The Past Mistake is the Future Wisdom: Error-driven Contrastive Probability Optimization for Chinese Spell Checking

Anonymous ACL submission

Abstract

Chinese Spell Checking (CSC) aims to detect and correct Chinese spelling errors, which are mainly caused by phonologically or visually similarity. Recently, due to the development of various pre-trained language models (PLMs), many CSC methods have achieved great progress. However, PLMs will pay more attention to common characters because of the pre-training settings. Therefore, there exists a gap between the learned knowledge of PLMs and the essential of CSC task. To address this issue, we propose an **E**rror-driven **CO**ntrastive **P**robability **O**ptimization (ECOPO) framework to refine the knowledge representation of PLMs for CSC. Particularly, ECOPO guides the model to avoid predicting common but improper characters through an error-driven way. Besides, ECOPO is model-agnostic so that it can be easily combined with existing CSC methods to achieve better performance. Extensive experiments[1] and detailed analysis on three standard benchmarks demonstrate that ECOPO is simple yet effective.

Phono-logical 83%	Input	希望您帮我素 (plain) 取公平。
	Correct	希望您帮我诉 (sue) 取公平。
	Candidate 1	希望您帮我争 (fight) 取公平。
	Candidate 2	希望您帮我谋 (plan) 取公平。
	Candidate 3	希望您帮我获 (acquire) 取公平。
	Translation	Hope you help me to sue and get justice.
Visual 48%	Input	我们为这个目标努力不解 (understand) 。
	Correct	我们为这个目标努力不懈 (slack) 。
	Candidate 1	我们为这个目标努力不休 (rest) 。
	Candidate 2	我们为这个目标努力不断 (break) 。
	Candidate 3	我们为这个目标努力不停 (stop) 。
	Translation	We fight for this goal without slack.

Figure 1: Examples of Chinese spelling errors. Previous research (Liu et al., 2021) shows that 83% of errors belong to phonological error and 48% belong to visual error. We give the characters with their pronunciation and translation. We mark the input confusing/golden confusing/common candidate characters in red/blue/orange. The characters in "Candidate" sentences are all predicted by fine-tuned BERT.

1 Introduction

Chinese Spell Checking (CSC) aims to detect and correct spelling errors in Chinese texts (Wu et al., 2013a). It is a crucial research field for various NLP downstream applications, such as Optical Character Recognition (OCR) (Afli et al., 2016), search query correction (Gao et al., 2010) and automatic essay scoring (Dong and Zhang, 2016). However, CSC is also a challenging task because it mainly suffers from confusing characters, such as phonologically and visually similar characters (Liu et al., 2010; Zhang et al., 2020). As illustrated in Figure 1, "素(sù, plain)" and "诉(sù, sue)" are confusing characters for each other due to the shared pronunciation "sù".

In recent years, pre-trained language models (PLMs) such as BERT (Devlin et al., 2019) have gradually been utilized in the CSC task and became the mainstream solutions (Zhang et al., 2020; Cheng et al., 2020; Xu et al., 2021). Although previous works have achieved good performance in the CSC task, they still have shortcomings to be improved. We notice that there exists a significant gap between the learned knowledge of PLMs and the essential of CSC task. The CSC task requires model to pay more attention to the confusing characters because the Chinese spelling errors are mainly caused by phonologically or visually similarity. However, limited by the masking strategy in pre-training procedure, general PLMs will be more inclined to common characters which would express the similar but improper semantic in the context. This kind of gap makes BERT-like PLMs be sub-optimal for CSC task (Liu et al., 2021).

Figure 1 presents two running examples of BERT to better understand the gap mentioned be-

[1]The source code will be available for reproducibility.

fore. The first phonological example is caused by the misuse of "素(sù, plain)" and "诉(sù, sue)". An ideal CSC model should pay attention to the pronunciation information "sù" and output the golden confusing character "诉(sue)" as a correction result for input confusing character. However, since BERT is pre-trained with a more general corpus, it will tend to predict more common but improper characters such as "争(zhēng, fight)", "谋(móu, plan)", "获(huò, acquire)". In the second visual example as well, BERT also ignores the visually similar information between "解(jiě, understand)" and "懈(xiè, slack)" and makes wrong correction.

To alleviate this gap, we propose to empower the PLMs to avoid predicting the above-mentioned common characters by optimizing the knowledge representation of PLMs. Intuitively, if we can guide the model to not make the same mistakes it would prone to make before, the performance of the model for the CSC task will be improved. Hence, the mistakes that the model has ever made can be utilized as constraints on the knowledge representation of the model. In other words, we hope the past mistakes that the model may make can be exploited to further enhance the model itself, this is the meaning of "the past mistake is the future wisdom". In our study, we perform error-driven optimization during the fine-tuning procedure of PLMs, thus narrowing the gap between the pre-trained knowledge of PLMs and the goal of CSC.

Motivated by the above intuition, we propose the **E**rror-driven **CO**ntrastive **P**robability **O**ptimization (ECOPO), a simple yet effective training framework which aims to refine the knowledge representation of models for CSC. The ECOPO consists of two stages: (1) *Negative samples selection*. Based on the model's prediction probability for different characters, we select the common but improper characters with high probability as negative samples. And we directly regard the golden confusing character as positive sample. (2) *Contrastive probability optimization*. After obtaining the positive/negative samples, we train the model by Contrastive Probability Optimization (CPO) objective which aims to optimize the prediction probability for different characters. Through this optimization process, we can finally adapt the model to the CSC task, and improve the model's performance.

