Spelling Corrector Is Just Language Learner

Anonymous ACL submission

Abstract

This paper studies spelling correction of purely unsupervised learning, which meanings there are no annotated errors within the training data, a pivotal issue that has received broad attention in the community. Our intuition is that humans are naturally good correctors with almost no exposure to parallel sentences, which contrasts to current unsupervised methods that are strongly reliant on the usage of confusion sets. In this paper, we demonstrate that learning a spelling correction model is identical to learning a language model from monolingual 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 solve out the underlying correct characters. Our method largely inspires the ability of language models to perform correction, including both BERT-based models and large language models (LLMs), and unlocks significant performances on Chinese and Japanese spelling correction benchmarks. We also show that this self-supervised learning manner generally outstrips using confusion sets, because it bypasses the need of introducing error characters to the training data which can impair the patterns in the target domains.

1 Introduction

Spelling correction stands as a fundamental task in natural language processing, supporting many downstream applications, e.g. web search (Martins and Silva, 2004; Gao et al., 2010), named entity recognition (Yang et al., 2023b), optical character recognition (Afli et al., 2016; Gupta et al., 2021). Recent studies (Wu et al., 2023; 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.

However, the great cost of annotation blames for the low accessibility of parallel sentences. These models remain mediocre in handling massive domains in real applications. This paper thus emphasizes the value of self-supervised learning, where only monolingual data is used to adapt models to specific target domains, which still achieves marginal progress in recent years.

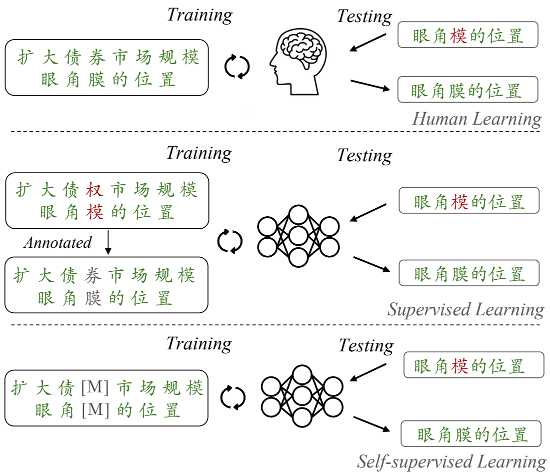

Figure 1: Comparison of human learning, supervised learning and proposed self-supervised learning process for spelling correction. [M] refers to the mask token.

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, a empirically constructed set of common misspelled cases. By replacing certain characters in the original sentences with the error characters 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 false positive rate (Wu et al., 2023). This paper raises a bold idea: *can machine spelling correction learn from monolingual data alone?*

Intriguingly, humans naturally learn to rectify mistakes in a sentence with minimal exposure to parallel data. We give an illustration in Figure 1, which shows that humans only learn to use the correct sentences (monolingual data) in daily life. When encountering a sentence with an error character "模" (*mold*), they are able to correct it to "膜" (*cornea*) with ease based on their knowledge. In contrast, the machine spelling correction models cannot do this only if it is 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 monolingual data, akin to a human learner, as illustrated in the bottom of Figure 1. The key is have the model learn semantics rather than character-to-character editing. In light of this, we find that rephrasing models (Liu et al., 2024), 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 low recall.

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

We summarize the contributions of this paper.

• We demonstrate the intrinsic transferability between learning spelling correction models and language, and spelling correction can be transferred by language modeling on monolingual data.

• With the propose novel decoding algorithm, we build an effective self-supervised learning manner, allowing the spelling correction models to adapt to target domains at a minimal expense.

2 Related Work

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 map each character in a given sentence to its accurate counterpart (Wang et al., 2018, 2019). On top of pre-trained language models (PLMs), a number of 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 (Wu et al., 2023; Liu et al., 2024) focus on rephrasing training objective and achieves notable results.

While in 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). We are the first to raise a notable self-supervised method with pure monolingual Chinese and Japanese spelling correction data in the community. Our method inherits the ability of PLMs and present a transferability from language modeling to spelling correction.

