Learning from the Dictionary: Heterogeneous Knowledge Guided Fine-tuning for Chinese Spell Checking

Yinghui Li[1]*, **Shirong Ma**[1]*, **Qingyu Zhou**[2]*, **Zhongli Li**[2], **Yangning Li**[1], **Shuling Huang**[1], **Ruiyang Liu**[5], **Chao Li**[4], **Yunbo Cao**[2] and **Hai-Tao Zheng**[1,3]†
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]Tencent Cloud Xiaowei, [3]Peng Cheng Laboratory , [4]Xiaomi Group
[5]Department of Computer Science and Technology, Tsinghua University
{liyinghu20, masr21}@mails.tsinghua.edu.cn, qingyuzhou@tencent.com

Abstract

Chinese Spell Checking (CSC) aims to detect and correct Chinese spelling errors. Recent researches start from the pretrained knowledge of language models and take multimodal information into CSC models to improve the performance. However, they overlook the rich knowledge in the dictionary, the reference book where one can learn how one character should be pronounced, written, and used. In this paper, we propose the LEAD framework, which renders the CSC model to **lea**rn heterogeneous knowledge from the **d**ictionary in terms of phonetics, vision, and meaning. LEAD first constructs positive and negative samples according to the knowledge of character phonetics, glyphs, and definitions in the dictionary. Then a unified contrastive learning-based training scheme is employed to refine the representations of the CSC models. Extensive experiments[1] and detailed analyses on the SIGHAN benchmark datasets demonstrate the effectiveness of our proposed methods.

Phonetically Similar Error	
Input	铁轨上有一列或(huò)车在行驶。
Candidate 1	铁轨上有一列货(huò)车在行驶。
	There is a *truck* running on the railway.
Candidate 2	铁轨上有一列火(huǒ)车在行驶。
	There is a *train* running on the railway.
Definition	【火车】一种交通工具，由机车牵引若干节车厢在铁路上行驶。
	A means of transportation in which a number of carriages are pulled by a locomotive to travel on a railway.
Visually Similar Error	
Input	炉子上正绕(rào)着水。
Candidate 1	炉子上正浇(jiāo)着水。
	Water is *pouring* on the stove.
Candidate 2	炉子上正烧(shāo)着水。
	Water is *burning* on the stove.
Definition	【烧】加热使物体发生变化。
	Change matters by heating.

Table 1: Examples of Chinese spelling errors. The wrong/candidate/golden characters are in red/purple/blue. The key information is in orange.

1 Introduction

As a crucial Chinese processing task, Chinese Spell Checking (CSC) aims to detect and correct Chinese spelling errors (Wu et al., 2013a), which are mainly caused by phonetically or visually similar characters (Liu et al., 2010). Recent researches propose to introduce phonetics and vision information to help pretrained language models (PLMs) deal with confusing characters (Liu et al., 2021; Xu et al., 2021; Huang et al., 2021). However, CSC is challenging because it requires not only phonetics/vision information but also complex definition knowledge to assist in finding the truly correct character. As shown in Table 1, the “货(huò)” and “火(huǒ)” are phonetically similar, and both are suitable collocations with “车”. But if the model pays attention to the keyword “铁轨(railway)” and knows the meaning of the “火车(train)”, then the model can not be disturbed by the “货” and easily make the correct judgment. The same situation also occurs in the visual case. For these hard samples, PLMs do not perform well in that the masked-language modeling objective determines their pretrained semantic knowledge is more about the collocation of characters, rather than the definitions of their meanings. Therefore, if the model understands the word meanings, it can be further enhanced to handle more hard samples and get performance improvements.

To help people learn Chinese, the meanings of Chinese characters and words have been pre-organized as the definition sentences in the dictionary. The dictionary contains a wealth of useful knowledge for CSC, including character phonetics, glyphs, and definitions. It is also the most impor-

* indicates equal contribution. Work is done during Yinghui's internship at Tencent Cloud Xiaowei.
† Corresponding author: Hai-Tao Zheng. (E-mail: zheng.haitao@sz.tsinghua.edu.cn)
[1]The source codes are available at https://github.com/geekjuruo/LEAD.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 238–249
December 7-11, 2022 ©2022 Association for Computational Linguistics

tant resource for Chinese beginners to learn how to pronounce, write, and use one character. Inspired by this, we focus on utilizing the rich knowledge in the dictionary to improve the CSC performance.

In this paper, we propose LEAD, a unified fine-tuning framework to guide the CSC models to **lea**rn heterogeneous knowledge from the **d**ictionary. In general, LEAD has one training paradigm but three different training objectives besides the traditional CSC objective. This enables models to learn three different kinds of knowledge, namely phonetics, vision, and definition knowledge. Specifically, we construct various positive and negative samples according to the respective characteristics of different knowledge, and then utilize these generated sample pairs to train models with our designed unified contrastive learning paradigm.

Through the optimization of LEAD, the fine-tuned model handles various phonetically/visually similar character errors as well as previous multimodal models, and goes a further step to deal with more confusing errors with the help of the definition knowledge contained in the dictionary. Additionally, LEAD is a model-agnostic fine-tuning framework, which has no restrictions on the fine-tuned models. In practice, we fine-tune BERT and a more complex multimodal CSC model (Xu et al., 2021) with LEAD, and experimental results on the SIGHAN datasets show consistent improvements.

To summarize, the contributions of our work are in three folds: (1) We focus on the importance of the dictionary knowledge for the CSC task, which is instructive for future CSC research. (2) We propose the LEAD framework, which fine-tunes the models to learn heterogeneous knowledge beneficial to the CSC task in a unified manner. (3) We conduct extensive experiments and detailed analyses on widely used SIGHAN datasets and LEAD outperforms previous state-of-the-art methods.

