AiM: Taking Answers in Mind to Correct Chinese Cloze Tests in Educational Applications

Yusen Zhang[1]*, **Zhongli Li**[2], **Qingyu Zhou**[2], **Ziyi Liu**[1]*, **Chao Li**[3], **Mina Ma**[2], **Yunbo Cao**[2], **Hongzhi Liu**[1]

[1]Peking University, [2]Tencent Cloud Xiaowei, [3]Xiaomi Group

{yusen-zhang0826@stu,lzymail@stu,liuhz@ss}.pku.edu.cn
{neutrali,qingyuzhou,minarma,yunbocao}@tencent.com
lichao51@xiaomi.com

Abstract

To automatically correct handwritten assignments, the traditional approach is to use an OCR model to recognize characters and compare them to answers. The OCR model easily gets confused on recognizing handwritten Chinese characters, and the textual information of the answers is missing during the model inference. However, teachers always have these answers in mind to review and correct assignments. In this paper, we focus on the Chinese cloze tests correction and propose a multimodal approach[1] (named AiM). The encoded representations of answers interact with the visual information of students' handwriting. Instead of predicting 'right' or 'wrong', we perform the sequence labeling on the answer text to infer which answer character differs from the handwritten content in a fine-grained way. We take samples of OCR datasets as the positive samples for this task, and develop a negative sample augmentation method to scale up the training data. Experimental results show that AiM outperforms OCR-based methods by a large margin. Extensive studies demonstrate the effectiveness of our multimodal approach.

1 Introduction

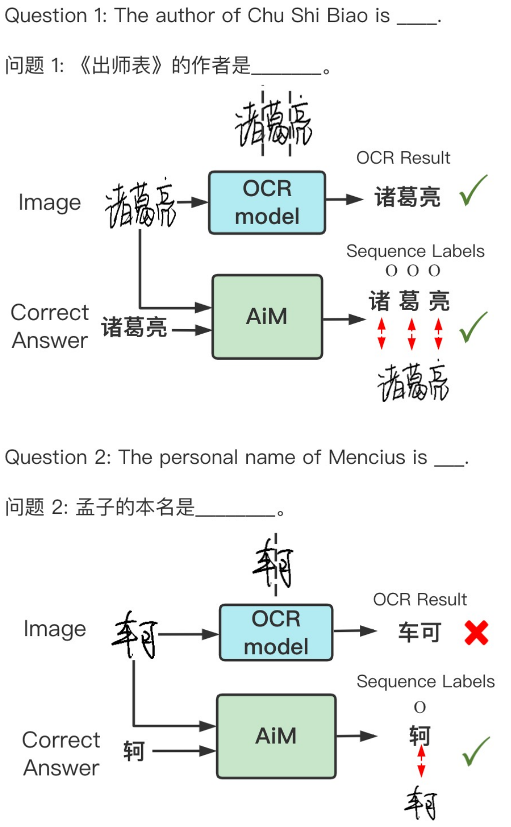

Question 1: The author of Chu Shi Biao is ____.
问题 1: 《出师表》的作者是______。
Image
OCR model
OCR Result
诸葛亮
✓
Sequence Labels
O O O
Correct Answer
诸葛亮
AiM
诸 葛 亮
✓
Question 2: The personal name of Mencius is ___.
问题 2: 孟子的本名是_______。
Image
OCR model
OCR Result
车可
✗
Sequence Labels
O
Correct Answer
轲
AiM
轲
✓

Figure 1: Task examples and correction methods. The OCR-based method misunderstands student's intent due to the width between "车" and "可" in Question 2. AiM can tackle this problem since it takes both the handwriting image and the textual answer into consideration.

The growing number of students has brought much pressure on teachers to correct assignments manually in the educational field. Recently, Optical Characters Recognition (OCR) based methods are widely used in several applications[2] to automatically complete this task. The OCR model first recognizes the text in the image, and then the OCR output is compared with the correct answer to feedback correction results. In this pipeline method, the post-comparison mainly relies on the pre-recognition. Without prior knowledge of answers, the OCR model easily gets confused when recognizing handwritten Chinese characters, especially for various handwriting styles, ligatures, and shape-similar characters. However, most of them can be distinguished by human teachers, because they take answers in mind at first.

In this paper, we focus on the Chinese cloze correction (CCC) and propose an **A**nswer-**i**n-**M**ind correction model (AiM) to tackle the above prob-

[*] Contribution done during internship at Tencent Cloud Xiaowei. Qingyu Zhou is the corresponding author.

[1] Our codes and data are available at https://github.com/YusenZhang826/AiM.

[2] Educational applications such as http://kousuan.yuanfudao.com, https://jiazhang.zuoyebang.com and https://www.zuoye.ai.

Proceedings of the 29th International Conference on Computational Linguistics, pages 3042–3053
October 12–17, 2022.

lem. Figure 1 shows examples of the CCC task and the comparison of the OCR-based method with our method. We look at Question 2. Obviously, the student knows the correct answer and writes down “轲”, but it is recognized as “车可” by the OCR model because of the slightly large width of the hand-written content. Consequently, the correction result is ‘wrong’. We hypothesize that the information of correct answers can help the neural model to understand student handwriting. As shown in Figure 1, AiM is a multimodal model, which takes the image and the answer text as input. Through the interaction of two modality information, our model understands the handwritten content and feeds back the ‘right’ correct result.

In the AiM model, the image is encoded to sequential feature representations through Resnet (He et al., 2016), where each of them represents a fixed-width pixels block. The answer text is encoded by word embedding. Then Transformer (Vaswani et al., 2017) self-attention is adopted to compute contextual representations for each modality. In order to fuse them, we develop a cross-modal attention. It renders the textual representations to interact with the visual information of students' handwriting. On the top of AiM, instead of predicting ‘right’ or ‘wrong’, our model performs sequence labeling on the answer text to infer which character differs from the handwritten content.

