LOP-OCR: A Language-Oriented Pipeline for Large-chunk Text OCR

Zijun Sun*, **Ge Zhang***, **Junxu Lu, and Jiwei Li**
Shannon.AI
{zijun_sun, ge_zhang, junxu_lu and jiwei_li}@shannonai.com

Abstract

Optical character recognition (OCR) for large-chunk texts (e.g., annuals, legal contracts, research reports, scientific papers) is of growing interest. It serves as a prerequisite for further text processing. Standard Scene Text Recognition tasks in computer vision mostly focus on detecting text bounding boxes, but rarely explore how NLP models can be of help.

It is intuitive that NLP models can significantly help large-chunk text OCR. In this paper, we propose LOP-OCR, a language-oriented pipeline tailored to this task. The key part of LOP-OCR is an error correction model that specifically captures and corrects OCR errors. The correction model is based on SEQ2SEQ models with auxiliary image information to learn the mapping between OCR errors and supposed output characters, and is able to significantly reduce OCR error rate.

LOP-OCR is able to significantly improve the performance of the CRNN-based OCR models, increasing sentence-level accuracy from 77.9 to 88.9, position-level accuracy from 91.8 to 96.5 and BLEU scores from 88.4 to 93.3.[1]

1 Introduction

The task of Optical character recognition (OCR) or scene text recognition (STR) is receiving increasing attentions (Deng et al., 2018; Zhou et al., 2017; Li et al., 2018; Liu et al., 2018). It requires recognizing scene images that varies in shape, font and color. The ICDAR competition[2] has become a world-wide competition and covered a wide range of real-world STR situations such as text in videos, incidental scene text, text extraction for biomedical figures, etc.

Different from standard STR tasks in ICDAR, in this paper, we specifically study the OCR task on scanned documents or PDFs that contain large chunks of texts, e.g., annuals, legal contracts, research reports, scientific papers, etc. There are several key differences between the tasks in ICDAR and large-chunk text OCR: firstly, ICDAR tasks focus on recognizing texts in scene images (e.g., images of a destination board). Texts are mixed with other distracting objects or embedded in the background (e.g., a destination board). The most challenging part of ICDAR tasks is separating text bounding boxes from other unrelated objects at the object detection stage. On the contrary, for OCR task on scanned documents, the key challenge lies in the identification of individual characters rather than text bounding boxes as since the majority of the image context is text. For alphabetical languages like English, character recognition might not be an issue since the number of distinct characters is small. But it could be a severe issue for logographic languages like Chinese or Korean, where the number of distinct characters are large (around 10,000 in Chinese) and many character shapes are highly similar; (2) In our task, since we

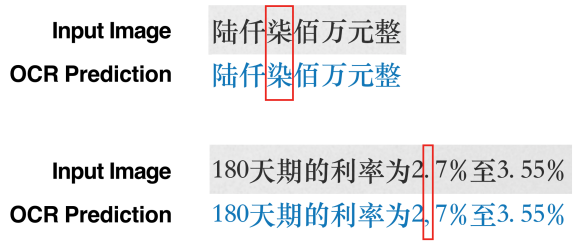

Figure 1: errors made by the CRNN-OCR model. Original input images are in black and output from OCR model is in blue. In the first example, 陆仟柒佰万元整 (67 million in English), the OCR model mistakenly recognize 柒 (the capital letter of 七 seven). In the second example 180天期的利率为2.7%至3.55% (The 180-day interest rate is from 2.7% to 3.55%) "." is mistakenly recognized as"," .

[1] Zijun Sun and Ge Zhang contribute equally to this paper.
[2] http://rrc.cvc.uab.es/

Task	Input	Output	Mapping Examples
En-Ch MT	English sen	Chinese sen	I → 我
grammar correction	sen with grammar errors	sen without grammar errors	are (from *I are a boy*) → am (from *I am a boy*)
spelling check	sen with spelling errors	sen without spelling errors	brake (from *I need to take a brake*) → break (from *I need to take a break*)
OCR correction	sen from the OCR model	sen without errors	陆仟染佰万元整→ 陆仟柒佰万元整

Table 1: The resemblance between the OCR correction task and other SEQ2SEQ generation tasks.

are trying to recognize large chunks of texts, predictions are dependent on surrounding predictions, it is intuitive that utilizing NLP models should significantly improve the performance. While for ICDAR tasks, texts are usually very short. NLP algorithms are thus of less importance.

