An End-to-End OCR Text Re-organization Sequence Learning for Rich-text Detail Image Comprehension

Liangcheng Li[1,2,3], Feiyu Gao[2], Jiajun Bu*[1,3,4], Yongpan Wang[1,2,3], Zhi Yu[1,3,4], and Qi Zheng[2]

[1] Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou, China
[2] Alibaba Group, Hangzhou, China
[3] Alibaba-Zhejiang University Joint Institute of Frontier Technologies, Hangzhou, China
[4] Ningbo Research Institute, Zhejiang University, Ningbo, China
liangcheng_li@zju.edu.cn,feiyu.gfy@alibaba-inc.com,bjj@zju.edu.cn,
yongpan@taobao.com,yuzhirenzhe@zju.edu.cn,yongqi.zq@taobao.com

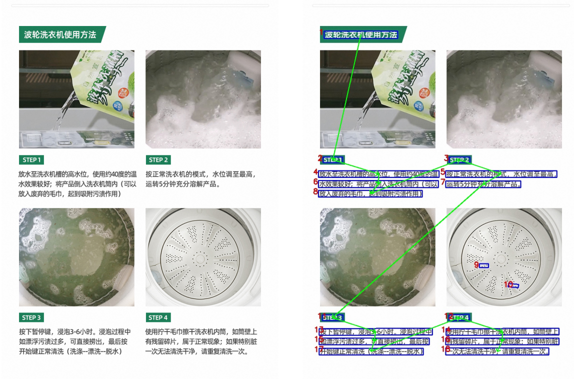
Abstract. Nowadays the description of detailed images helps users know more about the commodities. With the help of OCR technology, the description text can be detected and recognized as auxiliary information to remove the visually impaired users' comprehension barriers. However, for lack of proper logical structure among these OCR text blocks, it is challenging to comprehend the detailed images accurately. To tackle the above problems, we propose a novel end-to-end OCR text reorganizing model. Specifically, we create a Graph Neural Network with an attention map to encode the text blocks with visual layout features, with which an attention-based sequence decoder inspired by the Pointer Network and a Sinkhorn global optimization will reorder the OCR text into a proper sequence. Experimental results illustrate that our model outperforms the other baselines, and the real experiment of the blind users' experience shows that our model improves their comprehension.

Keywords: OCR Text Re-organization, Graph Neural Network, Pointer Network

1 Introduction

The internet era has given rise to the development of E-commerce and a large number of relevant platforms are springing up, such as Taobao, Jingdong and Amazon. Nowadays people are apt to participate in these websites for communications with online sellers and transactions on diverse commodities. To attract more consumers, these sellers take advantage of rich description text and commodity pictures to synthesize stylistic detail images, which help the consumers know their products as intuitive as possible.

* Corresponding author


波轮洗衣机使用方法
STEP 1
放水至洗衣机槽的高水位，使用约40度的温水效果较好；将产品倒入洗衣机筒内（可以放入废弃的毛巾，起到吸附污渍作用）
STEP 2
按正常洗衣机的模式，水位调至最高，运转5分钟充分溶解产品。
STEP 3
按下暂停键，浸泡3-6小时。浸泡过程中如漂浮污渍过多，可直接捞出，最后按开始键正常清洗（洗涤--漂洗--脱水）
STEP 4
使用拧干毛巾擦干洗衣机内筒，如筒壁上有残留碎片，属于正常现象；如果特别脏一次无法清洗干净，请重复清洗一次。

(a) (b)

Fig. 1. Example of a detail image (a) and the right reading order in (b). The blue boxes are the text blocks provided by OCR technology, the top-left red corner marks are the indexes of the text blocks. The green arrow lines in (b) show the proper reading route instead of reading from left to right and top to bottom simply.

Nevertheless, most detailed images are designed for healthy people who can comprehend both the image and text information directly. They ignore the demand of the visually impaired people who account for more than 27% of the world's population, such as the blind or the elderly. Since most existing screen readers cannot recognize the image format information, an interaction barrier between the visually impaired people and the e-commerce world has emerged. As the text is an essential tool for humankind's communication, it is an alternative to choose the description text in these detailed images for comprehension. Optical Character Recognition (OCR) technology devotes to mining the text information from several images, with its full application in scene text understanding[34], such as PhotoOCR [4], DocumentOCR [16]. Most classical and prevalent works on OCR concentrate on text detection [8, 13, 32] and recognition [1, 5, 14, 20]. They extract the characters in images and organize them into several text blocks according to semantic information, which performs well on many scene-text images, and detailed images are no exception.

