PLOME: Pre-training with Misspelled Knowledge for Chinese Spelling Correction

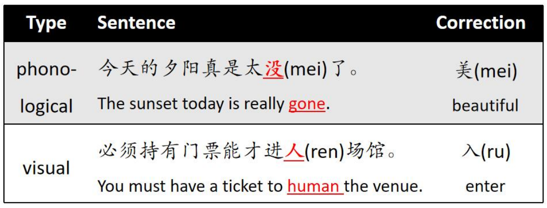
Shulin Liu, Tao Yang, Tianchi Yue, Feng Zhang, Di Wang
Tencent AI Platform Department, China
{forestliu, rigorosyang, tianchiyue}@tencent.com
{jayzhang, diwang}@tencent.com

Abstract

Chinese spelling correction (CSC) is a task to detect and correct spelling errors in texts. CSC is essentially a linguistic problem, thus the ability of language understanding is crucial to this task. In this paper, we propose a **P**re-trained masked **L**anguage m**O**del with **M**isspelled knowledg**E** (PLOME) for CSC, which jointly learns how to understand language and correct spelling errors. To this end, PLOME masks the chosen tokens with similar characters according to a confusion set rather than the fixed token "[MASK]" as in BERT. Besides character prediction, PLOME also introduces pronunciation prediction to learn the misspelled knowledge on phonic level. Moreover, phonological and visual similarity knowledge is important to this task. PLOME utilizes GRU networks to model such knowledge based on characters' phonics and strokes. Experiments are conducted on widely used benchmarks. Our method achieves superior performance against state-of-the-art approaches by a remarkable margin. We release the source code and pre-trained model for further use by the community[1].

1 Introduction

Chinese spelling correction (CSC) aims to detect and correct spelling errors in texts (Yu and Li, 2014). It is a challenging yet important task in natural language processing, which plays an important role in various NLP applications such as search engine (Martins and Silva, 2004) and optical character recognition (Afli et al., 2016). In Chinese, spelling errors can be mainly divided into two types: phonological errors and visual errors, which are separately caused by the misuse of phonologically similar characters and visually similar characters. According to Liu et al. (2010), about 83% of errors are phonological and 48% are visual. Figure 1 illustrates examples of such errors. The first case is caused by the misuse of "没(gone)" and "美(beautiful)" with the same phonics, and the second case is caused by the misuse of "人(human)" and "入(enter)" with very similar shape.



Type	Sentence	Correction
phono- logical	今天的夕阳真是太没(mei)了。 The sunset today is really gone.	美(mei) beautiful
visual	必须持有门票能才进人(ren)场馆。 You must have a ticket to human the venue.	入(ru) enter

Figure 1: Examples of Chinese spelling errors. Misspelling characters are marked in red, and the corresponding phonics are given in brackets.

Chinese spelling correction is a challenging task because it requires human-level language understanding ability to completely solve this problem (Zhang et al., 2020). Therefore, language model plays an important role in CSC. In fact, one of the mainstream solutions to this task is based on language models (Chen et al., 2013; Yu and Li, 2014; Tseng et al., 2015). Currently, the latest approaches (Zhang et al., 2020; Cheng et al., 2020) are based on BERT (Devlin et al., 2019), which is a masked language model. In these approaches, (masked) language models are independently pre-trained from the CSC task. As a consequence, they did not learn any task-specific knowledge during pre-training. Therefore, language models in these approaches are sub-optimal for CSC.

Chinese spelling errors are mainly caused by the misuse of phonologically or visually similar characters. Thus, knowledge of the similarity between characters is crucial to this task. Some work leveraged the confusion set, i.e. a set of similar characters, to fuse such information (Wang et al., 2018, 2019; Zhang et al., 2020). However, confu-

[1] https://github.com/liushulinle/PLOME

*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics
and the 11th International Joint Conference on Natural Language Processing*, pages 2991–3000
August 1–6, 2021. ©2021 Association for Computational Linguistics

sion set is usually generated by heuristic rules or manual annotations, thus its coverage is limited. To circumvent this problem, Hong et al. (2019) computed the similarity based on character's strokes and phonics. The similarity was measured via rules rather than learned by the model, therefore such knowledge was not fully utilized.

In this paper, we propose PLOME, a **P**re-trained masked **L**anguage m**O**del with **M**isspelled knowledg**E**, for Chinese spelling correction. The following characteristics make PLOME more effective than vanilla BERT for CSC. First, we propose the confusion set based masking strategy, where each chosen token is randomly replaced by a similar character according to a confusion set rather than the fixed token "[MASK]" as in BERT. Thus, PLOME jointly learns the semantics and misspelled knowledge during pre-training. Second, the proposed model takes each character's strokes and phonics as input, which enables PLOME to model the similarity between arbitrary characters. Third, PLOME learns the misspelled knowledge on both character and phonic level by jointly recovering the true character and phonics for masked tokens.