We show two errors from the OCR model in Figure 4. The outputs are from the widely used OCR model CRNN (convolutional recurrent neural networks) (Shi et al., 2017) (details shown in Section 3). The model makes errors due to the shape resemblance between the character 染 (dye) and 柒 (the capital letter of seven in Chinese) in the first example, and "." and "," in the second example. Given the fact that most errors that the OCR model makes is erroneously recognizing a word as another similarly-shaped one, there is an intrinsic mapping between OCR output errors and supposed output characters: for example, the character 柒 can only be mistakenly recognized as 染 or some other characters of similar shape, but not random ones. This mapping captures the mistake-making patterns of OCR models, which we can harness to build a post-processing method to correct these errors. This line of thinking immediately points to the sequence-to-sequence (SEQ2SEQ) models (Sutskever et al., 2014; Vaswani et al., 2017), which learn the mapping between source words and target words. Actually, our situation greatly mimics the task of grammar correction or spelling checking (Xie et al., 2016; Ge et al., 2018b; Grundkiewicz and Junczys-Dowmunt, 2018; Xie et al., 2018). In the grammar correction task, SEQ2SEQ models generate grammatical sentences based on ungrammatical ones by implicitly learning the mapping between grammar errors and their corresponding corrections in targets. This mapping is systematic rather than random: for the correct sequence *"I am a boy"*, the ungrammatical correspondence is usually "*I are a boy*" rather than a random one like "*I two a boy*". This property is very similar to OCR correction.

In this paper, we propose LOP-OCR, a language-oriented post-processing pipeline for large-chunk text OCR. The key part of LOP-OCR is a SEQ2SEQ OCR-correction model, which combines the idea of image-caption generation and sequence-to-sequence generation by integrating image information with OCR outputs. LOP-OCR not only corrects errors from the source-target error mapping perspective, but also from the language modeling perspective: the objective of SEQ2SEQ modeling $p(y|x)$ automatically considers the context evidence of language modeling $p(y)$. By combining other ideas like round-way corrections and reranking, we observe a significant performance boost, increasing sentence-level accuracy from 0.779 to 0.889, and the BLEU scores from 88.4 to 93.3.

The rest of this paper is organized as follows: we describe related work in Section 2. The CRNN model for OCR is presented in Section 3. The details of the proposed LOP-OCR model are presented in Section 4 and experimental results are shown in Section 5, followed by a brief conclusion.

2 Related Work

2.1 Scene Text Recognition

Recognizing texts from images is a classic problem in computer vision. With the rise of CNNs (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; He et al., 2016; Huang et al., 2017), text detection is receiving increasing attention. The task has a key difference from image classification (assigning a single label to an image regarding the category that current image belongs to) and object detection (detecting a set of regions of interest, and then assigning a single label to each of the detected regions): the system is required to recognize a sequence of characters instead of a single label. There are two reasons that deep models like CNNs (Krizhevsky et al., 2012) cannot be directly applied to the scene text recognition task: (1) the

length of texts to recognize vary significantly; and (2) vanilla CNN-based models operate on images with fixed length, and are not able to predict a sequence of labels of various length. Existing scene text recognition models can be divided into two different categories: CNN-detection-based models and Convolutional-Recurrent Neural Networks models.

Detection-based models use Faster-RCNN (Ren et al., 2015) or Mask-RCNN (He et al., 2017) as backbones. The model first detects text bounding boxes and then recognizes the text within the box. Based on how the bounding boxes are detected, the models can be further divided into *pixel-based models* and *anchor-based models*.

Pixel-based models predict text bounding boxes directly based on text pixels. This is done using a typical semantic segmentation method: classifying each pixel as text or non-text using FPN (Lin et al., 2017), an encoder-decoder model widely used for semantic segmentation. Popular pixel-based methods include Pixel-Link (Deng et al., 2018), EAST (Zhou et al., 2017), PSENet (Li et al., 2018), FOTS (Liu et al., 2018) etc. EAST and EAST predict a text bounding box at each text pixel and then connect then using a locality aware model NMS. For Pixel-Link and PSENet, adjacent text pixels are linked together. Pixel-Link and PSENet perform significantly better than EAST and EAST on longer texts, bur requires a complicated post-processing method.