However, the text in detail images has a flexible layout. It uses diverse typography structures to convey the product information, which causes the comprehending problem as the text blocks from OCR technology are discrete and lacking in **context order** without image structure. So it is often confusing for the visually impaired consumers when the screen reader reads the text blocks at an arbitrary order. Figure 1(a) shows an example of a detailed image, the blue boxes are the text blocks provided by OCR technology and the top-left red corner marks are the indexes of the text blocks. If the screen reader reads these text blocks from left to right and top to bottom, the visually impaired consumers are doomed to misinterpret even hardly comprehend the detailed images. Only the reading order in Figure 1(b) shows the same information that the raw detail image is expressed.

In this paper, we propose a novel end-to-end OCR text re-organization model for detailed image comprehension to tackle the problem as mentioned above. Based on the text detection feature extracted by a fully convolutional network (FCN), we use the text blocks to construct a graph structure and cast the problem to a graph to sequence model. Specifically, under the assumption that all the detailed images are probably be laid out regularly [15], we apply a graph convolution network (GCN) model with an attention mask to encode the logical layout information of the text blocks. A sequence decoder based on Pointer Network (PN) is proposed to obtain the text blocks' final order. We also introduce the Sinkhorn layer to make optimal global normalization by transforming the decoder predictions into doubly-stochastic matrices. Experiments on real-world detail image datasets have been conducted and show our method outperforms other sequence-oriented baselines both on local and global sequence evaluations. A real user experience test on blind people is also launched and shows the improvement of their comprehension.

Our contributions are threefold. First, to our best knowledge, it is the first time to propose the reading order problem for a rich-text detailed image based on OCR text blocks. Second, we propose an end-to-end graph to sequence model to solve the text blocks' re-organization problem using graph convolution network and pointer attention mechanism. Last, we design both quantitative sequence evaluation and real user experience tests among the blind people to convince our model's rationality and feasibility.

2 Related Work

Since the reading order re-organization problem is rarely mentioned and similar to the fields on sequence modeling, in this section, we briefly discuss related works on it. We also discuss traditional research on document analysis to show the similarities and differences with our work.

2.1 Sequence modeling

Sequence modeling has been widely researched in many fields. In computer vision, it aims to learn a proper order for a set of images according to some predefined rules [22]. A typical variation of this task is the jigsaw puzzle problem [18, 24], which needs to recover an image from a tile of puzzle segments. Jigsaw puzzle problems can be abstracted as ordering the image segments based on their shape or texture, especially on the boundaries [11, 19]. It is similar when regarding the OCR text blocks as sub-image regions and reconstructs their order. However, these methods are not suitable because OCR text blocks are discrete and isolated, with no joint boundary and continuous texture information.

Meanwhile, in natural language processing, RNN-based [21] Sequence-to-Sequence model (Seq2Seq) [27] and Neural Turing Machines [12] can solve most generative sequence tasks. However, they cannot solve the permutation problem where the outputs' size depends on the inputs directly. Vinyal et al. propose Pointer Network [29] which uses an attention mechanism to find the proper units from the input sequence and permute these as output. One of its application, text summarization, show the similarities of our work as they select some key information from the original text for summarization[7, 10]. Recently they are prevalent with the dynamic decision whether generating new words or permuting words from the original text inspired by the pointer mechanism[23, 33]. However, it is not suitable to generate the summarization of complete text information because the description text is carefully selected by the sellers to show the selling points [6], let alone the word deletion in extractive summarization. Meanwhile, as there remain some mistakes during the OCR text detection and recognition process, it is hard to guarantee the accuracy of the summarization under NLP features. Finally, sellers may tend to use concise and isolated phrases or words to describe their product, which has no grammar or syntax structure so that the summarization will fail to get whole sentences.

