Investigating Glyph-Phonetic Information for Chinese Spell Checking: What Works and What's Next?

Xiaotian Zhang[*], **Yanjun Zheng**[*], **Hang Yan**, **Xipeng Qiu**[†]
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University
{xiaotianzhang21, yanjunzheng21}@m.fudan.edu.cn {hyan19, xpqiu}@fudan.edu.cn

Abstract

While pre-trained Chinese language models have demonstrated impressive performance on a wide range of NLP tasks, the Chinese Spell Checking (CSC) task remains a challenge. Previous research has explored using information such as glyphs and pronunciations to improve the ability of CSC models to distinguish misspelled characters, with good results at the accuracy level on public datasets. However, the generalization ability of these CSC models has not been well understood: it is unclear whether they incorporate glyph-phonetic information and, if so, whether this information is fully utilized. In this paper, we aim to better understand the role of glyph-phonetic information in the CSC task and suggest directions for improvement. Additionally, we propose a new, more challenging, and practical setting for testing the generalizability of CSC models. Our code will be released at https://github.com/piglaker/ConfusionCluster.

1 Introduction

Spell checking (SC) is the process of detecting and correcting spelling errors in natural human texts. For some languages, such as English, SC is relatively straightforward, thanks to the use of tools like the Levenshtein distance and a well-defined vocabulary. However, for Chinese, Chinese spell checking (CSC) is a more challenging task, due to the nature of the Chinese language. Chinese has a large vocabulary consisting of at least 3,500 common characters, which creates a vast search space and an unbalanced distribution of errors (Ji et al., 2021). Moreover, substitutions or combinations of characters can significantly alter the meaning of a Chinese sentence while still being grammatically correct. The CSC task, therefore, requires requires the output to retain as much of the original meaning and wording as possible. Figure 1 shows different types of errors and corresponding target characters. Previous work has attempted to incorporate inductive bias to model the relationship between Chinese character glyphs, pronunciation, and semantics (Xu et al., 2021).

Glyphic Similar Case	
Misspelled character	我要去城南 **贝(bèi, *seashell*)** 我的奶奶。
Expected correction	我要去城南 **见(jiàn, *see*)** 我的奶奶。
Unexpected correction	我要去城南 **接(jiē, *pick up*)** 我的奶奶。
Translation	I'm going to **see** my grandmother in the south of the city.
Pronounced Similar Case	
Misspelled character	在医学上的应用范 **蠡(lǐ, *motheaten*)** 。
Expected correction	在医学上的应用范 **例(lì, *instance*)** 。
Unexpected correction	在医学上的应用范 **围(wéi, *range*)** 。
Translation	**Examples** of applications in medicine.

Figure 1: An example of different errors affecting CSC results. red/green/blue represents the misspelled character, the expected correction and the unexpected correction.

In recent years, pre-trained language models (PLMs) have shown great success in a wide range of NLP tasks. With the publication of BERT (Devlin et al., 2018), using PLMs for CSC tasks has become a mainstream approach, with examples including FASpell (Hong et al., 2019), Softmasked-BERT (Zhang et al., 2020), SpellGCN (Cheng et al., 2020), and PLOME (Liu et al., 2021). Some researchers have focused on the special features of Chinese characters in terms of glyphs and pronunciations, aiming to improve the ability to distinguish misspelled characters by incorporating glyph-phonetic information (Ji et al., 2021; Liu et al., 2021; Xu et al., 2021). However, despite these advances, the generalization of CSC models to real-world applications remains limited. How can we improve the generalization ability of CSC models? Can current models recognize and utilize glyph-phonetic information to make predictions? As we

[*] These two authors contributed equally.
[†] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2023, pages 1–13
July 9-14, 2023 ©2023 Association for Computational Linguistics

re-examine previous work, we have identified some previously unexplored issues and potential future directions for research.

Q1: ***Do existing Chinese pre-trained models encode the glyph-phonetic information of Chinese characters?*** Chinese writing is morpho-semantic, and its characters contain additional semantic information. Before studying existing CSC models, we seek to investigate whether existing mainstream Chinese pre-trained language models are capable of capturing the glyph-phonetic information.

Q2: ***Do existing CSC models fully utilize the glyph-phonetic information of misspelled characters to make predictions?*** Intuitively, introducing glyph-phonetic information in the CSC task can help identify misspelled characters and improve the performance of the model. However, there has been little research on whether existing CSC models effectively use glyph-phonetic information in this way.

Empirically, our main observations are summarized as follows:

- We show that Chinese PLMs like BERT encode glyph-phonetic information without explicit introduction during pre-training, which can provide insight into the design of future Chinese pre-trained models. We also propose a simple probe task for measuring how much glyph-phonetic information is contained in a Chinese pre-trained model.
- We analyze the ability of CSC models to exploit misspelled characters and explain why current CSC methods perform well on test sets but poorly in practice. We propose a new probe experiment and a new metric Correction with Misspelled Character Coverage Ratio (CCCR).
- We propose a new setting for the CSC task, called isolation correction, to better test the generalizability and correction performance of CSC models. This setting alleviates the shortcuts present in the original dataset, making the CSC task more challenging and realistic.