We conduct experiments on the widely used benchmark dataset SIGHAN (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015). Experimental results show that PLOME significantly outperforms all the compared approaches, including the latest Soft-masked BERT (Zhang et al., 2020) and SpellGCN (Cheng et al., 2020).

We summarize our contributions as follows: (1) PLOME is the first task-specific language model designed for Chinese spelling correction. The proposed confusion set based masking strategy enables our model to jointly learn the semantics and misspelled knowledge during pre-training. (2) PLOME incorporates phonics and strokes, which enables it to model the similarity between arbitrary characters. (3) PLOME is the first to model this task on both character and phonic level.

2 Related Work

Chinese spelling correction is a challenging task in natural language processing, which plays important roles in many applications, such as search engine (Martins and Silva, 2004; Gao et al., 2010), automatic essay scoring (Burstein and Chodorow, 1999; Lonsdale and Strong-Krause, 2003), and optical character recognition (Afli et al., 2016; Wang et al., 2018). It has been an active topic, and various approaches have been proposed in recent years (Yu and Li, 2014; Wang et al., 2018, 2019; Zhang et al., 2020; Cheng et al., 2020).

Early work on CSC followed the pipeline of error identification, candidate generation and selection. Some researchers focused on unsupervised approaches, which typically adopted a confusion set to find correct candidates and employed language model to select the correct one (Chang, 1995; Huang et al., 2000; Chen et al., 2013; Yu and Li, 2014; Tseng et al., 2015). However, these methods failed to condition the correction on the input sentence. In order to model the input context, discriminative sequence tagging methods (Wang et al., 2018) and sequence-to-sequence generative models (Chollampatt et al., 2016; Ji et al., 2017; Ge et al., 2018; Wang et al., 2019) were employed.

BERT (Devlin et al., 2019) is a bidirectional language model based on Transformer encoder (Vaswani et al., 2017). It has been demonstrated effective in a wide range of applications, such as question answering (Yang et al., 2019), information extraction (Lin et al., 2019), and semantic matching (Reimers and Gurevych, 2019). Recently, it has dominated the researches on CSC (Hong et al., 2019; Zhang et al., 2020; Cheng et al., 2020). Hong et al. (2019) adopted the DAE-Decoder paradigm with BERT as encoder. Zhang et al. (2020) introduced a detection network to generate the masking vector for the BERT-based correction network. Cheng et al. (2020) employed the graph convolution network (GCN) (Kipf and Welling, 2016) combined with BERT to model character interdependence. However, BERT is designed and pre-trained independently from the CSC task, thus it is sub-optimal. To improve the performance, we propose a task-specific language model for CSC.

3 Approach

We introduce PLOME and its detailed implementation in this section. Figure 2 illustrates the framework of PLOME. Similar to BERT (Devlin et al., 2019), the proposed model also follows the pre-training&fine-tuning paradigm. In the following subsections, we first introduce the confusion set based masking strategy, then present the architecture of PLOME and the learning objectives, finally show the details of fine-tuning.

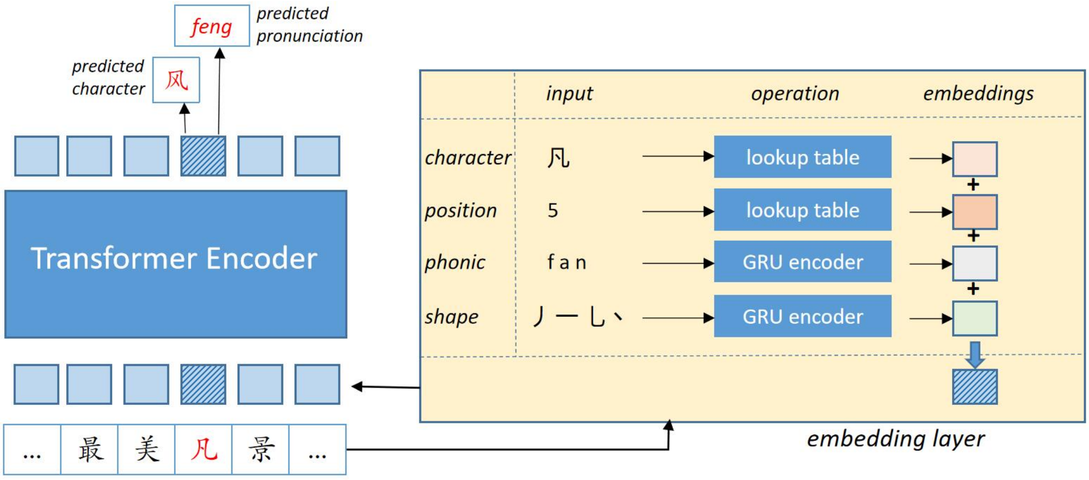


Figure 2: The framework of the proposed PLOME, where the masked token is marked in red. **Left:** This component illustrates the overall architecture of the proposed model. Input characters are processed by the transformer encoder to obtain semantic representation vectors. **Right:** This component collects different types of embeddings for each character to obtain the final embedding for the transformer encoder.