2 Related Work

2.1 Chinese Spell Checking

Recently, deep learning-based models have gradually become the mainstream CSC methods (Wang et al., 2018; Hong et al., 2019; Zhang et al., 2020; Li et al., 2022b). 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. GAD (Guo et al., 2021) proposes a global attention decoder method and pre-trains the BERT (Devlin et al., 2019) with a confusion set guided replacement strategy. Li et al. (2021) proposes a method that continually identifies the weak spots of a model to generate more valuable training samples, and applies a task-specific pre-training strategy to enhance the model. Additionally, many CSC works have focused on the importance of multimodal knowledge for CSC. DCN (Wang et al., 2021), MLM-phonetics (Zhang et al., 2021), and SpellBERT (Ji et al., 2021) all utilize phonetic features to improve CSC performance. PLOME (Liu et al., 2021) designs a confusion set-based masking strategy and introduces phonetics and stroke information. REALISE (Xu et al., 2021) and PHMOSpell (Huang et al., 2021) both employ kinds of encoders to learn multimodal knowledge. Different from previous works, our work is the first to introduce definition knowledge from the dictionary to enhance CSC models.

2.2 Contrastive Learning

Contrastive learning is a kind of representation learning method that has been widely used in NLP and CV (Chen et al., 2020; He et al., 2020a; Gao et al., 2021). The main motivation of contrastive learning is to attract the positive samples and repulse the negative samples in a certain space (Hadsell et al., 2006; Chen et al., 2020; Khosla et al., 2020). In the NLP field, various contrastive learning methods have been studied for learning all kinds of better representations, such as entity (Li et al., 2022a), sentence (Kim et al., 2021), and relation (Qin et al., 2021). To the best of our knowledge, we are the first to leverage the idea of contrastive learning to learn better phonetics, vision, and definition knowledge for CSC.

3 Methodology

In this section, we first introduce the overview of the LEAD framework, as illustrated in Figure 1, and describe our designed unified contrastive learning mechanism for heterogeneous dictionary knowledge. Then, for each knowledge-guided fine-tuning, we explain its motivation, positive/negative pairs construction, and representation metric which is used in the contrastive learning mechanism.

3.1 Overview of LEAD

In LEAD, in addition to using the CSC samples to train the traditional CSC objective, various positive and negative pairs are generated for the contrastive

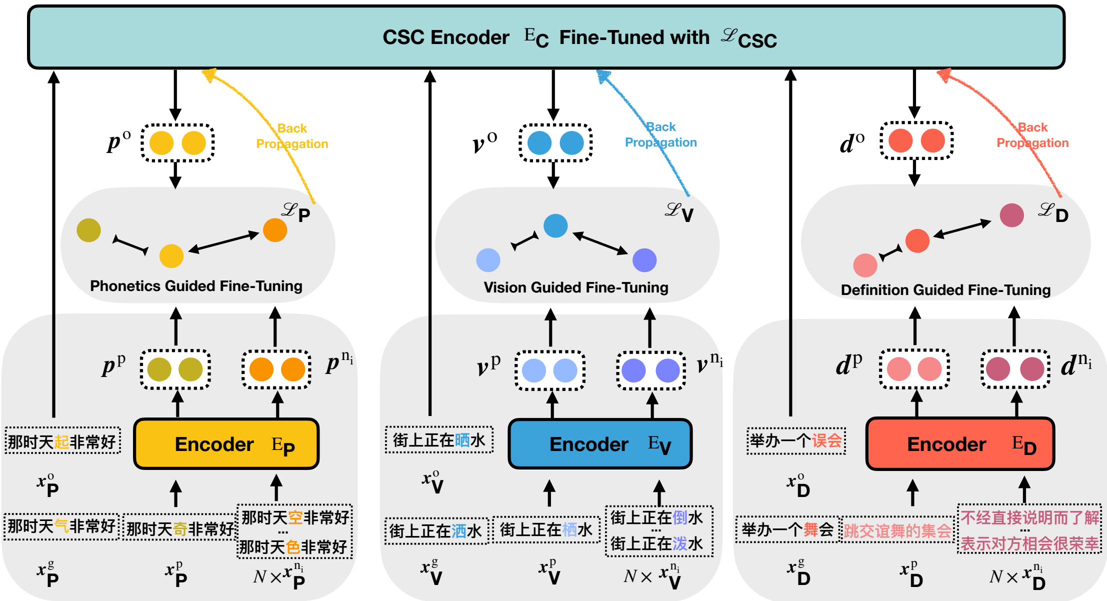


Figure 1: Overview of our LEAD framework. According to the contrastive learning mechanism proposed in (He et al., 2020b), the gradients of $\mathcal{L}_{\mathrm{P}}, \mathcal{L}_{\mathrm{V}}, \mathcal{L}_{\mathrm{D}}$ are propagated back to the CSC model so that it is optimized accordingly.

learning of three kinds of knowledge (i.e., phonetics, vision, and definition). Specifically, for a particular knowledge K, to achieve a training mini-batch, we construct a positive pair $(\boldsymbol{x}^{\mathrm{o}}_{\mathrm{K}}, \boldsymbol{x}^{\mathrm{p}}_{\mathrm{K}})$ and N negative pairs $\{(\boldsymbol{x}^{\mathrm{o}}_{\mathrm{K}}, \boldsymbol{x}^{\mathrm{n_i}}_{\mathrm{K}})\}_{i=0}^{N-1}$, where $\mathrm{K} \in \{\mathrm{P}, \mathrm{V}, \mathrm{D}\}$ represents "**P**honetics, **V**ision, **D**efinition" knowledge. Note that the original sample $\boldsymbol{x}^{\mathrm{o}}_{\mathrm{K}}$ is directly from the CSC samples, the positive sample $\boldsymbol{x}^{\mathrm{p}}_{\mathrm{K}}$ and negative samples $\{\boldsymbol{x}^{\mathrm{n_i}}_{\mathrm{K}}\}$ are generated from $\boldsymbol{x}^{\mathrm{o}}_{\mathrm{K}}$ according to the characteristics of the knowledge K.