We hope that this detailed empirical study will provide follow-up researchers with more guidance on how to better incorporate glyph-phonetic information in CSC tasks and pave the way for new state-of-the-art results in this area.

2 Related Work

2.1 Glyph Information

Learning glyph information from Chinese character forms has gained popularity with the rise of deep neural networks. After word embeddings (Mikolov et al., 2013b) were proposed, early studies (Sun et al., 2014; Shi et al., 2015; Yin et al., 2016) used radical embeddings to capture semantics, modeling graphic information by splitting characters into radicals. Another approach to modeling glyph information is to treat characters as images, using convolutional neural networks (CNNs) as glyph feature extractors (Liu et al., 2010; Shao et al., 2017; Dai and Cai, 2017; Meng et al., 2019). With pre-trained language models, glyph and phonetic information are introduced end-to-end. ChineseBERT(Sun et al., 2021) is a pre-trained Chinese NLP model that flattens the image vector of input characters to obtain the glyph embedding and achieves significant performance gains across a wide range of Chinese NLP tasks.

2.2 Phonetic Infomation

Previous research has explored using phonetic information to improve natural language processing (NLP) tasks. Liu et al. propose using both textual and phonetic information in neural machine translation (NMT) by combining them in the input embedding layer, making NMT models more robust to homophone errors. There is also work on incorporating phonetic embeddings through pre-training. Zhang et al. propose a novel end-to-end framework for CSC with phonetic pre-training, which improves the model's ability to understand sentences with misspellings and model the similarity between characters and pinyin tokens. Sun et al. apply a CNN and max-pooling layer on the pinyin sequence to derive the pinyin embedding.

2.3 Chinese Spell Checking

2.3.1 Task Description

Under the language model framework, Chinese Spell Checking is often modeled as a conditional token prediction problem. Formally, let $X = c_1, c_2, \ldots, c_T$ be an input sequence with potentially misspelled characters c_i. The goal of this task is to discover and correct these errors by estimating the conditional probability $P(y_i|X)$ for each misspelled character c_i.

2.3.2 CSC Datasets

We conduct experiments on the benchmark SIGHAN dataset (Wu et al., 2013; Yu et al., 2014; Tseng et al., 2015), which was built from foreigners' writings and contains 3,162 texts and 461 types of errors. However, previous studies have reported poor annotation quality in SIGHAN13 and SIGHAN14 (Wu et al., 2013; Yu et al., 2014), with many errors, such as the mixed usage of auxiliary characters, remaining unannotated (Cheng et al., 2020). To address these issues and enable fair comparisons of different models, we apply our probe experiment to the entire SIGHAN dataset and use only clean SIGHAN15 for metrics in our review. The statistics of the dataset are detailed in the appendix.

2.3.3 Methods for CSC

To investigate the role of glyph-phonetic information in CSC, we conduct a probe experiment using different Chinese PLMs as the initial parameters of the baseline. The models we use are detailed in the appendix. For our first probe experiment, we use the out-of-the-box BERT model as a baseline. We input the corrupted sentence into BERT and get the prediction for each token. If the predicted token for the corresponding output position is different from its input token, we consider BERT to have detected and corrected the error (Zhang et al., 2022). We also consider two previous pre-trained methods that introduced glyph and phonetic information for CSC. PLOME (Liu et al., 2021) is a pre-trained masked language model that jointly learns how to understand language and correct spelling errors. It masks chosen tokens with similar characters according to a confusion set and introduces phonetic prediction to learn misspelled knowledge at the phonetic level using GRU networks. RealiSe (Xu et al., 2021) leverages the multimodal information of Chinese characters by using a universal encoder for vision and a sequence modeler for pronunciations and semantics.

2.4 Metrics

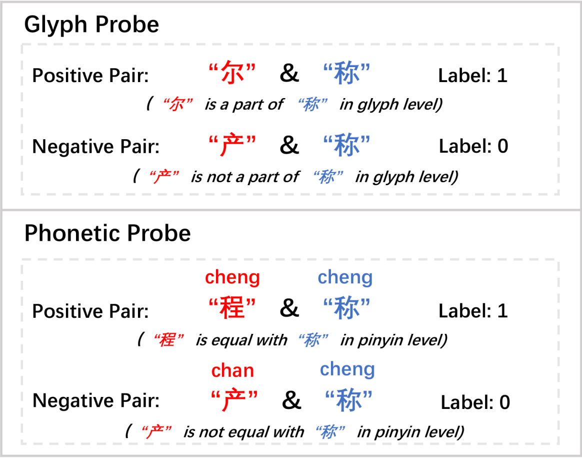
For convenience, all Chinese Spell Checking metrics in this paper are based on the sentence level score(Cheng et al., 2020). We mix the original SIGHAN training set with the enhanced training set of 270k data generated by OCR- and ASR-based approaches (Wang et al., 2018) which has been widely used in CSC task.