To train AiM, we collect EinkCC, a dataset containing about 5k handwriting images, answers, and correction results of cloze questions, from our educational application. Teachers distribute cloze tests in our app, and students practice on the electronic paper hardware. In addition to EinkCC, OCR datasets can be used for this task. We take samples of OCR datasets (Liu et al., 2011) as the positive samples, and construct negative samples by replacing the label characters with shape-similar ones derived from an open-sourced confusion set (Wu et al., 2013). The same method also augments EinkCC to scale up the training set.

We pretrain the image encoder in AiM to get better visual representations, and further train AiM with the correction objective. Experimental results show that compared with OCR-based methods (Liu et al., 2020; Du et al., 2021), AiM achieves 11% accuracy improvements. Extensive analyses verify that with the interactions between two modalities through our attention mechanism, AiM can understand student handwriting and ligatures, and more handwritten characters confused by OCR can be predicted well by AiM.

The main contributions are summarized as follows: i) We propose AiM, a multimodal model for Chinese cloze correction, to make up for shortages of OCR-based methods. ii) We extend OCR datasets using a negative sample augmentation method to fit this task. iii) Comprehensive experiments show that AiM achieves better performance compared with OCR-based methods, and it's effective and necessary to use a multimodal approach to correct Chinese cloze tests.

2 Preliminary

2.1 Chinese Cloze Correction

We first give the description of the Chinese cloze correction (CCC) task in this section. Given a handwriting image $\mathbf{I}$ and the textual answer $\mathbf{A} = [a_1, a_2, ..., a_m]$ of the corresponding question, assuming that the handwritten characters in the image is $\mathbf{C} = [c_1, c_2, ..., c_n]$, the target of the task is to predict a label $y \in \{0, 1\}$. The $y = 0$ indicates the correction result is ‘right’ (i.e. the handwritten content is a correct answer, $\mathbf{C} = \mathbf{A}$), otherwise it is ‘wrong’.

2.2 Transformer

Suppose the input of Transformer (Vaswani et al., 2017) is a pack of embeddings $\mathbf{X}^0 = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{|x|}]$. If we have L stacked Transformer blocks, the final output is like:

$$\mathbf{X}^l = \text{Transformer}_l(\mathbf{X}^{l-1}), l \in [1, L] \tag{1}$$

where each block consists of a self-attention layer, a feed-forward layer, residual connection (He et al., 2016) and layer normalization.

Self-Attention For the l-th block, the output $\mathbf{A}_l$ of a self-attention head is:

$$\begin{aligned}
&\mathbf{Q} = \mathbf{X}^{l-1}\mathbf{W}_l^Q, \mathbf{K} = \mathbf{X}^{l-1}\mathbf{W}_l^K \\
&\mathbf{M}_{i,j} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{forbid to attend} \end{cases} \\
&\mathbf{A}_l = \text{SelfAttention}(\mathbf{X}^{l-1}) \\
&\quad = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\intercal}{\sqrt{d_k}} + \mathbf{M})(\mathbf{X}^{l-1}\mathbf{W}_l^V)
\end{aligned} \tag{2}$$

where $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V$ can project previous output to queries, keys, and values, respectively. $\mathbf{M} \in \mathbb{R}^{|x| \times |x|}$ is a mask matrix that controls whether two tokens can attend each other.

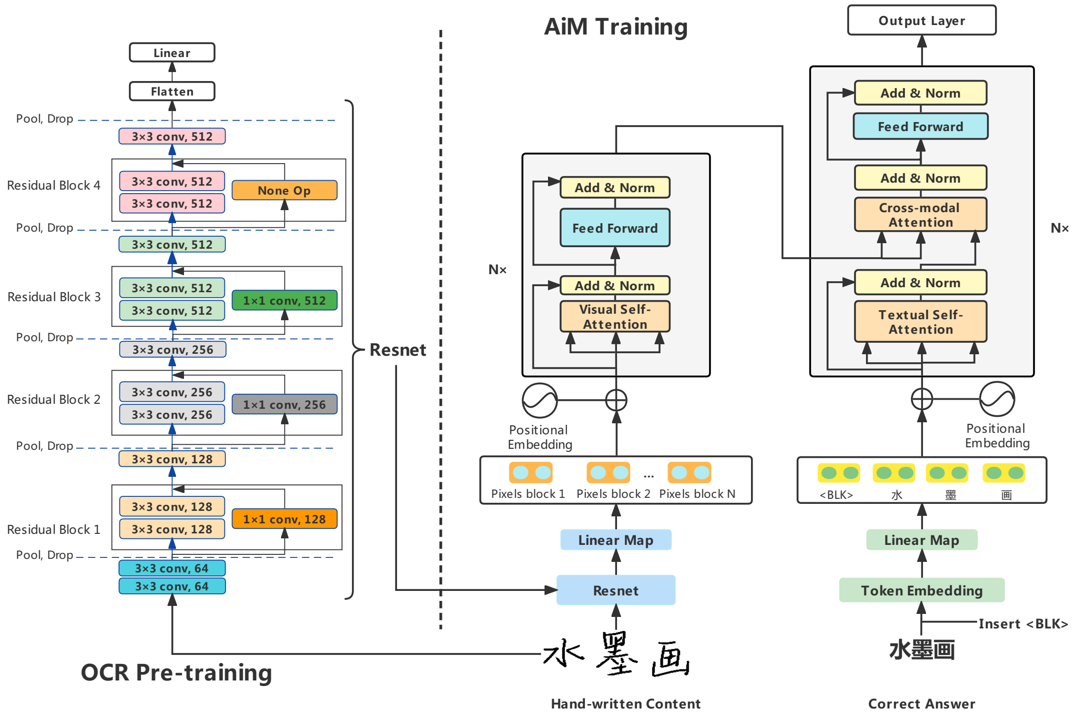


Figure 2: The architecture of AiM. The image encoder of AiM is initialized by OCR pretraining. In AiM, the visual and textual representations are fused by cross-modal attention.

