Chinese Spelling Correction as Rephrasing Language Model

Linfeng Liu*, **Hongqiu Wu***, **Hai Zhao**

Department of Computer Science and Engineering, Shanghai Jiao Tong University
Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
{linfengliu, wuhongqiu}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

This paper studies Chinese Spelling Correction (CSC), which aims to detect and correct the potential spelling errors in a given sentence. Current state-of-the-art methods regard CSC as a sequence tagging task and fine-tune BERT-based models on sentence pairs. However, we note a critical flaw in the process of tagging one character to another, that the correction is excessively conditioned on the error. This is opposite from human mindset, where individuals rephrase the complete sentence based on its semantics, rather than solely on the error patterns memorized before. Such a counter-intuitive learning process results in the bottleneck of generalizability and transferability of machine spelling correction. To address this, we propose *Rephrasing Language Model (ReLM)*, where the model is trained to rephrase the entire sentence by infilling additional slots, instead of character-to-character tagging. This novel training paradigm achieves the new state-of-the-art results across fine-tuned and zero-shot CSC benchmarks, outperforming previous counterparts by a large margin. Our method also learns transferable language representation when CSC is jointly trained with other tasks.

Introduction

Chinese Spelling Correction (CSC) is a fundamental natural language processing task to detect and correct the potential spelling errors in a given Chinese text (Yu and Li 2014; Xiong et al. 2015). It is crucial for many downstream applications, e.g, named entity recognition (Yang, Wu, and Zhao 2023), optical character recognition (Afli et al. 2016), web search (Martins and Silva 2004; Gao et al. 2010).

Current state-of-the-art methods regard CSC as a sequence tagging task and fine-tune BERT-based models on sentence pairs (Devlin et al. 2019). On top of this, phonological and morphological features are further injected to enhance the tagging process (Huang et al. 2021; Liu et al. 2021a; Lv et al. 2022).

While sequence tagging has become the prevailing paradigm in CSC, in this paper, we note a critical flaw coming with performing character-to-character tagging, which is counter to natural human mindset. CSC is a special form of tagging, where the majority of characters are the same between the source and target sentences. As a result, the model is allowed to greatly memorize those mappings between the error and correct characters during training and simply copy the other characters, to still achieve a decent score on the test set. It means that the resultant correction will be excessively conditioned on the original error itself, while ignoring semantics of the entire sentence. We showcase a concrete example in Figure 1. In (a), the model has been exposed to an edit pair (correct *age* to *remember*) during previous training. Now it encounters a new error *age* during testing and still corrects it to *remember*. It reflects that the tagging model doggedly memorizes the training errors and fails to fit in the context. However, this is far from human spelling correction. In (b), when a person sees a sentence, he first commits the underlying semantics to his mind. Then, he rephrases the sentence based on his linguistic knowledge from the past, though this process will not be explicitly written down, and eventually decides how the sentence should be corrected. For example, here, it is easy for a person to correct *age* to

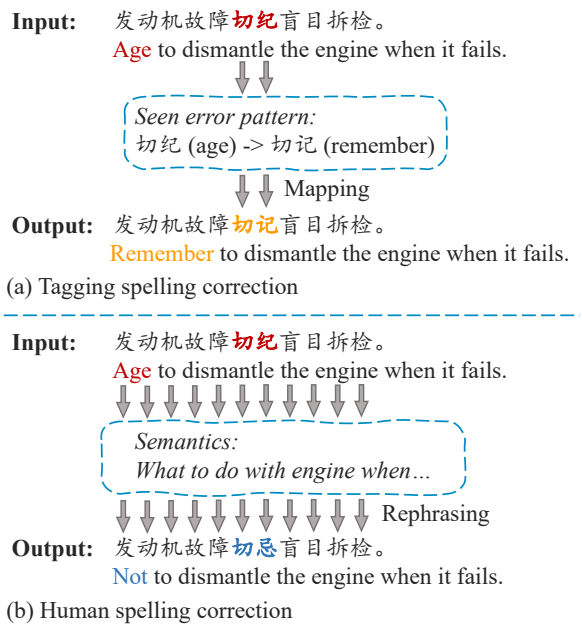


Figure 1: Comparison of tagging spelling correction and human spelling correction.

*These authors contributed equally.

not based on the given semantics.

Instead of character-to-character tagging, we propose to rephrase the sentence as the training objective for fine-tuning CSC models. We denote the resultant model as *Rephrasing Language Model (ReLM)*, where the source sentence will first be encoded into the semantic space, and then is rephrased to the correct sentence based on the given mask slots. ReLM is based on BERT (Devlin et al. 2019), and achieves the new state-of-the-art results on existing benchmarks, outperforming previous counterparts by a large margin. We find that the rephrasing objective also works on auto-regressive models like GPT (Brown et al. 2020) and Baichuan (Yang et al. 2023), but worse than ReLM.

As opposed to previous work, we also pay attention to the CSC performance in multi-task settings, where CSC is jointly trained with other tasks (e.g. sentiment analysis, natural language inference). We find that tagging-based fine-tuning leads to non-transferable language representation and the resultant CSC performance will significantly degenerate once there is another task. The explanation still lies in the excessive condition on the errors. This problematic property makes CSC hard to be incorporated into multi-task learning. Given the ongoing trend of instruction tuning across diverse tasks (OpenAI 2023), this phenomenon has a significant negative impact. We show that ReLM allows for better transferability between CSC and other tasks, building promising multi-task models.