3.1 Confusion Set based Masking Strategy

In order to train PLOME, we randomly mask some percentage of the input tokens and then recover them. Devlin et al. (2019) replaced the chosen tokens by a fixed token "[MASK]", which is nonexistent in downstream tasks. On the contrast, we remove this token and replace each chosen token by a random character that is similar to it. Similar characters are obtained from a publicly available confusion set (Wu et al., 2013), which contains two types of similar characters: phonologically similar and visually similar. Since phonological errors are two times more frequent than visual errors (Liu et al., 2010), these two types of similar characters are assigned different chance to be chosen during masking. Following Devlin et al. (2019), we totally mask 15% of tokens in the corpus. In addition, we use dynamic masking strategy (Liu et al., 2019), where the masking pattern is generated every time a sequence is fed into the model.

Always replacing chosen tokens by characters in a confusion set will cause two problems. (1). The model tends to make correction decision for all inputs since all the tokens to be predicted during pre-training are "misspelled". To circumvent this problem, some percentage of the selected tokens are unchanged. (2). The size of confusion set is limited, however misspelling may be caused by the misuse of an arbitrary pair of characters in real texts. To improve generalization ability, we replace some percentage of chosen tokens by random characters from the vocabulary. To sum up, if the i-th token is chosen, we replace it with (i) a random phonologically similar character 60% of the time (ii) a random visually similar character 15% of the time (iii) the unchanged i-th token 15% of the time (iv) a random token in the vocabulary 10% of the time. Table 1 presents examples of different masking strategies.

	Sentence
Original Sentence	他想明天去(qu)南京探望奶奶。
BERT Masking	他想明天[MASK]南京看奶奶。
Phonic Masking	他想明天曲(qu)南京看奶奶。
Shape Masking	他想明天丢(diu)南京看奶奶。
Random Masking	他想明天浩(hao)南京看奶奶。
Unchanging	他想明天去(qu)南京看奶奶。

Table 1: Examples of different masking strategies. The chosen token is marked in red, and the corresponding phonics is given in brackets.

3.2 Embedding Layer

As shown in Figure 2, the final embedding of each character is the sum of character embedding, position embedding, phonic embedding and shape embedding. The former two are obtained via looking up embedding tables, where the size of vocabulary and embedding dimension are the same as that in BERT$_{base}$ (Devlin et al., 2019).

Phonic Embedding In Chinese, phonics (also known as Pinyin) represents the pronunciation of a character, which is a sequence of lowercase letters

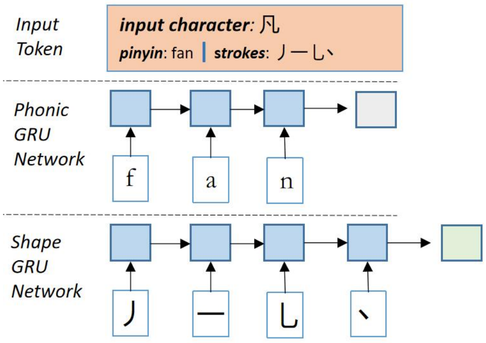


Figure 3: Illustration of phonic GRU network and shape GRU network.

with a diacritic[2]. In this paper, we use the Unihan Database[3] to obtain the character-phonics mapping (diacritic is removed). To model the phonological relationship between characters, we feed the letters of each character's phonics to a 1-layer GRU (Bahdanau et al., 2014) network to generate the phonic embedding, where similar phonics are expected to have similar embeddings. An example is given in the middle part in Figure 3.

Shape Embedding We use the Stroke Order[4] to represent the shape of a character, which is a sequence of strokes indicating the order in which the strokes of a Chinese character are written. A stroke is a movement of a writing instrument on a writing surface. In this paper, stroke data is obtained via Chaizi Database[5]. In order to model the visual relationship between characters, the Stroke order of each character is fed into another 1-layer GRU network to generate the shape embedding. An example is given in the bottom part in Figure 3.

3.3 Transformer Encoder

The transformer encoder has the same architecture as that in BERT$_{base}$ (Devlin et al., 2019). The number of transformer layers (Vaswani et al., 2017) is 12, the size of hidden units is 768 and the number of attention head is 12. For more detailed configurations please refer to Devlin et al. (2019).