Then, for the positive and negative sentences (i.e., $\boldsymbol{x}^{\mathrm{p}}_{\mathrm{K}}$ and $\{\boldsymbol{x}^{\mathrm{n_i}}_{\mathrm{K}}\}$) of length T, we use various encoders (i.e., $\mathrm{E_K} \in \{\mathrm{E_P}, \mathrm{E_V}, \mathrm{E_D}\}$) to map them to a sequence of representations $\boldsymbol{k}^{\mathrm{p}} = [k^{\mathrm{p}}_1, ..., k^{\mathrm{p}}_T], \{\boldsymbol{k}^{\mathrm{n_i}}\} = \{[k^{\mathrm{n_i}}_1, ..., k^{\mathrm{n_i}}_T]\}, k^{\mathrm{p}}_j, k^{\mathrm{n_i}}_j \in \mathbb{R}^h$, where h is the size of the $\mathrm{E_K}$'s hidden state:

$$\boldsymbol{k}^{\mathrm{p}} = \mathrm{E_K}(\boldsymbol{x}^{\mathrm{p}}_{\mathrm{K}}), \boldsymbol{k}^{\mathrm{p}} \in \{\boldsymbol{p}^{\mathrm{p}}, \boldsymbol{v}^{\mathrm{p}}, \boldsymbol{d}^{\mathrm{p}}\}, \quad (1)$$

$$\{\boldsymbol{k}^{\mathrm{n_i}}\} = \{\mathrm{E_K}(\boldsymbol{x}^{\mathrm{n_i}}_{\mathrm{K}})\}, \boldsymbol{k}^{\mathrm{n_i}} \in \{\boldsymbol{p}^{\mathrm{n_i}}, \boldsymbol{v}^{\mathrm{n_i}}, \boldsymbol{d}^{\mathrm{n_i}}\}. \quad (2)$$

For the original sentence $\boldsymbol{x}^{\mathrm{o}}_{\mathrm{K}}$, we utilize the encoder of CSC model (i.e., $\mathrm{E_C}$) to get its sentence representation $\boldsymbol{k}^{\mathrm{o}} = [k^{\mathrm{o}}_1, ..., k^{\mathrm{o}}_T], k^{\mathrm{o}}_j \in \mathbb{R}^h$, the $\mathrm{E_C}$'s hidden size is equal the dimension of the $\mathrm{E_K}$'s hidden state:

$$\boldsymbol{k}^{\mathrm{o}} = \mathrm{E_C}(\boldsymbol{x}^{\mathrm{o}}_{\mathrm{K}}), \boldsymbol{k}^{\mathrm{o}} \in \{\boldsymbol{p}^{\mathrm{o}}, \boldsymbol{v}^{\mathrm{o}}, \boldsymbol{d}^{\mathrm{o}}\}. \quad (3)$$

After obtaining the representations of our generated sentence pairs, following the widely used InfoNCE (van den Oord et al., 2018), we train these sample pairs in a contrastive manner:

$$\mathcal{L}_{\mathrm{K}} = -\log \frac{f_{\mathrm{K}}(\boldsymbol{k}^{\mathrm{o}}, \boldsymbol{k}^{\mathrm{p}}, s)}{f_{\mathrm{K}}(\boldsymbol{k}^{\mathrm{o}}, \boldsymbol{k}^{\mathrm{p}}, s) + \sum_{i=0}^{N-1} f_{\mathrm{K}}(\boldsymbol{k}^{\mathrm{o}}, \boldsymbol{k}^{\mathrm{n_i}}, s)}, \quad (4)$$

where the $\mathcal{L}_{\mathrm{K}}$ is the training objective of the knowledge K, and the f_{K} is the representation metric function in the respective space of each knowledge, which will be introduced in later sections. In the mini-batch, all sentences are of length T and their s-th character is the spelling error.

It is worth emphasizing that the three knowledge encoders (i.e., $\mathrm{E_P}$, $\mathrm{E_V}$, and $\mathrm{E_D}$) are frozen, while the $\mathrm{E_C}$ receives gradients from multiple dimensions and is optimized during the training process. Besides, our proposed LEAD is model-agnostic so that we can arbitrarily configure $\mathrm{E_P}$, $\mathrm{E_V}$, $\mathrm{E_D}$ and easily use previous CSC models as $\mathrm{E_C}$. The implementation details of various encoders in our experiments are shown in Appendix A.2.

Briefly, our proposed LEAD performs specific contrastive fine-tuning guided by heterogeneous knowledge, thereby introducing various beneficial information into CSC models to improve their performance. In the Sections 3.2- 3.4, we will detail the positive and negative pairs construction and representation metric we design for each knowledge.

3.2 Phonetics Guided Fine-tuning

According to the phonetics knowledge, Chinese characters are represented by Pinyin. Therefore, to make the model better handle phonetic errors, we aim to guide it to pay more attention to characters with similar Pinyin. To this end, we propose the *Phonetics Guided Fine-tuning*, which aims to refine the representation space learned by models so that the representations of the similar Pinyin characters are pulled closer while the representations of different Pinyin characters are pushed outward. Thus, when handling phonetically spelling errors, our model will preferentially associate with their corresponding phonetically similar characters.

Positive and Negative Pairs For the phonetics knowledge, we regard characters with similar Pinyin as positive pairs and characters with different Pinyin as negative pairs. As shown in Figure 1, given a training sample $\boldsymbol{x}_{\mathrm{P}}^{\mathrm{o}}$ "那时天起(qǐ, rise)非常好" that has a phonological spelling error, we replace "起(qǐ, rise)" with its phonetically similar character "奇(qí, strange)" to achieve a positive sample $\boldsymbol{x}_{\mathrm{P}}^{\mathrm{p}}$. To generate negative samples $\{\boldsymbol{x}_{\mathrm{P}}^{\mathrm{n_i}}\}$, we randomly select N characters with different Pinyin, such as "色(sè, color)", to replace "起(qǐ, rise)". Finally, we will get a positive pair $(\boldsymbol{x}_{\mathrm{P}}^{\mathrm{o}}, \boldsymbol{x}_{\mathrm{P}}^{\mathrm{p}})$ and N negative pairs $\{(\boldsymbol{x}_{\mathrm{P}}^{\mathrm{o}}, \boldsymbol{x}_{\mathrm{P}}^{\mathrm{n_i}})\}$ to form a mini-batch for the fine-tuning of phonetics knowledge.

