Computational Linguistics.

Harsh Gupta, Luciano Del Corro, Samuel Broscheit, Johannes Hoffart, and Eliot Brenner. 2021. Unsupervised multi-view post-OCR error correction with language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8647–8652, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Li Huang, Junjie Li, Weiwei Jiang, Zhiyu Zhang, Minchuan Chen, Shaojun Wang, and Jing Xiao. 2021. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5958–5967. Association for Computational Linguistics.

Chong Li, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2021. Exploration and exploitation: Two ways to improve chinese spelling correction models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, pages 441–446. Association for Computational Linguistics.

Piji Li. 2022. uChecker: Masked pretrained language models as unsupervised Chinese spelling checkers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2812–2822, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Linfeng Liu, Hongqiu Wu, and Hai Zhao. 2024. Chinese spelling correction as rephrasing language model. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024*. AAAI Press.

Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, and Di Wang. 2021. PLOME: pre-training with misspelled knowledge for chinese spelling correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2991–3000. Association for Computational Linguistics.

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*, volume 3230 of *Lecture Notes in Computer Science*, pages 372–383. Springer.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

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, Brussels, Belgium, October 31 - November 4, 2018*, pages 2517–2527. Association for Computational Linguistics.

Dingmin Wang, Yi Tay, and Li Zhong. 2019. Confusionset-guided pointer networks for chinese spelling check. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5780–5785. Association for Computational Linguistics.

Hongqiu Wu, Ruixue Ding, Hai Zhao, Boli Chen, Pengjun Xie, Fei Huang, and Min Zhang. 2022. Forging multiple training objectives for pre-trained language models via meta-learning. In *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 6454–6466. Association for Computational Linguistics.

Hongqiu Wu, Linfeng Liu, Hai Zhao, and Min Zhang. 2023a. Empower nested boolean logic via self-supervised curriculum learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 13731–13742. Association for Computational Linguistics.

Hongqiu Wu, Y. Wang, Xingyuan Liu, Hai Zhao, and Min Zhang. 2024. Instruction-driven game engines on large language models. *CoRR*, abs/2404.00276.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023b. Rethinking masked language modeling for chinese spelling correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 10743–10756. Association for Computational Linguistics.

Weijian Xie, Peijie Huang, Xinrui Zhang, Kaiduo Hong, Qiang Huang, Bingzhou Chen, and Lei Huang. 2015. Chinese spelling check system based on n-gram model. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2015, Beijing, China, July 30-31, 2015*, pages 128–136. Association for Computational Linguistics.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models.

Yifei Yang, Hongqiu Wu, and Hai Zhao. 2024. Attack named entity recognition by entity boundary interference. In *Proceedings of the 2024 Joint International*

Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy, pages 1734–1744. ELRA and ICCL.

Jui-Feng Yeh, Sheng-Feng Li, Mei-Rong Wu, Wen-Yi Chen, and Mao-Chuan Su. 2013. Chinese word spelling correction based on n-gram ranked inverted index list. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 43–48. Asian Federation of Natural Language Processing.

Junjie Yu and Zhenghua Li. 2014. 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, Wuhan, China, October 20-21, 2014*, pages 220–223. Association for Computational Linguistics.

Ruiqing Zhang, Chao Pang, Chuanqiang Zhang, Shuohuan Wang, Zhongjun He, Yu Sun, Hua Wu, and Haifeng Wang. 2021. Correcting chinese spelling errors with phonetic pre-training. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2250–2261. Association for Computational Linguistics.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 882–890. Association for Computational Linguistics.

Zewei Zhao and Houfeng Wang. 2020. Maskgec: Improving neural grammatical error correction via dynamic masking. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 1226–1233. AAAI Press.

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, Dublin, Ireland, May 22-27, 2022*, pages 1244–1253. 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 Prompts

A.1 Extract Words

Sentences are extracted from the original LEMON dataset.

```
1. Please extract the words in the given sentences
2. Your answer should be in Chinese and JSON format
{sentences}
Your answer format:
"words":
["word1", "word2",...],
["word1", "word2",...]
...
]}
```

A.2 Generate Data

We propose the GAM domain's prompt as an example.

```
1. You are a professional game writer. Try to use your professional knowledge and think step by step.
2. Please make your answers diverse in formats, words, and expressions.
3. Generate 5 smooth sentences Using the given word sets
4. Your answer should be abundant and include details, but not too long
5. Try to generate realistic and fluent sentences like a human writer
6. Your answer should be in Chinese in JSON format
7. Your generated sentence should follow the style of my given example sentences
This is my given word sets:
"words":
["word1", "word2",...],
["word1", "word2",...]
...
]}
This is my given example sentences:
{sentences}
Your answer:
[sentence1,sentence2,...]
```