Towards Real-World Writing Assistance: A Chinese Character Checking Benchmark with Faked and Misspelled Characters

Anonymous ACL submission

Abstract

Writing assistance aims to improve the correctness and quality of input texts, with character checking being crucial in detecting and correcting wrong characters. In the real world where handwriting occupies the vast majority, characters that humans get wrong include faked characters (i.e., untrue characters created due to writing errors) and misspelled characters (i.e., true characters used incorrectly due to spelling errors). However, existing datasets and related studies only focus on misspelled characters that can be represented by computer text encoding systems, thereby ignoring faked characters which are more common and difficult. To break through this dilemma, we present **Visual-C^3**, a human-annotated **Visual** Chinese **C**haracter **C**hecking dataset with faked and misspelled Chinese characters. To the best of our knowledge, Visual-C^3 is the first real-world visual and the largest human-crafted dataset for the Chinese character checking scenario. Additionally, we also propose and evaluate novel baseline methods on Visual-C^3. Extensive empirical results and analyses show that Visual-C^3 is high-quality yet challenging. As the first study focusing on Chinese faked characters, the Visual-C^3 dataset and the baseline methods will be publicly available to facilitate further research in the community.

Original: 人生就象(elephant)一场X戏
Correct : 人生就像(like)一场游戏
Trans. : Life is like a game

Figure 1: Examples of Chinese faked (错字) and misspelled (别字) characters.

1 Introduction

With texts on the Internet growing explosively every day, writing assistance that is to improve the correctness and quality of texts is becoming increasingly important (Strobl et al., 2019; Jourdan et al., 2023), and has received more and more attention from researchers. In the field of writing assistance, the character checking task aims to detect and correct wrong characters in the given text and occupies a crucial position, as it ensures the correctness of the minimum atom (i.e., the characters) of texts (Du et al., 2022). Large amounts of research are devoted to Chinese Character Checking, which is also well known as Chinese Spell Checking or Chinese Spelling Correction (CSC) (Wu et al., 2013a; Yu and Li, 2014). In this work, we also focus on the scene of Chinese Character Checking.

Since Chinese Character Checking is a daily application closely related to human life, to promote its progress and development, we must consider the real-world application needs of humans for it. Therefore, a natural question arises: **What are the types of erroneous Chinese characters that humans would produce during the writing process?** Based on the observation of human writing habits, it is well known that there exist two main types of Chinese characters that humans get wrong in the real world, namely faked characters (错字) and misspelled characters (别字) (Chen and Bai, 1998). As illustrated in Figure 1, the misspelled character itself is a character that exists but is used incorrectly, the faked character is a non-existent character caused by incorrect writing (e.g., wrong use of radicals or wrong number of strokes). Authoritative Chinese linguistics studies (Wang and Wu, 2023) have shown that faked characters appear more frequently than misspelled characters in the process of people's daily use of Chinese characters, and faked characters are often more difficult to detect than misspelled ones because faked characters are often caused by some very slight stroke errors.

Although faked characters are more common and challenging in the real world, researchers have not paid enough attention to how to handle the faked characters. The main reason for this dilemma is that the existing CSC data resources are all text-based. The main drawback of single text-modal data is its inability to represent characters beyond those encoded by computers. Fake characters are non-existent in computer text encoding systems. Hence, the traditional CSC datasets cannot represent faked characters, and the existing CSC models proposed cannot hold onto more complex and real scenarios. At this point, a pressing and significant problem is **how to expand and develop data resources for Chinese Character Checking to facilitate the automatic detection and correction of faked characters by models.**

Inspired by the enthusiasm to handle the faked characters, we propose to extend Chinese Character Checking to the visual modality, as images are the most direct form to represent the faked characters. We construct a large-scale human-annotated **V**isual **C**hinese **C**haracter **C**hecking dataset, **Visual-C**^{3}, which consists of 10,072 sentences represented by images and 12,019 wrong characters (including 5,670 misspelled and 6,349 faked characters) manually annotated by well-trained annotators. To the best of our knowledge, Visual-C^3 is the first real scene-oriented dataset that contains both faked and misspelled characters. Furthermore, to give future research on Visual-C^3 more possibilities, in addition to annotating sentence-level information (i.e., the golden sentence without error characters corresponding to the original content of the input image), we also annotate each image at the character level and provide the position and type information of each character on the image. Rich annotation information makes Visual-C^3 suitable for various NLP, CV, or multimodal studies.

Based on Visual-C^3, we design the benchmark tasks in which the model inputs an image containing sentences with wrong characters and outputs the correct sentence without wrong characters corresponding to the input image in the form of text. Through this task, Visual-C^3 effectively assesses the detection and correction ability of Chinese Character Checking methods, especially for faked characters. To verify the quality and challenge of Visual-C^3, we design and implement two baseline methods with different paradigms and evaluate them on Visual-C^3. Extensive experiments and detailed analyses demonstrate that Visual-C^3 is high-quality yet challenging. At the same time, the baselines also provide insightful and promising future directions. **Hopefully, we believe that the emergence of Visual-C**^{3} **could promote the research of writing assistance to better adapt to the intelligence needed in the real world.**

2 Related Works

2.1 Chinese Spell Checking

In recent years, several public CSC datasets have been proposed, which can be divided into two categories based on data content distribution: open-domain and specific-domain.