Representation Metric Note that the motivation of phonetics guided fine-tuning is to refine the character-level representation of CSC models under the constraints of phonetics knowledge, so we only need the representation of the spelling error position, i.e., the s-th character. Therefore, the representation metric of phonetics guided fine-tuning (i.e., f_{P}) is calculated as the dot product function:

$$f_{\mathrm{P}}(\boldsymbol{p}^{\mathrm{o}}, \boldsymbol{p}^{\mathrm{p}}, s) = \exp(p_s^{\mathrm{o}\top} p_s^{\mathrm{p}}), \quad (5)$$

$$f_{\mathrm{P}}(\boldsymbol{p}^{\mathrm{o}}, \boldsymbol{p}^{\mathrm{n_i}}, s) = \exp(p_s^{\mathrm{o}\top} p_s^{\mathrm{n_i}}). \quad (6)$$

3.3 Vision Guided Fine-tuning

Similar to the phonetics guided fine-tuning, we propose the *Vision Guided Fine-tuning* for better vision representations and improving the visual error correction ability of models. Specifically, based on the fact that Chinese characters are composed of strokes in the dimension of vision knowledge, the purpose of this module is to train models to represent characters with more similar strokes closer and characters with more different strokes farther away in the visual representation space.

Positive and Negative Pairs Based on the visual similarity between characters, for a specific Chinese character, we directly obtain its characters with similar strokes from the pre-defined confusion set widely used in previous works (Wang et al., 2019; Cheng et al., 2020; Zhang et al., 2020). Take Figure 1 as an example, for a training sample $\boldsymbol{x}_{\mathrm{V}}^{\mathrm{o}}$ "街上正在晒(shài, bask)水", its positive sample $\boldsymbol{x}_{\mathrm{V}}^{\mathrm{p}}$ is generated by replacing "晒(shài, bask)" with "栖(qī, habitat)". Similar to the phonetics guided fine-tuning, characters with different strokes are randomly selected to generate the $\{\boldsymbol{x}_{\mathrm{V}}^{\mathrm{n_i}}\}$.

Representation Metric Similar to the f_{P}, we also utilize the dot product metric to measure the representation distance in vision space:

$$f_{\mathrm{V}}(\boldsymbol{v}^{\mathrm{o}}, \boldsymbol{v}^{\mathrm{p}}, s) = \exp(v_s^{\mathrm{o}\top} v_s^{\mathrm{p}}), \quad (7)$$

$$f_{\mathrm{V}}(\boldsymbol{v}^{\mathrm{o}}, \boldsymbol{v}^{\mathrm{n_i}}, s) = \exp(v_s^{\mathrm{o}\top} v_s^{\mathrm{n_i}}). \quad (8)$$

3.4 Definition Guided Fine-tuning

As described in Section 1, the meanings of words in a structured dictionary are very useful for human spell checking when spelling errors cannot be corrected with only phonetics and vision information. To better utilize definition knowledge, we specially design the *Definition Guided Fine-tuning* to make the model better understand the word meanings. Benefiting from the enhancement of definition knowledge, our model will be human-like, seeing spelling errors and associating them with their definitions, and then making reasonable corrections based on the original word meanings.

Positive and Negative Pairs As shown in Figure 1, given a random training sample $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{o}}$ "举办一个误会" and its ground truth sentence $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{g}}$ "举办一个舞会". To get the word meaning, we must first get the original word that contains the wrong position s. Therefore, we tokenize[2] the $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{g}}$ into words "举办/一个/舞会" and index the original word (i.e., "舞会") in the dictionary[3] to get its corresponding definition sentence as a positive sample $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{p}}$. As for the negative samples $\{\boldsymbol{x}_{\mathrm{D}}^{\mathrm{n_i}}\}$, we will randomly select N definition sentences of other words.

Considering that some words have multiple definitions, we design different word definition selection strategies as follows:

1. **Select a random definition:** This is the easiest way to randomly select one sentence from multiple definition sentences.

[2] We utilize the HanLP to tokenize sentences into words.

[3] The pre-defined dictionary file we use is in the attachment.

2. **Select the first definition:** Through preliminary analysis of the dictionary, we find that when a word has multiple definitions, the more forwardly positioned definition is often the more commonly used meaning of the word. Based on this observation, we propose to select the first definition to be the word meaning.

3. **Select the most similar definition:** Intuitively, the meaning of a word can be revealed through its context. Therefore, we can also judge which definition sentence should be selected by the similarity between the sentence $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{g}}$ and the definition sentence. More practically, we obtain sentence representations through an encoder such as BERT (Devlin et al., 2019), and further use the distance metric such as the cosine function to calculate the similarity between sentence representations.

The effects of different word definition selection strategies will be analyzed in Section 4.6.2.

Representation Metric When we tokenize the $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{g}}$, we obtain the index position of the original word in the sentence at the same time. Thus, assuming that the index positions of the original word are $[s, ..., s+w]$, $s+w \leq T$, then we calculate the distance between representations as follows:

$$f_{\mathrm{D}}(\boldsymbol{d}^{\mathrm{o}}, \boldsymbol{d}^{\mathrm{p}}, s) = \cos(\mathrm{avg}([d_s^{\mathrm{o}}, ..., d_{s+w}^{\mathrm{o}}]), \mathrm{avg}(\boldsymbol{d}^{\mathrm{p}})), \tag{9}$$

$$f_{\mathrm{D}}(\boldsymbol{d}^{\mathrm{o}}, \boldsymbol{d}^{\mathrm{n_i}}, s) = \cos(\mathrm{avg}([d_s^{\mathrm{o}}, ..., d_{s+w}^{\mathrm{o}}]), \mathrm{avg}(\boldsymbol{d}^{\mathrm{n_i}})), \tag{10}$$

where the $\cos(y_1, y_2)$ is the cosine distance, and the $\mathrm{avg}([r_n, ..., r_m])$ is the mean pooling operation that calculates the average value of $[r_n, ..., r_m]$. In other words, the $\mathrm{avg}([d_s^{\mathrm{o}}, ..., d_{s+w}^{\mathrm{o}}])$ is the representation of the phrase at index positions $[s, ..., s+w]$ in the sentence $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{o}}$ and the $\mathrm{avg}(\boldsymbol{d}^{\mathrm{p}})$, $\{\mathrm{avg}(\boldsymbol{d}^{\mathrm{n_i}})\}$ are the sentence representations of $\boldsymbol{x}_{\mathrm{D}}^{\mathrm{p}}$, $\{\boldsymbol{x}_{\mathrm{D}}^{\mathrm{n_i}}\}$.