Furthermore, another line of research for sequence modeling has been devoted to converting other complex structures into sequences. Xu et al. [31] propose a graph to sequence model (Graph2Seq) with a GCN encoder and an attention Seq2Seq decoder to solve the bAbI artificial intelligence tasks [30]; Vinyals et al. [28] apply the attention mechanisms on input sets and propose the set to sequence model (Set2Seq) for language modeling and parsing tasks; Eriguchi et al. [9] design a tree to sequence (Tree2Seq) structure for extracting syntactic information for sentences. The commonality of these models is that their sequence decoders are all based on the Seq2Seq model, causing the limitation of output dictionary dependence.

2.2 Document analysis

Document analysis mainly includes two steps: document layout analysis and document understanding. The former process detects and annotates the physical structure of documents, and the latter process has several comprehension applications such as document retrieval, content categorization, text recognition[3]. However, most layout structure extraction and comprehension tasks on traditional documents are cast to a classification problem, which is different from text ordering tasks on scene-text images. It is hard to find homogeneous text regions and define semantic categories of the OCR text blocks with diverse layouts and open designs. Furthermore, scene texts with unique layouts and designs imply the visual cues and orders for comprehending the whole image, while document content analysis scheme is not suited for obtaining the order context.

3 Re-organization Model Architecture

Since the traditional sequence modeling methods cannot directly apply to the detailed image comprehension problem. This section sheds light on an end-to-end model to re-organize the OCR text block image regions for comprehension based on layout analysis. Specifically, we first define the re-organization task and then introduce the graph-based encoding method with an attention mask to get the layout embedding, finally we introduce a pointer-based attention decoder to solve the ordering problem.

3.1 Task definition

Given a set of text block images generated by OCR text detection and recognition from an original detail image, we need to generate a proper permutation of these blocks under which its text sequence can be comprehend. Formally, let us define an detail image with its OCR text block set $\mathcal{T} = \{t_1, t_2, \cdots, t_n\}$ where t$_i$ refers to the ith text block. Meanwhile, we also define an target permutation $\mathcal{P}^{\mathcal{T}} =< \mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_{m(\mathcal{T})} >$ where $m(\mathcal{T})$ is the indices of each unit in the text block set $\mathcal{T}$ between 1 and n. We are suggested to train an ordering model with the parameters w by maximizing the conditional probabilities for the training set as follows:

$$w^* = \arg\max_w \sum_{\mathcal{T}, \mathcal{P}^{\mathcal{T}}} \log p\left(\mathcal{P}^{\mathcal{T}} | \mathcal{T}; w\right) \tag{1}$$

where the sum operation means the sum of the total training examples. Actually, we cast the discrete image block re-organization process to a supervised sequence ordering problem.

3.2 Graph construction

We model each detail image as a graph of text blocks in which each independent text block are regarded as nodes with the image feature comprised for their attributes. We also take advantage of the geometric information (e.g. position) of the text blocks and construct edges to represent the original relations among them. Mathematically, we cast a detail image to a directed weighted graph structure $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{f(t_1), f(t_2), \cdots, f(t_n)\}$ is the set of n text blocks (i.e. nodes) and $f(t_i)$ stands for the attributes of the i^{th} text block, while $\mathcal{E} = \{r(e_{i,1}), r(e_{i,2}), \cdots, r(e_{i,n-1})\}$ is the set of edges and $e_{i,j}$ is the direct edge from node i to node j and $r(e_{i,j})$ stands for the attributes of the $e_{i,j}$ direct edge. In fact, we construct the fully connected graph for text blocks in a detail image primarily.

In order to obtain the attribute $f(t_i)$ for the i^{th} node, we consider the image feature which is related to the layout and image semantic feature instead of the text feature because the detail images do not have strict morphology and syntax structures. Given a detail image, we apply the Fully Convolutional Network (FCN) [17] model on detecting the text regions, then we extract its backbone and use the pretrained parameters from text detection to get the feature map of the total image. Combined with the text region bounding box, we get the text block feature as the node attributes with bi-linear interpolation technique.