3 Experiment-I: Probing for Character Glyph-Phonetic Information

In this section, we conduct a simple MLP-based probe to explore the presence of glyph and phonetic information in Chinese PLMs and to quantify the extent to which tokens capture glyph-phonetic information. We consider glyph and phonetic information separately in this experiment.

3.1 Glyph Probe

For glyphs, we train a binary classifier probe to predict if one character is contained within another character. We use the frozen embeddings of these characters from Chinese PLMs as input. That is, as shown in the upper part of Figure 2, if the probe is successful, it will predict that "称" contains a "尔" at the glyph level however not "产" (it is difficult to define whether two characters are visually similar, so we use this method as a shortcut).



Glyph Probe
Positive Pair: "尔" & "称" Label: 1
("尔" is a part of "称" in glyph level)
Negative Pair: "产" & "称" Label: 0
("产" is not a part of "称" in glyph level)
Phonetic Probe
cheng cheng
Positive Pair: "程" & "称" Label: 1
("程" is equal with "称" in pinyin level)
chan cheng
Negative Pair: "产" & "称" Label: 0
("产" is not equal with "称" in pinyin level)

Figure 2: Examples of the input and label in Experiment-I MLP Probe. We highlight the two characters in red/blue color.

For the glyph probe experiment, we consider the static, non-contextualized embeddings of the following Chinese PLMs: BERT (Cui et al., 2019), RoBERTa (Cui et al., 2019), Chinese-BERT (Sun et al., 2021), MacBERT (Cui et al., 2020), CPT (Shao et al., 2021), GPT-2 (Radford et al., 2019), BART (Shao et al., 2021), and T5 (Raffel et al., 2020). We also use Word2vec (Mikolov et al., 2013a) as a baseline and a completely randomized Initial embedding as a control. See Appendix C.1 for details on the models used in this experiment.

The vocabulary of different Chinese PLMs is similar. For convenience, we only consider the characters that appear in the vocabulary of BERT, and we also remove the characters that are rare and too complex in structure. The details of our datasets for the probe are shown in Appendix C.2.

We divide the character w into character component $\{u_1, u_2, \ldots, u_i\}$ using a character splitting tool[1]. That is, "称" will be divided into "禾" and "尔". The set of all characters (e.g. "称") is $\mathcal{W} = \{w_1, w_2, \ldots, w_d\}$, where d is number of characters. The set of all components of characters (e.g. "禾", "尔") is $\mathcal{U} = \{u_1, u_2, \ldots, u_c\}$, where c is the number of components of each character. If u_i exists in w_i, in other words, is a component of w_i in glyph level, then u_i, w_i is a positive example, and vice versa is a negative example. Then, we constructed a positive dataset $\mathcal{D}_{pos} = \{\{u_1, w_1\}, \{u_2, w_1\}, \ldots, \{u_i, w_d\}\}$, where the u corresponds to w separately. Also, we constructed a balanced negative dataset $\mathcal{D}_{neg} = \{\{u_1^n, w_1\}, \{u_2^n, w_1\}, \ldots, \{u_i^n, w_d\}\}$, where d is equal to $\mathcal{D}_{pos}$ and u^n is randomly selected in the set U. We mix $\mathcal{D}_{pos}$ and $\mathcal{D}_{neg}$ and split the dataset into training and test according to the ratio of 80:20 to ensure that a character only appears on one side.

We train the probe on these PLMs' static non-trainable embeddings. For every u_i, w_i, we take the embedding of u_i and w_i, and concatenation them as the input x_i. The classifier trains an MLP to predict logits $\hat{y_i}$, which is defined as :

$$\hat{y_i} = \text{Sigmoid}(\text{MLP}(x_i))$$

To control the variables as much as possible and mitigate the effects of other factors on the probe experiment, we also experimented with the number of layers of MLP. The results of this are detailed in Appendix C.3.

3.2 Phonetic Probe

For phonetics, we train another binary classifier probe to predict if two characters have the similar pronunciation, also using the frozen embeddings of these characters from Chinese PLMs as input. The meaning of 'similar' here is that the pinyin is exactly the same, but the tones can be different. That is, as shown in the lower part of Figure 2, if the probe is successful, it will predict that "称"*(cheng)* has the similar pronunciation with "程"*(cheng)* however not "产"*(chan)*. The pronunciation information for the Chinese characters comes from the pypinyin[2] toolkit.

We consider static non-contextualized embedding of Chinese PLMs, which are the same as the glyph probe. We also mainly analyze the characters in the vocabulary of BERT, and mainly consider common characters.

The dataset construction is also similar to the glyph probe. To create positive examples, for each character w_i in character list W, we find a character u_i which has the similar pronunciation as w_i, then u_i, w_i is a positive example. For each positive, we also find a character s_i which has a different pronunciation from w_i to construct negative example s_i, w_i. For example, the positive example is the two characters with similar pronunciation, such as "称" *(cheng)* and "程"*(cheng)*. And the negative example is the two characters with different pronunciation, such as "称"*(cheng)* and "产"*(chan)*. The divide ratio and other settings are the same as the glyph probe.

We train the probe on these PLMs' static non-trainable embeddings as the glyph probe and also concatenate the embeddings of the pairs as input.