Our contributions are summarized below: (1) We propose ReLM to narrow the gap between machine spelling correction and human spelling correction. (2) ReLM significantly enhances the generalizability of CSC models and refreshes the new state-of-the-art results across fine-tuned and zero-shot CSC benchmarks. (3) We probe into and enhance the transferability of CSC to other tasks. (4) Our analysis shows that ReLM effectively exploits and retains the pre-trained knowledge within PLMs, while tagging models do not. [1][2]

Related Work

Most of the early efforts in CSC focus on the unsupervised techniques by evaluating the perplexity of the sentence (Yeh et al. 2013; Yu and Li 2014; Xie et al. 2015). Zhou, Porwal, and Konow (2019) reformulates the spell correction problem as a machine translation task. Recent methods model CSC as a sequence tagging problem that maps each character in the sentence to the correct one (Wang et al. 2018; Wang, Tay, and Zhong 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. (2022) 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. 2021a; Cheng et al. 2020; Huang et al. 2021; Zhang et al. 2021). However, our method achieves the new state-of-the-art results over all these variants with the original BERT architecture, by repurposing the training objective.

Similar as previous methods, our method is based on the (PLMs) (Devlin et al. 2019; Brown et al. 2020; Liu et al. 2019; Wu et al. 2022; He, Gao, and Chen 2023). However, we maximize the pre-trained power by continually optimizing the language modeling objective, instead of sentence or token classification. Our method works on both encoder and decoder architectures, and furthermore, we discuss CSC as a sub-task in multi-task learning, which is not discussed in previous work.

More recently, Wu et al. (2023c) decompose a CSC model into two parallel models, a language model (LM) and an error model (EM), and find that tagging models lean to overfit the error model while under-fit the language model. An effective technique masked-fine-tuning is thus proposed to facilitate the learning of LM. While the masking strategy is still effective in our method, their work differs from ours in terms of bottom logic. The masked-fine-tuned CSC model remains a tagging model, which partially mitigates the negative effect of EM. More importantly, our method is a language model alone, instead of two parallel models. It indicates that with an effective training objective, LM can possess the functionality of EM, essentially solving the overfitting to EM.

Method

Problem Formulation

Chinese Spelling Correction (CSC) aims to correct all misspelled characters in the source sentence. Given a source sentence $X = \{x_1, x_2, \cdots, x_n\}$ of n characters with potential spelling errors, the model seeks to generate the target sentence $Y = \{y_1, y_2, \cdots, y_n\}$ of the same length with all potential errors corrected. The above process can be formulated as a conditional probability $P(Y|X)$. Specifically for x_i, suppose that it is an error character and its ground truth is y_i, then the probability to correct x_i to y_i can be written as $P(y_i|X)$.

Tagging

Sequence tagging is a common model in many natural language processing tasks, where the model is trained to map one character to another correspondingly, e.g. named entity recognition, part-of-speech tagging. All these tasks share an pivotal property in that they strongly rely on the alignment information between input and output characters. However, deep neural models like Transformer (Vaswani et al. 2017) are always good at exploiting spurious clues if it is possible to achieve lower training loss, especially when the training data is not big enough (Wu et al. 2023a,b). For example, *Norway* is always a geopolitical entity (GPE) in entity recognition. Consequently, the model can memorize such a character-to-character mapping and make correct predictions in most situations. Similar in CSC, the model can greatly memorize the trivial edit pair of correcting x_i to y_i

[1]https://github.com/Claude-Liu/ReLM

[2]https://github.com/gingasan/lemon

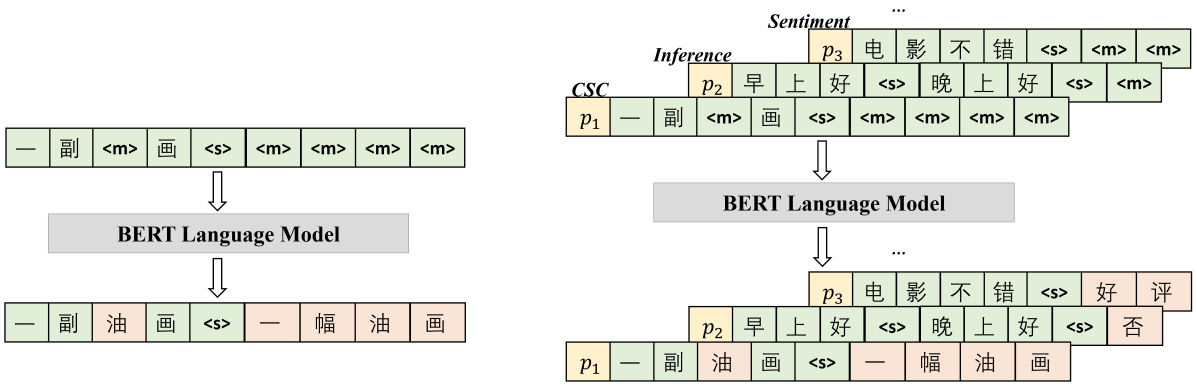


Figure 2: Paradigm of ReLM in single-task (left) and multi-task (right) settings. The source sentence for CSC is "taking a pair (→ piece) of painting", and ⟨m⟩ and ⟨s⟩ refer to the mask and separate character respectively. On the right, we depict three tasks as a representative, CSC, language inference, and sentiment analysis, and p refers to the prompt for each task.

and continue to apply it in a different context of x_i, without referring to the semantics. Hence, the original training objective degenerates to:

$$P(y_i|X) \approx P(y_i|x_i) \tag{1}$$

where we suppose x_i is an error character.

However, the errors in CSC are much more diverse. As previously shown in Figure 1, *age to* can be both corrected to *remember to* or *not to*, which significantly relies on the immediate context. The resultant tagging model can hardly be generalized to unseen errors.

Rephrasing

In this paper, we propose to substitute sequence tagging with rephrasing as the primary training objective for CSC.