For open-domain, the most widely used are the SIGHAN datasets, which include SIGHAN13 (Wu et al., 2013b), SIGHAN14 (Yu et al., 2014), and SIGHAN15 (Tseng et al., 2015). In particular, SIGHAN datasets come from mistakes in essays written by teenage students (SIGHAN13) or Chinese as foreign language learners (SIGHAN14 and SIGHAN15). As for the specific-domain CSC datasets, MCSCSet (Jiang et al., 2022) is a large-scale specialist-annotated dataset containing about 200K samples from a real-world medical application named Tencent Yidian. ECSpell (Lv et al., 2023) is a CSC dataset with three domains, law, medical, and official document. LEMON (Wu et al., 2023) is a large-scale multi-domain dataset with natural spelling errors.

However, the existing CSC datasets have one major limitation that cannot be ignored, that is, **the modality of these datasets is limited to the single text modality**. The immediate dilemmas posed by this limitation are twofold. First, all existing CSC datasets do not cover text in images, while spelling errors in the real world do not only exist in text but also more widely in images. The second dilemma is the inability to handle the faked characters, whereas humans are more likely to make in daily life. The existing CSC datasets are all constructed in text form, so they cannot contain the faked characters at all. Therefore, to overcome the limitations described above, we construct Visual-C^3, the first real-world visual and the largest human-crafted dataset for the Chinese Character Checking scenario.

2.2 OCR Error Correction

OCR error correction is somewhat related to Visual Chinese Character Checking. Therefore, it is necessary to introduce the related data resources of OCR error correction. HANDS-VNOnDB3 (Nguyen

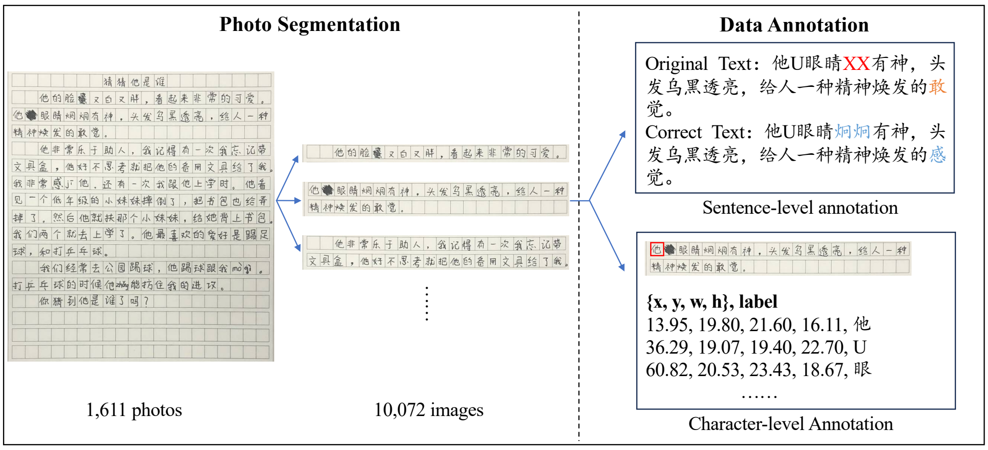


Figure 2: The annotation schema. "U" represents the unknown character and "X" represents the faked character.

et al., 2018) has been presented to promote the studies on Vietnamese handwritten text recognition. It has handwritten images that contain 1,146 Vietnamese paragraphs of handwritten text comprising 7,296 lines. Tanaka et al. (2022) constructed a dataset based on the historical newspaper database Trove (Cassidy, 2016; Sherratt, 2021) and public meeting articles in Australian historical newspapers (Fujikawa, 1990), which contains 719 public meeting articles including 13,543 lines.

To the best of our knowledge, the existing OCR error correction datasets noticeably lack Chinese resources. More importantly, the OCR task is different from what we focus on. When there are wrong characters in the image, OCR models try to directly predict the original correct characters, but we hope that Chinese character checking models can point out which characters in the image are wrong and further correct the wrong characters.

3 The Visual-C³ Dataset

3.1 Dataset Construction

Data Collection We cooperate with a Chinese language teaching and research group in a middle school and take anonymized photos of their students' handwritten essays as the raw data [1]. There are two main reasons why we chose the photos of middle school students' handwritten essays as the raw data: (1) Photos of handwritten text are most consistent with real scenes and they can display faked and misspelled characters at the same time, while data in text format cannot represent faked characters. (2) The average Chinese character writing mastery level of middle school students determines that they will neither make simple mistakes that are too low-level nor make no mistakes at all, which ensures the challenge and usability of our data set. The entire data collection process lasted for 3 months, and we finally collected the photos of 5,692 handwritten essays from 389 students as our raw data.

Data Preprocessing In order to ensure the quality of our dataset, we carefully check and filter the 5,692 original photos collected one by one. In particular, after observing the raw data, we identified three main categories of situations that we think may affect the dataset quality: (1) Students excessively daube and modify some characters during their writing process, seriously affecting the clarity of the photos and their characters. (2) Some photos contain the teacher's red markings for faked and misspelled characters, which we believe will cause information leakage in the data sample. (3) Some photos are affected by many factors such as location and light during the shooting process, which affects the clarity of the photos and the completeness of the content of the essay. After our careful data cleaning, we finally retained 1,611 high-quality photos for the next step of annotation.

Annotation Schema To obtain sentence-level data, we segmented the 1,611 original photos into 10,072 images containing only one semantically complete sentence, as illustrated in Figure 2. For the sentence level, we annotate both the original text and the correct text. Note that the original text contains faked and misspelled characters. Particularly, for the faked characters, we marked them using the symbol "X". And for some characters

[1]We have signed a legal intellectual property agreement with the school and paid a data purchase fee of $5 per essay.

that are difficult to recognize, we will directly mark them as “U”. At the character level, we annotate the position information for each character on the image. Specifically, we annotate the coordinate values (x, y) representing the top-left corner of each character, along with the length and width dimensions(w,h), as depicted in Figure 2.