3 Method

We first introduce the sequence labeling conversion for the CCC task in Section 3.1. Model architecture of AiM is shown in Section 3.2 and Figure 2. Finally, we describe the data augmentation and pre-training methods in Section 3.3.

3.1 Label Space

The CCC task is to feedback 'right' or 'wrong' on the handwritten content. In this paper, we perform sequence labeling on the answer text, and the corresponding labels are defined as:

- `del`: the current character does not appear in the image.
- `add`: compared to the handwritten content, one or more characters should be inserted between the current and the next character[3].
- `sub`: compared to the handwritten content, the current character should be substituted by another one.

We get these labels by calculating the edit distance between the answer and the handwritten content[4].

The BIO annotation (Ratinov and Roth, 2009) is adopted that the label space is {`O`, `B-sub`, `I-sub`, `B-del`, `I-del`, `B-add`}. It is similar to the label space of the Grammatical Error Detection/Correction (GED/GEC) task, but our method compares the answer text to the handwritten content, instead of the erroneous sentence to the correct sentence. After this conversion, AiM is trained in a fine-grained way. If the predicted sequence only contains '`O`', the correction result of AiM is 'right', otherwise it is 'wrong'.

3.2 Model Architecture

The model architecture of AiM is shown in Figure 2. Components include: i) an image encoder with Resnet and the self-attention mechanism to extract the visual features, ii) a fusion module with the cross-modal attention to mine the interactions between modalities, and iii) an output layer to predict the label sequence.

Image Encoder To understand the handwritten content in images, we follow Liu et al. (2020) to adopt Resnet (He et al., 2016) as the image encoder.

[3] A placeholder `<BLK>` is inserted at the beginning of the textual answers to handle the character missing at the first position.

[4] The handwritten text is annotated or taken from the OCR dataset for calculating the label sequence of AiM.

The image encoder maps the input image $\mathbf{I}$ to a sequence of visual features $\mathbf{H}_v \in \mathbb{R}^{N_v \times d}$ with a linear transformation:

$$\begin{aligned} \mathbf{H'}_v &= \text{Linear}\left(\text{ResNet}(\mathbf{I})\right) \\ \mathbf{H}_v &= \mathbf{H'}_v + \mathbf{P}_v \end{aligned} \tag{3}$$

where $N_v = \frac{max\_width}{32}$, max_width is the maximum width of input images and d is the hidden size of the visual representation. Each element in sequence represents a fixed-width pixels block in the image. Besides, there is a learnable positional embedding matrix $\mathbf{P}_v \in \mathbb{R}^{N_v \times d}$ where each row is a positional representation for each element in $\mathbf{H}_v$ to capture the location information.

We perform a padding operation on the image with extra white pixels blocks to ensure all images have the same width. Then we adopt Transformer blocks to capture the contextual information in visual modality and the l-th output is:

$$\mathbf{S}_v^l = \text{Transformer}(\mathbf{S}_v^{l-1}), l \in [1, L_v]. \tag{4}$$

where L_v is the number of Transformer blocks. Notes that $\mathbf{S}_v^0 = \mathbf{H}_v$ and $\mathbf{S}_v = \mathbf{S}_v^{L_v}$.

Fusion Module Assuming that the input answer has N_t characters, the answer characters are first converted to dense vectors $\mathbf{X} = [x_1, x_2, ..., x_{N_t}]$ through a word embedding. The linear transformation and positional embedding are also employed to compute the textual representation $\mathbf{H}_t \in \mathbb{R}^{N_t \times d}$ as follows:

$$\begin{aligned} \mathbf{H'}_t &= \text{Linear}(\mathbf{X}) \\ \mathbf{H}_t &= \mathbf{H'}_t + \mathbf{P}_t \end{aligned} \tag{5}$$

where $\mathbf{P}_t$ is the positional embedding matrix.

Then our fusion block further encodes and merges the information of textual and visual modality. As shown in Figure 2, each fusion block contains a textual self-attention layer and a cross-modal attention layer. Self-attention mechanism is employed to encode textual representations:

$$\begin{aligned} \mathbf{S}'_t &= \text{SelfAttention}(\mathbf{H}_t) \\ \mathbf{S}_t &= \text{LayerNorm}(\mathbf{H}_t + \mathbf{S}'_t) \end{aligned} \tag{6}$$

To capture the interactions between them, we develop a cross-modal attention as follows:

$$\begin{aligned} &\mathbf{Q} = \mathbf{S}_t\mathbf{W}^Q, \mathbf{K} = \mathbf{S}_v\mathbf{W}^K \\ &\mathbf{M}^f_{i,j} = \begin{cases} -\infty, & \text{padding token or pixels block} \\ 0, & \text{otherwise} \end{cases} \\ &\mathbf{S}_f = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{M}^f)(\mathbf{S}_v\mathbf{W}^V) \end{aligned} \tag{7}$$

where $\mathbf{W}^Q$ projects $\mathbf{S}_t$ to queries, $\mathbf{W}^K$ and $\mathbf{W}^V$ project $\mathbf{S}_v$ to keys and values, and $\mathbf{M}^f \in \mathbb{R}^{N_t \times N_v}$ is the mask matrix to ensure only valid tokens and pixels blocks can attend to each other. The fused representations $\mathbf{S}_f$ are followed by the feed-forward layer, residual connection, and layer normalization.

Finally, we stack our fusion blocks to obtain more informative fused features. It can be seen that our fusion module is similar to the Transformer decoder. Different from the cross-attention of Transformer, our cross-modal attention merges the visual and textual information, which allows each valid answer character to attend to all valid pixels blocks, without the order restriction of language modeling. Besides, the Transformer decoder decodes words one by one, but our module outputs fused features at once.