In summary, our contributions are in three folds: (1) We firstly empirically observe and focus on the negative impact of the gap between the knowledge of PLMs and the CSC task. (2) We propose ECOPO, an error-driven optimization framework for CSC, which can teach the models to grow and progress with their own past mistakes. (3) We conduct extensive experiments and detailed analysis on three public datasets and achieve state-of-the-art performance with only a very thin model.

2 Related Work

2.1 Chinese Spell Checking

Chinese Spell Checking (CSC) is a promising task because of its broad application, such as OCR (Afli et al., 2016), Search Engine (Martins and Silva, 2004; Gao et al., 2010) and various education scenarios (Burstein and Chodorow, 1999; Lonsdale and Strong-Krause, 2003; Dong and Zhang, 2016). CSC has attracted more and more researchers, especially because of the recent rapid development of the education industry (Yu et al., 2014; Wang et al., 2018; Zhang et al., 2020; Cheng et al., 2020).

Previous CSC methods can be divided into three categories: rule-based methods, machine learning-based methods and deep learning-based methods. Early works in CSC mainly focus on designing heuristic rules to detect different kinds of errors (Chang et al., 2015; Chu and Lin, 2015). Most of these methods rely on solid linguistic knowledge and manually designed features, and thus do not have the generalization performance required for large-scale application. Next, various traditional machine learning algorithms, such as Conditional Random Field (CRF) and Hidden Markov Model (HMM), are applied in CSC task (Wang and Liao, 2015; Zhang et al., 2015). Then, deep learning-based models have gradually become the mainstream of the CSC field in recent years (Wang et al., 2021; Guo et al., 2021; Zhang et al., 2021).

Wang et al. (2018) utilize a BiLSTM trained on an automatically generated dataset to convert CSC to sequence labeling problem. Hong et al. (2019) propose to generate and curtail the candidate characters through a BERT-based denoising autoencoder. The Soft-Masked BERT model (Zhang et al., 2020) uses two separate networks for detection and correction. Then SpellGCN (Cheng et al., 2020) uses GCN (Kipf and Welling, 2017) to fuse character embedding with similar pronunciation and shape, explicitly modeling the relationship between characters. Additionally, REALISE (Xu et al., 2021) verifies that the multimodal knowledge can be leveraged to improve CSC performance.

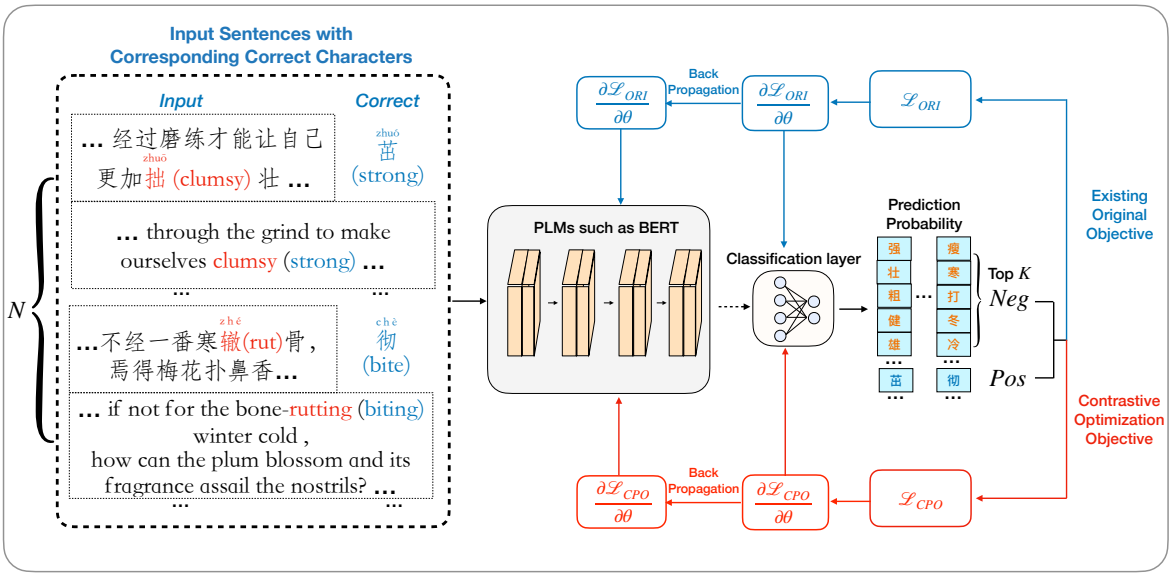


Figure 2: Overview of ECOPO framework. We select negative samples according to the original prediction probability of PLMs (e.g, for the position of "拙", PLMs predicts the Top 5 characters as "强", "壮", "粗", "健", and "雄".), then optimize the PLMs with the contrastive optimization objective and traditional original objective.