As for the directed edge attributes, we consider the geometric information and take advantage of the position coordinates of the text blocks. Since the rectangle text regions are in difference size, we apply the relative position inspired by [16] to represent the edge attribute between node t_i and t_j as follows:

$$r\left(e_{i,j}\right)=\left[\mathit{\Delta}_{i,j}X,\mathit{\Delta}_{i,j}Y,\frac{l_i}{h_i},\frac{l_j}{h_i},\frac{h_j}{h_i},\frac{h_j}{l_i},\frac{l_j}{l_i}\right] \tag{2}$$

where $\mathit{\Delta}_{i,j}X$ and $\mathit{\Delta}_{i,j}Y$ stand for the horizontal and vertical euclidean distance of two text blocks based on their top-left coordinates, while l_i and h_i stand for the width and height of the i^{th} text block respectively. The third to eighth values of the attributes are the shape ratio of the node t^i, with four relative height and width of node t^j. Because the text blocks are not single points and have different region shapes, it is necessary to consider the impact of the shape instead of only using the euclidean distance of vertexes.

To summarize, we construct the graph of text blocks in a detail image with its node embedded the image textual features and its edge embedded the geometric features primarily, as Figure 2 depicts.

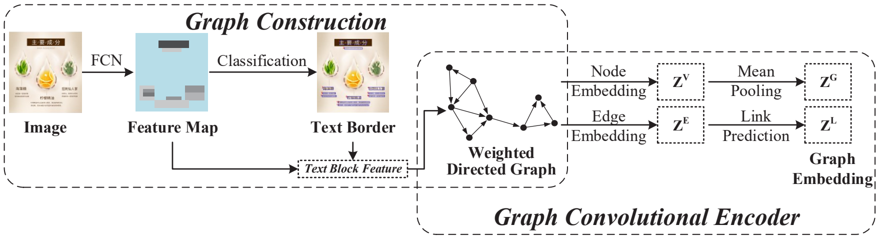


Fig. 2. The framework of graph construction and graph convolutional encoder module

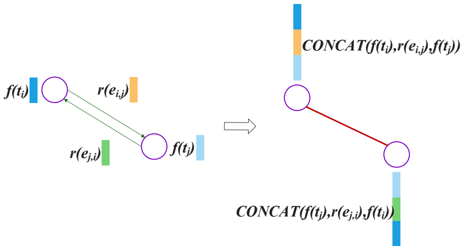

f(t_i)
r(e_{i,j})
r(e_{j,i})
f(t_j)
CONCAT(f(t_i),r(e_{i,j}),f(t_j))
CONCAT(f(t_j),r(e_{j,i}),f(t_i))

Fig. 3. The transformation of the directed weighted graph. The new feature contains the concatenation of two node feature vectors with the edge feature vector of their directed link.

3.3 Graph convolutional encoder

Compared to the traditional convolutional network, graph convolution is applied to the discrete data structure and learn the embeddings of nodes through the aggregation of their local neighbors. In this paper, we simultaneously perform the convolution operation on both nodes and edges. Because two directed edges link every two nodes, we deal with the node feature vector with the concatenation of two-node feature vectors and an edge feature vector that links them, as Figure 3 depicts. That is, for two text blocks' nodes t_i and t_j with two edges e_i and e_j between them, we define a new compound node $c_{i,j}$ with its feature vector $\boldsymbol{h}_{i,j}^{0}$ at 0^{th} layer as follows:

$$\boldsymbol{h}_{i,j}^{0} = \text{CONCAT}\left(f^{0}\left(t_i\right), r^{0}\left(e_{i,j}\right), f^{0}\left(t_j\right)\right) \tag{3}$$

then we can iteratively compute the l_{th} layer feature $h_{i,j}^{l}$ as follows:

$$\boldsymbol{h}_{i,j}^{l} = \sigma\left(\left(\boldsymbol{W}_{v}^{l}\right)^{T} \cdot \boldsymbol{h}_{i,j}^{l-1}\right) \tag{4}$$

where σ refers to the nonlinear activation function, and W_{v}^{l} refers to the node weight parameters of the l^{th} layer. However, to get the hidden representation of node t_i instead of compound node $c_{i,j}$, we also need to analyze and aggregate the proper local neighbors of the node t_i. Instead of using the traditional aggregator architectures like mean or LSTM aggregators, we use the self-attention mechanism on different hidden layers. Mathematically, the attention output embedding $f^{l}\left(t_i\right)$ for the node t_i at l^{th} layer can be calculated as follows:

$$f^{l}\left(t_i\right) = \sigma\left(\sum_{j \in \{k \mid \forall k \in NB(i)\}} \alpha_{i,j}^{l} \boldsymbol{h}_{i,j}^{l}\right) \tag{5}$$

where σ is a nonlinear activation function. Since we will mask the node with very low attention value and do not regard them as a proper local neighbor, $NB(i)$ refers to the local neighbors of the node t_i. Likewise, $\alpha_{i,j}^{l}$ refers to the attention coefficient between node t_i and t_j. Based on the [2], the attention coefficient can be defined as follows:

$$\alpha_{i,j}^{l} = \frac{\exp\left(\sigma\left(\left(\boldsymbol{w}_{a}^{l}\right)^{T} \boldsymbol{h}_{i,j}\right)\right)}{\sum_{u \in \{k \mid \forall k \in NB(i)\}} \exp\left(\sigma\left(\left(\boldsymbol{w}_{a}^{l}\right)^{T} \boldsymbol{h}_{i,u}\right)\right)} \tag{6}$$

where the σ refers to the LeakyReLU activation function, w_{a}^{l} is a attention weight vector of the l^{th} layer.

Meanwhile, we perform the edge embedding with more easier operation as we find that the compound node $c_{i,j}$ represents the edge link information of two nodes, so we define the convolution output embedding $r^l\left(e_{i,j}\right)$ for the edge $e_{i,j}$ at l^{th} layer as follows:

$$r^l\left(e_{i,j}\right)=\sigma\left(\left(\boldsymbol{W}_e^l\right)^T\cdot\boldsymbol{h}_{i,j}^{l-1}\right) \tag{7}$$

where σ is a nonlinear activation function, and $\boldsymbol{W}_e^l$ refers to the edge weight parameters of the l^{th} layer.

The intermediate output $f^l\left(t_i\right)$, $r^l\left(e_{i,j}\right)$ and $f^l\left(t_j\right)$ can be send to the next graph convolution layer as inputs according to Eq. 3. After K graph convolution operations, we can obtain the final node embedding feature matrix $\boldsymbol{Z}^V$ which combined by $f^K\left(t_i\right),\forall t_i\in\mathcal{N}$ and edge embedding feature matrix $\boldsymbol{Z}^E$ which combined by $r^K\left(e_{i,j}\right),\forall e_{i,j}\in\mathcal{E}$. Finally, we perform mean pooling operation on the node embedding to obtain the final graph representation $\boldsymbol{Z}^G$ as sequence, which is fed to the downstream pointer-based sequence decoder for the result order. Meanwhile, we use a fully-connected neural network to perform link prediction task for obtaining the relation features $\boldsymbol{Z}^L$ of the text blocks, which implies the layout constraints for the downstream decoder task. In Section 3.5 we will illustrate more about the layout constraints. The right blocks of Fig 2 shows the process of the encoder.

3.4 Pointer-based attention decoder

As for a sequence problem, the decoder of the text block re-organization task happens sequentially. That is, at each time step s, the decoder will output the node t_s according to the embeddings of the encoder and the previous output $t_{\mathrm{s}'}$ which $\mathrm{s}'<\mathrm{s}$. In this task, we have no output vocabulary and the nodes in the output sequence are just from the inputs. Therefore we apply a pointer-based decoder with a single-head attention mechanism. Figure 4 depicts the decoding process.

The information considered by the decoder at each time step s includes three embeddings, the graph embeddings from the encoder including node embeddings and layout constraints, and the previous (last) node embedding. Hence that at the first step we will use a special start label and learn the first node v^{input} as input placeholder. Formally, we define this information as a concatenating context vector $\boldsymbol{h}_c$ and compute as follows:

$$\boldsymbol{h}_{\mathrm{c}}=\begin{cases}[\boldsymbol{Z}^G,\boldsymbol{Z}^L,\boldsymbol{h}_{t_{s-1}}],s>1\\ [\boldsymbol{Z}^G,\boldsymbol{Z}^L,\boldsymbol{v}^{input}],s=1\end{cases} \tag{8}$$