Output Layer We denote the above fused features as $\mathbf{H}_f$. To project the hidden representation to the space of labels, the fusion features is fed to the fully-connected layer and a softmax function to get the final output $\mathbf{O} \in \mathbb{R}^{N_t \times m}$:

$$\mathbf{O} = \text{softmax}(\mathbf{H}_f\mathbf{W}_o) \tag{8}$$

where $\mathbf{W_o} \in \mathbb{R}^{d \times m}$ is the weight matrix and m is the number of labels. During the training, we apply the cross-entropy function as our correction objective and the training loss is computed as:

$$\mathcal{L} = -\sum_{i=1}^{N_t} \log p(i, k_i), \;\; p(i, k_i) \in \mathbf{O} \tag{9}$$

where k_i is the label of the i-th character and $p(i, k_i)$ is the probability of the i-th character being predicted to label k_i.

3.3 Data Augmentation and Pretraining

Data Augmentation The sample of the OCR dataset contains an image and the corresponding written text, which can be easily extended for the CCC task. We directly take the annotated text as the answer to augment our positive CCC samples. Then we develop a negative sample augmentation method. Given a CCC sample, we keep the image unchanged and modify the answer text. The modifications include random character insertion, deletion and substitution. Especially for the character substitution, we attempt to construct hard negative samples by replacing the original character

Algorithm 1: Negative Sample Augmentation

```
Input: $\mathcal{S} = \{(\mathbf{I}_i, \mathbf{C}_i, \mathbf{A}_i, \mathbf{L}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{I}_i$ is
       the image, $\mathbf{C}_i$ is the handwritten content, $\mathbf{A}_i$
       is the answer text, $\mathbf{L}_i$ is the label sequence of
       AiM and $y_i$ is the correction result.
$D \leftarrow \mathcal{S}$
for $(\mathbf{I}_i, \mathbf{C}_i, \mathbf{A}_i, \mathbf{L}_i, y_i)$ in $\mathcal{S}$ do
    repeat
        $\hat{\mathbf{A}}_i \leftarrow \mathbf{A}_i$
        Randomly select an index $j$ in $\hat{\mathbf{A}}_i$
        $\hat{\mathbf{A}}_i[j] \leftarrow \mathbf{A}_i[j]$ 's shape-similar character
        Get $\hat{\mathbf{L}}_i$ comparing $\mathbf{C}_i$ and $\hat{\mathbf{A}}_i$
        $\hat{y}_i = 1$
        $D$.append $((\mathbf{I}_i, \mathbf{C}_i, \hat{\mathbf{A}}_i, \hat{\mathbf{L}}_i, \hat{y}_i))$
    until Random times
    repeat
        $\hat{\mathbf{A}}_i \leftarrow \mathbf{A}_i$
        Randomly select an index $j$ in $\hat{\mathbf{A}}_i$
        Delete $\hat{\mathbf{A}}_i[j]$
        Get $\hat{\mathbf{L}}_i$ comparing $\mathbf{C}_i$ and $\hat{\mathbf{A}}_i$
        $\hat{y}_i = 1$
        $D$.append $((\mathbf{I}_i, \mathbf{C}_i, \hat{\mathbf{A}}_i, \hat{\mathbf{L}}_i, \hat{y}_i))$
    until Random times
    repeat
        $\hat{\mathbf{A}}_i \leftarrow \mathbf{A}_i$
        Randomly select an index $j$ in $\hat{\mathbf{A}}_i$
        Insert a common Chinese character at $\hat{\mathbf{A}}_i[j]$
        Get $\hat{\mathbf{L}}_i$ comparing $\mathbf{C}_i$ and $\hat{\mathbf{A}}_i$
        $\hat{y}_i = 1$
        $D$.append $((\mathbf{I}_i, \mathbf{C}_i, \hat{\mathbf{A}}_i, \hat{\mathbf{L}}_i, \hat{y}_i))$
    until Random times
Output: the augmented dataset $D$
```

with shape-similar ones. The shape-similar characters are derived from an open-sourced confusion set[5] (Wu et al., 2013). The pseudo code of negative sample augmentation is presented in Algorithm 1.

Pretraining Before the AiM training, we first pretrain our image encoder with the OCR objective. We follow the Liu et al. (2020) to use CTC (Graves et al., 2006) as the loss function and apply high dropout rates after each max-pooling layer. As shown on the left side in Figure 2, the input image is converted to a sequence of dense representations through Resnet. A linear layer is added on the top of Resnet to transform the space dimension to the size of the vocabulary.

4 Experiment

In this section, we first introduce datasets, baselines, and other details of our experiments. Then we show experimental results and perform analyses from different views.

Dataset	Train set		Dev set		Test set	
	#img	#sample	#img	#sample	#img	#sample
EinkCC	4256	10610	-	-	673	673
HWCC	41781	166984	10499	41767	-	-
SynCC	150594	602376	-	-	-	-
Total	196631	779970	10499	41767	673	673

Table 1: Data statistics. "#img" means the number of handwriting images. "#sample" means the number of CCC samples. In each train or dev set, the "#sample" is larger than "#img" because of our data augmentation in Section 3.3.

Dataset	OCR Model	CER
HWCC	PP-OCRv2	0.3936
	CNN-CTC-CBS	0.2943
	Resnet-CTC	**0.0663**
EinkCC	PP-OCRv2	0.3115
	CNN-CTC-CBS	0.2325
	Resnet-CTC	**0.1661**

Table 2: The OCR performance on dev and test sets.

4.1 Dataset

EinkCC EinkCC is collected from our educational application. Teachers distribute cloze tests in our app, and students practice on the e-ink display hardware. When students finish tests, teachers can correct them and feedback to students. Each sample in EinkCC contains a student's handwriting image, the answer text and the correction result marked by teachers. It mainly covers the dictation of ancient poetry, the idiom application, and the reading comprehension. Besides, we manually annotate the image text for the OCR training and the AiM label sequence calculation.