3.4 Output Layer

As illustrated in Figure 2, our model makes two predictions for each chosen character.

Character Prediction Similar to BERT, PLOME predicts the original character for each masked token based on the embedding generated by the last transformer layer. The probability of the character predicted for the i-th token in a given sentence is defined as:

$$p_c(y_i = j|X) = softmax(\mathbf{W_c h_i} + \mathbf{b_c})[j] \quad (1)$$

where $p_c(y_i = j|X)$ is the conditional probability that the true character of the i-th token x_i is predicted as the j-th character in vocabulary, $\mathbf{h_i}$ denotes the embedding output from the last transformer layer for x_i, $\mathbf{W_c} \in \mathbf{R}^{n_c \times 768}$ and $\mathbf{b_c} \in \mathbf{R}^{n_c}$ are parameters for character prediction, n_c is the size of the vocabulary.

Pronunciation Prediction Chinese totally has about 430 different pronunciations (represented by phonics) but has more than 2,500 common used characters. Thus, many characters share the same pronunciation. Moreover, some pronunciations are so similar that it is easy to be misused, such as "jing" and "jin". Therefore, phonological error dominates Chinese spelling errors. In practice, about 80% of spelling errors are phonological (Zhang et al., 2020). In order to learn the misspelled knowledge on phonic level, PLOME also predicts the true pronunciation for each masked token, where pronunciation is presented by phonics without diacritic. The probability of pronunciation prediction is defined as:

$$p_p(g_i = k|X) = softmax(\mathbf{W_p h_i} + \mathbf{b_p})[k] \quad (2)$$

where $p_p(g_i = k|X)$ is the conditional probability that the correct pronunciation of the masked character x_i is predicted as the k-th phonics in the phonic vocabulary, $\mathbf{h_i}$ denotes the embedding output from the last transformer layer for x_i, $\mathbf{W_c} \in \mathbf{R}^{n_p \times 768}$ and $\mathbf{b_p} \in \mathbf{R}^{n_p}$ are parameters for pronunciation prediction, n_p is the size of the phonic vocabulary.

3.5 Learning

The learning process is driven by optimizing two objectives, corresponding to character prediction and pronunciation prediction, respectively.

$$\mathcal{L}_c = -\sum_{i=1}^{n} \log p_c(y_i = l_i|X) \quad (3)$$

$$\mathcal{L}_p = -\sum_{i=1}^{n} \log p_p(g_i = r_i|X) \quad (4)$$

where $\mathcal{L}_c$ is the objective for character prediction, l_i is the true character for x_i, $\mathcal{L}_p$ is the objective for

[2] https://en.wikipedia.org/wiki/Pinyin
[3] http://www.unicode.org/charts/unihan.html
[4] https://en.wikipedia.org/wiki/Stroke_order
[5] https://github.com/kfcd/chaizi

pronunciation prediction, r_i is the true pronunciation. The overall objective is defined as:

$$\mathcal{L} = \mathcal{L}_c + \mathcal{L}_p \tag{5}$$

3.6 Fine-tuning Procedure

Above subsections present the details of the pre-training procedure. In this subsection, we introduce the fine-tuning procedure. PLOME is designed for the CSC task, which aims to detect and correct spelling errors in Chinese texts. Formally, given a character sequence $\mathbf{X} = \{x_1, x_2, ..., x_n\}$ consisting of n characters, the model is expected to generate a target sequence $\mathbf{Y} = \{y_1, y_2, ..., y_n\}$, where errors are corrected.

Training The learning objective is exactly the same as that in the pre-training procedure(see Section 3.5). This procedure is similar to pre-training except that: (1). the masking operation introduced in Section 3.1 is eliminated. (2). all input characters require to be predicted rather than only chosen tokens as in pre-training.

Inference As illustrated in Section 3.4, PLOME predicts both the character distribution and pronunciation distribution for each masked token. We define the joint distribution as:

$$p_j(y_i = j|X) = p_c(y_i = j|X) \times p_p(g_i = j^p|X) \tag{6}$$

where $p_j(y_i = j|X)$ is the probability that the original character of x_i is predicted as the j-th character jointly considering the character and pronunciation predictions, p_c and p_p are separately defined in Equation 1 and Equation 2, j^p is the pronunciation of the j-th character. To this end, we construct an indicator matrix $\mathbf{I} \in \mathbf{R}^{n_c \times n_p}$, where $\mathbf{I}_{i,j}$ is set to 1 if the pronunciation of the i-th character is the j-th phonics, otherwise set to 0. Then the joint distribution can be computed by:

$$\mathbf{p_j}(y_i|X) = [\mathbf{p_p}(g_i|X) \cdot \mathbf{I}^\mathsf{T}] \odot \mathbf{p_c}(y_i|X) \tag{7}$$

where $\odot$ is the element-wise production.