3.5 Summary of Methodology

In the above Sections 3.2-3.4, we describe in detail the contrastive learning objectives designed for the three types of knowledge. *The purpose of these three kinds of contrastive learning objectives is to let the CSC model learn the external knowledge of phonetics, vision, and definition, and finally improve the model's CSC performance.* Additionally, because the model is to be used for the CSC task, it is still necessary to train the CSC training objective $\mathcal{L}_{\mathrm{CSC}}$ with the CSC training data. So finally we have the following training loss:

$$\mathcal{L} = \lambda_1\mathcal{L}_{\mathrm{CSC}} + \lambda_2\mathcal{L}_{\mathrm{P}} + \lambda_3\mathcal{L}_{\mathrm{V}} + \lambda_4\mathcal{L}_{\mathrm{D}}, \tag{11}$$

where λ_i is the task weighting. The $\mathcal{L}_{\mathrm{CSC}}$ is the traditional CSC objective and the $\mathcal{L}_{\mathrm{P}}, \mathcal{L}_{\mathrm{V}}, \mathcal{L}_{\mathrm{D}}$ are the contrastive objectives we design for "Phonetics, Vision, Definition" knowledge respectively.

4 Experiments

In this section, we first introduce the experiment settings and the main performance of LEAD. Then we conduct detailed discussions and analyses to verify the effectiveness of our proposed methods.

4.1 Datasets

Training Data In all our experiments, we use the widely used training data of most previous works (Zhang et al., 2020; Liu et al., 2021; Xu et al., 2021), including the training sentences from SIGHAN13 (Wu et al., 2013b), SIGHAN14 (Yu et al., 2014), SIGHAN15 (Tseng et al., 2015), and the generated training sentences (the size of this part data is 271K, we denote them as Wang271K in our paper) (Wang et al., 2018).

Test Data To ensure the fairness of our experiments, we use the exact same test data as the baseline methods, which are from the SIGHAN13/14/15 test datasets. The details of the training/test data we use in our experiments are presented in Appendix A.1.

4.2 Baseline Methods

To evaluate the performance of LEAD, we select several latest CSC models as our baselines, including the previous state-of-the-art methods on SIGHAN13/14/15 datasets: **BERT** (Devlin et al., 2019) is fine-tuned on the training data only with the cross-entropy. **SpellGCN** (Cheng et al., 2020) introduces the confusion set information through GCNs. **GAD** (Guo et al., 2021) combines a global attention decoder with BERT and trains the model under a confusion set guided replacement strategy. **Two-Ways** (Li et al., 2021) continually identifies the model's weak spots to generate more valuable training sentences. **DCN** (Wang et al., 2021) utilizes the Pinyin enhanced candidate generator and proposes the dynamic connected networks to build the dependencies. **MLM-phonetics** (Zhang et al., 2021) introduces the phonetic features into

Dataset	Method	Detection Level Pre	Rec	F1	Correction Level Pre	Rec	F1
SIGHAN13	SpellGCN (Cheng et al., 2020)	80.1	74.4	77.2	78.3	72.7	75.4
	MLM-phonetics (Zhang et al., 2021)	82.0	78.3	80.1	79.5	77.0	78.2
	DCN (Wang et al., 2021)	86.8	79.6	83.0	84.7	77.7	81.0
	GAD (Guo et al., 2021)	85.7	79.5	82.5	84.9	78.7	81.6
	REALISE (Xu et al., 2021)	**88.6**	<u>82.5</u>	<u>85.4</u>	**87.2**	<u>81.2</u>	84.1
	Two-Ways (Li et al., 2021)	-	-	84.9	-	-	<u>84.4</u>
	BERT (Xu et al., 2021)	85.0	77.0	80.8	83.0	75.2	78.9
	LEAD	88.3	**83.4**	**85.8**	**87.2**	**82.4**	**84.7**
SIGHAN14	SpellGCN (Cheng et al., 2020)	65.1	69.5	67.2	63.1	67.2	65.3
	DCN (Wang et al., 2021)	67.4	70.4	68.9	65.8	68.7	67.2
	GAD (Guo et al., 2021)	66.6	71.8	69.1	65.0	70.1	67.5
	REALISE (Xu et al., 2021)	<u>67.8</u>	71.5	69.6	<u>66.3</u>	70.0	68.1
	Two-Ways (Li et al., 2021)	-	-	<u>70.4</u>	-	-	68.6
	MLM-phonetics (Zhang et al., 2021)	66.2	<u>**73.8**</u>	69.8	64.2	<u>**73.8**</u>	<u>68.7</u>
	BERT (Xu et al., 2021)	64.5	68.6	66.5	62.4	66.3	64.3
	LEAD	**70.7**	71.0	**70.8**	**69.3**	69.6	**69.5**
SIGHAN15	GAD (Guo et al., 2021)	75.6	80.4	77.9	73.2	77.8	75.4
	SpellGCN (Cheng et al., 2020)	74.8	80.7	77.7	72.1	77.7	75.9
	DCN (Wang et al., 2021)	77.1	80.9	79.0	74.5	78.2	76.3
	PLOME (Liu et al., 2021)	77.4	81.5	79.4	75.3	79.3	77.2
	MLM-phonetics (Zhang et al., 2021)	<u>77.5</u>	<u>**83.1**</u>	<u>80.2</u>	74.9	<u>80.2</u>	77.5
	REALISE (Xu et al., 2021)	77.3	81.3	79.3	<u>75.9</u>	79.9	77.8
	Two-Ways (Li et al., 2021)	-	-	80.0	-	-	<u>78.2</u>
	BERT (Xu et al., 2021)	74.2	78.0	76.1	71.6	75.3	73.4
	LEAD	**79.2**	82.8	**80.9**	**77.6**	**81.2**	**79.3**