HWCC CASIA-HWDB (Liu et al., 2011) is a benchmark for the handwritten Chinese text recognition(HCTR) task. We select the HWDB 2.x set to be the positive CCC samples.

SynCC To further enlarge the scale of CCC datasets, we first build a synthetic OCR dataset. The details of the data construction are presented in Appendix. Then all OCR samples are taken as the positive CCC samples.

We split EinkCC into train and test sets, and split HWCC into train and dev sets. All train sets and dev sets are extended by our negative sample augmentation method of Section 3.3. The data statistics are shown in Table 1.

[5] It is taken from URL.

Dataset	Model	Sequence level			Binary level			
		P	R	F1	P	R	F1	Acc
HWCC	PP-OCRv2	0.3590	0.4661	0.4057	0.7508	**0.9997**	0.8576	0.7510
	CNN-CTC-CBS	0.4584	0.5563	0.5026	0.7604	0.9994	0.8637	0.7635
	Resnet-CTC	0.8157	0.8877	0.8502	0.8578	0.9989	0.9230	0.8751
	AiM$_{\text{wo-PT}}$	0.4893	0.5602	0.5223	0.9083	0.9306	0.9193	0.8571
	AiM	**0.9735**	**0.9717**	**0.9726**	**0.9926**	0.9937	**0.9932**	**0.9898**
EinkCC	PP-OCRv2	0.1146	0.6847	0.1964	0.2128	**1.0000**	0.3509	0.4502
	CNN-CTC-CBS	0.1829	**0.8108**	0.2985	0.2710	**1.0000**	0.4264	0.6003
	Resnet-CTC	0.2311	0.7748	0.3561	0.3322	0.9964	0.4987	0.7013
	AiM$_{\text{wo-PT}}$	0.0568	0.2252	0.0907	0.2025	0.6497	0.3088	0.5676
	AiM	**0.2795**	0.5766	**0.3765**	**0.4262**	0.7799	**0.5512**	**0.8113**

Table 3: Main results on dev and test sets. AiM$_{\text{wo-PT}}$ is the AiM model without the OCR pretraining. Resnet-CTC is trained using the same data source as AiM, which is a strong OCR baseline in a fair comparison.

Model	CER
Resnet-CTC$_{\text{wo-Syn}}$	0.1916
Resnet-CTC$_{\text{wo-E}}$	0.2995
Resnet-CTC$_{\text{wo-H}}$	0.3304
Resnet-CTC	**0.1661**

Table 4: Data ablation of the Resnet-CTC OCR performance on the EinkCC test set.

Model	Sequence Level		
	P	R	F1
AiM$_{\text{wo-Syn}}$	0.0212	0.1712	0.0378
AiM$_{\text{wo-E}}$	0.1857	0.3964	0.2529
AiM$_{\text{wo-H}}$	0.2660	0.4865	0.3440
AiM	**0.2795**	**0.5766**	**0.3765**

Table 5: Data ablation of the AiM performance at the sequence level on the EinkCC test set.

4.2 Evaluation Metrics

We use the character error rate (CER) to evaluate the OCR performance in the pretraining stage, and compute the following metrics to evaluate the CCC performance:

Sequence level We use the widely-used metrics, Precision (P), Recall (R), and $F1$, to measure the quality of label sequences outputted by AiM.

Binary level We transform label sequences to binary labels ('right' or 'wrong') to compute the accuracy of binary classification. We report the $P, R,$ and $F1$ of negative samples (i.e. $y = 1$) to evaluate model's ability to identify wrong answers.

4.3 Implementation Details and Baselines

Our training dataset $\{(\mathbf{I}_i, \mathbf{C}_i, \mathbf{A}_i, \mathbf{L}_i, y_i)\}_{i=1}^{N}$ contains the image $\mathbf{I}_i$, the handwritten content $\mathbf{C}_i$, the answer text $\mathbf{A}_i$, the label sequence of AiM $\mathbf{L}_i$ and the correction result y_i. We pretrain the image encoder Resnet on the subset $\{(\mathbf{I}_i, \mathbf{C}_i)\}_{i=1}^{N}$, and train the AiM model on the subset $\{(\mathbf{I}_i, \mathbf{A}_i, \mathbf{L}_i)\}_{i=1}^{N}$.

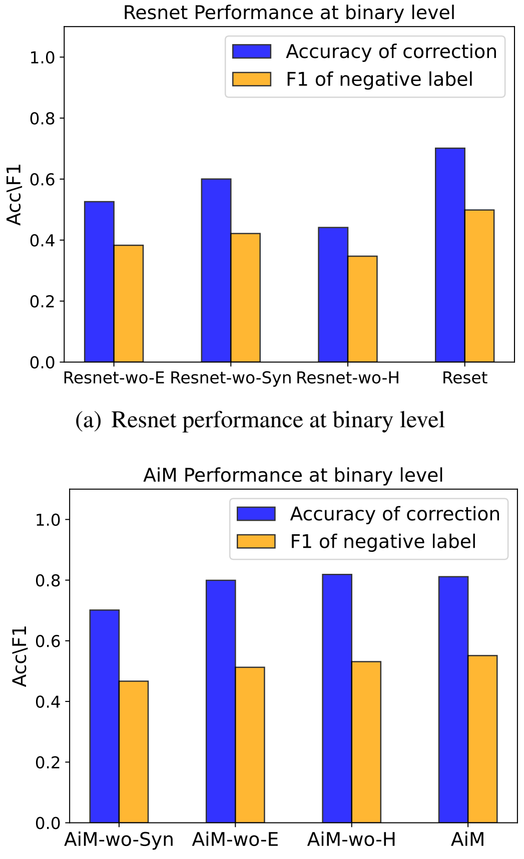
In the OCR pretraining, we follow Liu et al. (2020) to set the number of convolution blocks to {2,3,1,4}. The height of every input image is set to 128 pixels and the width is scaled to the corresponding value, so the shape of each pixels block is 128 × 32. The learning rate is set to 1e-3, the batch size is 8 and the training epoch is 15.