Annotation Workflow Our workflow is divided into two parts:

(1) For the sentence-level annotation, we arranged 30 annotators and 10 senior annotation experts who are native Chinese speakers and are instructed in the guidelines of annotation in detail. Specifically, each segmented image is independently annotated by three annotators and double-checked by one senior expert. The annotator is responsible for transcribing the content in the image into the original sentence containing faked characters represented by the symbol “X” and misspelled characters, and is responsible for modifying the original sentence into a correct sentence. Then, one annotator expert carefully checks the original/correct sentences for possible wrong or omissive annotations and makes the final decision in case three annotators have inconsistent correction results.

(2) For the character-level annotation, we employed 10 annotators and 2 senior experts professionally serving image segmentation. Therefore, each image is annotated by an annotator using the tool to achieve the specific coordinate position information of each character on it, and then a senior expert checks the accuracy of the annotated coordinate information.

To ensure the annotation quality, we paid annotators according to their workload (the number of images per hour). In addition, we divided the entire raw data into 10 batches. In the annotation workflow, we will randomly select 20% of the data submitted by senior annotation experts for sampling check. If the check accuracy is lower than 98%, this batch will be returned for re-annotation. Overall, the entire annotation process lasted about 4 months.

3.2 Dataset Analysis

Dataset Statistics Visual-C^3 consists of 10,072 sentences represented by images and 12,019 wrong characters. We randomly divided the training set, validation set, and test set according to the ratio of 3:1:1. We counted three attributes, namely average length, number of misspellings, and number of faked characters, respectively. As compared with previous CSC datasets in Table 1, our Visual-C^3 is not only the first dataset containing faked characters, but its data size is also very competitive.

Dataset	#Sent	Avg.Length	#Misspelled	#Faked
SIGHAN2013	1,700	60.9	1,567	-
SIGHAN2014	4,499	49.7	5,893	-
SIGHAN2015	3,439	31.1	3,740	-
Visual-C^3	10,072	40.4	5,670	6,349

Table 1: Statistics of CSC datasets. Column Sentence represents the number of samples in this dataset.

Dataset Quality Considering that the batch annotation method we designed has guaranteed annotation accuracy to a certain extent, we further measure the agreements between multiple annotators. In particular, we calculate the Fleiss’ kappa (Moons and Vandervieren, 2023) to verify the annotator agreement of labeling the original/correct sentences of images, the result is 85.20%, which indicates that our annotation can be regarded as “almost perfect agreement” (Landis and Koch, 1977).

3.3 Benchmark Settings

Task Formulation Visual-C^3 focuses on Visual Chinese Character Checking. To fully exploit Visual-C^3 so that it more comprehensively evaluates the model’s Chinese Character Checking capabilities, especially the processing capabilities of the faked characters, we divide Visual Chinese Character Checking into two subtasks based on Visual-C^3.

(1) **Detection Subtask**: The inputs are images from Visual-C^3, and the ideal outputs are corresponding text marked with faked and misspelled character positions. The core of the detection subtask is to accurately identify which characters in the image are faked and which characters are misspelled. It does not require the model to know the correct characters corresponding to the faked or misspelled characters.

(2) **Correction Subtask**: After the detection subtask has located which characters in the image, the correction subtask further requires the model to output a text with no wrong characters at all, that is, to correct the detected faked and misspelled characters.

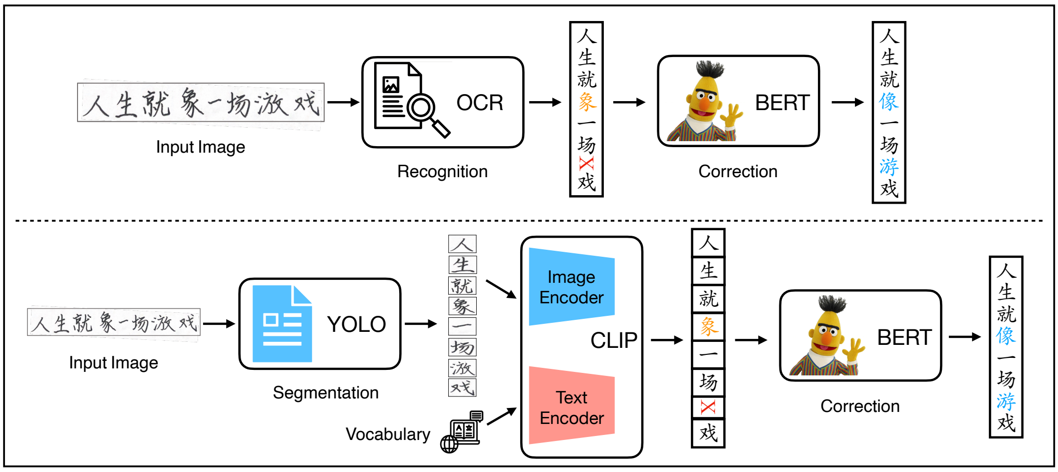


Figure 3: Illustration of our designed baselines, namely OCR-based method (top) and CLIP-based method (bottom).

Evaluation Metrics For evaluation granularity, there are two levels, i.e., character and sentence levels. The sentence-level metric requires that all the wrong characters in a sentence are successfully detected and corrected. So the sentence-level metric is more difficult than the character-level metric, because a sentence may have multiple wrong characters. We calculate the Precision, Recall, and F1 score for the detection and correction subtasks. Besides, to evaluate the model's capabilities for different characters, we also calculate metrics separately for faked and misspelled characters.