Anchor-based models detect bounding boxes based on anchors (which can be thought as regions that are potentially of interest), the key idea of which was first proposed in Faster-RCNN (Ren et al., 2015). Faster-RCNN generates anchors from features in the fully connected layer. Then the object offsets relative to the anchors are then predicted using another regression model. Anchor-based text detection models include Textboxes (Liao et al., 2017) and Textboxes++ (Liao et al., 2018). Textboxes propose modifications to Faster-RCNN and these modifications are tailored to text detection. More advanced versions such as DMPNet (Liu and Jin, 2017) and RRPN (Ma et al., 2018) are proposed.

Convolutional Recurrent Neural Networks (CRNNs) CRNNs combine CNNs and RNNs, and are tailored to predict a sequence of labels (Shi et al., 2017) from the images. An input image is first split into same-sized frames called receptive fields and the CNN layer extracts image features from each frame using convolutional and max-pooling layers with fully-connected layers being removed. Frame features are used as inputs to the bidirectional LSTM layers. The recurrent layers predict a label distribution of characters for each frame in the feature sequence. The idea of sequence label prediction is similar to CRFs: the predicted label for each frame is dependent on the labels of sounding frames. CRNN-based models outperform detection-based models on cases where texts are more densely distributed. In this paper, our OCR system uses CRNNs as backbones.

2.2 Sequence-to-Sequence Models

The SEQ2SEQ model (Sutskever et al., 2014; Vaswani et al., 2017) is a general encoder-decoder framework in NLP that generate a sequence of output tokens (targets) given a sequence of input tokens (sources). The model automatically learns the semantic dependency between source words and target words, and can be applied to a variety of generation tasks, such as machine translation (Luong et al., 2015b; Wu et al., 2016; Sennrich et al., 2015), dialogue generation (Vinyals and Le, 2015; Li et al., 2016a, 2015), parsing (Vinyals et al., 2015a; Luong et al., 2015a), grammar correction (Xie et al., 2016; Ge et al., 2018b,a; Grundkiewicz and Junczys-Dowmunt, 2018) etc.

The structure of SEQ2SEQ has kept evolving over the years, from the original LSTM recurrent models (Sutskever et al., 2014), to LSTM recurrent models with attentions (Luong et al., 2015b; Bahdanau et al., 2014), to CNN based models (Gehring et al., 2017), to transformers with self attentions (Vaswani et al., 2017).

2.3 Image Caption Generation

The image-caption generation task (Xu et al., 2015; Vinyals et al., 2015b; Chen et al., 2015) aims at generating a caption (which is a sequence of words) given an image. It is different from SEQ2SEQ tasks in that the input is an image rather than another sequence of words. Normally, image features are extracted using CNNs, based on which an decoder is used to generate the caption word by word. Attention models (Xu et al., 2015) is widely applied to map each caption token to a specific image region.

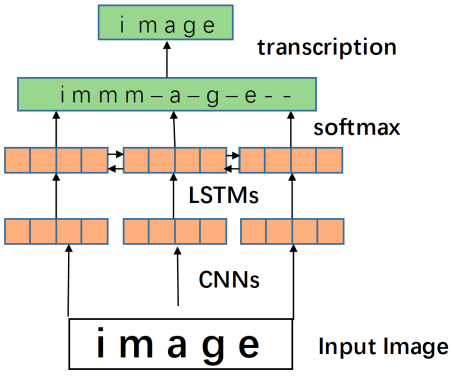

Figure 2: The RCNN model for object character recognition.

2.4 OCR Using Language Information

Using text information to post-process OCR outputs has long been existing (Tong and Evans, 1996; Nagata, 1998; Zhuang et al., 2004; Magdy and Darwish, 2006; Llobet et al., 2010). Specifically, Tong and Evans (1996) used language modeling probabilities to rerank OCR outputs. Nagata (1998) combined various features including morphology and word clusterings to correct OCR outputs. Finite-state transducers were used in Llobet et al. (2010) for post-processing. As far as we are concerned, our work is the first one that aims at learning to capture the error-making patterns of the OCR model. Additionally, the text-based model and the OCR model are pipelined and thus independent in previous work. Our work bridges this gap by combing the image information and OCR outputs together to generate corrections.

3 CRNNs for OCR

In this paper, we use the CRNN model (Shi et al., 2017) as the backbone for OCR. The model takes as input an image and output a sequence of characters. It consists of three major components: CNNs for feature extraction, LSTMs for sequence labeling and transcription .