In the AiM training, the learning rate is 1e-4, the batch size is 8 and the training epoch is set to 14. The dimension of hidden states d is 768. The number of image encoder blocks N_{enc} and fusion module blocks N_{fus} in AiM are both 2. We adopt AdamW optimizer (Loshchilov and Hutter, 2018) and cosine-annealing strategy. To accelerate training, the parameters in Resnet are frozen when training AiM. We ignore punctuation in the text.

OCR-based methods are our baselines, where the correction results are derived from the post-comparison between the OCR results and answers. The following OCR models are evaluated: **Resnet -CTC** is the image encoder of AiM (Resnet) trained with the CTC function on the subset $\{(\mathbf{I}_i, \mathbf{C}_i)\}_{i=1}^{N}$. **PP-OCRv2** (Du et al., 2021) is an open-sourced widely-used OCR model in Chinese recognition, and we take its text recognition server model[6] for evaluation. **CNN-CTC-CBS** (Liu et al., 2020) is a handwritten text-line recognition model, and we take the released well-trained model[7] for evaluation. Table 2 shows the OCR performance of baselines on our datasets.

[6] https://github.com/PaddlePaddle/PaddleOCR.

[7] https://github.com/intel/handwritten-chinese-ocr-samples.



(a) Resnet performance at binary level

(b) AiM performance at binary level

Figure 3: Data ablation of the AiM and its baseline performance at the binary level on the EinkCC test set.

4.4 Results and Analyses

The main results are shown in Table 3. We evaluate OCR models at the OCR level and the binary level. For AiM and its variants, we evaluate them at the sequence level and the binary level. We describe our observations from the following perspectives:

- How do OCR models perform in CCC?
- Does AiM improve the performance and whether the two modalities are well fused?
- Do data augmentation and pretraining work and what's the impact of data source?

4.4.1 Limitation of OCR-based Method

As shown on Table 3, the recall of OCR-based methods is extremely high (nearly 1.0), but the precision and F1 are much lower. This means that OCR models can correct almost all wrong handwritten answers but mark many students' right text as wrong. Oppositely, AiM improves the precision, F1, and accuracy with large margins.

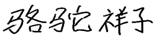
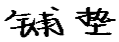
Image:		
Handwritten:	骆驼祥子	铺垫
Answer:	骆驼祥子	铺垫
OCR result:	骆马它祥子	铺挚
AiM output labels:	O O O O O	O O O
Correction label:	✓	✓
Correcting by OCR:	×	×
Correcting by AiM:	✓	✓

Table 6: Examples on EinkCC test set. Resnet gives the wrong outputs in all examples. The mistakes made by Resnet are shown in bold with underline, while AiM predicts correctly. Notes that the first label is for the placeholder '<BLK>'.

4.4.2 Influence of Data Source

We conduct an ablation study on the training data. The suffix '-wo-E', '-wo-Syn', '-wo-H' means the model is trained without the EinkCC training set, SynCC set and HWCC training set, respectively. The performance of Resnet and its variants is shown in Table 4 and Figure 3(a). The CER reaches the lowest level when Resnet is trained with all training sets, which can prove the necessity of data extension for OCR models. Table 5 and Figure 3(b) show the performance of AiM. Removing any training set, all metrics at sequence level drops, as well as the binary level. It means sufficient data is valuable in the CCC task. Meanwhile, comparing Figure 3(a) and 3(b), although the data scale decreases, the performance of AiM is always better than Resnet, which demonstrates the AiM is more robust and stable to data variation.

Notice that the model performance are always better on the HWCC dev set than it on the EinkCC test set. This is because all negative samples of HWCC are synthetic samples that are augmented in the same way. This suggests that AiM learns the error patterns well from our negative sample augmentation.

4.4.3 Effectiveness of Multimodal

As shown in Table 3, with AiM, the recall of negative label decreases[8] but the precision and F1-score increase significantly at the binary level. We give two examples on EinkCC in Table 6 to further analyze the effectiveness of AiM. Obviously, Resnet generates wrong outputs while AiM makes no mistake. In the first example, Resnet considers one left-right structured character as two independent characters, which never happens when manually correcting. In the second example, Resnet gets

[8] We perform an analysis on the recall drop in Appendix.

			Sequence level			Binary level			
N_{enc}	N_{fus}	Text Self-Att in Fusion	P	R	F1	P	R	F1	Acc
1	1	×	0.1150	0.3513	0.1733	0.3515	0.7099	0.4702	0.7623
1	1	✓	0.1773	0.4775	0.2585	0.3794	**0.7908**	0.5128	0.7727
2	2	×	0.1235	0.3694	0.1851	0.3630	0.7161	0.4818	0.7667
2	2	✓	**0.2795**	**0.5766**	**0.3765**	**0.4262**	0.7799	**0.5512**	**0.8113**
3	3	✓	0.2501	0.5495	0.3436	0.3682	0.7398	0.4917	0.7727

Table 7: An ablation study of components in AiM on EinkCC test set.