Table 2: The performance of LEAD and baselines. For each dataset, we rank baselines from low to high performance according to the most important metric (i.e., correction level F1 score). Note that all results of baselines are directly from published papers. We <u>underline</u> the previous state-of-the-art performance for convenient comparison.

the ERNIE (Sun et al., 2020) and uses the enhanced ERNIE model for CSC. **PLOME** (Liu et al., 2021) pre-trains BERT with a confusion set-based masking strategy and utilizes GRU (Dey and Salem, 2017) to encode phonetics/strokes as input. **REALISE** (Xu et al., 2021) is a multimodal model which mixes the semantic, phonetic, and graphic information to improve the model performance.

4.3 Experimental Setup

The character/sentence-level metrics are both used in the CSC task. According to the sentence-level metric, one test sentence will be judged to be correct only when all the wrong characters in it are detected and corrected successfully. Therefore, the sentence-level metric is stricter than the character-level metric because some sentences may have multiple wrong characters. So we report the sentence-level metrics for the evaluation in all our experiments, this setting is also widely used in previous works (Li et al., 2021; Liu et al., 2021; Xu et al., 2021). More specifically, we report the metrics including Precision, Recall, and F1 score for detection and correction levels. At the detection level, all positions of wrong characters in a test sample should be detected correctly. At the correction level, we require the model must not only detect but also correct all the spelling errors. Additionally, other implementation details of our experiments are shown in Appendix A.2.

4.4 Main Results

From Table 2, we observe that:

1. Because LEAD is essentially a fine-tuning framework of BERT, its direct baseline should be the BERT. The comparison results between LEAD and BERT show that LEAD outperforms BERT significantly on SIGHAN13/14/15, which verifies the effectiveness of our proposed heterogeneous knowledge guided fine-tuning methods.

2. Compared with previous state-of-the-art models (i.e., Two-Ways, REALISE, and MLM-phonetics), our model utilizes only a thin BERT as the main body to achieve better performance, while REALISE and MLM-phonetics both explicitly introduce multi-

modal information into their inference process, which demonstrates the competitive performance of our proposed methods.

3. Considering the effect of different knowledge, LEAD is trained under the guidance of phonetics, vision, and definition knowledge, while most baselines (e.g., SpellGCN, DCN, and PLOME) also use different mechanisms to leverage the phonetics and vision knowledge. That our method outperforms these baselines indicates that the unique definition knowledge we focus on is very important for CSC.

4.5 Ablation Study

We explore the effectiveness of each contrastive learning objective in LEAD by conducting ablation studies with different variants. Specifically, in Table 3, MODEL + K, K $\in$ {P, V, D} means that we use the CSC training objective $\mathcal{L}_{\text{CSC}}$ and corresponding contrastive training objective $\mathcal{L}_{\text{K}}$ to train the MODEL. Besides, because REALISE has its own way of using vision/phonetics features, which makes $\mathcal{L}_{\text{V}}$ and $\mathcal{L}_{\text{P}}$ not meaningful, so we only perform $\mathcal{L}_{\text{D}}$ on REALISE.

From the three rows of results using a single training objective (i.e., BERT+V/P/D), we know that each of our proposed contrastive learning strategies leads to significant performance improvements when applied to BERT alone. Particularly, the phenomenon that BERT+P outperforms BERT+V at the correction level is in line with the fact that 83% of errors belong to phonological errors and 48% belong to visual errors in the real scene (Liu et al., 2021). Furthermore, we also see that all methods including the previous state-of-the-art model (i.e., REALISE) have further improvements after adding our proposed definition guided fine-tuning objective, which demonstrates that the definition information we focus on is very useful for enhancing CSC models.

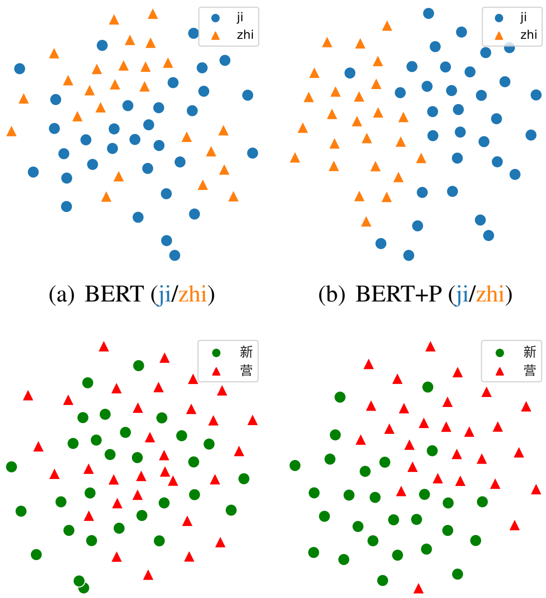
4.6 Analysis and Discussion

4.6.1 Visualization of Better Phonetic/Vision Representations

The key motivation of our proposed phonetics/vision guided fine-tuning is to refine the representations of the models for characters on different dimensions of knowledge. We hope that through the phonetics/vision guided fine-tuning, the model can be guided to represent characters with similar

Method	Det-F1	Cor-F1
BERT	76.1	73.4
+ V(ision)	78.4	77.1
+ P(honetics)	78.2	77.3
+ D(efinition)	79.0	77.4
+ V(ision) + P(honetics)	79.6	78.1
+ V(ision) + D(efinition)	78.9	78.2
+ P(honetics) + D(efinition)	80.3	78.3
REALISE	79.3	77.8
+ D(efinition)	80.3	78.6
LEAD	**80.9**	**79.3**

Table 3: Ablation results on the SIGHAN15 test set. Note that the LEAD is equivalent to BERT+V+P+D.