3.3 Results and Analysis

The following conclusions can be drawn from Figure 3.

The Chinese PLMs encoded the glyph information of characters From the results, we can see that for glyphs, all models outperform the control model. The results of the control are close to 50% that there is no glyph information encoded in the input embedding, and the model guesses the result randomly. Comparing Word2vec and other Chinese PLMs side-by-side, we find that the large-scale pre-trained model has a significant advantage over Word2vec, suggesting that large-scale pre-training can lead to better representation of characters. In addition, we find that the results of these Chinese PLMs are concentrated in a small interval. ChineseBERT boasts of introducing glyph-phonetic information, which do not have advantages in glyph.

PLMs can hardly distinguish the phonetic features of Chinese characters In our experiments, the control group performed similarly to the phonetic probe, with an accuracy of approximately 50%. Unlike the glyph probe, the accuracy of Word2vec and other Chinese PLMs are also low in this probe. However, the introduction of phonetic embedding allowed ChineseBERT to perform significantly better than the other models. Our anal-

[1] https://github.com/howl-anderson/hanzi_chaizi

[2] https://github.com/mozillazg/python-pinyin

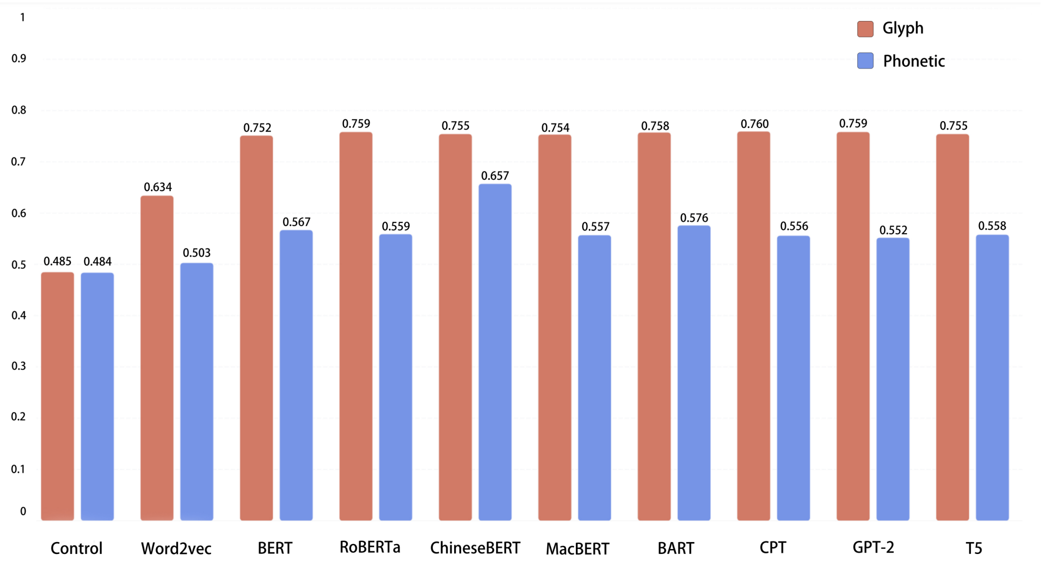


Figure 3: Results of Probe for Chinese PLMs. We found that the language models modeled by different paradigms are roughly close in perceiving graphical information but weak in speech. It is worth noting that ChineseBERT performs more significantly on this probe, probably because it explicitly introduces graphical and pronunciation information from the embedding stage.

ysis suggests that current Chinese PLMs may have limited phonetic information.

Method	Acc.
Control	0.485
Word2vec	0.634
BERT	0.752
RoBERTa	0.759
ChineseBERT	0.755
BERT-trained	0.756
RoBERTa-trained	0.757
ChineseBERT-trained	0.759

Table 1: Results of Probe for Models trained on the CSC task.We find that training on spell checking dataset does not enhance the graphical perception capability of models.

Model training on the CSC task does not enrich glyph and phonetic information We perform the same two probes using models fine-tuned on the SIGHAN dataset. We aim to investigate whether the training for the CSC task could add glyph and phonetic information to the embeddings, and the results are shown in Table 1. We found that the difference between the fine-tuned and untrained models is almost negligible, indicating that the relevant information is primarily encoded during the pre-training stage.

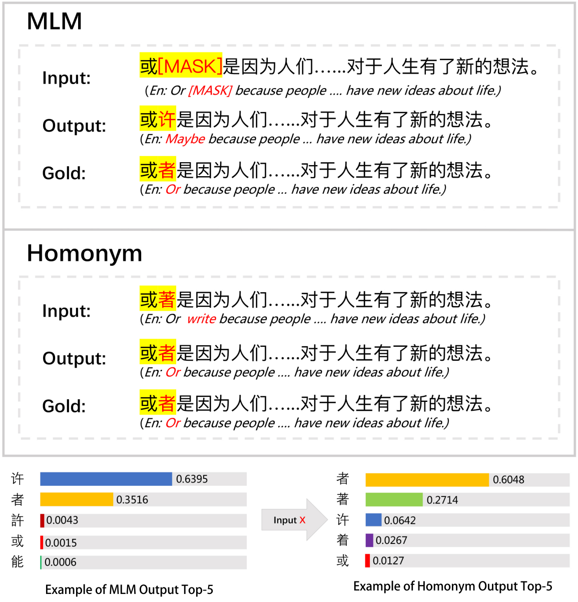
4 Experiment-II: Probing for Homonym Correction

In this experiment, we aim to explore the extent to which existing models can make use of the information from misspelled characters. To do this, we propose a new probe called Correction with Misspelled Character Coverage Ratio(CCCR), which investigates whether the model can adjust its prediction probability distribution based on the glyph-phonetic information of misspelled characters when making predictions.