We use the joint probability as the predicted distribution. For each input token, the character with the highest joint probability is selected as the final output: $\widehat{y}_i$ =argmax $\mathbf{p_j}(y_i|X)$. The joint distribution simultaneously takes the character and pronunciation predictions into consideration, thus is more accurate. We will verify it in Section 4.5.

4 Experiments

In this section, we present the details for pre-training PLOME and the fine-tuning results on the most widely used benchmark dataset.

4.1 Pre-training

Dataset We use wiki2019zh[6] as the pre-training corpus, which consists of one million Chinese Wikipedia[7] pages. Moreover, we also collect three million news articles from a Chinese news platform. We split those pages and articles into sentences and totally obtain 162.1 million sentences. Then we concatenate consecutive sentences to obtain text fragments with at most 510 characters, which are used as the training instances.

Parameter Settings We denote the dimension of character embeddings, letter (in phonics) embeddings and stroke embeddings as d_c, d_l, d_s, respectively, the dimension of hidden states in phonic and shape GRU networks as h_p, and h_s. Then we have $d_c = 768, d_l = d_s = 32, h_p = h_s = 768$. The configuration of transformer encoder is exactly the same as that in BERT$_{base}$ (Devlin et al., 2019), and the learning rate is set to 5e-5. These parameters are set based on experience because of the large cost of pre-training. Better performance could be achieved if parameter tuning technique (e.g. grid search) is employed. Moreover, instead of training PLOME from scratch, we adopt the parameters of Chinese BERT released by Google[8] to initialize the Transformer blocks.

4.2 Fine-tuning

Training Data Following Cheng et al. (2020), the training data is composed of 10K manually annotated samples from SIGHAN (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015) and 271K automatically generated samples from Wang et al. (2018).

Evaluation Data We use the latest SIGHAN test dataset (Tseng et al., 2015) as in Zhang et al. (2020) to evaluate the proposed model, which contains 1100 texts and 461 types of errors.

Evaluation Metrics Following previous work (Cheng et al., 2020; Zhang et al., 2020), we use the

[6] https://github.com/suzhoushr/nlp_chinese_corpus

[7] https://zh.wikipedia.org/wiki/

[8] https://github.com/google-research/bert

Category	Method	Character-level (%)						Sentence-level (%)					
		Detection-level			Correction-level			Detection-level			Correction-level		
		P	R	F	P	R	F	P	R	F	P	R	F
SOTA	*Hybrid* (Wang et al., 2018)	54.0	69.3	60.7	-	-	52.1	-	-	-	-	-	-
	PN (Wang et al., 2019)	66.8	73.1	69.8	71.5	59.5	69.9	-	-	-	-	-	-
	FASPell (Hong et al., 2019)	-	-	-	-	-	-	67.6	60.0	63.5	66.6	59.1	62.6
	SKBERT (Zhang et al., 2020)	-	-	-	-	-	-	73.7	73.2	73.5	66.7	66.2	66.4
	SpellGCN (Cheng et al., 2020)	88.9	87.7	88.3	95.7	83.9	89.4	74.8	80.7	77.7	72.1	77.7	75.9
Pretrain	*cBERT-Pretrain*	64.2	83.2	72.5	85.6	71.2	77.7	37.9	49.5	42.9	32.1	42.0	36.4
	PLOME-Pretrain	68.1	74.2	71.0	83.2	61.7	70.9	41.8	47.5	44.5	34.2	38.9	36.4
Finetune	*BERT-Finetune*	90.9	84.9	87.8	95.6	81.2	87.8	68.4	77.6	72.7	66.0	74.9	70.2
	cBERT-Finetune	92.4	**87.7**	90.0	96.2	**84.4**	89.9	75.3	78.9	77.1	72.7	76.1	74.4
	PLOME-Finetune	**94.5**	87.4	**90.8**	**97.2**	84.3	**90.3**	**77.4**	**81.5**	**79.4**	**75.3**	**79.3**	**77.2**

Table 2: The performance of our approach and baseline models. Results in the latter two groups are from our implementation. Following Cheng et al. (2020), we run the experiments 4 times and report the average metrics.

precision, recall and F1 scores as the evaluation metrics. Besides character-level evaluation, we also report sentence-level metrics on the detection and correction sub-tasks. We evaluate these metrics using the script from Cheng et al. (2020)[9].

Parameter Settings Following Cheng et al. (2020), we set the maximum sentence length to 180, batch size to 32 and the learning rate to 5e-5. All experiments are conducted for 4 runs and the averaged metric is reported. The code and trained models will be released (currently the code is attached in the supplementary files).