4 Models and Baselines

To reflect the usability of Visual-C^3 and provide reference ideas for future research on Visual-C^3, we design two baselines, namely OCR-based and CLIP-based methods, as illustrated in Figure 3.

4.1 OCR-based Method

The OCR-based method consists of two modules, namely the recognition module and the correction module. The recognition module is responsible for obtaining text content by identifying the characters in the input images, while the correction module corrects based on the output of the recognition module and outputs text without incorrect characters.

Recognition Module To recognize the Chinese characters on the images, we utilize an OCR model which has the ability to convert images into texts. Specifically, the input of this module is the image I with n characters and the output is the text $X = (x_1, x_2, \ldots, x_n)$ with faked and unknown characters. Consistent with the dataset annotation, the faked character is marked as "X" and the unknown character is marked as "U".

In particular, for traditional OCR methods, a great challenge with Visual-C^3 is how to recognize the faked characters. Therefore, we propose two strategies to solve this dilemma. First, we heuristically treat any characters whose recognition module output confidence score is below a reasonable threshold thr as faked characters. Additionally, we also employ our training datasets with customized vocabulary to fine-tune the OCR model. After fine-tuning, the OCR model in the recognition module will have the ability to recognize faked characters without any artificially set heuristic thresholds.

Correction Module The correction module is a sequential multi-class labeling model based on transformers such as BERT (Devlin et al., 2019). The input is the sentence $X = (x_1, x_2, \ldots, x_n)$ and the output is a character sequence $Y = (y_1, y_2, \ldots, y_n)$. For a character of the sequence, its correction probability is defined as:

$$P(y_i = j|X) = \text{softmax}(Wh_i + b)[j] \quad (1)$$

where $P_c(y_i = j|X)$ is the conditional probability that character x_i is corrected as the character j in the vocabulary, h_i denotes the hidden state, W and b are learnable parameters. It is worth noting that the vocabulary of the correction module is extended with the special tokens "U" and "X" to facilitate it to receive the output of the recognition module.

4.2 CLIP-based Method

The CLIP-based method is divided into three modules, which are the segmentation module, retrieval module, and correction module.

Methods	Detection (Character-Level)			Correction (Character-Level)			Detection (Sentence-Level)			Correction (Sentence-Level)		
	Prec.	**Rec.**	**F1.**	**Prec.**	**Rec.**	**F1.**	**Prec.**	**Rec.**	**F1.**	**Prec.**	**Rec.**	**F1.**
OCR-Based Method	3.6	42.2	6.6	2.0	23.5	3.7	0.8	2.8	1.3	0.3	0.9	0.4
+ Fine-tuned Recognition	16.0	56.3	25.0	14.1	**49.3**	21.9	11.6	23.4	15.5	9.4	19.0	12.6
+ Fine-tuned Recognition/Correction	**16.2**	55.8	**25.1**	**14.2**	**49.3**	**22.0**	**12.4**	**24.7**	**16.6**	**10.0**	**20.0**	**13.4**
CLIP-Based Method	9.8	55.7	16.8	8.5	48.3	14.5	5.4	13.5	7.7	4.3	10.6	6.0
+ Fine-tuned Correction	10.1	**56.9**	17.2	8.7	48.9	14.8	5.5	13.5	7.8	4.7	11.5	6.7

Table 2: Performance of different methods on the misspelled characters of Visual-C^3 test set.

Methods	Detection (Character-Level)			Correction (Character-Level)			Detection (Sentence-Level)			Correction (Sentence-Level)		
	Prec.	**Rec.**	**F1.**	**Prec.**	**Rec.**	**F1.**	**Prec.**	**Rec.**	**F1.**	**Prec.**	**Rec.**	**F1.**
OCR-Based Method	3.6	**36.0**	6.5	0.3	2.9	0.5	0.3	0.8	0.4	0.0	0.0	0.0
+ Fine-tuned Recognition	13.1	20.9	**16.1**	5.9	9.3	7.2	**8.6**	**9.1**	**8.8**	4.7	5.0	4.9
+ Fine-tuned Recognition/Correction	13.1	20.9	**16.1**	7.1	**11.4**	**8.8**	**8.6**	**9.1**	**8.8**	**6.1**	**6.4**	**6.2**
CLIP-Based Method	**14.3**	15.5	14.9	6.7	7.2	6.9	7.6	8.5	8.0	4.4	4.9	4.6
+ Fine-tuned Correction	**14.3**	15.5	14.9	**9.2**	9.9	**9.5**	7.6	8.5	8.0	5.8	**6.4**	6.1

Table 3: Performance of different methods on the faked characters of Visual-C^3 test set.

Segmentation Module In the segmentation module, our objective is to identify and arrange each character present in the image, following a traditional left-to-right and top-down ordering scheme.

Specifically, we employ an object detection approach capable of identifying all characters within the image. This method enables us to extract the individual characters present in the image. Specifically, given the image I, we can obtain the coordinates of the upper left corner $(\boldsymbol{L}_X, \boldsymbol{L}_Y)$, as well as the width $\boldsymbol{W}$ and height $\boldsymbol{H}$ of each of the n character-level sub-images segmented.

While the object detection model effectively identifies the characters, arranging them in the correct order poses a challenge. Consequently, we design a regularization sorting algorithm to establish the ordered character sequence. Due to page limits, the details of this algorithm are presented in Appendix A. Finally, the segmentation module will get a sequence of character-level images sorted according to the order of characters in the sentence.