4.1 Correction with Misspelled Character Coverage Ratio

Measure models utilizing the misspelled characters In this paper, we propose a method to evaluate the ability of a model to make predictions using additional information from misspelled characters, as well as to assess whether the model contains glyph-phonetic information.

Assume that $\mathcal{C}$ is a combination set of all possible finite-length sentence C_i in the languages L, $\mathcal{C} = \{C_0, ..., C_i, ...\}$, $C_i = \{c_{i,1}, ..., c_{i,n}, ...\}$, while $c_{i,j} \in L$. Let sentence $C_i^{n,a}$ be $C_i^{n,a} = \{c_{i,1}, ..., c_{i,n-1}, a, c_{i,n+1}, ...\}$, then assume that the representation learning model, let $H^w(C)$ be the hiddens of model w, X_i is an example in $\mathcal{C}$, For model w, the probability of token in position i should be:



MLM

Input: 或[MASK]是因为人们……对于人生有了新的想法。
(*En: Or [MASK] because people .... have new ideas about life.)*

Output: 或许是因为人们……对于人生有了新的想法。
(*En: Maybe because people ... have new ideas about life.)*

Gold: 或者是因为人们……对于人生有了新的想法。
(*En: Or because people ... have new ideas about life.)*

Homonym

Input: 或著是因为人们……对于人生有了新的想法。
(*En: Or write because people .... have new ideas about life.)*

Output: 或者是因为人们……对于人生有了新的想法。
(*En: Or because people .... have new ideas about life.)*

Gold: 或者是因为人们……对于人生有了新的想法。
(*En: Or because people .... have new ideas about life.)*

Figure 4: Take BERT as an example. The first half shows examples of MLM and Homonym respectively. The bottom half shows the change in the probability distribution predicted by the model in this example.

$$P\left(y_i = j|X_i, w\right) = \text{softmax}\left(WH^w(X_i) + b\right)[j]$$

Dataset $\mathcal{D}$ is a subset of $\mathcal{C}$, Then we can approximate the probability of the model. The CCCR is composed of $\mathcal{MLM}$ and $Homonym$. The former indicates which samples need the information on misspelled characters to be corrected while the latter shows which sample models adjust the output distribution. We take the intersection to get the frequency of whether the model is adjusted for the samples whose distribution should be adjusted.

$\mathcal{MLM}$ MLM is a subset of dataset $\mathcal{D}$. For all input sentence $C_i \in \mathcal{D}, C_i = \{c_1, c_2, [MASK], \dots, c_T\}$ and the position of $[MASK]$ is spelling error, let special token $mask = [MASK]$, $C_i \in MLM$ if:

$$P\left(y_i = noise\middle|C_i^{n,mask}, w\right) > P\left(y_i = Y_i\middle|C_i^{n,mask}, w\right)$$

$Homonym$ Same to MLM, For input sentence $C_i \in \mathcal{D}, C_i = \{c_1, c_2, c_{misspelled}, \dots, c_T\}$ and the position of $c_{misspelled}$ is spelling error. For all sentences C_i in the dataset $\mathcal{D}$, $C_i \in Homonym$ if:

$$P(y_i = Y_i|C_i^{n,c_{misspelled}}, w)) > P(y_i = noise|C_i^{n,c_{misspelled}}, w)$$

Correction with Misspelled Character Coverage Ratio (CCCR) The measured ratio is used to describe the lower bound of the probability that the model uses the information of the misspelled characters for the sentences C_i in the dataset $\mathcal{C}$.

$$CCCR = \frac{|\{C_i|C_i \in MLM \wedge C_i \in Homonym\}|}{|\{C_i|C_i \in MLM\}|}$$

Baseline Independently, we give an estimation method for the base value. Given model w, $noise$, dataset $\mathcal{D}$, ground truth correction y. The baseline of CCCR should be estimated as:

$$guess_i = \frac{P(y_i = noise|C_i^{n,mask}, w)}{1 - P(y_i = noise|C_i^{n,mask}, w)}$$

$$CCCR_{baseline} = \frac{\sum_{i \in S} \{1 * guess_i\}}{|\{C_i \mid C_i \in MLM\}|}$$

The baseline can be understood as a model with no glyph-phonetic information at all, and the probability of being able to guess the correct answer. But no such language model exists. For this purpose, instead of inputting the misspelled characters into the model, we artificially design strategies for the model to randomly guess answers by weight from the remaining candidates, which is equivalent to the probability of being able to guess correctly.