2.2 Pre-Trained Language Models

Recently, pre-trained language models (PLMs) have gained good improvements on various NLP tasks. The paradigm of fine-tuning PLMs for specific tasks has been widely used. In this paradigm, a model with fixed architecture is used to predict the probability of observed text data by pre-training as a language model. Take the pre-training setting of BERT (Devlin et al., 2019) as an example, Cui et al. (2020) use the Wikipedia dump which is general but not task-sensitive for CSC. Motivated by that BERT is designed and pre-trained independently from the CSC task, PLOME (Liu et al., 2021) is proposed to be a task-specific pre-trained language model for CSC. But unlike our method, PLOME designs a confusion set based masking strategy and introduces various external knowledge.

3 Methodology

In this section, we introduce the proposed ECOPO in details, as illustrated in Figure 2. ECOPO aims to refine the knowledge representation of PLMs to narrow the gap between it and the essential of CSC task. As mentioned in Section 1, with the model before our optimization process, we select the mistakes generated by this model itself to be the negative samples. Then through the Contrastive Probability Optimization objective, we maximize the prediction probability of the model for correct answers and minimize the prediction probability of the model for negative samples. In this error-driven way, the original prediction probability of the model is refined, improving the performance of the model on the CSC task. *Therefore, the model will grow and progress after making mistakes again and again, just as humans do.* Note that the proposed ECOPO is a model-agnostic framework, we can choose different PLMs or CSC models to be optimized in practice for better performance.

3.1 Observation and Intuition

To present our approach more clearly, we will firstly describe our observation, then we will give our explanation of the observation and intuition.

Based on our preliminary experiment of applying BERT to the SIGHAN13/14/15 datasets, we notice that out of the total 491 wrong correction samples, 383 (78%) samples fail due to BERT predicting common but improper characters. Note that if a character co-occurs with the character before or after the error position more than 1000 times in wiki2019zh[2], we regard it as a common character. Therefore, the key empirical observation that ECOPO builds on is that PLMs such as BERT cannot focus well on the confusing characters that need to be paid more attention in the CSC task, as illustrated in Figure 1. We think that this gap comes mainly from the general corpus and paradigm used in the pre-training process of models. Taking the

[2]The general pre-training corpus which is from Wikipedia dump (as of February 7, 2019) and contains one million pages.

BERT as an example, its pre-training corpus is mainly from the text in wikipedia, which has a very low proportion of contexts containing confusing characters, as verfied in Section 4.7. Additionally, Devlin et al. (2019) randomly choose 15% of tokens in the entire corpus to be masked by a fixed token "[MASK]" and then recover them. This masking-recovering strategy makes the knowledge acquired by PLMs in pre-training process discontinuous in the CSC task (Liu et al., 2021). Because the size of confusing characters will be lower in the 15% of characters that are randomly selected.

In fact, there also exists the same challenge when humans correct spelling errors. When only given the context of input sentence without seeing the misspelling, they tend to associate the common character rather than the confusing character with the context. Therefore, humans or models would wrongly predict common characters. *Intuitively, if the model can be optimized with common characters through an error-driven way, then the model can certainly be further enhanced, just as humans get progress from the mistakes they have made.*

3.2 Stage 1: Negative Samples Selection

We define the negative samples in CSC as those common characters that will be incorrectly assigned high prediction probability by PLMs before our optimization process. According to our observation, negative samples that can form common collocations with the context tend to be assigned higher probability than the golden confusing character, leading the model to make wrong corrections. Therefore, we use a simple strategy based on the prediction probability to select the negative samples which we will utilize in the next stage.

Specifically, we use PLMs such as BERT to predict the original character for each input token based on the output of the last transformer layer. The prediction probability of the i-th token x_i in a sentence X is defined as:

$$p\left(y_i = j \mid X\right) = \text{softmax}\left(\boldsymbol{W}\boldsymbol{h_i} + \boldsymbol{b}\right)[j], \quad (1)$$

where $p\left(y_i = j \mid X\right)$ means the conditional probability that the i-th token x_i is predicted as the j-th character in the vocabulary of PLMs, $\boldsymbol{W} \in \boldsymbol{R}^{vocab \times hidden}$ and $\boldsymbol{b} \in \boldsymbol{R}^{vocab}$ are learnable parameters, $vocab$ is the size of vocabulary and the $hidden$ is the size of hidden state, $\boldsymbol{h_i} \in \boldsymbol{R}^{hidden}$ is hidden state output of PLMs for the i-th token x_i.

Based on the prediction probability, we can select the negative samples according to the magnitude of the probability. The negative samples set Neg is selected from the candidate set T as:

$$T = \{t \mid t \in V \ \ and \ \ t \neq t^{+}\}, \quad (2)$$

$$Neg = \underset{T' \subset T, |T'|=K}{\arg\max} \sum_{t^{-} \in T'} p\left(y_i = t^{-} \mid X\right), \quad (3)$$

where t^{-} and t^{+} mean the negative and positive samples, respectively. The negative samples t^{-} are selected from those tokens whose prediction probability is in the Top K of the vocabulary V, and the best value of K is selected empirically. It is worthy noted that the training process is supervised in the CSC task, so we can regard the golden confusing character as the positive sample t^{+}.