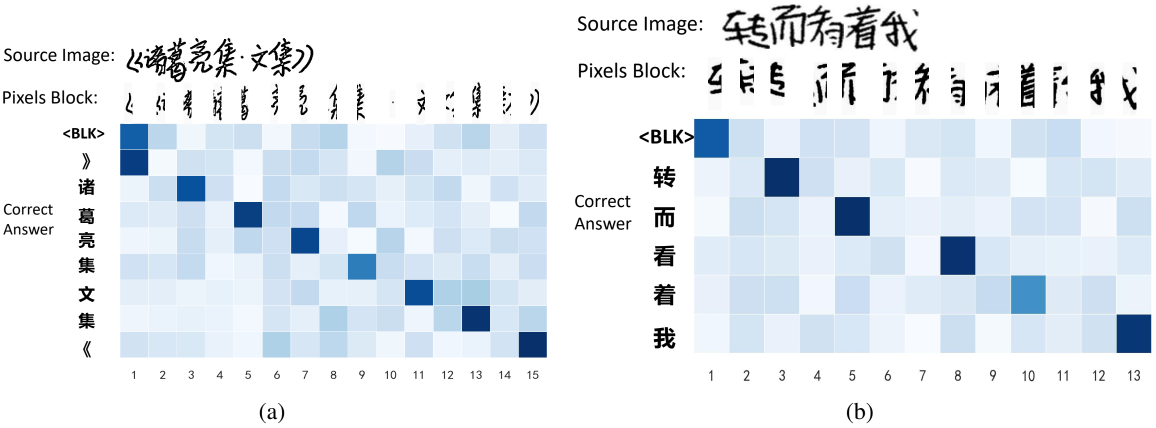


Figure 4: Cross-modal attention visualization. The figure shows the cross-attention scores between characters in answer and the pixels blocks in image.

confused by the shape-similar characters. AiM predicts successfully in these examples. More predictions on the dev and test sets are presented in Appendix to illustrate that AiM is able to identify all situations of the mismatch between the image and answer text.

Comparing AiM and AiM$_{\text{wo-PT}}$ in Table 3, without the pretraining, the performance of AiM drops significantly, even worse than Resnet. It demonstrates that visual modality features can be better understood through pretraining.

4.4.4 Learning of Modal Fusion

We conduct experiments to explore the contribution of each component. Experimental results are shown in Table 7. AiM reaches a better performance when encoding text information with the self-attention than only using token embeddings. This suggests that self-attention brings better textual representation. Meanwhile, stacking each module with a proper number of layers helps AiM capture interactions between modalities. We visualize the cross-modal attention scores in Figure 4. We find that characters manage to attend the most relative pixels block. For instance, in this sub-figure (a), the character “集” appears twice in different positions, and the model captures the context feature and both of them attend the corresponding pixels block in order. It shows that cross-modal attention can match multimodal information effectively.

5 Related Work

5.1 OCR Model

OCR models have evolved for a long period. Researchers usually used hybrid CNN and RNN architectures (Breuel, 2017) with CTC (Connectionist temporal classification) loss (Graves et al., 2006). For handwritten Chinese text recognition (HCTR) task, Liu et al. (2020) has achieved the state-of-the-art performance. They use the simple end-to-end CNN-CTC method and ease the overfitting problem with a high-rate dropout strategy.

5.2 Multimodal Model

Multimodal models have attracted more and more attention and have been applied in many fields, such as visual question answering (Antol et al., 2015), Chinese spell checking (Xu et al., 2021; Li et al., 2022) and other applications (Toto et al., 2021; Hu et al., 2021; Aguilar et al., 2019). Unsupervised pretraining (Kenton and Toutanova, 2019) has provided informative representations and fine-tuning techniques (Cui et al., 2019; Li et al., 2021) have brought further performance gains. Several large-scale pre-trained models are utilized in multimodal models (Anderson et al., 2018; Toto et al., 2021; Xu et al., 2021). To capture multimodal features, one common method is to concatenate encoders' output from two sides and then feed to the downstream multi-layer perceptrons (Nie et al., 2021). While some works use attention mechanism to get fusion representation (Lu et al., 2019; Tan and Bansal, 2019; Tsai et al., 2019).

6 Conclusion

In this paper, we propose a multimodal model AiM to effectively correct Chinese cloze tests. AiM employs cross-modal attention to understand the correlation between modalities. We collect data from different sources and develop a data augmentation method. Experiments show that AiM outperforms traditional OCR-based methods with over 11% accuracy improvements.

References

Gustavo Aguilar, Viktor Rozgić, Weiran Wang, and Chao Wang. 2019. Multimodal and multi-view models for emotion recognition. *arXiv preprint arXiv:1906.10198*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: visual question answering. *CoRR*, abs/1505.00468.

Thomas M Breuel. 2017. High performance text recognition using a hybrid convolutional-lstm implementation. In *2017 14th IAPR international conference on document analysis and recognition (ICDAR)*, volume 1, pages 11–16. IEEE.

Baiyun Cui, Yingming Li, Ming Chen, and Zhongfei Zhang. 2019. Fine-tune BERT with sparse self-attention mechanism. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3548–3553, Hong Kong, China. Association for Computational Linguistics.

Yuning Du, Chenxia Li, Ruoyu Guo, Cheng Cui, Weiwei Liu, Jun Zhou, Bin Lu, Yehua Yang, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. 2021. Pp-ocrv2: Bag of tricks for ultra lightweight OCR system. *CoRR*, abs/2109.03144.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Chuanbo Hu, Minglei Yin, Bin Liu, Xin Li, and Yanfang Ye. 2021. Detection of illicit drug trafficking events on instagram: A deep multimodal multilabel learning approach. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3838–3846.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

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

Zhongli Li, Qingyu Zhou, Chao Li, Ke Xu, and Yunbo Cao. 2021. Improving BERT with syntax-aware local attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 645–653, Online. Association for Computational Linguistics.

Brian Liu, Xianchao Xu, and Yu Zhang. 2020. Offline handwritten chinese text recognition with convolutional neural networks. *arXiv preprint arXiv:2006.15619*.

Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. 2011. Casia online and offline chinese handwriting databases. In *2011 International Conference on Document Analysis and Recognition*, pages 37–41. IEEE.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Yixin Nie, Linjie Li, Zhe Gan, Shuohang Wang, Chenguang Zhu, Michael Zeng, Zicheng Liu, Mohit Bansal, and Lijuan Wang. 2021. Mlp architectures for vision-and-language modeling: An empirical study. *arXiv preprint arXiv:2112.04453*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111.