4.3 Baseline Models

We use the following methods for comparison.

Hybird (Wang et al., 2018) uses a BiLSTM-based model trained on an automatically generated dataset.

PN (Wang et al., 2019) is a Seq2Seq model incorporating a pointer network.

FASPell (Hong et al., 2019) adopts the DAE-Decoder paradigm and employs BERT as the denoising auto-encoder.

SKBERT (Zhang et al., 2020) introduces the **S**oftmas**K**ing strategy in BERT to improve the performance of error detection.

SpellGCN (Cheng et al., 2020) combines a GCN network with BERT to model the relationship between characters in the given confusion set.

Besides, we implement a baseline model *cBERT* (**c**onfusion set based **BERT**), whose input and encoder layers are the same as that in BERT$_{base}$ (Devlin et al., 2019). The output layer is similar to *PLOME*, but only has the character prediction as defined in Equation 1. *cBERT* is also pre-trained via the confusion set based masking strategy.

4.4 Main Results

Table 2 illustrates the performance of the proposed method and baseline models. The results of recently proposed models are presented in the first group. The results of pre-trained and fine-tuned models are presented in the second and third group, respectively. From this table, we observe that:

1) Without fine tuning, pre-trained models in the middle group achieve relatively good results, even outperform the supervised approach *PN* with remarkable gains. This indicates that the confusion set based masking strategy enables our model to learn task-specific knowledge during pre-training.

2) Compared the fine-tuned models, *cBERT* outperforms *BERT* on all metrics. Especially, the *F* score of sentence-level evaluations are improved by more than 4 absolute points. The improvement is remarkable with such a large amount of training data (281k texts), which indicates that the proposed masking strategy provides essential knowledge and it can not be learned from fine tuning.

3) With the incorporation of phonic and shape embeddings, *PLOME-Finetune* outperforms *cBERT-Finetune* by 2.3% and 2.8% absolute improvements in sentence-level detection and correction. This indicates that characters' phonics and strokes provide useful information and it can hardly be learned from the confusion set.

4) *SpellGCN* and our approach use the same con-

[9] https://github.com/ACL2020SpellGCN/SpellGCN

Method	Character-level on Whole Set						Sentence-level via Official Tool								
	Detection-level			Correction-level				Detection-level				Correction-level			
	P	R	F	P	R	F	FPR	A	P	R	F	A	P	R	F
SpellGCN	77.7	85.6	81.4	96.9	82.9	89.4	13.2	83.7	85.9	80.6	83.1	82.2	85.4	77.6	81.3
BERT-Finetune	76.2	83.1	79.5	96.5	80.3	87.6	14.7	81.7	85.2	76.0	80.3	80.3	84.7	73.5	78.7
cBERT-Finetune	83.0	**87.8**	85.3	96.0	83.9	89.5	**10.6**	84.5	**88.1**	79.6	83.6	82.9	87.6	76.3	81.5
PLOME-Finetune	**85.2**	86.8	**86.0**	**97.2**	**85.0**	**90.7**	10.9	**85.0**	87.9	**80.9**	**84.3**	**83.7**	**87.6**	**78.3**	**82.7**

Table 3: Experimental results evaluated on the whole test set. FPR denotes the false positive rate and A denotes the accuracy, which are evaluated by official tools on SIGHAN2015.

Prediction	Character-level						Sentence-level					
	Detection-level			Correction-level			Detection-level			Correction-level		
	P	R	F	P	R	F	P	R	F	P	R	F
p_c (Equation 1)	83.5	86.8	85.1	96.4	84.7	90.2	76.5	81.1	78.7	74.0	78.5	76.2
p_j (Equation 6)	**85.2**	**86.8**	**86.0**	**97.2**	**85.0**	**90.7**	**77.4**	**81.5**	**79.4**	**75.3**	**79.3**	**77.2**

Table 4: The performance of *PLOME* with the character prediction p_c and the joint prediction p_j as output.

fusion set from Wu et al. (2013), but adopt different strategies to learn the knowledge contained in it. *SpellGCN* built a GCN network to model this information, whereas *PLOME* learned it from huge scale data during pre-training. *PLOME* achieves better performance on all metrics, indicating that our approach is more effective to model such knowledge.

Previous work (Wang et al., 2019; Cheng et al., 2020) conducted the character-level evaluation on positive sentences which contain at least one error (sentence-level metrics were evaluated on the whole test set). Thus, the precision score is very high. The character-level results in table 2 are also evaluated in such manner for fair comparison. To make more comprehensive evaluation, we report the results evaluated on the whole test set in table 3. Moreover, following Cheng et al. (2020), we also report the sentence-level results evaluated by SIGHAN official tool. We observe that *PLOME* consistently outperforms *BERT* and *SpellGCN* on all metrics.