Our intuition is to eliminate the trend that the model fits the training data by naively memorizing the errors. To do this, we train the pre-trained language models (PLMs) to rephrase the source sentence following it. Concretely, the Transformer layers first transfer the source sentence to the semantic space. Then, the model generates a new sentence while correcting all errors in it, based on the semantics. This process is consistent with the human process of doing spelling correction. When a person sees a sentence, he will first commit the sentence to his mind (akin to encoding it to the semantic space), and then transform the semantics to a new sentence based on his linguistic instinct (the pre-trained weights in the PLM). We see that the pre-training knowledge offers a great foundation for such learning rephrasing, which is aside from learning sequence tagging. Our following experiments show that sequence tagging does not make good use of the benefits from pre-training.

The process of rephrasing can be modeled based on the auto-regressive architecture with a decoder to generate the output characters one by one, e.g. GPT (Brown et al. 2020). Specifically, we concatenate the source characters X and the target characters Y as the input sentence, i.e. $\{x_1, x_2, \cdots, x_n, \langle \mathrm{s} \rangle, y_1, y_2, \cdots, y_n, \langle \mathrm{eos} \rangle\}$, where $\langle \mathrm{s} \rangle$ and $\langle \mathrm{eos} \rangle$ refers to the separate token and wrap token, and train the model to predict all the target characters y_i auto-regressively. Hence, rephrasing-based spelling correction seeks to solve the following probability for $y_i, i >= 1$:

$$P(y_i|X) \approx P(y_i|X, y_1, y_2, \cdots, y_{i-1}). \tag{2}$$

Rephrasing Language Model

Based on the BERT-based architecture, we propose *Rephrasing Language Model* (*ReLM*), a non-auto-regressive rephrasing model.

Eq 2 generates the sentences free of lengths. Given that the length of the target sentence is fixed in CSC, equal to that of the source sentence, such freedom may bring a negative impact. BERT is an encoder-only architecture (Devlin et al. 2019), pre-trained by randomly setting a portion of characters to a mask symbol $\langle \mathrm{m} \rangle$. In contrast to the auto-regressive model, which keeps generation until $\langle \mathrm{eos} \rangle$, BERT is programmed to only infill the pre-set slots of mask.

As shown in Figure 2, we concatenate the source characters X and a sequence of mask characters $M = \{m_1, m_2, \cdots, m_n\}$ of the same length as the input sentence, i.e. $\{x_1, x_2, \cdots, x_n, \langle \mathrm{s} \rangle, m_1, m_2, \cdots, m_n\}$, where m_i refers to the mask character for y_i, and train the model to infill all the mask characters m_i following X. Since BERT can see both the left-side and right-side context, ReLM seeks to solve the following probability for $y_i, i = 1 \sim n$:

$$P(y_i|X) \approx P(y_i|X, m_1, m_2, \cdots, m_n). \tag{3}$$

ReLM is superior to the auto-regressive model as it can always generate the output sentence of the same length as the input, which makes it more accurate. In our following experiments, we find that both auto-regressive rephrasing and ReLM outweigh previous tagging models, and the latter achieves more powerful results.

Auxiliary Masked Language Modeling As opposed to tagging models, fine-tuned ReLM on CSC is still a language model as its core. However, there remains a chance that the model can learn the alignment of source and target sentences. We thus propose a key strategy, that is to uniformly mask a fraction of the non-error characters in the

source sentence with an unused token, to greatly regularize the model against learning character-to-character alignment. ReLM eventually necessitates correcting potential typos while simultaneously restoring the entire sentence.

Distinguish from Sequence Tagging ReLM is a biased estimation of $P(y_i|X)$, which optimizes $P(y_i|X, m_1, m_2, \cdots, m_n)$ instead. The resultant model is forced to rely on the entire semantics. The key property is, Eq. 3 predicts y_i conditioned on the entire source sentence X, in contrast to Eq. 1. More concretely, there is no alignment of characters in ReLM, and the model is not allowed to find a shortcut to perform character-to-character mapping as it does in sequence tagging.

ReLM for Multi-Task

The emerging large language models (OpenAI 2023; Touvron et al. 2023) tend to handle diverse tasks at the same time, and it is time to study the incorporation of CSC into other tasks. In typical multi-task learning, we add a specific classification head for each task and train a shared encoder for all tasks. For instance, CSC will share the same language representation within the encoder with sentence classification. However, our empirical analysis shows that the performance of conventional tagging-based CSC may largely deteriorate when it is jointly trained with other tasks. The corresponding probing is in the following analysis section.

In contrast, ReLM, still a language model as its core, naturally suits the multi-task learning on top of language modeling, while tagging-based CSC does not. Concretely, each individual task is templated to the format of masked language modeling, as shown in Figure 2. In general, all tasks are unified to a rephrasing-like format, which enhances the transferablity of CSC to various tasks. In addition, ReLM supports prompt tuning (Lester, Al-Rfou, and Constant 2021; Liu et al. 2021b). We prefix a sequence of trainable characters to the input sentence as the prompt steering the model for different tasks, and optimize the corresponding prompt for each task. We find that introducing prompts can further improve the outcome, but to a slight extent.

Experiment

In this section, we compare ReLM with a line of tagging-based methods on existing benchmarks. We also evaluate the CSC performance in multi-task learning, where all the models are jointly trained on three different tasks, CSC, semantic similarity, and news classification.

Dataset

ECSpell ECSpell (Lv et al. 2022) is a CSC benchmark with three domains, LAW (1,960 training and 500 test samples), MED (3,000 training and 500 test samples), and ODW (1,728 training and 500 test samples).

LEMON Large-scale multi-domain dataset with natural spelling errors (LEMON) (Wu et al. 2023c) is a novel CSC benchmarks with diverse real-life spelling errors. It spans 7 different domains with totally 22,252 test samples. It typically measures the open-domain generalizability of a CSC model in a zero-shot setting.