3 Transfer Language Modeling to Spelling Correction

This section serves as the preliminary of our work. The basic effort is to learn spelling correction from monolingual data. We call it self-supervised spelling correction. We first discuss the transferability between language modeling to spelling correction. Second, we point out that rephrasing is the primary training objective for self-supervised spelling correction.

3.1 Language Modeling

Given an input sentence $Y = \{y_1, y_2, \cdots, y_n\}$ of n characters, (auto-regressive) language modeling seeks to solve the character $\mathbf{y}_i$ based on its left context, namely $P(y_i|y_1, y_2, \cdots, y_{i-1})$. A spelling correction model can be learned by two dominant objectives, sequence tagging and rephrasing.

3.2 Spelling Correction

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 after it, 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 that of language modeling, where the model is trained on monolingual 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 means that the spelling correction model can better utilize the knowledge in a pre-trained language model and be transferred from it more easily.

Due to the lack of parallel sentences, we let $X = Y$, so that the rephrasing objective is modified to $P(y_i|\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_n, y_1, y_2, \cdots, y_{i-1})$. It means that the model learns to rephrase the sentence based on its semantics and we hope that the resultant model can be generalizable to rephrase a sentence with typos to its correct.

We evaluate the tagging model and rephrasing model on unsupervised spelling correction and present empirical results in Table 1 (Mono.). It shows that the tagging model trained on monolingual data is powerless. We conjecture that the model only learns point-to-point copying since the source is always the same as it target, thus losing the ability to make modification to the source sentence. In contrast, the rephrasing model can learn well even with monolingual data, suggesting that the model is well transferred from the pre-training process. It paves the wave for us that pre-trained language models can learn spelling correction from solely monolingual data.

	Method	LAW	MED	ODW
Mono.	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 1: Comparison (F1) of tagging and rephrasing on unsupervised spelling correction / shuffled characters. The details of the models and dataset are in Sec. 5.

3.4 Shuffling of Characters

We conduct a second tiny experiment to strengthen the idea. Specifically, we shuffle the characters in the source and target sentences parallelly to spoil the semantics of them and use these samples to fine-tune the models. From Table 1 (Shuf.), we find that the tagging model outperforms the rephrasing model on samples that do not convey semantic information. It inversely verifies that the tagging model learns more of point-to-point editing at the expense of semantics. As aforementioned, it is the semantics that are the key to learning spelling correction from monolingual data.

3.5 Vanilla Pre-trained Language Models

The second tiny experiment is to probe the pre-trained knowledge in pre-trained language models. We hypothesize that, after large-scale pre-training, the language model already contains the literal knowledge needed for spelling correction. What we do is to mask the the error characters in the source sentence and have the vanilla model (non-fine-tuned one) to predict that. From Table 2, 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 has covered the correct answer.

To sum, this section provides evidence that spelling correction can be learned with monolin-

Method	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 2: Accuracy of the top-k predictions of MLM from the vanilla BERT model.

gual data from pre-trained language models:

• rephrasing-based spelling correction shares the same objective as language modeling;

• pre-trained language models have already possessed the needed knowledge for spelling correction.

4 Method

In this section, we propose an enhanced decoding method to further unleash the potential of pre-trained language models.

4.1 Two Rephrasing Architectures

The method focuses on rephrasing-based spelling correction, which can be achieved in two architectures, non-auto-regressive rephrasing and auto-regressive rephrasing.

Auto-regressive models Auto-regressive model is the primary choice to generate the rephrasing following the input sentence, represented by GPT-like models (Brown et al., 2020) and large language models (LLMs).

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. In this paper, we denote the masked source sentence as $\tilde{Y} = \{\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_n\}$.

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 infilling the mask slots. Specifically, the model is fed with the concatenation of the source sentence and the a sequence of mask tokens. Due to the bidirectional nature of BERT, the rephrasing process can be written as $P(y_i|\tilde{y}_1, \tilde{y}_2, \cdots, \tilde{y}_n, m_1, m_2, \cdots, m_n)$, where m_i refers to the mask token. As opposed to auto-regressive models, ReLM predict all characters at once.