3.3 Stage 2: Contrastive Probability Optimization

After obtaining the positive/negative samples and their corresponding prediction probability, we train the model by Contrastive Probability Optimization (CPO) objective which is defined as:

$$\mathcal{L}_{CPO} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{K}\sum_{k=1}^{K}\{p\left(y_i = t^{+} \mid X\right) - p\left(y_i = t_k^{-} \mid X\right)\}, \quad (4)$$

where N is the batch size, K is the selected negative samples size, t_k^{-} is the k-th negative sample in Neg. The CPO objective aims to teach the model to increase the prediction probability for positive sample (i.e, confusing character) and decrease the prediction probability for negative samples (i.e, common characters) by the maximum likelihood of the difference between the original probability for positive and negative samples.

To preserve the generalization performance of the model, we can train both the existing original objective $\mathcal{L}_{ORI}$ and the CPO objective $\mathcal{L}_{CPO}$. The overall objective is defined as:

$$\mathcal{L} = \lambda_1\mathcal{L}_{ORI} + \lambda_2\mathcal{L}_{CPO}, \quad (5)$$

where λ_1 and λ_2 are weighting factors for two objectives. We use CrossEntropy loss function as the $\mathcal{L}_{ORI}$ for BERT in our experiments.

In practice, the training pseudocode of ECOPO is shown in Appendix A. As described in Equation 5, we can replace the $\mathcal{L}_{ORI}$ with other models' training objectives, so ECOPO is model-agnostic and it can be easily used in other PLMs or previous CSC methods to achieve further improvement.

4 Experiments

In this section, we will introduce the details of experiments and main results we obtained firstly. Then we will conduct detailed analysis and discussion to verify the effectiveness of our method.

4.1 Datasets

Training data. We conduct extensive experiments to investigate the effectiveness of our proposed ECOPO. Following most previous works (Zhang et al., 2020; Cheng et al., 2020; Liu et al., 2021; Xu et al., 2021), we use the same training data as them, including the training samples from SIGHAN13 (Wu et al., 2013b), SIGHAN14 (Yu et al., 2014), SIGHAN15 (Tseng et al., 2015) and the pseudo training samples (size of 271K, we denote this part of training samples as Wang271K in our paper) automatically generated by OCR-based and ASR-based methods (Wang et al., 2018).

Test data. In order to ensure the fairness of the experiments, we use the exact same test data as the baseline methods, from the test datasets of SIGHAN13, SIGHAN14 and SIGHAN15. Noted that the text of original SIGHAN datasets is in the Traditional Chinese, we pre-process these original datasets to the Simplified Chinese using the OpenCC[3]. This data conversion procedure has been widely used in previous works (Wang et al., 2019; Cheng et al., 2020; Zhang et al., 2020). The detailed statistic of the training/test data we use in our experiments is presented in Appendix B.

4.2 Baseline Methods

To evaluate the performance of ECOPO better, we select several advanced strong baseline methods:

- BERT (Devlin et al., 2019): The BERT is directly fine-tuned on the training data.
- Hybrid (Wang et al., 2018): It casts CSC into a sequence labeling problem and implements a supervised model, i.e., BiLSTM trained on an automatically generated dataset.
- FASpell (Hong et al., 2019): This model consists of a denoising autoencoder (DAE) and a decoder, where the DAE curtails the number of candidate characters.
- Soft-Masked BERT (Zhang et al., 2020): A neural architecture consists of a detection network and a correction network, where the detection network can help the correction network to learn the right context.
- SpellGCN (Cheng et al., 2020): An end-to-end model to integrate the confusion set to the correction model through GCNs.
- REALISE (Xu et al., 2021): A multimodal model which captures and mixes the semantic, phonetic and graphic information to improve the performance of CSC. It is the current state-of-the-art method on SIGHAN13/14 datasets.
- PLOME (Liu et al., 2021): The task-specific pre-trained masked language model which jointly learns how to understand language and correct spell errors. It is the current state-of-the-art method on SIGHAN15 dataset.

4.3 Evaluation Metrics

In terms of evaluation granularity, there are two levels of metrics, namely character/sentence-level. Obviously, the sentence-level metric is stricter than the character-level metric because there may be multiple wrong characters in a sentence. One sentence sample is considered to be correct only when all the wrong characters in it are detected and corrected successfully. Therefore, we report the sentence-level metrics for evaluation, which are widely used in previous works (Li et al., 2021; Huang et al., 2021; Xu et al., 2021).

Specifically, the metrics we report include Accuracy, Precision, Recall and F1 score for detection and correction levels. At the detection level, all locations of wrong characters in a sentence should be identical successfully. At the correction level, the model must not only detect but also correct all the erroneous characters with the gold standard.

4.4 Experimental Setup

All the source code of our experiments is implemented using Pytorch (Paszke et al., 2019) based on the Huggingface's implementation of Transformer library[4] (Wolf et al., 2020). The architecture of the BERT encoder we use in the related models is same as the $BERT_{BASE}$ model, which has 12 transformers layers with 12 attention heads and its hidden state size is 768. We initialize the BERT encoder with the weights of Chinese BERT-wwm model (Cui et al., 2020). We train ECOPO with the