	Method	Prec.	Rec.	F1
LAW	$\text{GPT2}_{\text{Tagging}}$	37.7	32.5	35.0
	$\text{BERT}_{\text{Tagging}}$	43.3	36.9	39.8
	$\text{GPT2}_{\text{Rephrasing}}$	**61.6**	**84.3**	**71.2**$_{\uparrow 31.4}$
	$\text{BERT}_{\text{Tagging}}$-MFT	73.2	79.2	76.1
	$\text{MDCSpell}_{\text{Tagging}}$-MFT	77.5	83.9	80.6
	ReLM	**89.9**	**94.5**	**91.2**$_{\uparrow 10.6}$
	$\text{Baichuan2}_{\text{Rephrasing}}$	85.1	87.1	86.0
	ChatGPT-10 shot	46.7	50.1	48.3
MED	$\text{GPT2}_{\text{Tagging}}$	23.1	16.7	19.4
	$\text{BERT}_{\text{Tagging}}$	25.3	20.0	22.3
	$\text{GPT2}_{\text{Rephrasing}}$	**29.6**	**44.7**	**35.6**$_{\uparrow 13.3}$
	$\text{BERT}_{\text{Tagging}}$-MFT	57.9	58.1	58.0
	$\text{MDCSpell}_{\text{Tagging}}$-MFT	69.9	69.3	69.6
	ReLM	**79.2**	**85.9**	**82.4**$_{\uparrow 12.8}$
	$\text{Baichuan2}_{\text{Rephrasing}}$	72.6	73.9	73.2
	ChatGPT-10 shot	21.9	31.9	26.0
ODW	$\text{GPT2}_{\text{Tagging}}$	26.8	19.8	22.8
	$\text{BERT}_{\text{Tagging}}$	30.1	21.3	25.0
	$\text{GPT2}_{\text{Rephrasing}}$	**46.2**	**64.3**	**53.8**$_{\uparrow 28.8}$
	$\text{BERT}_{\text{Tagging}}$-MFT	59.7	58.8	59.2
	$\text{MDCSpell}_{\text{Tagging}}$-MFT	65.7	68.2	66.9
	ReLM	82.4	**84.8**	**83.6**$_{\uparrow 16.7}$
	$\text{Baichuan2}_{\text{Rephrasing}}$	**86.1**	79.3	82.6
	ChatGPT-10 shot	56.5	57.1	56.8

Table 1: Precison, recall, and F1 results on ECSpell. We mark the performance improvement of GPT2-rephrasing over BERT-tagging and ReLM over previous SotA.

SIGHAN SIGHAN (Tseng et al. 2015) is a CSC benchmark collected from the Chinese essays written by foreign speakers. Following Wu et al. (2023c), we evaluate the model on SIGHAN as zero-shot learning.

AFQMC Ant Financial Question Matching (AFQMC) (Xu et al. 2020) is a Chinese semantic similarity dataset that requires the model to predict whether the given two questions are semantically similar. It contains 34,334 training samples and 3,861 test samples.

TNEWS TouTiao Text Classification for News Titles (TNEWS) (Xu et al. 2020) is a text classification dataset, requiring to link each given title to 15 news categories. It contains 53,360 training samples and 10,000 test samples.

Methods to Compare

$\text{BERT}_{\text{Tagging}}$ We fine-tune the original BERT model as sequence tagging[3].

$\text{MDCSpell}_{\text{Tagging}}$ It is an enhanced BERT-based model with a detector-corrector design (Zhu et al. 2022).

$\text{GPT2}_{\text{Tagging}}$ We initialize a new classifier following the pre-trained Chinese GPT2 model and fine-tune it as sequence tagging[4].

Masked-Fine-Tuning (MFT) It is a simple and effective fine-tuning technique when fine-tuning tagging models,

[3]https://huggingface.co/bert-base-chinese

[4]https://huggingface.co/uer/gpt2-chinese-cluecorpussmall

	GAM	ENC	COT	MEC	CAR	NOV	NEW	SIG	Avg
Previous tagging models SotA (Wu et al. 2023c)									
BERT	27.1	41.6	63.9	47.9	47.6	34.2	50.7	50.6	45.5
BERT-MFT	**33.3**	45.5	64.1	50.9	52.3	36.0	56.0	53.4	48.9
Soft-Masked-MFT	29.8	44.6	65.0	49.3	52.0	37.8	55.8	53.4	48.4
MDCSpell-MFT	31.2	45.9	65.4	52.0	52.6	**38.6**	57.3	54.7	49.7
CRASpell-MFT	30.7	48.1	66.0	51.7	51.7	**38.6**	55.9	55.1	49.7
ReLM (*Ours*)	33.0	**49.2**$_{\uparrow 3.7}$	**66.8**$_{\uparrow 2.7}$	**54.0**$_{\uparrow 3.1}$	**53.1**$_{\uparrow 0.8}$	37.8$_{\uparrow 1.8}$	**58.5**$_{\uparrow 2.5}$	**57.0**$_{\uparrow 3.6}$	**51.2**$_{\uparrow 2.3}$

Table 2: Performances (F1) of ReLM and previous SotA tagging models on LEMON, where SIG refers to SIGHAN. We mark the performance improvement of ReLM over BERT-MFT.

which achieves the previous state-of-the-art (SotA) results on ECSpell and LEMON (Wu et al. 2023c).

Baichuan2-7b We fine-tune Baichuan2 (Yang et al. 2023), one of the strongest Chinese open source LLMs, with LoRA (Hu et al. 2022).

ChatGPT We instruct ChatGPT (OpenAI 2023) to correct samples by in-context learning with 10 shots, using the *openai* API[5].

ReLM We train ReLM based on the same BERT model in BERT-tagging.

Fine-tuned CSC on ECSpell

We fine-tune each model separately on the three domains for 5000 steps, with the batch size selected from {32, 128} and learning rate from {2e-5, 5e-5}.