CNNs for feature extraction Using CNNs with layers of convolution, pooling and element-wise activation, an input image D is first mapped to a matrix $M \in \mathcal{R}^{k \times T}$ matrix. Each column of the matrix m_t corresponds to a rectangle region of the original image in the same order to their corresponding columns from left to right. m_t is considered as the image descriptor for the corresponding receptive field. It is worth noting that one character might correspond to multiple receptive fields.

LSTMs for Sequence Labeling The goal of sequence labeling is to predict a label q_t for each frame representation m_t. q_t takes the value of the index of a character from the vocabulary or a BLANK label indicating the current receptive field does not correspond to any character. We use Bidirectional LSTMs, obtaining c_t^{left} from a left-to-right LSTM and c_t^{right} from a right-to-left LSTM for each receptive field. c_t is then obtained by concatenating both:

$$\begin{aligned} c_t^{\text{left}} &= \text{LSTM}^{\text{left}}(c_{t-1}^{\text{left}}, m_t) \\ c_t^{\text{right}} &= \text{LSTM}^{\text{right}}((c_{t+1}^{\text{right}}, m_t) \\ c_t &= [c_t^{\text{left}}, c_t^{\text{right}}] \end{aligned} \tag{1}$$

The label q_t is predicted using c_t:

$$p(q_t|c_t) = \text{softmax}(W \times c_t) \tag{2}$$

The sequence labeling model outputs a distribution matrix to the transcription layer: the probability of each receptive field being labeled as each label.

Transcription The output distribution matrix from the sequence labeling stage gives a probability for any given sequence or path $Q = \{q_1, q_2, ..., q_t\}$. Since each character from the original image can sit across multiple receptive fields, the output from LSTMs might contain repeated labels or blanks, for example, Q can be *hhh-e-l-ll-oo–*. Here we define a mapping $\mathcal{B}$ which removes repeated characters and blanks. $\mathcal{B}$ maps the output format from the sequence labeling stage Q to the format L. For example,

$$\mathcal{B}(\text{ Q:–hhh-e-llll-oo–}) = \text{L: hello}$$

The training data for OCR does not specify which character corresponds for which receptive field, but rather, a full string for the whole input image. This means that we have gold labels for L rather than Q. Multiple Qs thus can be transformed to one same gold L. The Connectionist Temporal Classification (CTC) layer proposed in Graves et al. (2006) is adopted to bridge this gap. The probability of generating sequence label L given the image D is the sum of probability of all paths Q (computed from the sequence labeling layer) given by that image:

$$p(L|D) = \sum_{\pi:\mathcal{B}(Q)=L} p(Q|D) \tag{3}$$

Directly computing Eq.3 is computationally infeasible because the number of Q is exponential to the number of its containing characters. Forward-backward model is used to efficiently compute Eq.3. Using CTC, the system can be trained based on image-string pairs in an end-to-end fashion. At test time, a greedy best-path-decoding strategy is usually adopted, in which the model calculates the best path by generating the most likely character at each time-step.

4 LOP-OCR

In this section, we describe the LOP-OCR model in detail.

4.1 Text2Text Correction

To learn the mistake-making pattern of the OCR model, we need to construct mappings between OCR errors and correct outputs. We can achieve this goal by directly training a Text2Text correction model using SEQ2SEQ models. The correction model takes as inputs the outputs of the OCR model and generate correct sequences. Suppose that $L = \{l_1, l_2, ..., l_{N_l}\}$ is an output from the CRNN model. L is the source input to the OCR-correction model. Each source word l is associated with a k-dimensional vector representation x. We use $X = [x_1, x_2, ..., x_{N_l}]$ to denote the concatenation of all input word vectors. $X \in \mathcal{R}^{k \times N_L}$. $Y = \{y_{1,}, y_{2,}, ..., y_{N_y}\}$ is the output of OCR-correction model. The SEQ2SEQ model defines the probability of generating Y given L:

$$p(Y|L) = \sum_{t \in [1, N_y]} p(y_{t,}|L, y_{1,t-1}) \quad (4)$$

It is worth noting that the length of the source N_l and that of the target N_y might not be the same. This stems from the fact that CRNNs at the transcription stage might mistakenly map a blank to a character, or a character to a blank, leading the total length to be different.