This probability is comparable to CCCR. CCCR restricts the condition for y to overtake $noise$. In the case of baseline, considering rearranging the candidates, the probability of y overtaking noise can also be re-normalized by probability.

4.2 Isolation Correction Setting Experiment

In the previous section, we test CCCR on the model finetuned on the SIGHAN dataset then found the CCCR of the models approached 92%. The results are shown in Table 3. As shown in Table 4, we analyze the overlap of correction pairs in the training and test sets in the SIGHAN dataset.

To test the model generalization ability, we design Isolation Correction Task, which removes all overlapped pairs in the training set and duplicate pairs in the test set. With isolation, the training set is reduced by about 16%. We believe that such a setup can better test the generalizability of the model and is more challenging and practical. Within the CCCR probe, We explore the ability of the model whether rely on its information, not just the ability to remember the content on the isolated SIGHAN dataset. The result is shown in Table 2

Method	MLM	Homonym	CCCR	Precision	Recall	F1
Baseline	-	-	15.61	-	-	-
BERT-Initial	45.58	64.87	34.57	-	-	-
RoBERTa-Initial	46.53	60.19	28.17	-	-	-
ChineseBERT-Initial	44.97	62.22	31.17	-	-	-
BERT	48.57	67.73	41.67	43.72	26.93	33.32
RoBERTa	48.70	64.80	36.12	39.82	27.14	32.27
ChineseBERT	46.33	67.39	40.32	42.56	27.26	33.23
PLOME	55.63	88.38	80.83	42.63	37.15	39.70
ReaLiSe	51.29	84.23	78.14	52.26	19.23	28.11

Table 2: Model performance in the isolation correction setting of SIGHAN15. '-Initial' means without any training.

Method	MLM	Homonym	CCCR	Precision	Recall	F1
Baseline	-	-	15.61	-	-	-
BERT	52.64	95.78	92.1	70.15	75.46	72.71
RoBERTa	47.07	95.92	91.77	70.49	74.91	72.63
ChineseBERT	48.57	97.62	96.83	73.24	76.75	74.59

Table 3: Model performance in the original version of SIGHAN15, which is finetuned. We found that the CCCR of the model fine-tuned on the CSC dataset is very high. We found that this is caused by overlapped pairs in the training data.

	#Pairs Count	#sent
Training Set	23140	284196
Test Set	824	2162
Training Set ∩ Test Set	799	-
Training Set ∪ Test Set	23165	-
Isolation Training Set	20758	230525
Isolation Test Set	824	2162

Table 4: The overlap of the correction pairs in the train and test sets and the statistics of the isolation SIGHAN set.

Between CCCR and F1 score, the mismatch phenomenon we refer to as stereotype is observed. The correction pair remembered while training harms the generalization of models.

4.3 Results and Analysis

We conducted experiments on three generic Chinese PLMs, BERT, RoBERTa, and ChineseBERT, and two CSC Models, PLOME, and Realise. We compare the metrics difference between the Initial model and the model after finetuning the isolation training set. The result is shown in Table 2.

CCCR and F1 values mismatch Our experimental results show that the CCCR and F1 values mismatch for CSC models. In the isolation training setting, we observed that the F1 values of PLOME and ReaLise are both significantly lower than their performance in Table 2, indicating that their ability to make correct predictions is primarily based on the memory of correction pairs in the training set. However, their CCCR values remained high, suggesting that they are able to discriminate glyph-phonetic information but are not able to correct it effectively.

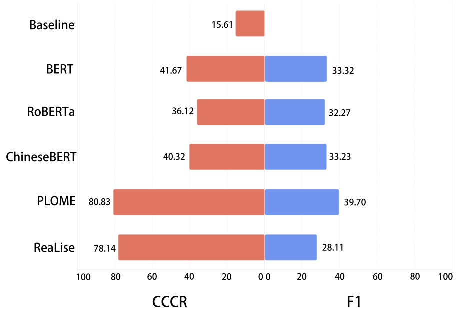


Figure 5: Results of CCCR Probe. We observe CCCR and F1 values mismatch. and for the pre-trained CSC model, we observe a phenomenon we call stereotype, which maintains a high CCCR under the isolation setting while performing worse on the F1 score, implying that stereotyping during pre-training weakens the generalization of the model.

Stereotype harm the generalization ability of the model in Isolation Correction Experiments These results suggest that the correction performance of the models is primarily dependent on their memory ability and that a strong reliance on memory can hinder generalization. The poor performance in the isolation setting indicates that none of the current methods generalize well, which presents a significant challenge for future CSC research. We recommend that future research in this field follow the isolation experiment setting to address this challenge.

5 Conclusion

In this paper, we have explored the role of glyph-phonetic information from misspelled characters in Chinese Spell Checking (CSC). Based on our experimental results, we have reached the following conclusions:

- Current Chinese PLMs encoded some glyph information, but little phonetic information.
- Existing CSC models could not fully utilize the glyph-phonetic information of misspelled characters to make predictions.
- There is a large amount of overlap between the training and test sets of SIGHAN dataset, which is not conducive to testing the generalizability of the CSC model. We propose a more challenging and practical setting to test the generalizability of the CSC task.