Table 1 summarizes the results on ECSpell. We find that naive tagging models of BERT and GPT2 perform poorly on all of three domains, while BERT performs slightly better than GPT2. However, ReLM yields amazing performance improvement. Concretely, it achieves the new SotA result on every domain, significantly outperforming the previous SotA masked-fine-tuned MDCSpell by 10.6, 12.8, and 16.7 absolute points respectively. We also apply the rephrasing objective to GPT2. We find that even GPT2-rephrasing outperforms BERT-tagging by a large margin (e.g. F1 39.8 $\rightarrow$ 71.2 on LAW), demonstrating the great superiority of the rephrasing objective, which prevents the model from simply memorizing errors.

However, we see ReLM is more powerful. It indicates that fixed-length rephrasing is naturally matched with CSC, while the auto-regressive one is also promising for future study. On the other hand, it is worth noting that ReLM surpasses all other enhanced architectures by simply training based on the original Transformer architecture. It highlights the pivotal role of the rephrasing objective, while the common tagging objective does not exploit the full power of PLMs, incurring the performance bottleneck.

We report the results of Baichuan2 and ChatGPT as representatives for LLMs. We find that ChatGPT does not work well on CSC even in a 10-shot setting. We speculate that this is due to the lack of high-quality annotated data for CSC on the web. However, fine-tuned Baichuan2 achieves promising results, outperforming GPT2-rephrasing by a large margin.

[5]gpt-3.5-turbo

Zero-Shot CSC on LEMON

On LEMON, we evaluate models as a zero-shot learner. Following Wu et al. (2023c), we collect 34 million monolingual sentences and synthesize training sentence pairs using the confusion set. We train the model with the batch size 4096 and learning rate 5e-5 on 8 A800 sheets for 60,000 steps.

Table 2 compares the zero-shot performance of ReLM to previous SotA tagging models. We find that, though each LEMON domain varies greatly, ReLM brings a significant performance boost in almost every domain, and reaches the new SotA results, raising the previous SotA from 49.7 to 51.2. It indicates that ReLM is more generalizable to out-of-distribution errors over all other BERT-tagging variants.

CSC in Multi-Task

We train the multi-task model on three distinct tasks, ECSpell for CSC, AFQMC for semantic similarity, and TNEWS for news classification. The three datasets are mixed together and we uniformly sample one batch from them during training. We fine-tune each model on all tasks for 15 epochs, with the batch size selected from {32, 128} and learning rate from {2e-5, 5e-5}. For the tagging models, we train three task-specific linear classifiers and one shared encoder. For ReLM, we share the entirety of model parameters for all three tasks. For ReLM with prompt embeddings, we train an additional prompt embeddings for each task. The MFT technique is exclusively applied on CSC.

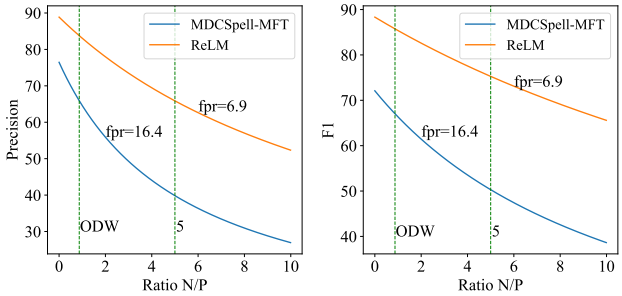
Table 3 compares the results on multiple tasks. We find that the performances of two text classification tasks vary only marginally between multi-task and single-task settings. We speculate that these two tasks are less challenging, and the model can fit them well more easily. In contrast, the performance of CSC is strongly affected by other tasks, where both tagging models meet a great performance drop. However, ReLM can largely maintain the CSC performance and achieve competitive results on all three domains, almost without compromising other tasks. Adding additional prompt characters can further improve the performance. It suggests that ReLM contributes to better collaboration between different tasks, on top of templating all tasks to the MLM format, while tagging-based CSC is incompatible to such a training paradigm. The logic behind is that ReLM retains the useful features within the pre-trained language representation of PLMs. In our following analysis, we show that tagging-based CSC learns non-transferable features.

	CSC Method	*CSC*		*News Classification*		*Semantic Similarity*		Avg
		F1	Δ F1$_{\text{single}}$ (%)	F1	Δ F1$_{\text{single}}$	F1	Δ F1$_{\text{single}}$	
LAW	BERT$_{\text{Tagging}}$	34.5	- 13.3%	56.2	- 0.4	72.6	**- 1.3**	54.4
	BERT$_{\text{Tagging}}$-MFT	62.0	- 18.9%	55.0	- 1.6	71.0	- 2.9	62.6
	ReLM	84.2	- 8.7%	56.9	+ 0.3	71.6	- 2.3	70.9
	ReLM (prompt)	**87.6**	**- 4.9%**	56.9	**+ 0.3**	72.4	- 1.5	**72.3**
MED	BERT$_{\text{Tagging}}$	15.1	- 32.2%	56.3	- 0.3	72.5	- 1.4	48.0
	BERT$_{\text{Tagging}}$-MFT	48.8	- 15.9%	55.8	- 0.8	72.5	**- 1.3**	59.0
	ReLM	76.1	- 10.9%	57.1	**+ 0.5**	70.7	- 3.2	68.0
	ReLM (prompt)	**80.8**	**- 4.7%**	56.6	+ 0	71.8	- 2.1	**69.7**
ODW	BERT$_{\text{Tagging}}$	16.8	- 32.2%	56.6	+ 0	73.3	**- 0.6**	48.9
	BERT$_{\text{Tagging}}$-MFT	52.4	- 11.5%	55.9	- 0.7	73.2	- 0.7	60.5
	ReLM	75.0	- 13.5%	56.9	**+ 0.3**	71.8	- 2.1	67.9
	ReLM (prompt)	**78.0**	**- 10.0%**	56.8	+ 0.2	72.5	- 1.4	**69.1**

Table 3: Results of different CSC training methods in the multi-task setting (ECSpell, TNEWS, AFQMC from left to right), where Δ F1$_{\text{single}}$ refers to the performance difference form multi-task to single-task, and % means the relative difference.