For the SEQ2SEQ structure, we use transformers (Vaswani et al., 2017) as a backbone. Specifically, the encoder consists of 3 layers, and each layer consists of a multi-head self attention layer, a residual connection layer and a positionwise fully connected layer. For the purpose of illustration, we only use n_{head}=1 for illustration. In practice, we set the number of multi-heads to be 8. Let $h_t^i \in \mathcal{R}^{K \times 1}$ denote the vector for time step t on the i^{th} layer. The operation at the self-attention layer and the feed-forward layer are shown as follows:

$$\begin{aligned} \text{atten}^i &= \text{softmax}(h_t^i \times W^{iT}) W^i \\ h_t^{i+1} &= \text{FeedForward}(\text{atten}^i + h_t^i) \end{aligned} \quad (5)$$

At the encoding time, W^i is the stack of vectors for all source words. At the decoding times, W^i is the stack of vectors for all source words plus words that have been generated, as being referred to as masked self-attention in Vaswani et al. (2017).

4.2 Text+Image2Text Correction

The issue with the Text2Text correction model is that corrections are conducted only based on OCR outputs, and that the model ignores important evidence provided by the original image. As will be shown in the experiment section, a correction model only based on text context might change correct outputs wrongly: changing correct OCR outputs to sequences that are highly grammatical but contain characters irrelevant to the image. The image information is crucial in providing guidance for error corrections.

One direct way to handle this issue is to use the concatenation the image matrix D and input string embeddings X as inputs to the SEQ2SEQ model. The disadvantage of doing so is obvious: we are not able to harness any information from the pre-trained OCR model. We thus use intermediate representations from the RCNN-OCR model rather than the image matrix D as SEQ2SEQ inputs. Recall that receptive fields d from the original image is mapped to vector representations using CNNs, and then a Bidirectional LSTM integrates context information and obtains vector representations $c = \{c_1, c_2, ..., c_{C_N}\}$ for the corresponding receptive fields. We use the combination of X and C as SEQ2SEQ model inputs.

There are two ways to combine C and X: vanilla concatenation (vanilla-concat for short) and aligned concatenation (aligned-concat for short), as will be described in order below.

vanilla-concat directly concatenates $C \in \mathcal{R}^{k \times N_C}$ and $X \in \mathcal{R}^{k \times N_L}$ along the horizontal axis. This makes the dimensionality of the input representation to be $k \times (N_L + N_C)$. One can think this strategy as the input containing $N_L + N_C$ words. At the encoding time, self-attention operations are performed between each pair of inputs at the complexity of

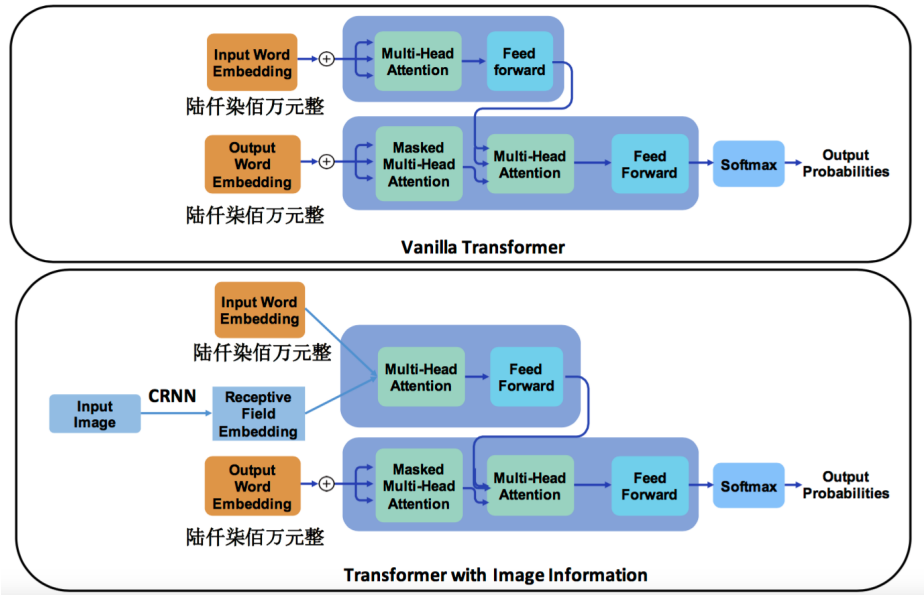

Figure 3: Illustration of the OCR-correction model with vanilla transformers and transformers using image information.