(a) BERT (ji/zhi) (b) BERT+P (ji/zhi)

(c) BERT (新/营) (d) BERT+V (新/营)

Figure 2: Visualization (t-SNE) of phonetically/ visually similar characters.

Pinyin/strokes closer, and characters with different Pinyin/strokes to represent farther. Therefore, it is necessary to visualize the representations of the characters before and after the model is combined with our methods. Specifically, we randomly select two groups of phonetically/visually similar characters (e.g., characters with similar Pinyin to “ji/zhi” and similar strokes to “新/营”), then apply BERT and BERT+P/V to obtain their representations. Finally, we use t-SNE to visualize these high-dimensional representations of characters.

Figure 2 shows the representation distribution of BERT and BERT+P/V for phonetically/visually similar characters. From the comparison of Figures 2(a) and 2(b), Figure 2(a)’s representation of characters is messy, while in 2(b), it can even be

Method	Pre	Rec	F1
	Detection Level		
BERT	74.2	78.0	76.1
LEAD (Random)	77.7	81.3	79.5
LEAD (First)	77.4	82.3	79.8
LEAD (Similar)	**79.2**	**82.8**	**80.9**
	Correction Level		
BERT	71.6	75.3	73.4
LEAD (Random)	75.8	80.6	78.1
LEAD (First)	76.7	80.2	78.4
LEAD (Similar)	**77.6**	**81.2**	**79.3**

Table 4: The results of LEAD on SIGHAN15 when using different word definition selection strategies.

seen that there is a clear boundary between the two kinds of characters, which indicates that after the optimization of phonetics guided fine-tuning, it does represent the phonetically similar characters closer. Also in the visual comparison, we see that the points of the two colors in Figure 2(c) are significantly more scattered, while Figure 2(d) is more orderly, which also verifies our motivation for proposing vision guided fine-tuning.

4.6.2 Effects of Different Word Definition Selection Strategies

As mentioned in Section 3.4, we design three different word definition selection strategies for the definition guided fine-tuning, namely "select a random definition" (Random), "select the first definition" (First), and "select the most similar definition" (Similar). To further empirically explain why these strategies we proposed are effective, we conduct analyses as shown in Table 4. We apply LEAD with different strategies on the SIGHAN15 dataset and observe the performance change.

From Table 4, we know that LEAD (Similar) has the best performance, followed by LEAD (First), and LEAD (Random) has the lowest improvement. Such results are consistent with the mechanism of these strategies. The better performance of LEAD (First) than LEAD (Random) shows that our observation on the dictionary is correct, that is, the first of multiple definitions of a word is often the most representative in most cases. Additionally, the best performance of LEAD (Similar) also proves the effectiveness of our designed selection strategy that is based on sentence similarity. It is worth mentioning that although the three strategies have different effects on the model performance, they all bring steady performance improvements

Input 1:	要永(yǒng)于面对逆境。 Please always face adversity.
Output 1:	要勇(yǒng)于面对逆境。 Please face adversity bravely.
Input 2:	秋天己经无声的来到了。 Autumn self come silently.
Output 2:	秋天已经无声的来到了。 Autumn has come silently.
Input 3:	迎接每一个固难，并克服它。 Meet every hardship and overcome it.
Output 3:	迎接每一个困难，并克服它。 Meet every difficulty and overcome it.
Definition:	【困难】: (名) 工作、生活中遇到的不易解决的问题或障碍, 克服～ (noun) Problems or obstacles in work and life that are not easy to solve, overcome～

Table 5: Examples of the input/output of LEAD. We mark the wrong/correct/key characters.

compared to the baseline method (i.e., BERT).

4.7 Case Study

From the first/second cases in Table 5, we know that our LEAD perceives the phonetic and visual similarity of Chinese characters, so as to accurately detect the wrong positions and make reasonable corrections. Particularly, for the first example, if ignoring the phonetic similarity, there are other candidate characters such as "乐(lè)" and "敢(yǒng)". But the LEAD's output is the best correction because it is more in line with the essential of CSC. Additionally, in the third example, "固(gù)" and "困(kùn)" are neither phonetically nor visually similar, and LEAD successfully correct this case because it perceives the definition of "困难" in the dictionary. Without the help of the definition, we can replace the "固(gù)" with the "苦(kǔ)" which is more phonetically similar to "固(gù)". However, in daily use of Chinese, the combination of "克服" and "苦难" is not common. Therefore, this example just reflects the importance of definition knowledge we are concerned with for CSC.

5 Conclusion

In this paper, we propose to promote CSC by utilizing various knowledge contained in the dictionary. We introduce LEAD, a unified fine-tuning framework that aims to perform contrastive learning for three kinds of heterogeneous knowledge. Extensive experiments and empirical analyses verify the motivation of our study and the effectiveness of our proposed methods. The dictionary knowledge we focus on is not only beneficial for CSC, but also crucial for other Chinese text processing tasks. Therefore, in the future, we will continue to mine the knowledge contained in the dictionary to improve other Chinese text processing tasks.

6 Limitations

In this section, we discuss the limitations of our work in detail and propose corresponding solutions that we believe are feasible.

6.1 Language Limitation

Our work and the proposed method focus on the Chinese Spell Checking (CSC) task. The language characteristics of Chinese are very different from other languages such as English. For example, the phonetically or visually characters, which bring great challenge to CSC, does not exist in English. Therefore, the limitation of language characteristics prevents our method from being directly transferable to English scenarios. However, we also believe that the definition knowledge in the dictionary we are concerned with still has important implications for English text error correction.