Method	LAW	MED	ODW	Avg
BERT$_{\text{Tag}}$	13.1	9.1	13.9	12.0
BERT$_{\text{Tag}}$ (multi-task)	13.8	10.2	15.5	13.2$_{\uparrow}$
BERT$_{\text{Tag}}$-MFT	14.7	11.2	15.5	13.8
BERT$_{\text{Tag}}$-MFT (multi-task)	14.7	11.6	18.5	14.9$_{\uparrow}$
MDCSpell$_{\text{Tag}}$-MFT	14.3	10.5	16.4	13.7
ReLM	**8.4**	**5.0**	**6.9**	**6.8**
ReLM (multi-task)	**7.4**	**6.5**	**2.2**	**5.4**$_{\downarrow}$

Table 4: Comparison of false positive rate (FPR) on ECSpell. It is excepted to be lower for a better CSC system.



(a) Precision (b) F1

Figure 3: Performance variation (precision and F1) with the proportion of negative and positive samples.

Further Analysis

False Positive Rate

False positive rate (FPR) is a measurement to evaluate a CSC system in real-world applications, which refers to the ratio that the model mistakenly modifies an otherwise correct sentence, which is also known as over-correction. Table 4 shows that ReLM greatly reduces the FPR compared to tagging models. It suggests the tagging models are overly conditioned on the seen errors and thus tend to modify some new expressions to familiar ones, while ReLM does not. Additionally, we find that ReLM produces even lower FPR in multi-task learning. It indicates that by ReLM, the language representation learned from CSC and other tasks can complement each other, while sequence tagging cannot.

We further demonstrate that a high FPR may result in a gap between the development performance and real-world practice. Mathematically, we have $1/\mathrm{p} \propto \frac{N}{P} \cdot \mathrm{FPR}$, where N and P refer to the number of negative samples and positive samples, and p is the precision score. We can find that p is negatively correlated with the ratio $\frac{N}{P}$, which means more negative samples lead to lower precision under the same FPR. However, negative samples are dominant in real-world situations ($\frac{N}{P}$ is large), since humans do not misspell very frequently. We can derive similarly results for the F1 score. Consequently, a higher FPR may exacerbate the decrease of the overall performance of the CSC system.

CSC Method	LAW→ TNEWS	MED→ TNEWS	ODW→ TNEWS	TNEWS
BERT$_{\text{Tag}}$	13.2$_{\downarrow 43.4}$	14.8$_{\downarrow 41.8}$	15.7$_{\downarrow 40.9}$	56.6
BERT$_{\text{Tag}}$-MFT	16.1$_{\downarrow 40.5}$	17.6$_{\downarrow 39.0}$	18.5$_{\downarrow 38.1}$	56.6
ReLM	54.1$_{\downarrow 2.5}$	53.7$_{\downarrow 2.9}$	49.2$_{\downarrow 7.4}$	56.6

Table 5: Results where we intend to transfer the learned language representation from CSC to news classification.

Figure 3 depicts the variation of the precision and F1 with the proportion of positive and negative samples, comparing MDCSpell-MFT to ReLM on ECSpell-ODW. We find that both F1 and precision curves of ReLM are more gentle, which decrease more slowly with the increase of $\frac{N}{P}$. It highlights the practical value of ReLM for real applications.

Probing in Multi-Task

To investigate the transferability from CSC to other tasks, we perform a linear probing experiment (Aghajanyan et al. 2021). First, we fine-tune the model on CSC data (ECSpell). Second, we freeze the parameters of the encoder and initialize a new linear classifier following it. We fine-tune this

	LAW	MED	ODW	Avg
MDCSpell$_{\text{Tagging}}$-MFT	80.6	69.6	66.9	72.4
mask any	90.5	84.0	82.9	85.8
mask non error	92.2	85.4	86.7	88.1

Table 6: Comparison of different mask strategies.

classifer only on another task (TNEWS). The results of the linear probing reflect whether the learned features within the encoder are generalized to transfer to new tasks.

From Table 5, we find that both tagging models suffer from a severe drop when transferring their learned language representation from CSC to TNEWS. It suggests that sequence tagging does not learn any generalized features from CSC, even degrades the language representation of the original PLM. In contrast, we find that ReLM can transfer much better, suggesting that it retains generalized features within the language representation during fine-tuning.

Mask Strategy

We investigate two mask strategies of the auxiliary MLM when training ReLM. The first is to uniformly mask any characters in the sentence and the second is to mask non-error characters only. From Table 6, we can see that both mask strategies are effective, while masking non-error characters works better. This is because masking error characters can reduce the amount of the real errors within the training data, which the model needs for learning correction. Additionally, we find that adding learnable prompts can further improve the performance of ReLM. The results in Table 6 and Table 7 are based on ReLM with learnable prompts.

Mask Rate

We also investigate the impact of the mask rate. From Table 7, we find that the performance on ECSpell keeps improving when the mask rate grows from 0% to 30%, and 30% is the best choice. While it still relies on specific data, it shows that the mask rate for ReLM is higher than that for MLM (Devlin et al. 2019). This is because ReLM is essentially a further refinement of PLMs, which allows a higher mask rate.

Case Study

We illustrate the superiority of ReLM over tagging through a number of cases selected from the evaluation results of masked-fine-tuned BERT-tagging and ReLM in Figure 4.

For the first case (*Director of People's Hospital in Daxingan Ridge.*), BERT-tagging overly corrects a geopolitical place "Daxingan Ridge" to "Daxingan Mountain", which is due to the fact that it doggedly memorizes a previous edit pair "ridge" → "mountain" during training. However, we can see that ReLM does not make this mistake.