4.2 Denoising Decoding Correction

In self-supervised spelling correction, where the source sentence equals to the target sentence, the resultant model trained with rephrasing still suffers from a low recall when testing on real sentences that need to be corrected. A more severs situation happens when there are multiple errors in one sentence. The cascade effect of errors makes it even harder to correct the sentence. To this end, we propose a novel decoding algorithm, where we actively introduce noise to the source sentence and 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 (D^2C).*

Concretely, we first mask the characters in source sentence from the left side during each iteration if the character's confidence is bigger than β (0.995). The character in such a position is regarded as a potential error. To determine which character to be updated, we send this sentence to the model and figure out whether the original character appears in its **top-k** candidates. If it does, we remain the original character, else we record the new character and its confidence, if this confidence is bigger than a **threshold** ϵ. After each iteration, we choose the character with the biggest confidence recorded before and update the original sentence with it. We do the iteration continually until there is nothing to update after an iteration. Note that once a character is updated, the confidences of the other characters will change correspondingly, so this iterative decoding is robust to multiple errors.

Accelerating We notice that picking a character with the biggest confidence each iteration costs large decoding overhead. Given that there are always a small number of errors in a sentence, we rank the characters in the sentence by their confidences from the lowest to highest, mask top α of them respectively and send the sentence to model. Figure out whether the original character appears in its top-k candidates. If it does, we remain the original character (same as original D^2C strategy), else we update it with a new character with the highest confidence, if this confidence is bigger than a threshold ϵ.

Pseudo code The overall procedure of D^2C is described in Algorithm 1.

Algorithm 1: D^2C

Input: Source sentence Y; threshold ϵ, top-k.
Output: predict result Z

Sort the characters in Y on their confidences ascendingly and record the indices I;
for $i \in I$ **do**
  Mask y_i;
  Get top-k predictions $\{y_i^1, y_i^2, \cdots, y_i^k\}$;
  Get confidences $\{p_i^1, p_i^2, \cdots, p_i^k\}$;
  if $y_i \notin \{y_i^1, \cdots, y_i^k\}$ ***and*** $p_i^1 > \epsilon$ **then**
    Replace y_i with y_i^1;
    Decode the new Y and update it;
  else
    Keep y_i unchanged;
  end
end
$Z = Y$;

5 Experiments

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

We focus on two languages:

- *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 in that there are a large number of errors in the test set that do not appear in the training set;
- *MCSC* (Jiang et al., 2022): a large-scale Chinese spelling correction dataset specialized in medicine, with more than 200k training samples;
- *JWTD* (Tanaka et al., 2020): a Japanese spelling correction dataset, which is extracted from the revision update of wikipedia.

We consider the following methods:

- *BERT* (Devlin et al., 2019): the fine-tuned tagging model based on BERT-base;
- *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., 2023): a simple yet effective fine-tuning technique on tagging models to uniformly masking 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-auto-regressive manner;
- *Baichuan2-7b* (Yang et al., 2023a): one of the strongest Chinese LLMs following the auto-regressive architecture;
- *User Dictionary (UD)* (Lv et al., 2023): an enhanced decoding method that leverage an expertise dictionary (law, medical treatment, and official document writing) to bias the beam search.

5.1 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 learn 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.

5.2 Results on ECSpell

Table 3 summarizes the performances of different training methods on ECSpell and we also report the supervised performances for reference. For self-supervised spelling correction, 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, suggesting the great promise of rephrasing models. 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 on the recall rate, which is consistent with the design of D^2C. Furthermore, we find that D^2C is competitive against using user dictionary (UD), or even more powerful. It suggests the some of the domain knowledge in the user dictionary has already stored in the pre-trained language models, and D^2C plays a key role to unlock the great power of pre-training.