$(N_L + N_C) \times (N_L + N_C)$. This process can be thought as learning to construct links between source words and their corresponding receptive fields in the original image.

aligned-concat aligns intermediate representations of CRNNs c with corresponding input words x based on results from the CRNN model. Recall that at decoding time of CRNN, the model calculates the best path by selecting the most likely character at each time-step: c is first translated to the most likely token q at the LSTM sequence labeling process. Then the sequence of Q is mapped to L based on the mapping pattern $\mathcal{B}$ by removing repeated characters and blanks. This means that there is a direct correspondence between each decoded word $l \in L$ and receptive field representation c. The key idea of aligned-concat is to concatenate each source word x with corresponding receptive fields c. Since one x can be mapped to multiple receptive fields, we use one layer of convolution with max pooling to map a stack of c to a vector with invariant length k. This vector is then concatenated with x along the vertical axis, which makes the dimensionality of the input to transformers to be $2k \times N_L$.

For both vanilla-concat and aligned-concat models, inputs are normalized using layer normalizations since C and X might be of different scales. The SEQ2SEQ training errors are also back-propagated to the RCNN model. At decoding time, for all models (Text2Text and Text+Image2Text), we use beam search with a beamsize of 15.

4.3 Two-Way Corrections and Data Noising

The proposed OCR-correction model generates sentences from left to right. Therefore, errors are corrected based on left-to-right language models. This naturally points to its disadvantage: the model ignores the right-sided context.

To take advantage of the right-sided context information, we trained another OCR-correction model, with the only difference being that the token is generated from right to left. The right-to-left model shares the same structure with the left-to-right model. At both training and test time, the right-to-left model takes as input the output from the left-to-right model, and generate corrected sequences. Such strategy has been used in the literature of grammar correction (Ge et al., 2018b).

We also adopted the data noising strategy for data augmentation, which is proposed in SEQ2SEQ models (Xie et al., 2018). We implemented a backward SEQ2SEQ model to generate sources (sequences with errors) from targets (sequences without errors). We used the diverse decoding strategy (Li et al., 2016b) to map one correct sentence to multiple sentences with errors. This will increase the model's ability to generalize since the grammar correction model is exposed to more errors.

Model	ave edit dis	sen-acc	pos-acc	BLEU-4	Rouge-L
OCR	1.133	77.9	91.8	88.4	77.9
Text2Text	1.052	84.1	93.2	90.5	84.0
Text+Image2Text (vanilla concat)	1.002	86.2	94.2	91.6	85.9
Text+Image2Text (aligned concat)	0.984	87.0	95.0	92.2	86.7
Text+Image2Text (aligned concat)+roundway	0.970	88.0	95.9	92.7	87.3
Text+Image2Text (aligned concat)+roundway+noise	0.962	88.9	96.5	93.3	87.8

Table 2: Performances for different models.

ex1: Ranked top among 70 fund companies.			
gold	在70家基金公司中高居榜首。	OCR	在70家基金公司中高居悖首。
Text2Text	在70家基金公司中高居榜首。	Text+Image2Text	在70家基金公司中高居榜首。
ex2: Unanimous voice of the vegetable farmers.			
gold	是菜农的一致呼声。	OCR	是莱农的一致呼声。
Text2Text	是菜农的一致呼声。	Text+Image 2Text	是菜农的一致呼声。
ex3: Joined in the rescue missions.			
gold	又投入到紧张的救援中。	OCR	又投入到紧张的救援中。
Text2Text	又投入到紧张的恢复中。	Text+Image2Text	又投入到紧张的救援中。
ex4: Received the reward of free electricity from the Municipal Electric Power Bureau.			
gold	并受到市电业局力率电费嘉奖。	OCR	并受到市电业局力率电费嘉奖。
Text2Text	并受到市电业局力争电费嘉奖。	Text+Image2Text	并受到市电业局力率电费嘉奖。
ex5: Stimulated the activity of the entire banking sector.			
gold	刺激了整个银行股的活跃。	OCR	刺激了整个银行股的活题。
Text2Text	刺激了整个银行股的活题。	Text+Image2Text	刺激了整个银行股的活跃。

Table 3: Results give by the OCR model, the correction model only based on seq2seq correction models (denoted by vanilla-correct) and the seq2seq model with image information being considered. Characters marked in Blue denote correct characters, while those marked in red denote errors.