For the second case (*Worried it develops towards muscle and bone.*), it highlights the ability of ReLM to utilize the semantics of global context. Two expressions "like" and "towards" are all locally correct, while to reach the correct result, the model should refer to the word "develop" located at the end of the sentence (the Chinese order).

	LAW	MED	ODW	Avg
BERT$_{\text{Tagging}}$	37.9	22.3	25.0	28.4
ReLM-0%	57.6	56.9	59.0	57.8
10%	90.0	84.2	82.5	85.6
20%	91.3	84.8	**86.9**	87.7
30%	**92.2**	**85.4**	86.7	**88.1**
40%	91.3	82.8	84.9	86.3
60%	86.7	81.7	78.8	82.4

Table 7: Comparison of different mask rates.

Figure 4: Cases selected from ECSpell.

The third case is quite puzzling (*Judicial power is not execution, but judgemental power.*), especially the first error. The correct answer not only necessitates the semantics but also a legal principle that "judicial power is judgemental power", which can only be attained through the pre-training process. We find that the tagging model does not possess such expertise and its answer is "law enforcement power is judgemental power". It suggests that ReLM effectively inherits the knowledge of PLMs, while the tagging model does not even enhanced with masked-fine-tuning.

Conclusion

This papers notes a critical flaw in current CSC learning, that is conventional sequence tagging allows the correction excessively conditioned on errors, leading to limited generalizability. To address this, we propose *ReLM*, where rephrasing acts as the training objective, akin to human spelling correction. ReLM greatly outweighs previous methods on prevailing benchmarks and facilitates multi-task learning.

Acknowledgements

This paper was partially supported by Joint Research Project of Yangtze River Delta Science and Technology Innovation Community (No. 2022CSJGG1400).

References

Afli, H.; Qiu, Z.; Way, A.; and Sheridan, P. 2016. Using SMT for OCR Error Correction of Historical Texts. In Calzolari, N.; Choukri, K.; Declerck, T.; Goggi, S.; Grobelnik, M.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016*. European Language Resources Association (ELRA).

Aghajanyan, A.; Shrivastava, A.; Gupta, A.; Goyal, N.; Zettlemoyer, L.; and Gupta, S. 2021. Better Fine-Tuning by Reducing Representational Collapse. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Cheng, X.; Xu, W.; Chen, K.; Jiang, S.; Wang, F.; Wang, T.; Chu, W.; and Qi, Y. 2020. SpellGCN: Incorporating Phonological and Visual Similarities into Language Models for Chinese Spelling Check. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 871–881. Association for Computational Linguistics.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Gao, J.; Li, X.; Micol, D.; Quirk, C.; and Sun, X. 2010. A Large Scale Ranker-Based System for Search Query Spelling Correction. In Huang, C.; and Jurafsky, D., eds., *COLING 2010, 23rd International Conference on Computational Linguistics, Proceedings of the Conference, 23-27 August 2010, Beijing, China*, 358–366. Tsinghua University Press.

Guo, Z.; Ni, Y.; Wang, K.; Zhu, W.; and Xie, G. 2021. Global Attention Decoder for Chinese Spelling Error Correction. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, 1419–1428. Association for Computational Linguistics.

He, P.; Gao, J.; and Chen, W. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 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.

Huang, L.; Li, J.; Jiang, W.; Zhang, Z.; Chen, M.; Wang, S.; and Xiao, J. 2021. PHMOSpell: Phonological and Morphological Knowledge Guided Chinese Spelling Check. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *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*, 5958–5967. Association for Computational Linguistics.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In Moens, M.; Huang, X.; Specia, L.; and Yih, S. W., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, 3045–3059. Association for Computational Linguistics.

Li, C.; Zhang, C.; Zheng, X.; and Huang, X. 2021. Exploration and Exploitation: Two Ways to Improve Chinese Spelling Correction Models. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *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*, 441–446. Association for Computational Linguistics.

Liu, S.; Yang, T.; Yue, T.; Zhang, F.; and Wang, D. 2021a. PLOME: Pre-training with Misspelled Knowledge for Chinese Spelling Correction. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *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*, 2991–3000. Association for Computational Linguistics.

Liu, X.; Zheng, Y.; Du, Z.; Ding, M.; Qian, Y.; Yang, Z.; and Tang, J. 2021b. GPT Understands, Too. *CoRR*, abs/2103.10385.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Lv, Q.; Cao, Z.; Geng, L.; Ai, C.; Yan, X.; and Fu, G. 2022. General and Domain Adaptive Chinese Spelling Check with Error Consistent Pretraining. *CoRR*, abs/2203.10929.

Martins, B.; and Silva, M. J. 2004. Spelling Correction for Search Engine Queries. In González, J. L. V.; Martínez-Barco, P.; Muñoz, R.; and Saiz-Noeda, M., eds., *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*, 372–383. Springer.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *CoRR*, abs/2302.13971.

Tseng, Y.; Lee, L.; Chang, L.; and Chen, H. 2015. Introduction to SIGHAN 2015 Bake-off for Chinese Spelling Check. In Yu, L.; Sui, Z.; Zhang, Y.; and Ng, V., eds., *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing,*

SIGHAN@IJCNLP 2015, Beijing, China, July 30-31, 2015, 32–37. Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Wang, D.; Song, Y.; Li, J.; Han, J.; and Zhang, H. 2018. A Hybrid Approach to Automatic Corpus Generation for Chinese Spelling Check. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2517–2527. Association for Computational Linguistics.

Wang, D.; Tay, Y.; and Zhong, L. 2019. Confusionset-guided Pointer Networks for Chinese Spelling Check. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 5780–5785. Association for Computational Linguistics.