Retrieval Module After obtaining the images of each character sequentially, we carry out the image-text retrieval task based on CLIP (Radford et al., 2021). CLIP usually has a text encoder and an image encoder to obtain representations of texts and images, and then we can retrieve texts based on images according to the similarity between their representations. Particularly, we train the CLIP model from scratch on Visual-C^3, giving it the ability to retrieve Chinese characters based on images, especially the ability to identify faked characters.

For training of CLIP, we instruct the text encoder to align itself with the image embedding by maximizing the cosine similarity between paired image/text embeddings, while simultaneously minimizing the cosine similarity of unpaired image/text within the batch. We optimize the CLIP model with the similarity score utilizing the contrastive loss:

$$L = -\frac{1}{n}\sum_{j=1}^{n} \log \frac{exp(sim(z_j^t, z_i^t)/\tau)}{\sum_{k=1}^{n} exp(sim(z_j^t, z_i^k)/\tau)} \quad (2)$$

where $z^t = [z_1^t, z_2^t, \ldots, z_n^t]$ represents the latent representations of texts, while $z^i = [z_1^i, z_2^i, \ldots, z_n^i]$ represents those of images within a mini-batch comprising n samples.

Through the retrieval module, we obtain the text including misspelled and fake characters.

Correction Module The function and implementation of this part module are the same as the correction module of the OCR-based method.

5 Experiments and Analyses

5.1 Main Results

The implementation details and hyper-parameter selection are shown in Appendix B. From Table 2 an Table 3, we have the following observations:

1. When not fine-tuned, the pre-trained OCR model performs poorly on Visual-C^3, which indicates that existing OCR methods cannot

work well on our dataset and reflects the challenge of our dataset. In particular, for the faked characters, the performance of our proposed baselines is still unsatisfactory even after fine-tuning. Therefore, studying how to handle faked characters is very urgent.

2. For the misspelled characters, we find that the models' recall is much higher than its precision. This is because the model corrects a large number of characters, thereby incorrectly modifying many correct characters. Therefore, the poor performance of the BERT-based correction module on the misspelled characters indicates that the text content of Visual-C^3 is very difficult.

3. We are surprised to find that the CLIP-based method' performance is not very poor, which shows that our idea of identifying the faked characters through retrieval is feasible.

5.2 Performance Analysis

Methods	Misspelled	Faked	Correct	Average
Fine-tuned OCR	0.694	**0.209**	**0.944**	**0.929**
Fine-tuned CLIP	**0.732**	0.155	0.929	0.914

Table 4: The numbers of misspelled, faked, and correct characters in the test set are 788, 1,223, and 79,141.

The OCR and CLIP Performance: Table 4 reports the performance of our fine-tuned OCR model and CLIP model, i.e., their character recognition (or retrieval) accuracy. After fine-tuning on Visual-C^3, both the OCR model and the CLIP model have a certain ability to distinguish the faked characters. Of course, we have to admit that compared with the misspelled and correct characters, our fine-tuned models' processing ability for the faked characters is still much inferiorc. We encourage subsequent researchers to make greater innovations in the model structure to obtain better performance of the faked characters on the Visual-C^3 dataset.

Character Type	Correction (Character-Level)			Correction (Sentence-Level)		
	Prec.	Rec.	F1.	Prec.	Rec.	F1.
Misspelled	**72.7**	47.3	57.3	52.7	40.4	45.8
Faked	63.8	**63.8**	**63.8**	**58.4**	**58.4**	**58.4**

Table 5: The correction performance upper bounds.

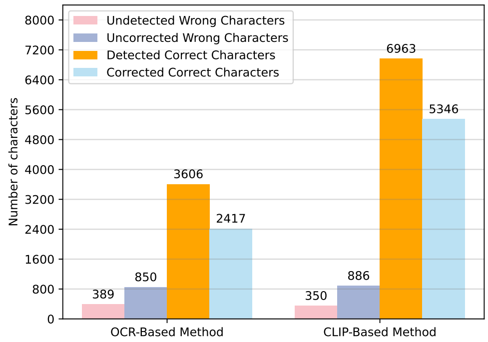


Figure 4: The numbers of wrong and correct characters in the test set are 2,011 and 79,141.

Correction Upper Bound: To further measure the difficulty of the text content of our dataset for existing error correction methods, we study the performance upper bound of our correction module. Specifically, we input the annotated original text into the correction module. Note that we only report the performance of the fine-tuned correction module. From Table 5, we know that BERT's performance on Visual-C^3 is not very high, and BERT achieves at least a score of 63.4 or more on sentence-level correction F1 on widely used SIGHAN13/14/15 (Wu et al., 2013b; Yu et al., 2014; Tseng et al., 2015). This performance gap indicates that the text content of our dataset is more complex than the previous CSC datasets and we think this difficulty stems from the fact that our dataset is collected from completely real scenes.