Our detailed observations can provide valuable insights for future research in this field. It is clear that a more explicit treatment of glyph-phonetic information is necessary, and researchers should consider how to fully utilize this information to improve the generalizability of their CSC models. We welcome follow-up researchers to verify the generalizability of their models using our proposed new setting.

6 Limitation

6.1 Limited number of CSC models tested

During our research, we encountered difficulties in reproducing previous models due to unmaintained open source projects or the inability to reproduce the results claimed in the papers. As a result, we are unable to test all of the available models.

6.2 Limited datasets for evaluating model performance

There are currently few datasets available for the CSC task, and the mainstream SIGHAN dataset is relatively small. The limited size of the data used to calculate the metrics may not accurately reflect the performance of the models. Furthermore, we found that the quality of the test set is poor, the field is narrow, and there is a large gap between the test set and real-world scenarios.

Acknowledgments

This work was supported by the National Key Research and Development Program of China (No.2020AAA0106700) and National Natural Science Foundation of China (No.62022027). We would like to express our gratitude to all the reviewers for their diligent, careful, and responsible feedback.

References

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. *arXiv preprint arXiv:2004.14166*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. *arXiv preprint arXiv:2004.13922*.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2019. Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.

Falcon Dai and Zheng Cai. 2017. Glyph-aware embedding of Chinese characters. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 64–69, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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.

Hairong Liu, Mingbo Ma, Liang Huang, Hao Xiong, and Zhongjun He. 2019. Robust neural machine translation with joint textual and phonetic embedding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3044–3049, Florence, Italy. Association for Computational Linguistics.

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.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for chinese character representations. *Advances in Neural Information Processing Systems*, 32.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 173–183, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yunfan Shao, Zhichao Geng, Yitao Liu, Junqi Dai, Fei Yang, Li Zhe, Hujun Bao, and Xipeng Qiu. 2021. Cpt: A pre-trained unbalanced transformer for both chinese language understanding and generation. *arXiv preprint arXiv:2109.05729*.

Xinlei Shi, Junjie Zhai, Xudong Yang, Zehua Xie, and Chao Liu. 2015. Radical embedding: Delving deeper to Chinese radicals. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 594–598, Beijing, China. Association for Computational Linguistics.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-enhanced chinese character embedding. In *International Conference on Neural Information Processing*, pages 279–286. Springer.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. ChineseBERT: Chinese pretraining enhanced by glyph and Pinyin information. 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 2065–2075, Online. Association for Computational Linguistics.

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, SIGHAN@IJCNLP 2015, Beijing, China, July 30-31, 2015*, pages 32–37. 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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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.

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, SIGHAN@IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 35–42. 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. *arXiv preprint arXiv:2105.12306*.

Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. 2016. Multi-granularity Chinese word embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 981–986, Austin, Texas. 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, Wuhan, China, October 20-21, 2014*, pages 126–132. 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.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. *arXiv preprint arXiv:2005.07421*.

Xiaotian Zhang, Hang Yan, Sun Yu, and Xipeng Qiu. 2022. Sdcl: Self-distillation contrastive learning for chinese spell checking. *arXiv preprint arXiv:2210.17168*.

Zhe Zhao, Hui Chen, Jinbin Zhang, Xin Zhao, Tao Liu, Wei Lu, Xi Chen, Haotang Deng, Qi Ju, and Xiaoyong Du. 2019. Uer: An open-source toolkit for pre-training models. *EMNLP-IJCNLP 2019*, page 241.

A The Statistic of SIGHAN Dataset

Training Set	#Sent	Avg. Length	#Errors
SIGHAN14	3,437	49.6	5,122
SIGHAN15	2,338	31.3	3,037
Wang271K	271,329	42.6	381,962
Total	277,104	42.6	390,121
Test Set	**#Sent**	**Avg. Length**	**#Errors**
SIGHAN14	1,062	50.0	771
SIGHAN15	1,100	30.6	703
Total	2,162	40.5	1,474

Table 5: Statistics of the SIGHAN datasets.

		Para 1			Para 2			Para 3		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
SIGHAN14	BERT	65.7	68.7	67.2	65.3	70.1	67.6	60.2	63.7	61.9
	RoBERTa	64.9	69.3	67.1	64.0	67.6	65.7	58.8	64.9	62.7
	ChineseBERT	63.5	68.2	65.7	62.1	66.6	64.3	65.5	70.3	67.8
SIGHAN15	BERT	74.1	78.4	76.2	71.8	76.9	74.3	70.1	72.6	71.3
	RoBERTa	73.9	78.0	75.9	71.9	76.0	74.9	68.0	73.8	70.7
	ChineseBERT	73.3	78.5	75.8	72.4	77.4	74.8	73.2	76.7	74.9

Table 6: All results for fine-tuning pre-trained models in raw data.

B The Experimental Results of Different Parameters

In Experiment I, we use the average of three sets of training parameters as the final result, which is due to the large fluctuation of performance on the test set during the experiment.