5.3 Results on MCSC

Table 4 summarizes the results on MCSC. In contrast to ECSpell, we find that the self-supervised performances are much worse than supervised ones. There are two reasons. The first reason is that the annotated samples in MCSC are sufficient enough so that the supervised fine-tuning results in nice outcomes. The evidence is that all methods achieve closer results on it compared to those on ECSpell. The second is that MCSC is there are a great number of samples than contain more than one errors. It is still a big challenge to handle these samples in self-supervised spelling correction even with D^2C.

	Method	EC-LAW (%) F1	P	R	FPR	EC-MED (%) F1	P	R	FPR	EC-ODW (%) 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
	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-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 3: Results on ECSpell, where F1, P, R, FPR refers to the F1 score, precision, recall, and false positice rate.

	Method	MCSC (%) F1	P	R	FPR
Supervised	BERT	70.7	70.8	70.7	2.9
	BERT-MFT	73.3	73.4	73.1	2.9
	MDCSpell-MFT	78.5	78.5	78.6	2.5
	Baichuan2	75.5	76.3	74.7	1.2
	ReLM	**83.2**	**82.9**	**83.6**	2.5
Self-supervised	BERT	0.6	1.7	0.4	0.2
	BERT-MFT	1.6	2.9	1.1	1.3
	MDCSpell-MFT	7.1	13.7	4.7	0.7
	Baichuan2	3.6	10.7	2.2	0.1
	Baichuan2-D^2C	20.0	37.4	13.7	0.8
	ReLM	21.8	29.9	17.2	1.3
	ReLM-UD	30.8	36.3	26.8	3.4
	ReLM-D^2C	**37.9**	**38.9**	**37.0**	4.3

Table 4: Results on MCSC.

	Method	JWTD(%) F1	P	R	FPR
Supervised	BERT	65.0	75.6	56.8	22.0
	BERT-MFT	68.0	79.8	59.2	29.3
	MDCSpell-MFT	73.0	81.8	**65.9**	26.8
	ReLM	**73.6**	**84.8**	65.1	7.5
Self.	BERT	0.0	0.0	0.0	0.0
	BERT-MFT	1.7	30.2	0.9	4.9
	MDCSpell-MFT	2.3	52.4	1.2	4.9
	ReLM	10.8	**77.1**	5.8	1.1
	ReLM-D^2C	**29.0**	39.3	**23.0**	38.0

Table 5: Results on Japanese spelling correction.

Dataset	ReLM-Conf. (%) F1	FPR	ReLM-D^2C (%) F1	FPR
EC-LAW	80.0	40.2	**90.2**	8.6
EC-ODW	67.0	38.9	**85.9**	7.3
MCSC	**39.8**	9.9	37.9	4.3

Table 6: Comparison with the confusion set and D^2C. Conf. means confusion.

For self-supervised spelling correction, we find that D^2C similarly achieves significant performance gain on both Baichuan2 and ReLM, lifting the F1 scores by 16.4 and 16.1 respectively. We also find that the user dictionary does not work very well, because of the weak alignment between the dictionary and MCSC data, incurring unstable gain on different data. However, D^2C play its role conditioned on the pre-trained knowledge.

5.4 Results on Japanese

From Table 5, we find that D^2C also works well on Japanese, outperforming the base decoding on ReLM by 18.2% points of F1.

6 Discussion

6.1 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 6. We find that D^2C outperforms using the confusion set on two of the chosen datasets. It suggests that the non-matching segments in the confusion set can cause gaps to the real error patterns in the testing time. However, the monolingual data used in self-supervised learning bypasses this risk.