[3]https://github.com/BYVoid/OpenCC

[4]https://github.com/huggingface/transformers

Dataset	Method	Detection Level				Correction Level			
		Acc	Pre	Rec	F1	Acc	Pre	Rec	F1
SIGHAN13	Hybrid (Wang et al., 2018)	-	54.0	69.3	60.7	-	-	-	52.1
	FASpell (Hong et al., 2019)	63.1	76.2	63.2	69.1	60.5	73.1	60.5	66.2
	SpellGCN (Cheng et al., 2020)	-	80.1	74.4	77.2	-	78.3	72.7	75.4
	BERT (Devlin et al., 2019)	77.0	85.0	77.0	80.8	77.4	83.0	75.2	78.9
	ECOPO (BERT)	81.7$^{\uparrow}$	87.2$^{\uparrow}$	81.7$^{\uparrow}$	84.4$^{\uparrow}$	80.7$^{\uparrow}$	86.1$^{\uparrow}$	80.6$^{\uparrow}$	83.3$^{\uparrow}$
	REALISE (Xu et al., 2021)	82.1	87.2	82.0	84.5	80.7	85.7	80.5	83.0
	ECOPO (REALISE)	**82.8**$^{\uparrow}$	**88.6**$^{\uparrow}$	**82.7**$^{\uparrow}$	**85.6**$^{\uparrow}$	**81.4**$^{\uparrow}$	**87.1**$^{\uparrow}$	**81.3**$^{\uparrow}$	**84.1**$^{\uparrow}$
SIGHAN14	Hybrid (Wang et al., 2018)	-	51.9	66.2	58.2	-	-	-	56.1
	FASpell (Hong et al., 2019)	70.0	61.0	53.5	57.0	69.3	59.4	52.0	55.4
	SpellGCN (Cheng et al., 2020)	-	65.1	69.5	67.2	-	63.1	67.2	65.3
	BERT (Devlin et al., 2019)	75.3	63.4	68.8	66.0	74.2	61.2	66.5	63.8
	ECOPO (BERT)	76.7$^{\uparrow}$	65.8$^{\uparrow}$	69.0$^{\uparrow}$	67.4$^{\uparrow}$	75.7$^{\uparrow}$	63.7$^{\uparrow}$	66.9$^{\uparrow}$	65.3$^{\uparrow}$
	REALISE (Xu et al., 2021)	78.3	67.2	71.5	69.3	77.3	65.2	69.4	67.2
	ECOPO (REALISE)	**78.9**$^{\uparrow}$	**68.2**$^{\uparrow}$	**72.1**$^{\uparrow}$	**70.1**$^{\uparrow}$	**78.0**$^{\uparrow}$	**66.4**$^{\uparrow}$	**70.2**$^{\uparrow}$	**68.2**$^{\uparrow}$
SIGHAN15	Hybrid (Wang et al., 2018)	-	56.6	69.4	62.3	-	-	-	57.1
	FASpell (Hong et al., 2019)	74.2	67.6	60.0	63.5	73.7	66.6	59.1	62.6
	SpellGCN (Cheng et al., 2020)	-	74.8	80.7	77.7	-	72.1	77.7	75.9
	PLOME (Liu et al., 2021)	-	77.4	**81.5**	79.4	-	75.3	79.3	77.2
	Soft-Masked BERT (Zhang et al., 2020)	79.7	69.8	73.4	71.6	76.9	64.4	67.7	66.0
	ECOPO (Soft-Masked BERT)	81.2$^{\uparrow}$	70.9$^{\uparrow}$	76.6$^{\uparrow}$	73.6$^{\uparrow}$	79.1$^{\uparrow}$	67.0$^{\uparrow}$	72.3$^{\uparrow}$	69.6$^{\uparrow}$
	BERT (Devlin et al., 2019)	82.4	74.2	78.0	76.1	81.0	71.6	75.3	73.4
	ECOPO (BERT)	**85.5**$^{\uparrow}$	**79.0**$^{\uparrow}$	81.3$^{\uparrow}$	**80.2**$^{\uparrow}$	**84.4**$^{\uparrow}$	**76.8**$^{\uparrow}$	79.1$^{\uparrow}$	**78.0**$^{\uparrow}$
	REALISE (Xu et al., 2021)	84.1	76.3	80.8	78.5	83.5	75.0	79.5	77.2
	ECOPO (REALISE)	84.8$^{\uparrow}$	76.6$^{\uparrow}$	**81.5**$^{\uparrow}$	79.0$^{\uparrow}$	84.3$^{\uparrow}$	75.5$^{\uparrow}$	**80.4**$^{\uparrow}$	77.9$^{\uparrow}$

Table 1: The performance of ECOPO and all baseline methods. Note that all baseline results are directly from other published paper, except for the results of Soft-Masked BERT and REALISE which are from our own re-implementation experiments. ECOPO (model-X) means that we perform ECOPO framework on model-X. We underline the previous state-of-the-art performance for convenient comparison. "$^{\uparrow}$" indicates that the corresponding baseline method receives a further performance improvement after optimization by ECOPO.

AdamW (Loshchilov and Hutter, 2018) optimizer for 10 epochs. The training batch size N is set to 64 and the evaluation batch size is set to 50. The negative samples size K is set to 5 by default. The weighting factors λ_1, λ_2 are both set to 1. The initial learning rate is set to 5e-5. We set the maximum sentence length to 128. The model is trained with learning rate warming up and linear decay.