We use the pre-trained weight realized by (Cui et al., 2020). For all of our models, we use the AdamW optimizer (Loshchilov and Hutter, 2019) to optimize our model for 20 epochs, the learning rate is set to be 5e-5, the batch size is 48 and the warm-up ratio is set to be 0.3.

C Probe details

Our implementation uses PyTorch(Paszke et al., 2019) and HuggingFace(Wolf et al., 2020). The probes for each MLP are trained separately starting with random initialization weights. We train the probe via a binary classification task, using the Adam optimizer and Cross Entropy Loss.

C.1 PLMs considered

We selected several mainstream Chinese PLMs as our research objects, along with their model card on Huggingface:

BERT-Chinese (Cui et al., 2019) consists of two pre-training tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP), and introducing a strategy called whole word masking (wwm) for optimizing the original masking in the MLM task. We consider the base model with 110 Million parameters. Model Card:'hfl/chinese-bert-wwm-ext' under Joint Laboratory of HIT and iFLYTEK Research.

RoBERTa-Chinese (Cui et al., 2019) removes the next sentence prediction task and uses dynamic masking in the MLM task. We also consider the base model. Model Card:'hfl/chinese-roberta-wwm-ext' under Joint Laboratory of HIT and iFLYTEK Research.

ChineseBERT (Sun et al., 2021) proposes to integrate the glyph-phonetic information of Chinese characters into the Chinese pre-training model to enhance the ability to model the Chinese corpus. We consider the base model. Model Card:'junnyu/ChineseBERT-base' under Joint Laboratory of HIT and iFLYTEK Research.

MacBERT (Cui et al., 2020) suggests that $[MASK]$ token should not be used for masking, but similar words should be used for masking because $[MASK]$ has rarely appeared in the fine-tuning phase. We also consider the base model. Model Card:'hfl/chinese-macbert-base' under Joint Laboratory of HIT and iFLYTEK Research.

CPT (Shao et al., 2021) proposes a pre-trained model that takes into account both understanding and generation. Adopting a single-input multiple-output structure, allows CPT to be used flexibly in separation or combination for different downstream tasks to fully utilize the model potential. We consider the base model. Model Card:'fnlp/cpt-base' under Fudan NLP.

BART-Chinese (Lewis et al., 2019; Shao et al., 2021) proposes a pre-training model that combines bidirectional and autoregressive approaches. BART first uses arbitrary noise to corrupt the original text and then learns the model to reconstruct the original text. In this way, BART not only handles the text generation task well but also performs well on the comprehension task. We consider the base model. Model Card:'fnlp/bart-base-chinese' under Fudan NLP.

T5-Chinese (Raffel et al., 2020; Zhao et al., 2019) leverages a unified text-to-text format that treats various NLP tasks as Text-to-Text tasks, i.e., tasks with Text as input and Text as output, which attains state-of-the-art results on a wide variety of NLP tasks. We consider the base model. Model Card:'uer/t5-base-chinese-cluecorpussmall' under UER.

C.2 The Statistics of Probe Dataset

We remove some rare characters for two reasons. Firstly, these characters are rarely encountered as misspellings in CSC task. Secondly, these characters appeared infrequently in the training corpus of the PLMs, which we believe would make it excessively challenging for the PLMs to learn effectively. The statistics are shown in Table 7 and Table 8.

C.3 Probing Results from Models with Different Numbers of MLP Layers

From the experimental results, it can be seen that the number of layers of MLP has little effect on the results, and most of the results of the pre-training models are finally concentrated in the interval of 0.75-0.76. The Chinese pre-training models of the BERT family are slightly less effective when the number of layers is relatively small and similar to other Chinese pre-training models after more than three layers.

	#Pos.	#Neg.	#Total
Training Set	7968	7968	15936
Test Set	1992	1992	3984

Table 7: The statistics of the dataset for the glyph probe.

	#Pos.	#Neg.	#Total
Training Set	8345	8345	16690
Test Set	2087	2087	4174

Table 8: The statistics of the dataset for the phonetic probe.

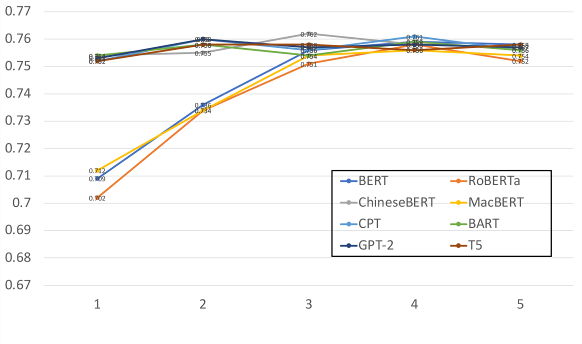


Figure 6: Results for each model in the case of 1-5 layers of MLP.

ACL 2023 Responsible NLP Checklist

A For every submission:

□ A1. Did you describe the limitations of your work?
Left blank.

□ A2. Did you discuss any potential risks of your work?
Left blank.

□ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

□ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B □ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C □ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D □ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.