6.2 Seen and Unseen Errors

To take a closer look at the correction ability, we divide test set into two subsets, exclusive (E) and inclusive (I) set, which refer to the test errors that occur or not occur in the training set. From table 7, it is discernible that supervised models fit internal error set well but the performances drop sharply

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

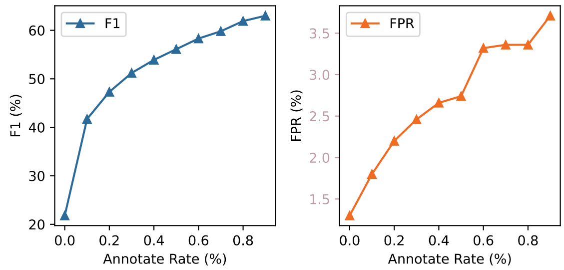

Figure 2: ReLM's F1 and FPR scores with different amount of annotated data on MCSC.

on external error set. While models trained with monolingual data have a high degree of similarity between the performance on external error set and internal error set and D^{2}C boosts the performance on external and internal set simultaneously. Surprisingly, we find that MDCSpell-MFT performs even better on self-supervised learning than supervised on the exclusive set. It suggests that the tagging objective degenerates the learned representation in the pre-trained language model, incurring the drop of generalizablity.

6.3 Effect of Annotated Data

We investigate the variation of F1 and FPR when increasingly adding annotated data on top of monolingual data. From 2, it is discernible that eventual performance can be boosted greatly with a small amount of annotated data, which is about under 20%. It offers a promising signal that monolingual data, which can be achieved with a low cost, combined with a smaller amount of annotation, can lead to nice outcomes in real applications. Meanwhile, we notice that the false positive rate also increases with the increase of annotated data.

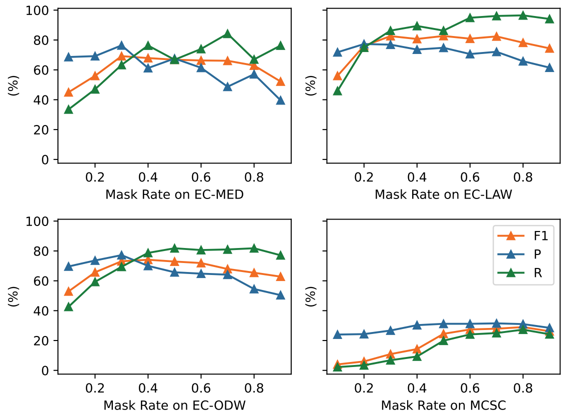

Figure 3: Performances with different mask rates.

6.4 Mask Rate

We also investigate the impact of mask rate. From Figure 3 it is apparent that the F1 scores on ECSpell keep improving when the mask rate grows from 0% to about 30%, and than drop slightly. Besides, the F1 score on MCSC keeps increasing until mask rate meets 80%, which is much higher than ECSpell. To dig further, a phenomenon observed across all datasets is that an increase in the mask rate uplifts recall (R) scores more apparently than precision (P) scores while P scores either lean to unchanged or even decline with an increase in the mask rate. Because monolingual fine-tuning process is designed to shield models from error patterns and introduces noise solely through mask tokens, the models are more inclined to preserve the source sentences without modification, which means a lower R scores. During the evaluation stage, error characters serve as noise for the model, therefore a higher mask rate boost models' performances on R scores. It also indicates that mask rate relies on specific data and the dataset MCSC which has shorter sentences and multi-typos leans to perform better under higher mask rate.

6.5 Effect of Hyperparameters

We access the effect of hyperparameters in D^{2}C. As a representative, we depict the curves on ReLM in Figure 4.

Threshold Figure 4 shows that different datasets are suitable with different threshold (ϵ). For example, D^{2}C with higher ϵ (0.9) gains better performances on LAW, MED and ODW domains. However, D^{2}C with about 0.6 threshold have higher F1 scores on MCSC. It reveals that ϵ should be set based on different datasets.

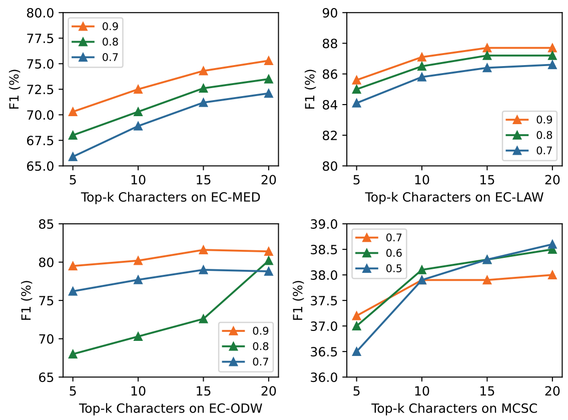

Figure 4: Non-auto-regressive D^{2}C's performance with different hyperparameters.