Wu, H.; Ding, R.; Zhao, H.; Chen, B.; Xie, P.; Huang, F.; and Zhang, M. 2022. Forging Multiple Training Objectives for Pretrained Language Models via Meta-Learning. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Findings of the Association for Computational Linguistics: EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, 6454–6466. Association for Computational Linguistics.

Wu, H.; Ding, R.; Zhao, H.; Xie, P.; Huang, F.; and Zhang, M. 2023a. Adversarial Self-Attention for Language Understanding. In Williams, B.; Chen, Y.; and Neville, J., eds., *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence, IAAI 2023, Thirteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2023, Washington, DC, USA, February 7-14, 2023*, 13727–13735. AAAI Press.

Wu, H.; Liu, L.; Zhao, H.; and Zhang, M. 2023b. Empower Nested Boolean Logic via Self-Supervised Curriculum Learning. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, 13731–13742. Association for Computational Linguistics.

Wu, H.; Zhang, S.; Zhang, Y.; and Zhao, H. 2023c. Rethinking Masked Language Modeling for Chinese Spelling Correction. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 10743–10756. Association for Computational Linguistics.

Xie, W.; Huang, P.; Zhang, X.; Hong, K.; Huang, Q.; Chen, B.; and Huang, L. 2015. Chinese Spelling Check System Based on N-gram Model. In Yu, L.; Sui, Z.; Zhang, Y.; and Ng, V., eds., *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2015, Beijing, China, July 30-31, 2015*, 128–136. Association for Computational Linguistics.

Xiong, J.; Zhang, Q.; Zhang, S.; Hou, J.; and Cheng, X. 2015. HANSpeller: A Unified Framework for Chinese Spelling Correction. *Int. J. Comput. Linguistics Chin. Lang. Process.*, 20(1).

Xu, L.; Hu, H.; Zhang, X.; Li, L.; Cao, C.; Li, Y.; Xu, Y.; Sun, K.; Yu, D.; Yu, C.; Tian, Y.; Dong, Q.; Liu, W.; Shi, B.; Cui, Y.; Li, J.; Zeng, J.; Wang, R.; Xie, W.; Li, Y.; Patterson, Y.; Tian, Z.; Zhang, Y.; Zhou, H.; Liu, S.; Zhao, Z.; Zhao, Q.; Yue, C.; Zhang, X.; Yang, Z.; Richardson, K.; and Lan, Z. 2020. CLUE: A Chinese Language Understanding Evaluation Benchmark. In Scott, D.; Bel, N.; and Zong, C., eds., *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, 4762–4772. International Committee on Computational Linguistics.

Yang, A.; Xiao, B.; Wang, B.; Zhang, B.; Bian, C.; Yin, C.; Lv, C.; Pan, D.; Wang, D.; Yan, D.; Yang, F.; Deng, F.; Wang, F.; Liu, F.; Ai, G.; Dong, G.; Zhao, H.; Xu, H.; Sun, H.; Zhang, H.; Liu, H.; Ji, J.; Xie, J.; Dai, J.; Fang, K.; Su, L.; Song, L.; Liu, L.; Ru, L.; Ma, L.; Wang, M.; Liu, M.; Lin, M.; Nie, N.; Guo, P.; Sun, R.; Zhang, T.; Li, T.; Li, T.; Cheng, W.; Chen, W.; Zeng, X.; Wang, X.; Chen, X.; Men, X.; Yu, X.; Pan, X.; Shen, Y.; Wang, Y.; Li, Y.; Jiang, Y.; Gao, Y.; Zhang, Y.; Zhou, Z.; and Wu, Z. 2023. Baichuan 2: Open Large-scale Language Models. *CoRR*, abs/2309.10305.

Yang, Y.; Wu, H.; and Zhao, H. 2023. Attack Named Entity Recognition by Entity Boundary Interference. *CoRR*, abs/2305.05253.

Yeh, J.; Li, S.; Wu, M.; Chen, W.; and Su, M. 2013. Chinese Word Spelling Correction Based on N-gram Ranked Inverted Index List. In Yu, L.; Tseng, Y.; Zhu, J.; and Ren, F., eds., *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing, SIGHAN@IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, 43–48. Asian Federation of Natural Language Processing.

Yu, J.; and Li, Z. 2014. Chinese Spelling Error Detection and Correction Based on Language Model, Pronunciation, and Shape. In Sun, L.; Zong, C.; Zhang, M.; and Levow, G., eds., *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, October 20-21, 2014*, 220–223. Association for Computational Linguistics.

Zhang, R.; Pang, C.; Zhang, C.; Wang, S.; He, Z.; Sun, Y.; Wu, H.; and Wang, H. 2021. Correcting Chinese Spelling Errors with Phonetic Pre-training. In Zong, C.; Xia, F.; Li, W.; and Navigli, R., eds., *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, 2250–2261. Association for Computational Linguistics.

Zhang, S.; Huang, H.; Liu, J.; and Li, H. 2020. Spelling Error Correction with Soft-Masked BERT. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 882–890. Association for Computational Linguistics.

Zhou, Y.; Porwal, U.; and Konow, R. 2019. Spelling Correction as a Foreign Language. In Degenhardt, J.; Kallumadi, S.; Porwal, U.; and Trotman, A., eds., *Proceedings of the SIGIR 2019 Workshop on eCommerce, co-located with the 42st International ACM SIGIR Conference on Research and Development in Information Retrieval, eCom@SIGIR 2019, Paris, France, July 25, 2019*, volume 2410 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Zhu, C.; Ying, Z.; Zhang, B.; and Mao, F. 2022. MDCSpell: A Multi-task Detector-Corrector Framework for Chinese Spelling Correction. In Muresan, S.; Nakov, P.; and Villavicencio, A., eds., *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, 1244–1253. Association for Computational Linguistics.