5.3 Error Analysis

As shown in Figure 4, we count the cases where different methods mishandle wrong characters (including misspelled and faked characters) and correct characters. We notice that whether it is the OCR-based or CLIP-baed method, they tend to detect or correct a large number of correct characters wrongly (it should be emphasized that the proportion of correct characters that are mishandled is not high). Based on our observations, we think that this kind of error mainly comes from the recognition module of the OCR-based method or the segmentation and retrieval modules of the CLIP-based method. Especially for the CLIP-based method, although we innovatively propose the method of image and text retrieval to identify the faked characters in images, the pipeline paradigm of first segmenting the sentence-level image into small

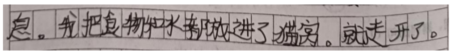
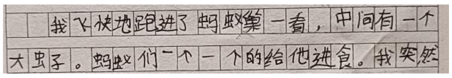
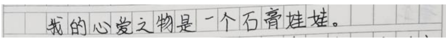
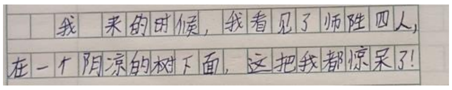
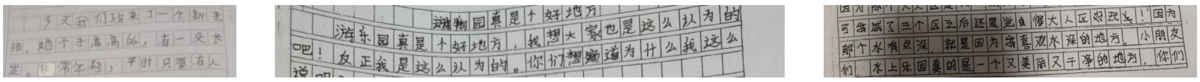
Image & Original Text	Output	Result
 息。我把食物和水都放进了猫X。就走开了。	**OCR:** 息。我把食物和水都放进了猫X。就走开了。	✓
	CLIP: 息。我把食物和水都放进了猫X。就走开了。	✓
 我飞快地跑进了蚂蚁巢一看，中间有一个大虫子。蚂蚁们一个一个的给他进食。我X然	**OCR:** 我飞快地跑进了蚂蚁巢一看，中间有一个大虫子。蚂蚁们一个一个的给他进食。我突然	✗
	CLIP: 我飞快地跑进了蚂蚁巢一看，中间有一个大虫子。蚂蚁们一个一个的给他进食。我X然	✓
 我的心X之物是一种石X娃娃。	**OCR:** 我的心X之物是一种石X娃娃	✓
	CLIP: 我的心X之物是一种石高娃娃	✗
 我来的时候，我看见了师X四人，在一个阴凉的树下面，这把我都惊呆了！	**OCR:** 我来的时候，我看见了师陆四人，在一个阳凉的树下面，这把我都惊呆了！	✗
	CLIP: 我来的时候，我看见了师性四人，在一个X凉的树下面，这把我都惊呆了：	✗

Figure 5: Some cases from our designed baselines. Other cases from multimodal LLMs are presented in Appendix C.



(a) The ink is not clear. (b) The photo angel is inappropriate. (c) The light and shadow is uneven.

Figure 6: Representatives of hard samples from the Visual-C^3 dataset.

character-level images and then retrieving will result in a certain degree of error accumulation. For the OCR-based method, the accuracy of the recognition module also determines the performance ceiling of the entire method to a certain extent.

5.4 Case Study

Model Cases: From Figure 5, we know that after fine-tuning on Visual-C^3, both the OCR model and our proposed CLIP-based model can recognize the faked characters in images. We also run advanced multimodal LLMs on Visual-C^3, as shown in Appendix C. For future studies on Visual-C^3, we think there are two ideas that can improve model performance. First, how can we make the model better handle complex characters with many strokes, such as “突” in the second case and “膏” in the third case? Second, it is crucial to improve the model to distinguish between the faked characters and misspelled characters with similar strokes. For example, in the fourth case, the model should detect the character in the image as a faked character, but it instead gives a “陆” with similar strokes as the output, which would lead to a decrease in the model’s faked character detection performance.

Dataset Challenges: During constructing the Visual-C^3 dataset, some hard samples are observed by our annotators, as shown in Figure 6. For the part of hard samples, we do not exclude them from our dataset because we think that the situations represented by these samples are exactly what the model would encounter when deployed in real scenarios. Therefore, compared with previous related datasets, the fact that the data comes entirely from the real world is a major advantage of Visual-C^3.

6 Conclusion

In this paper, we pay attention to the faked characters, which have never been focused on in previous works. To empower machines to automatically process the faked characters, we construct Visual-C^3, a large-scale visual Chinese Character Checking dataset with faked and misspelled characters. Furthermore, we design two baseline methods with different ideas. In particular, we first propose the idea of using image-text retrieval to detect the faked characters in the images. Experimental results and detailed analyses indicate that our proposed baselines are effective and Visual-C^3 is challenging and of great research value.

Limitations

We conduct experiments on Visual-C^3 employing two proposed baselines. Due to hardware resource limitations, we only use the base-level pre-trained weights to initialize each module in our baseline methods. In addition, because the collection and annotation of the dataset cost a lot of money, we do not have enough financial budget to fully test the performance of multimodal LLMs such as GPT-4v on our dataset. Of course, the main contribution of our work is to provide new research directions and data resources. Our designed baselines are also mainly to verify the usability of the dataset itself and to provide model design ideas for subsequent researchers to refer to. Therefore, We believe that using larger scale models to obtain better performance can be left as future work.

Ethics Statement

In this paper, we present the human-annotated Visual-C^3, which focuses on real-world writing assistance scenes. We have described the details of the collection, preprocessing, and annotation of our dataset in the main text of our paper. It is worth noting that all data in our dataset has obtained authorization from its providers and is desensitized before annotation to ensure that the privacy of the data providers would not be leaked. Besides, the Chinese Character Checking task itself comes from very common and important application requirements in daily life and is designed to be convenient for human daily life. Therefore, neither the task on which our work focuses nor the dataset presented poses potential harm to human society.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Steve Cassidy. 2016. Publishing the trove newspaper corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4520–4525.

Keh-Jiann Chen and Ming-Hong Bai. 1998. Unknown word detection for chinese by a corpus-based learning method. *Int. J. Comput. Linguistics Chin. Lang. Process.*, 3(1).

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.