5 Experimental Results

In this section, we first describe the details for dataset construction, and then we report experimental results.

5.1 Dataset Construction

Since there is no publicly available datasets for large-chunk text OCR, we create a new benchmark. we generate image datasets using large-scale corpora. Images are generated and augmented dynamically during training. Two corpora are used for data generation: (1) Chinese Wikipedia: a complete copy of Chinese Wikipedia collected by Dec 1st, 2018 (448,858,875 Chinese characters in total) (2) Financial News: containing 200,000 financial related news collected from several Chinese News websites (308,617,250 characters in total). The CRNN model detects 8384 distinct characters, including common Chinese characters, English alphabet, punctuations and special symbols. We split the corpus into a set of short texts with smaller size (12-15 characters), and then we separated the text set into training, validation and test subsets with a proportion of 8:1:1. Within each subset of short texts, an image is generated for each short text by the following process:f (1) randomly picked a background color, a text color, a Chinese font and a font size for the image; (2) draw the short text on a 32 $\times$300 pixel RGB image with the attributes given in (1) and make sure the text is within image boundaries; (3) used a combination of 20 augmentation functions (including blurring, adding noises, affine transformations, adding color filters, etc.) to reduce the fidelity of the image so as to increase the robustness of CRNN model. The benchmark will be released upon publication.

5.2 Results

For correction models, we train a three-layer transformer with the number of multi-head set to 8.

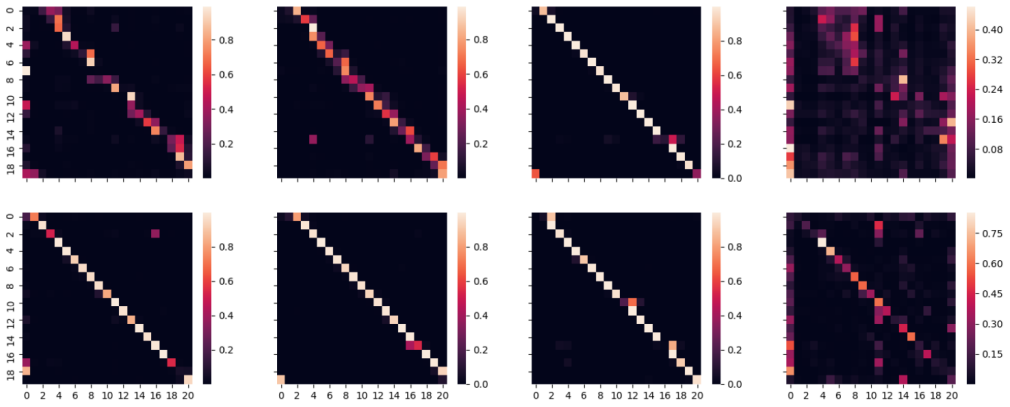

Figure 4: Illustrations for the 8 multi-head attentions for decoding. x axis corresponds to the source sentence: <s>180天期的利率为2,7%至3.55%</s> with length being 20. y axis corresponds to the target sentence: 180天期的利率为2.7%至3.55%</s> with length being 19. The erroneously decoded token by the OCR model "," is at the 12th position in the source. The corrected token "." is at the 11st position in the target.

We report the following numbers for evaluation: (1) average edit distance; (2) pos-acc: position-level accuracy, indicating whether in the corresponding positions of the decoded sentence and the reference sits the same character; (3) sen-acc: sentence-level accuracy, taking the value of 1 if the decoded sentence is exactly the same as the gold one, 0 otherwise; (4) BLEU-4: the four-gram precision of generated sentences (Papineni et al., 2002); and (5) Rouge-L: the recall of generated sentences (Lin, 2004).

Results are shown in Table 2. The Text2Text model takes outputs from the CRNN-OCR models as inputs, and feeds them to a vanilla transformer for correction. As can be seen, it outperforms the original OCR model by a large margin, increasing sentence-level accuracy from 77.9 to 84.1, and BLEU-4 score from 88.4 to 90.5. Figure 4 shows attention values between sources and targets at decoding time. We can see that the correction model is capable of learning the mapping between ground truth characters and errors, and consequently introduces significant benefits. ex1 and ex2 in Table 3 illustrate the cases where correction models are able to correct mistakes from the OCR model: in ex1, 悖 in 悖首 is corrected to 榜 in 榜首(rank top); in ex2, 莱 in 莱农 is corrected to 菜 in 菜农(vegetable farmers).