	Dataset	Original (s)	Accelerate (s)
ReLM	EC-MED	0.024	0.048
	EC-LAW	0.022	0.038
	EC-ODW	0.022	0.044
Baichuan	EC-MED	1.0	3.2
	EC-LAW	0.6	1.6
	EC-ODW	0.7	2.2

Table 8: Comparison between accelerated D^{2}C and directly decoding on ReLM and Baichuan, measured by second per sample.

Top-k There is a common phenomenon in Figure 4 that a higher top-k characters uplifts F1 score under different change confidence ϵ. Considering that a high top-k characters brings decline in running speed, it is a trade off between speed and accuracy for users.

6.6 Efficiency

We compare the decoding efficiency of accelerated D^{2}C and decoding directly in Table 8. We can observe that compared with decoding each sentence directly, D^{2}C requires about twice the time on ReLM and three times the time on Baichuan.

7 Case Study

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

SRC	伴月板改化的病因有哪些
TRG	半月板钙化的病因有哪些
ReLM	伴月板改化的病因有哪些
ReLM-D^{2}C	半月板钙化的病因有哪些

Table 9: Multi-typo case can be better corrected by D^{2}C. Blue characters are right and red are wrong.

SRC	小孩休重怎么计算
TRG	小孩体重怎么计算
ReLM	小孩休重怎么计算
ReLM-D^{2}C	小孩体重怎么计算

Table 10: D^{2}C improves the recall rate.

Multi-typo In this case, (How does calcification (钙化) of the meniscus (半月板) occur), error characters are (钙→ 改) and (半→ 伴) , which are very similar in pronunciation but meaningless as words in the sentence. We noticed in experiment that ReLM without D^{2}C failed to correct this sentences with two error characters while success with 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.

Not recall Considering sentences in spelling correction sometimes have short length, models receive limited semantics information and tend to under-correct error characters just like case in Table 10. This case (How to calculate children'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 position to incorporate more suitable characters and effectively mitigating the issue of under-correction.

8 Conclusion

This paper studies self-supervised spelling correction based on the rephrasing-based models. We demonstrate that machine spelling correction does not necessitate parallel data, and can be learned from monolingual data alone. We propose a novel decoding algorithm named *D^{2}C* to effectively enhance the recall ability of the self-supervised model. Results on Chinese and Japanese spelling correction showcase the significant improvement brought by our method. We hope that this paper can bring new insight and vigour to future research on unsupervised spelling correction.

Limitations

Our work focuses on Chinese and Japanese. Other language such as Korean have not been studied in this work. D^{2}C cost a decline in the speed of single sentence processing. Our self-supervised method's performances is sensitive to multi-typo data.

References

Haithem Afli, Zhengwei Qiu, Andy Way, and P‘ąraic 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.

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.

Wangjie Jiang, Zhihao Ye, Zijing Ou, Ruihui Zhao, Jianguang Zheng, Yi Liu, Bang Liu, Siheng Li, Yujiu Yang, and Yefeng Zheng. 2022. Mcscset: A specialist-annotated dataset for medical-domain chinese spelling correction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM ’22, page 4084‘Ă‘Ş4088, New York, NY, USA. Association for Computing Machinery.

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.

Yu Tanaka, Yugo Murawaki, Daisuke Kawahara, and Sadao Kurohashi. 2020. Building a japanese typo dataset from wikipedia's revision history. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, ACL 2020, Online, July 5-10, 2020*, pages 230–236. Association for Computational Linguistics.

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, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023. 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. 2023a. Baichuan 2: Open large-scale language models.

Yifei Yang, Hongqiu Wu, and Hai Zhao. 2023b. Attack named entity recognition by entity boundary interference. *CoRR*, abs/2305.05253.

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.