6.2 Encoder Selection

Our proposed LEAD framework is a unified fine-tuning framework to guide the CSC models to learn heterogeneous knowledge. The unified paradigm allows LEAD to impose no strict restrictions on the various encoders used in it. To verify the effectiveness of LEAD, in our experiments, we just choose the simple configuration as E_P, E_V, E_D (see Appendix A.2). In the future, we suggest that more complex models and configurations can be used for more performance improvements.

6.3 Running Efficiency

As academic verification experiments, we do not consider the running efficiency of our proposed methods in the specific code implementation. Specifically, it takes about 10 hours on 1 V100 GPU to finish the training process and it takes up to 24G GPU memory. We think that there are at least two solutions to improve efficiency: (1) Deploying the model training process to multi-GPUs and using data-parallel operations can increase the training batch size and shorten the training time. (2) Change the online positive and negative sample construction to offline, that is, various positive and negative sample pairs for training are constructed and stored in advance, which can also greatly save the time cost during training.

Acknowledgement

This research is supported by National Natural Science Foundation of China (Grant No.62276154 and 62011540405), Beijing Academy of Artificial Intelligence (BAAI), the Natural Science Foundation of Guangdong Province (Grant No. 2021A1515012640), Basic Research Fund of Shenzhen City (Grant No. JCYJ20210324120012033 and JSGG20210802154402007), and Overseas Cooperation Research Fund of Tsinghua Shenzhen International Graduate School (Grant No. HW2021008).

References

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

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.

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.

Rahul Dey and Fathi M Salem. 2017. Gate-variants of gated recurrent unit (gru) neural networks. In *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, pages 1597–1600. IEEE.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

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.

R. Hadsell, S. Chopra, and Y. LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020a. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020b. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9726–9735. Computer Vision Foundation / IEEE.

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.

Tuo Ji, Hang Yan, and Xipeng Qiu. 2021. SpellBERT: A lightweight pretrained model for Chinese spelling check. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3544–3551, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*.

Taeuk Kim, Kang Min Yoo, and Sang-goo Lee. 2021. Self-guided contrastive learning for BERT sentence representations. 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 2528–2540, 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.

Yinghui Li, Yangning Li, Yuxin He, Tianyu Yu, Ying Shen, and Hai-Tao Zheng. 2022a. Contrastive learning with hard negative entities for entity set expansion. *arXiv preprint arXiv:2204.07789*.

Yinghui Li, Qingyu Zhou, Yangning Li, Zhongli Li, Ruiyang Liu, Rongyi Sun, Zizhen Wang, Chao Li, Yunbo Cao, and Hai-Tao Zheng. 2022b. The past mistake is the future wisdom: Error-driven contrastive probability optimization for Chinese spell checking. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3202–3213, Dublin, Ireland. 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.

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

Boer Lyu, Lu Chen, and Kai Yu. 2021. Glyph enhanced Chinese character pre-training for lexical sememe prediction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4549–4555, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Yujia Qin, Yankai Lin, Ryuichi Takanobu, Zhiyuan Liu, Peng Li, Heng Ji, Minlie Huang, Maosong Sun, and Jie Zhou. 2021. ERICA: Improving entity and relation understanding for pre-trained language models via contrastive learning. 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 3350–3363, Online. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie 2.0: A continual pre-training framework for language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8968–8975.

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.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748.

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.

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.

A Appendix

A.1 Datasets Details

Please kindly note that the original sentences of SIGHAN datasets are in Traditional Chinese, so we need to convert these original texts to Simplified Chinese using the OpenCC tool[4]. This data pre-process procedure has been widely used in previously published works (Wang et al., 2019; Cheng et al., 2020; Zhang et al., 2020). The details of the datasets we use in our experiments are presented in Table 6.

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 6: Statistics of the datasets that we use in experiments. We report the number of sentences (#Sent), the average sentence length (Avg.Length), and the number of spelling errors (#Errors).

A.2 Implementation Details

In our experiments, all the source code is implemented using Pytorch (Paszke et al., 2019) based on the Huggingface's Transformer library[5] (Wolf et al., 2020). For the implementation of $\mathrm{E_C}$, we use the cross-entropy function as the $\mathcal{L}_{\mathrm{CSC}}$ and BERT as the main CSC model. The BERT's architecture we use in our experiments is the same as the $BERT_{BASE}$, which has 12 transformers layers with 12 attention heads and its hidden state size is 768. And the initial weights of BERT are from the weights of Chinese BERT-wwm (Cui et al., 2020). For the implementation of $\mathrm{E_P}$, $\mathrm{E_V}$, $\mathrm{E_D}$, we preliminarily select the BERT consistent with the above description as $\mathrm{E_P}$ and $\mathrm{E_D}$, and we use the glyph enhanced pre-training model proposed in Lyu et al. (2021) as $\mathrm{E_V}$ to obtain the strokes representations of Chinese characters.

We set the maximum sentence length to 128. We train LEAD with the AdamW optimizer (Loshchilov and Hutter, 2018) for 10 epochs and set the training batch size to 32. The model is trained with learning rate warming up and linear decay, while the initial learning rate is set to 5e-5. The negative pairs size N of a mini-batch is set to 8 when we report the main results of LEAD. Besides, the weighting factors λ_i of $\mathcal{L}$ are all set to 1.

As mentioned in (Cheng et al., 2020; Xu et al., 2021; Li et al., 2022b), lots of the mixed usage of auxiliary (such as “的”, “地”, and “得”) are wrongly annotated, which makes the quality of the SIGHAN13 test dataset very poor. To alleviate this problem and more accurately evaluate the performance of models on SIGHAN13, there exist two main solutions in previous works. To avoid the over-fitting problem brought by the method proposed in (Cheng et al., 2020) that continues to fine-tune the trained model on the SIGHAN13 training data before testing, we follow the post-processing method implemented in (Xu et al., 2021; Li et al., 2022b) and don't consider all the detected/corrected mixed auxiliary, which will not compromise the fairness of our experiments and can better reflect the model's real performance.

[4]https://github.com/BYVoid/OpenCC
[5]https://github.com/huggingface/transformers