The Text+Image2Text models, both the vanilla-concat and the aligned-concat models, significantly outperform the Text2Text model, introducing an increase of 2.1 and 2.9 respectively with respect to sentence-level accuracy, and +1.1 and +1.7 with respect to BLEU-4 scores. This is in accord with our expectation: information from the original input image provides guidance for the correction model. Tangible comparisons between the Text2Text model and Text+Image2Text model are show in ex3, ex4 and ex5 of Table 3. For ex3 and ex4, the OCR model actually outputs correct outputs. But the Text2Text correction model changes the OCR output mistakenly. This is because the model is prune to making mistakes when image information is lost and context information dominates. The Text+Image2Text model doesn't have the above issues since a character is to be corrected only when the image provides strong evidence. In ex5, the Text+Image2Text model is able correct the mistake that the Text2Text model fails to correct.

Additional performance boosts are observed when using round-way corrections and adding noise to perform data augmentation. When combing all strategies, LOP-OCR is able to increase sentence-level accuracy from 77.9 to 88.9, position-level accuracy from 91.8 to 96.5 and BLEU score from 88.4 to 93.3.

6 Conclusion

In this paper, we propose LOP-OCR: A Language-Oriented Pipeline for Large-chunk Text OCR. The major component of LOP-OCR is an error correction model, which incorporates image information into the seq2seq model. LOP-OCR is able to significantly improve the performance of the CRNN-based OCR models, increasing sentence-level accuracy from 77.9 to 88.9, position-level accuracy from 91.8 to 96.5, BLEU scores from 88.4 to 93.3.

References

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

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*.

Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. 2018. Pixellink: Detecting scene text via instance segmentation. *arXiv preprint arXiv:1801.01315*.

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

Tao Ge, Furu Wei, and Ming Zhou. 2018b. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

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. ACM.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. *arXiv preprint arXiv:1804.05945*.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE.

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.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *CVPR*, volume 1, page 3.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2015. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Xiang Li, Wenhai Wang, Wenbo Hou, Ruo-Ze Liu, Tong Lu, and Jian Yang. 2018. Shape robust text detection with progressive scale expansion network. *arXiv preprint arXiv:1806.02559*.

Minghui Liao, Baoguang Shi, and Xiang Bai. 2018. Textboxes++: A single-shot oriented scene text detector. *IEEE Transactions on Image Processing*, 27(8):3676–3690.

Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. 2017. Textboxes: A fast text detector with a single deep neural network. In *AAAI*, pages 4161–4167.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. *Text Summarization Branches Out*.

Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. 2017. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4.

Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. 2018. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5676–5685.

Yuliang Liu and Lianwen Jin. 2017. Deep matching prior network: Toward tighter multi-oriented text detection. In *Proc. CVPR*, pages 3454–3461.

Rafael Llobet, Jose-Ramon Cerdan-Navarro, Juan-Carlos Perez-Cortes, and Joaquim Arlandis. 2010. Ocr post-processing using weighted finite-state transducers. In *2010 International Conference on Pattern Recognition*, pages 2021–2024. IEEE.

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015b. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. 2018. Arbitrary-oriented scene text detection via rotation proposals. *IEEE Transactions on Multimedia*.

Walid Magdy and Kareem Darwish. 2006. Arabic ocr error correction using character segment correction, language modeling, and shallow morphology. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 408–414. Association for Computational Linguistics.

Masaaki Nagata. 1998. Japanese ocr error correction using character shape similarity and statistical language model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2*, pages 922–928. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Baoguang Shi, Xiang Bai, and Cong Yao. 2017. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Xiang Tong and David A Evans. 1996. A statistical approach to automatic ocr error correction in context. In *Fourth Workshop on Very Large Corpora*.

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

Oriol Vinyals, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2015a. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, pages 2773–2781.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015b. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727*.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 619–628.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. East: an efficient and accurate scene text detector. In *Proc. CVPR*, pages 2642–2651.

Li Zhuang, Ta Bao, Xioyan Zhu, Chunheng Wang, and Satoshi Naoi. 2004. A chinese ocr spelling check approach based on statistical language models. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 5, pages 4727–4732. IEEE.