To make more comprehensive comparisons, we also evaluate the proposed model on SIGHAN13(Wu et al., 2013) and SIGHAN14(Yu et al., 2014). Following Cheng et al. (2020), we performed 6 additional fine-tuning epochs on SIGHAN13 as its data distribution differs from other datasets. Table5 illustrates the results, from which we observe that *PLOME* consistently outperforms all the compared models.

Method	Detection-level			Correction-level		
	P	R	F	P	R	F
SIGHAN14						
BERT	82.9	77.6	80.2	96.8	75.2	84.6
SpellGCN	83.6	78.6	81.0	97.2	76.4	85.5
PLOME	**88.5**	**79.8**	**83.9**	**98.8**	**78.8**	**87.7**
SIGHAN13						
BERT	80.6	88.4	84.3	98.1	87.2	92.3
SpellGCN	82.6	88.9	85.7	98.4	88.4	93.1
PLOME	**85.0**	**89.3**	**87.1**	**98.7**	**89.1**	**93.7**

Table 5: The character-level performance of *PLOME* on SIGHAN13 and SIGHAN14.

4.5 Effects of Prediction Strategy

As illustrated in Section 3.4 and 3.6, *PLOME* predicts three distributions for each character: the character distribution p_c, the pronunciation distribution p_p and the joint distribution p_j. The latter two distributions are related to pronunciation prediction, which is first to be introduced in this work. In this subsection, we investigate the performance of *PLOME* with each of them as the final output. The CSC task requires character prediction, thus we only compare the effects of the character prediction p_c and the joint prediction p_j.

Table 4 presents the experimental results, from which we observe that the joint distribution outperforms the character distribution on all evaluation metrics. Especially, the gap of precision scores is more obvious. The joint distribution simultaneously takes the character and pronunciation predic-

Method	Character-level						Sentence-level					
	Detection-level			Correction-level			Detection-level			Correction-level		
	P	R	F	P	R	F	P	R	F	P	R	F
cBERT-Rand	**81.8**	86.2	83.9	**96.3**	83.0	89.2	73.7	77.0	75.3	70.0	73.9	71.9
cBERT-BERT	83.0	**87.8**	**85.3**	96.0	**83.9**	**89.5**	**75.3**	**78.9**	**77.1**	**72.7**	**76.1**	**74.4**
PLOME-Rand	83.4	86.6	84.9	96.8	83.9	89.9	75.9	80.7	78.2	73.6	78.3	75.9
PLOME-BERT	**85.2**	**86.8**	**86.0**	**97.2**	**85.0**	**90.7**	**77.4**	**81.5**	**79.4**	**75.3**	**79.3**	**77.2**

Table 6: The performance of *cBERT* and *PLOME* with different initialization strategies. **-Rand* denotes that all the parameters are randomly initialized and **-BERT* denotes parameters are initialized by BERT.

tions into consideration, thus the predicted results are more accurate.

4.6 Effects of Initialization Strategy

Generally speaking, initialization strategy has a great influence on the performance for deep models. In this subsection, we investigate the effects of different initialization strategies in the pre-training procedure. For comparison, we implement four baselines based on *cBERT* and *PLOME*.

Table 6 illustrates the results, where methods named with "**-Rand*" initialize all the parameters randomly and methods named with "**-BERT*" initialize the transformer encoder by BERT released by Google. From the table we observe that both *cBERT* and *PLOME* initialized with BERT achieve better performance. Especially, the recall score improves significantly for all evaluations. We believe the following two reasons may explain this phenomenon. 1) The rich semantic information in BERT can effectively improves the generalization ability. 2) *PLOME* is composed of two 1-layer GRU networks and a 12-layer transformer encoder, and totally contains more than 110M parameters. It is easily trapped into local optimization when training such a large-scale model from scratch.

4.7 Phonic/Shape Embedding Visualization

In this subsection, we investigate whether the phonic and shape GRU networks learned meaningful representations for characters. To this end, we generate the phonic and shape embeddings for each character by the GRU networks in Figure 2 and then visualize them.

Figure 4 illustrates 30 characters nearest to '锭' according to the cosine similarity of the 768-dim embeddings generated by GRU networks, which is visualized via t-SNE (Maaten and Hinton, 2008). On one hand, nearly all the characters similar to '锭', such as '啶' and '绽', are included in this figure. On the other hand, similar characters are very close to each other (labeled by circles). These phenomena indicate that the learned shape embedding well models the shape similarity. Figure 5 shows the same situation for the phonic embedding related to 'ding' and also demonstrates its ability in modeling phonic similarity.