It is worthy noted that the annotation quality of SIGHAN13 test dataset is relatively poor. As we have observed and mentioned in (Cheng et al., 2020; Xu et al., 2021), quite lots of the mixed usage of auxiliary (such as "的", "地", and "得") don't have correct annotations. Therefore, the evaluation metrics we use may not accurately reflect the real model performance on SIGHAN13. To alleviate this problem, there are two main solutions in previous works. Cheng et al. (2020) propose to continue fine-tuning well-trained models on the SIGHAN13 training dataset before testing, which we think will suffer from the over-fitting problem. Therefore, we follow the post-processing method proposed in (Xu et al., 2021) and don't consider all the detected and corrected mixed auxiliary. This approach does not compromise the fairness of the evaluation process and can better reflect the model performance.

4.5 Experimental Results

From Table 1, we can observe that:

1. The ECOPO (BERT) performs better than BERT on all test sets. At the correction level, ECOPO (BERT) exceeds BERT by 4.4% F1 on SIGHAN13, 1.5% F1 on SIGHAN14, and 4.6% F1 on SIGHAN15. Specifically, ECOPO (BERT) achieves significant improvements on SIGHAN13/SIGHAN15, and outperforms the previous state-of-the-art models with a very thin model, while REALISE and PLOME are two complex models with some auxiliary modules. Note that ECOPO (BERT) only consists of a BERT encoder.

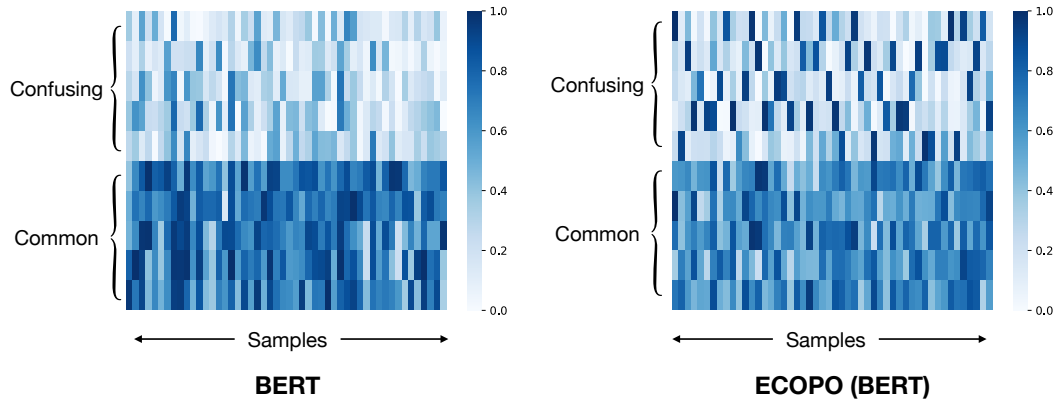


Figure 3: Heat map visualization of probability. The darker the blue, the higher the model's prediction probability for a particular character (vertical axis) given the input of samples containing misspelled characters (horizontal axis). The selected samples are from SIGHAN15, and the original BERT would make wrong corrections for them.

2. From the results on the SIGHAN14 test set, we can see that the performance improvement of ECOPO (BERT) based on BERT is not as large as on the other two test sets, but still effective. Additionally, due to the model-agnostic advantage of ECOPO, it can be simply combined with not only BERT but also other previous state-of-the-art models such as REALISE and get further enhancement to obtain better results, which are presented in the rows of REALISE and ECOPO (REALISE).

3. Considering the impact of external knowledge, several previous works exploit various additional information to improve performance. For example, FASpell and SpellGCN introduce character similarity to CSC, REALISE and PLOME propose to leverage multimodal knowledge such as phonetic and graphic information. Unlike the aforementioned models, ECOPO (BERT) achieves competitive performance without any additional knowledge and optimizing only based on the mistakes that the original BERT itself has made.

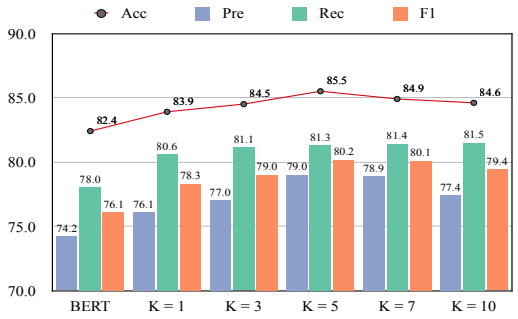
4. To verify the expandability of ECOPO, we choose two other existing models including Soft-Masked BERT and REALISE to be optimized. Practically, we train the combined model with the joint objective, as described in Equation 5. From the results of Table 1, we can see that ECOPO's improvement is stable and significant over the three models. In summary, comparison results of the three models demonstrate the effectiveness and model-agnostic characteristic of our method.