Wanyu Du, Zae Myung Kim, Vipul Raheja, Dhruv Kumar, and Dongyeop Kang. 2022. Read, revise, repeat: A system demonstration for human-in-the-loop iterative text revision. *CoRR*, abs/2204.03685.

Takao Fujikawa. 1990. Public meetings in new south wales, 1871/1901. *Journal of the Royal Australian Historical Society*, 76(1):45–61.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Wangjie Jiang, Zhihao Ye, Zijing Ou, Ruihui Zhao, Jianguang Zheng, Yi Liu, Bang Liu, Siheng Li, Yujiu Yang, and Yefeng Zheng. 2022. Mcscset: A specialist-annotated dataset for medical-domain chinese spelling correction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4084–4088.

Léane Jourdan, Florian Boudin, Richard Dufour, and Nicolas Hernandez. 2023. Text revision in scientific writing assistance: An overview. *CoRR*, abs/2303.16726.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. *biometrics*, pages 159–174.

Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. 2022. Pp-ocrv3: More attempts for the improvement of ultra lightweight OCR system. *CoRR*, abs/2206.03001.

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

Qi Lv, Ziqiang Cao, Lei Geng, Chunhui Ai, Xu Yan, and Guohong Fu. 2023. General and domain-adaptive chinese spelling check with error-consistent pretraining. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(5):1–18.

Filip Moons and Ellen Vandervieren. 2023. Measuring agreement among several raters classifying subjects into one-or-more (hierarchical) nominal categories. a generalisation of fleiss' kappa.

Hung Tuan Nguyen, Cuong Tuan Nguyen, Pham The Bao, and Masaki Nakagawa. 2018. A database of unconstrained vietnamese online handwriting and recognition experiments by recurrent neural networks. *Pattern Recognition*, 78:291–306.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Tim Sherratt. 2021. Glam workbench–using the trove newspaper & gazette harvester (the web app version).

Carola Strobl, Emilie Ailhaud, Kalliopi Benetos, Ann Devitt, Otto Kruse, Antje Proske, and Christian Rapp. 2019. Digital support for academic writing: A review of technologies and pedagogies. *Comput. Educ.*, 131:33–48.

Koji Tanaka, Chenhui Chu, Tomoyuki Kajiwara, Yuta Nakashima, Noriko Takemura, Hajime Nagahara, and Takao Fujikawa. 2022. Corpus construction for historical newspapers: A case study on public meeting corpus construction using ocr error correction. *SN Computer Science*, 3(6):489.

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.

Wei Wang and Jianming Wu. 2023. Chinese linguistics: An introduction. *Open Journal of Modern Linguistics*, 13:515–522.

Hongqiu Wu, Shaohua Zhang, Yuchen Zhang, and Hai Zhao. 2023. Rethinking masked language modeling for chinese spelling correction. *arXiv preprint arXiv:2305.17721*.

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.

Junjie Yu and Zhenghua Li. 2014. Chinese spelling error detection and correction based on language model, pronunciation, and shape. In *Proceedings of The Third CIPS-SIGHAN Joint Conference on Chinese Language Processing, Wuhan, China, October 20-21, 2014*, pages 220–223. 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.

A Regularization Sorting Algorithm

The pseudo-code is shown in Algorithm 1.

Algorithm 1 Regularization Sorting

Input: $\boldsymbol{L}_X$, $\boldsymbol{L}_Y$, $\boldsymbol{W}$, $\boldsymbol{H}$
Output: Sorted $\tilde{\boldsymbol{L}_X}$, $\tilde{\boldsymbol{L}_Y}$, $\tilde{\boldsymbol{W}}$, $\tilde{\boldsymbol{H}}$

$\tilde{\boldsymbol{L}_X} \leftarrow \emptyset$, $\tilde{\boldsymbol{L}_Y} \leftarrow \emptyset$, $\tilde{\boldsymbol{W}} \leftarrow \emptyset$, $\tilde{\boldsymbol{H}} \leftarrow \emptyset$
repeat
    Calculate the average value $\bar{\boldsymbol{M}}$ within the range of α for the minimum values of $\boldsymbol{L}_Y$
    Treat the index $\boldsymbol{i}$ of characters that are within a distance of β from the mean $\bar{\boldsymbol{M}}$
    Sort $\boldsymbol{i}$ according to horizontal coordinate from small to large, it is put into $\tilde{X}$
    Take sorted coordinates according to $\boldsymbol{i}$ into $\tilde{\boldsymbol{L}_X}$, $\tilde{\boldsymbol{L}_Y}$, $\tilde{\boldsymbol{W}}$, $\tilde{\boldsymbol{H}}$
    Remove the coordinates already taken from $\boldsymbol{L}_X$, $\boldsymbol{L}_Y$, $\boldsymbol{W}$, $\boldsymbol{H}$
until $|\boldsymbol{L}_X| \leq 0$
return $\tilde{\boldsymbol{L}_X}$, $\tilde{\boldsymbol{L}_Y}$, $\tilde{\boldsymbol{W}}$, $\tilde{\boldsymbol{H}}$

B Implementation Details

All the models presented in this paper are implemented using Python (Version 3.7.15) and the PyTorch framework (Version 1.12.1). For the OCR-based method, we select the PaddleOCRv3 of handwriting (Li et al., 2022) to be the recognition module. If the recognition module is not fine-tuned, the faked characters will be classified by the thr of 0.2. We utilize the advanced and widely used YOLOv8 model [2] to segment sentence-level images into character-level images. For the implementation of our CLIP model in the retrieval module, we initialize the image encoder and text encoder with the ResNet-50 (He et al., 2016) and RoBERTa-base (Liu et al., 2019). As for the correction module, we utilize the $BERT_{BASE}$ (Devlin et al., 2019) which has 12 transformer layers with 12 attention heads.