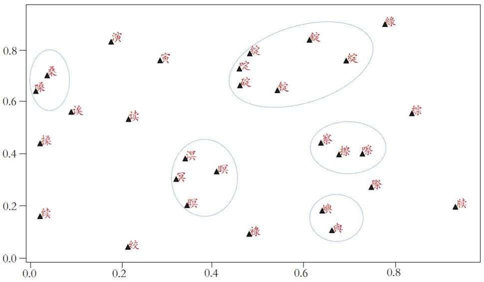


Figure 4: The visualization of shape embeddings.

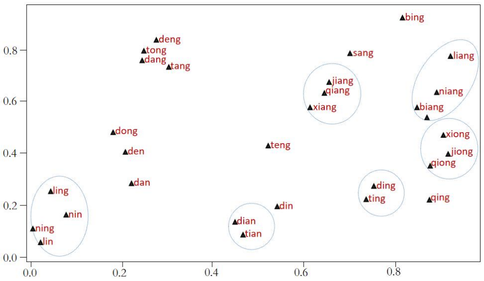


Figure 5: The visualization of phonic embeddings.

4.8 Converging Speed of Various Models

In this subsection, we investigate the converging speed of various models in the fine-tuning procedure. Figure 6 shows the test curves for character-level detection metrics of *BERT*, *cBERT* and *PLOME*. Thanks to the confusion set based masking strategy, *cBERT* and *PLOME* learned task-specific knowledge in the pre-training procedure, therefore they achieve much better performance than *BERT* at the beginning of the training. As the training went on, the gap gradually narrowed dur-

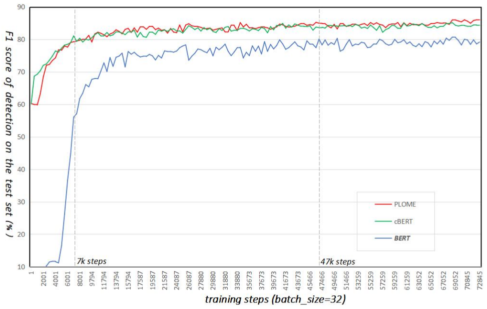


Figure 6: The test curves for character-level detection metrics of various models in the fine-tuning procedure.

ing the first 35,000 steps and then remained stable with a gap of 6%(86% vs. 80%). In addition, the proposed model needs much less training steps to achieve a relatively good performance. *PLOME* needs only 7k steps to achieve the score of 80%, whereas *BERT* needs 47k steps.

5 Conclusions

We propose PLOME, a pre-trained masked language model with misspelled knowledge for CSC. To the best of our knowledge, PLOME is the first task-specific language model for CSC, which jointly learns semantics and misspelled knowledge thanks to the confusion set based masking strategy. Previous work demonstrated that phonological and visual similarity between characters is essential to this task. We introduce phonic and shape GRU networks to model such features. Moreover, PLOME is also the first model that makes decision via jointly considering the target pronunciation and character distributions. Experimental results showed that PLOME outperforms all the compared models with remarkable gains.

Acknowledgments

We thank Lei He, Suncong Zheng and Weikang Wang for helpful discussions, and anonymous reviewers for their insightful comments.

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).

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jill Burstein and Martin Chodorow. 1999. Automated essay scoring for nonnative english speakers. In *Computer mediated language assessment and evaluation in natural language processing*.

Chao-Huang Chang. 1995. A new approach for automatic chinese spelling correction. In *Proceedings of Natural Language Processing Pacific Rim Symposium*, volume 95, pages 278–283. Citeseer.

Kuan-Yu Chen, Hung-Shin Lee, Chung-Han Lee, Hsin-Min Wang, and Hsin-Hsi Chen. 2013. A study of language modeling for Chinese spelling check. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 79–83, Nagoya, Japan. Asian Federation of Natural Language Processing.

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.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016. Neural network translation models for grammatical error correction. *arXiv preprint arXiv:1606.00189*.

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

Jianfeng Gao, Chris Quirk, et al. 2010. A large scale ranker-based system for search query spelling correction. In *23rd International Conference on Computational Linguistics*.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Fluency boost learning and inference for neural grammatical error correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1055–1065, Melbourne, Australia. 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.

Changning Huang, Haihua Pan, Zhou Ming, and Lei Zhang. 2000. Automatic detecting/correcting errors in chinese text by an approximate word-matching algorithm.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. pages 753–762.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Chen Lin, Timothy Miller, Dmitriy Dligach, Steven Bethard, and Guergana Savova. 2019. A BERT-based universal model for both within- and cross-sentence clinical temporal relation extraction. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 65–71, Minneapolis, Minnesota, USA. 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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

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.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. pages 3982–3992.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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.

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

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. pages 72–77.

Junjie Yu and Zhenghua Li. 2014. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 220–223.

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.

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.