4.6 Analysis and Discussion

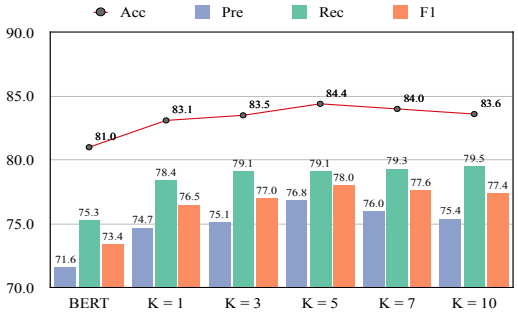
4.6.1 Visualization of Common/Confusing Character Probability

The key objective of ECOPO is to optimize the prediction probability of the PLMs for two different kinds of characters, i.e., **common characters** which original PLMs would be more inclined and **confusing characters** which CSC task should pay more attention to. Therefore, we visualize the probability optimization effect of ECOPO in this part of experiment. Specifically, we ask BERT and ECOPO (BERT) to predict the character which should appear at the position of the misspelled character based on its context. We select the Top-5 characters co-occurring with the context of the misspelled character as the common characters, and 5 confusing characters from the widely used confusion set (Wu et al., 2013b). Note that we ensure that the common and confusing characters selected are not duplicated, and the golden confusing character must be in the selected 5 confusing characters. Then we visualize the prediction probability of common/confusing characters as a heat map.

Figure 3 shows the prediction probability distributions of BERT and ECOPO (BERT) for the common/confusing characters. By comparison, we can see that BERT would assign higher probability to common characters than confusing characters, and ECOPO (BERT) will focus more on confusing characters which are similar to the golden confusing character. This difference in BERT before and after ECOPO's optimization is consistent with our study motivation and design objective, we can see that ECOPO does refine the knowledge representation



(a) Detection Performance



(b) Correction Performance

Figure 4: The results on SIGHAN15 test set, using different values of K in Equation 3 in ECOPO (BERT).

and prediction probability of BERT for different characters. After ECOPO's optimization, BERT is able to assign higher probability to the confusing characters that should receive more attention, thus improving its performance on CSC.

4.6.2 Effects of Negative Samples Size

As different amounts of negative samples can affect ECOPO's performance, it is essential to study the impact of negative samples size K in Equation 3.

Figure 4 illustrates the performance change from the perspective of detection and correction. We find that the Recall performance of the model exhibits incremental increases when more negative samples are used in the optimization process. This phenomenon is intuitive, as introducing more negative samples allows the model to focus on more possibilities. Besides, when the value of K reaches a certain value (e.g., $K > 5$), the overall performance of the model (F1 score) does not improve anymore. This is because ECOPO optimizes the model based on the probability representation, when the value of K becomes very large, the predicted probability of samples becomes so small that they have almost no effect on the probability optimization of the positive sample. Therefore, choosing an appropriate K value is critical to the performance improvement of ECOPO, although ECOPO has significant improvement based on BERT at all values of K.

4.7 Case Study for Probability Optimization

Input:	与其自暴自气 (弃) 不如往好处想。 It's better to think for the good than to be angry (give up).
BERT:	[己(own), 大(big), 利(benefit)]
ECOPO:	[弃(give up), 尊(respect), 强(strong)]
Input:	我努力打败数不进 (尽)的风雨。 I try to beat the enter (endless) storms.
BERT:	[起(raise), 上(up), 得(get)]
ECOPO:	[尽(endless), 得(get), 完(end)]

Table 2: Examples of spelling errors and corresponding output (Top 3 candidates) of original BERT and ECOPO (BERT). We mark the input confusing/golden confusing/wrong correction characters in red/blue/orange.

Table 2 shows the comparisons between the correction results of BERT and ECOPO (BERT). In the first examples, the output of BERT such as "己", "大" and "利" all can form a correct Chinese phrase with "自", but they cause a semantic incoherence for the whole sentence. The statistics of the general pre-training corpus wiki2019zh show that "自己" co-occurs 136318 times and "自弃" co-occurs 119 times, which verifies the intuition about common/confusing characters described in Section 3.1. In the second example as well, the output of BERT can be formed with "数不" as reasonable phrases. From the two examples, we can see that ECOPO does guide the BERT to accurately predict the ideal confusing characters by the highest probability and make the right corrections. Such experimental results are in line with our work's core motivation.

5 Conclusion

In this paper, we introduce to promote the CSC task by narrowing the gap between the knowledge of PLMs and the goal of CSC. We propose the ECOPO, a simple yet effective training framework that aims to perform an error-driven optimization for the PLMs based on their original probability representation. Extensive experiments and empirical results show the competitive performance of our method. In the future, we will study how to automatically measure the quality of negative samples to further enhance our method. Additionally, applying our core idea and motivation to kinds of other tasks will be an interesting direction.

References

Haithem Afli, Zhengwei Qiu, Andy Way, and Páraic Sheridan. 2016. Using SMT for OCR error correction of historical texts. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 962–966, Portorož, Slovenia. European Language Resources Association (ELRA).

Jill Burstein and Martin Chodorow. 1999. Automated essay scoring for nonnative English speakers. In *Computer Mediated Language Assessment and Evaluation in Natural Language Processing*.

Tao-Hsing Chang, Hsueh-Chih Chen, and Cheng-Han Yang. 2015. Introduction to a proofreading tool for Chinese spelling check task of SIGHAN-8. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 50–55, Beijing, China. Association for Computational Linguistics.

Xingyi Cheng, Weidi Xu, Kunlong Chen, Shaohua Jiang, Feng Wang, Taifeng Wang, Wei Chu, and Yuan Qi. 2020. SpellGCN: Incorporating phonological and visual similarities into language models for Chinese spelling check. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 871–881, Online. Association for Computational Linguistics.