Regarding the fine-tuning details, the recognition module of the OCR-based baseline is trained over 500 epochs, with a learning rate of 4e-5 and a batch size of 50. For the CLIP-based baseline, the detection module is trained for 2,000 epochs, employing a learning rate of 5e-5 and a batch size of 256. Additionally, the correction module is fine-tuned for 10 epochs, using a learning rate of 5e-5 and a batch size of 4.

[2] https://github.com/ultralytics/ultralytics

C Running Cases of Multimodal LLMs

To further reflect the challenge of Visual-C^3, we also select advanced and popular multimodal LLMs to run on Visual-C^3 to observe the performance of multimodal LLMs. Specifically, we choose the most widely studied GPT-4V (Achiam et al., 2023) and the Qwen-VL-Max [3] (Bai et al., 2023) model newly released in the Chinese community for experiments. Limited by the price of GPT-4V services and the access method of Qwen-VL-Max, we only test and observe them using a small number of samples in Visual-C^3. Our input text prompt for these two models is "首先我们分别定义错字：因为一些偏旁部首搭配错误产生的一些字典中不存在的字；别字：因为音近或形似产生的一些字典中存在的字。现在请你识别我给你的手写图片，请告诉我图中哪些字是错字，哪些字是别字？(First, we define the faked characters: some characters that do not exist in the dictionary due to incorrect strokes of some radicals; misspelled characters: some characters that exist in the dictionary due to similar sounds or similar shapes. Now please identify the handwritten image I give you. Please tell me which characters in the image are faked and which characters are misspelled characters?)"

From Figure 7, we can see that multimodal LLMs such as GPT-4V and Qwen-VL-Max encounter great difficulties on Visual-C^3. For GPT-4V, we find that it has almost no ability to correctly recognize Chinese characters from images, and instead gives responses that have nothing to do with the content of the images. Nevertheless, GPT-4V is not without its merits. We see that it still understands our definitions of the faked and misspelled characters in the text prompt, and tries to detect the faked characters at a more fine-grained level of strokes and radicals. We guess that the reason for the poor performance of GPT-4V is mainly due to its lack of OCR capabilities for Chinese characters. On the other hand, for Qwen-VL-Max, we know that it does have excellent Chinese OCR capabilities, but unfortunately, it believes that there are no faked characters in all images because of its lack of processing capabilities for the faked characters. In addition, even if Qwen-VL-Max recognizes most of the content in the image, it cannot correctly detect the misspelled characters in the image due to its lack of Chinese semantic understanding.

In summary, even in the era of LLMs, our pro-

[3] The Qwen-VL-Max model was just introduced and available on Jan 18, 2024 at https://github.com/QwenLM/Qwen-VL.

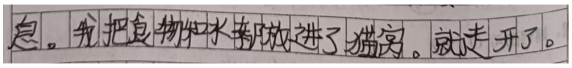
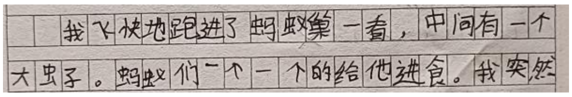
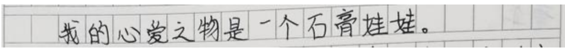
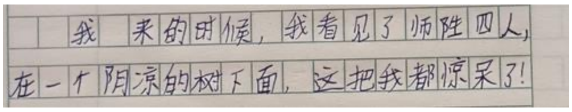
Image & Original Text	Output
 息。我把食物和水都放进了猫X。就走开了。	**GPT-4V:** 这幅图中包含一个错字和一个别字。错字是“掉”字，因为它的偏旁部首搭配错误，正确的偏旁部首应该是“辶”而不是“忄”。别字是“霊”字，这个字本身存在于字典中，但在这个上下文中，正确的字应该是“领”字，因为它们音近。 **Qwen-Vl-Max:** 图中没有发现明显的错字，别字为 "息" 和 "扔"。
 我飞快地跑进了蚂蚁巢一看，中间有一个 大虫子。蚂蚁们一个一个的给他进食。我X然	**GPT-4V:** 这段手写字中有一个错字“猪”写成了“蒙”，这是一个错字。这段手写字中也有一个别字，“大家”写成了“大夹”，“夹”是一个别字。 **Qwen-Vl-Max:** 图中没有发现明显的错字，别字为 "蚂"、"巢"、"虫" 和 "给"。
 我的心X之物是一种石X娃娃。	**GPT-4V:** 这句话中的“细”字是别字，正确的应该是“悉”字，因为它与“细”字发音相近。这句话没有错 **Qwen-Vl-Max:** 图中没有发现明显的错字，别字为 "石" 和 "膏"。
 我来的时候，我看见了师X四人， 在一个阴凉的树下面，这把我都惊呆了！	**GPT-4V:** 错字：“发”写错了，应该是“法”，“要”写错了，应该是“邀”。别字：“可以”的“以”误写成“已”，“一下”的“下”误写成“夏”。 **Qwen-Vl-Max:** 图片中所有的字都是正确的，没有需要更正的地方。

Figure 7: Cases from GPT-4V and Qwen-VL-Max.

posed Visual-C^3 is still a very challenging dataset worthy of further research, because Visual-C^3 comprehensively requires multimodal LLMs to have excellent image OCR capabilities, fine-grained stroke perception capabilities, and text semantic understanding ability.