Ermal Toto, ML Tlachac, and Elke A Rundensteiner. 2021. Audibert: A deep transfer learning multimodal classification framework for depression screening. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4145–4154.

Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2019. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

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

A Annotation Process

We present the process of annotation in our sequence labeling task in Figure 5. We put a label to individual character in textual answer referring to hand-written content and each label stands for a kind of editing operation.

B SynCC Data Construction

To build a synthetic OCR dataset, we first collect handwritten character images from the HWDB 1.0 set of CASIA-HWDB (Liu et al., 2011) and our educational application. Then the character images are spliced together into text-line images according to the sentences or clauses of our online essay corpus. We also replace characters with shape-similar ones and their images to enhance OCR models to recognize them. The synthetic OCR examples are shown on Table 8. These samples are taken as the positive CCC samples, and negative sample augmentation introduced in Section 3.3 is applied to construct SynCC dataset.

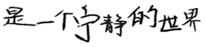
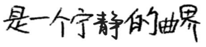
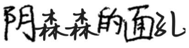
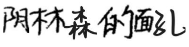
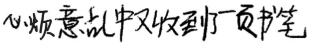
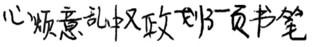
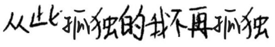
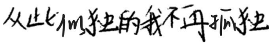
Image:	
Content:	是一个宁静的世界
Image:	
Content:	是一个宁静的曲界
Image:	
Content:	阴森森的面孔
Image:	
Content:	阴林森的面孔
Image:	
Content:	心烦意乱中又收到了一页书笔
Image:	
Content:	心烦意乱中又政到了一页书笔
Image:	
Content:	从此孤独的我不再孤独
Image:	
Content:	从此似独的我不再孤独

Table 8: Synthetic OCR examples of SynCC.

C Recall Drop on EinkCC

Table 3 shows that AiM improves the correction precision and F1 but the recall drops on EinkCC. We present several false-positive predictions on Table 9. All examples are annotated as wrong by teachers but are predicted as right by AiM. In the first example, the student writes down “斯” but there is a extremely large margin between the left and right sides. In the rest examples, students write down shape-similar characters but AiM can’t detect them and tend to predict as right. Thus, even though we use a confusion set to enhance AiM to detect shape-similar characters, there is still a lot of room for improvement in this regard.

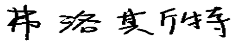
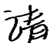
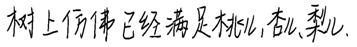
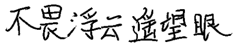
Image:	
Answer:	弗洛斯特
Labels:	O O O B-sub O
Image:	
Answer:	清
Labels:	O B-sub
Image:	
Answer:	... 已经满是桃儿...
Labels:	... O O O B-sub O O ...
Image:	
Answer:	不畏浮云遮望眼
Labels:	O O O O O B-sub O O

Table 9: False-positive predictions on EinkCC test set. All above answer characters are predicted to ‘O’ by AiM. The character marked by the underline indicates why the correction result should be wrong.

D Predictions of AiM

Table 10 shows several cases of AiM’s predictions. AiM is able to identify all kinds of modifications between the answer and hand-written content including substitutions, deletions and insertions, which proves that it’s reasonable to format the fill-in-the-blank assignments correction task to sequence labeling. In this way, AiM can not only indicate the correctness of students’ answers, but also can locate where the errors occur.

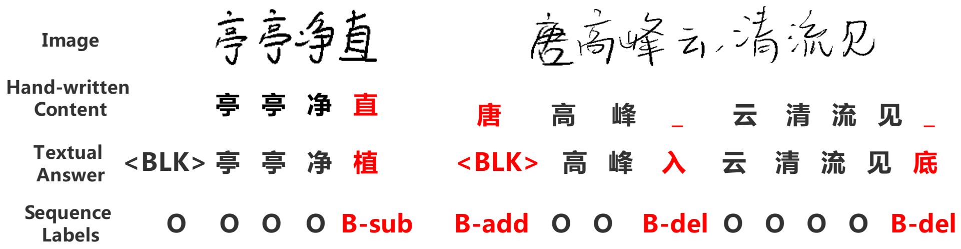


Figure 5: Examples of sequence labeling annotation. For the first sample, to convert the answer to the hand-written content, the last character in the answer “植” should be replaced by “直”. So the label ‘B-sub’ is annotated to “植” and ‘O’ for the rest. For the latter one, the character “入” and “底” do not appear in the hand-written content so the labels are both ’B-del’. Moreover, the character “唐” should be inserted at the first position in answer for conversion, so we put a ‘B-add’ label to the placeholder ‘<BLK>’.

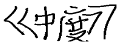
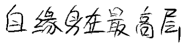
Image:	
Handwritten Content:	前 认 为 的 早 5 亿 年
Correct Answer:	麻 前 锢 认 为 的 早 5 亿 材 嚓 年
AiM Labels:	O B-del O B-del I-del O O O O O O B-del I-del O
AiM Predictions:	O B-del O B-del I-del O O O O O O B-del I-del O
Image:	
Handwritten Content:	《 中 度 》
Correct Answer:	《 中 庸 》
AiM Labels:	O O O B-sub O
AiM Predictions:	O O O B-sub O
Image:	
Handwritten Content:	自 缘 身 在 最 高 层
Correct Answer:	不 畏 浮 云 遮 望 眼
AiM Labels:	O B-sub I-sub I-sub I-sub I-sub I-sub I-sub
AiM Predictions:	O B-sub I-sub I-sub I-sub I-sub I-sub I-sub
Image:	
Handwritten Content:	自 己 寻 找 工 作
Correct Answer:	自 寻 工 作
AiM Labels:	O B-add B-add O O
AiM Predictions:	O B-add B-add O O

Table 10: Several outputs of AiM on dev and test sets. Some answers are generated through our data augmentation. AiM can identify editing operations between answers and hand-written content. Characters are shown in bold with underline if the corresponding label is not ‘O’.