Wei-Cheng Chu and Chuan-Jie Lin. 2015. NTOU Chinese spelling check system in sighan-8 bake-off. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 137–143, Beijing, China. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

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

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring – an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1072–1077, Austin, Texas. 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*, pages 1419–1428, Online. Association for Computational Linguistics.

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

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 (Volume 1: Long Papers)*, pages 5958–5967, Online. Association for Computational Linguistics.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

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 (Volume 2: Short Papers)*, pages 441–446, Online. Association for Computational Linguistics.

Chao-Lin Liu, Min-Hua Lai, Yi-Hsuan Chuang, and Chia-Ying Lee. 2010. Visually and phonologically similar characters in incorrect simplified Chinese words. In *Coling 2010: Posters*, pages 739–747, Beijing, China. Coling 2010 Organizing Committee.

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 (Volume 1: Long Papers)*, pages 2991–3000, Online. Association for Computational Linguistics.

Deryle Lonsdale and Diane Strong-Krause. 2003. Automated rating of ESL essays. In *Proceedings of the HLT-NAACL 03 Workshop on Building Educational Applications Using Natural Language Processing*, pages 61–67.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Bruno Martins and Mário J Silva. 2004. Spelling correction for search engine queries. In *International Conference on Natural Language Processing (in Spain)*, pages 372–383. Springer.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037.

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

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

Dingmin Wang, Yan Song, Jing Li, Jialong Han, and Haisong Zhang. 2018. A hybrid approach to automatic corpus generation for Chinese spelling check. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2517–2527, Brussels, Belgium. 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 Annual Meeting of the Association for Computational Linguistics*, pages 5780–5785, Florence, Italy. Association for Computational Linguistics.

Yih-Ru Wang and Yuan-Fu Liao. 2015. Word vector/conditional random field-based Chinese spelling error detection for SIGHAN-2015 evaluation. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 46–49, Beijing, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Jian-cheng Wu, Hsun-wen Chiu, and Jason S. Chang. 2013a. Integrating dictionary and web n-grams for Chinese spell checking. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 18, Number 4, December 2013-Special Issue on Selected Papers from ROCLING XXV*.

Shih-Hung Wu, Chao-Lin Liu, and Lung-Hao Lee. 2013b. Chinese spelling check evaluation at SIGHAN bake-off 2013. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 35–42, Nagoya, Japan. Asian Federation of Natural Language Processing.

Heng-Da Xu, Zhongli Li, Qingyu Zhou, Chao Li, Zizhen Wang, Yunbo Cao, Heyan Huang, and Xian-Ling Mao. 2021. Read, listen, and see: Leveraging multimodal information helps Chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 716–728, Online. Association for Computational Linguistics.

Liang-Chih Yu, Lung-Hao Lee, Yuen-Hsien Tseng, and Hsin-Hsi Chen. 2014. Overview of SIGHAN 2014 bake-off for Chinese spelling check. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 126–132, Wuhan, China. 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*, pages 2250–2261, Online. 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*, pages 882–890, Online. Association for Computational Linguistics.

Shuiyuan Zhang, Jinhua Xiong, Jianpeng Hou, Qiao Zhang, and Xueqi Cheng. 2015. HANSpeller++: A unified framework for Chinese spelling correction. In *Proceedings of the Eighth SIGHAN Workshop on Chinese Language Processing*, pages 38–45, Beijing, China. Association for Computational Linguistics.

```
# vocab_prob : the prediction probability for all characters in vocabulary
# pos_idx    : the index of positive sample (golden character) in vocabulary
# K          : the selected negative samples amount

# Negative Samples Selection
pos_prob = vocab_prob[pos_idx]
neg_prob = torch.topk(vocab_prob, K)[0]
neg_idx = torch.topk(vocab_prob, K)[1].tolist()

# Contrastive Probability Optimization Objective
loss_list = []
for x in range(0, K):
   if neg_idx[x] != pos_idx:
      loss_list.append(pos_prob - neg_prob[x])
loss = - torch.stack(loss_list).mean()
```

Figure 5: Pseudo-code of our practical implementation.

A Pseudo-code of ECOPO

Figure 5 shows the Pytorch-style pseudo-code for the ECOPO. As described in Section 3, our proposed ECOPO consists of two stages, namely Negative Samples Selection and Contrastive Probability Optimization. It is worthy noting that in the pseudo-code, we only show the process of calculating the loss of one training sample.

B Datasets Details

Table 3 shows the detailed statistics of our used datasets. We report the number of sentences in the datasets (#Sent), the average sentence length of the datasets (Avg.Length), and the number of misspellings the datasets contains (#Errors).

Training Data	#Sent	Avg. Length	#Errors
SIGHAN13	700	41.8	343
SIGHAN14	3,437	49.6	5,122
SIGHAN15	2,338	31.3	3,037
Wang271K	271,329	42.6	381,962
Total	277,804	42.6	390464
Test Data	**#Sent**	**Avg. Length**	**#Errors**
SIGHAN13	1,000	74.3	1,224
SIGHAN14	1,062	50.0	771
SIGHAN15	1,100	30.6	703
Total	3,162	50.9	2,698

Table 3: Statistics of the datasets that we use in experiments. All the training data are merged to train the models in our experiments. The test sets are used separately to evaluate performance.