where $[\cdot,\cdot,\cdot]$ is the horizontal concatenation. With the context vector, we will decode the corresponding node and use the result to update itself for the next prediction. Under the attention mechanism, we can compute a single query q_c from the context vector as follows:

$$\boldsymbol{q}_c=W^Q\boldsymbol{h}_c,\boldsymbol{k}_i=W^K\boldsymbol{h}_i,\boldsymbol{v}_i=W^V\boldsymbol{h}_i \tag{9}$$

where W^Q, W^K, W^V are the learning parameters and $\boldsymbol{h}_i$ is the node embedding, from which we get its key $\boldsymbol{k}_i$ and value $\boldsymbol{v}_i$. After that, we can compute the relation score of the query with all nodes, and mask the already visited nodes. the score $a_{c,i}$ is defined as follows:

$$a_{c,i} = \begin{cases} \dfrac{\boldsymbol{q}_c^T \boldsymbol{k}_i}{\sqrt{d_h}}, & if\ i \neq s', \forall s' < s \\ -\inf, & otherwise \end{cases} \tag{10}$$

where d_h is the node embedding dimentionality. Then we can compute the output softmax probability p_i of node t_i as follows:

$$p_i = \frac{\exp(a_{c,i})}{\sum_j \exp(a_{c,j})} \tag{11}$$

the decoder will choose the node with max probability as the output of each time step.

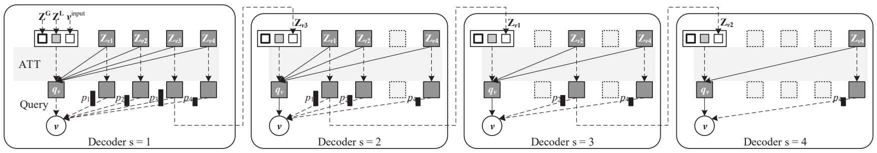


Fig. 4. The framework of pointer-based attention decoder. The decoder takes the graph embeddings including node embeddings and layout constraints. At each time step s, the decoder takes advantage of the graph embeddings and the last output node embedding where the learned placeholder is used at the first step. Once a node has been output, it will be masked and cannot be considered anymore. The example depicts that the output sequence $< t_3, t_1, t_2, t_4 >$ is decoded sequentially.

3.5 Sinkhorn global optimization

To improve the efficiency and make the max probability more significant, Sinkhorn normalization algorithm can be applied in the attention matrix. Because each text block has unique link to the next one, we can cast the attention matrix into a double-stochastic matrix with rows and columns summing to one. In Sinkhorn theory, any non-negative square matrix can be transformed into a double-stochastic matrix via iteratively scaling its rows and columns to one alternatively [25, 26]. Consider the attention matrix $\boldsymbol{A}^{n\times n}$ before the final prediction, and it can be transformed to a double-stochastic matrix by alternatively performing row and column normalization until its rows and columns summing to one. the row R and column C normalizing operations are defined as follows:

$$R_{i,j}(A) = \frac{A_{i,j}}{\sum_{k=1}^{n} A_{i,k}}; C_{i,j}(A) = \frac{A_{i,j}}{\sum_{k=1}^{n} A_{k,j}}. \tag{12}$$

And the Sinkhorn normalization SH for the l-th iteration is operated recursively by the following rules:

$$SH^n(\boldsymbol{A}) = \begin{cases} \boldsymbol{A}, & \text{if } n = 0 \\ C\left(R\left(SH^{n-1}(\boldsymbol{A})\right)\right), & \text{otherwise} \end{cases} \tag{13}$$

Then we can add Sinkhorn normalization for global optimal max probability of the output text block at each time step.

4 Experiments

In this section, we apply our model on real Detailed Image (DI) datasets with several types of products and use both global and local sequence evaluation methods to compare our model with other baselines. Furthermore, we launch a real user experience test on blind people and analyze their feedbacks.

4.1 Dataset

Since there is no work on re-organizing OCR text blocks for proper reading order on detail image, we first collect and label detail images from e-commerce platforms to construct the DI datasets. DI consists of about 10k detail images with more than 130k text blocks from several product types such as cosmetics, daily necessities, detergents, and the number of text block ranges from 5 to 50 for each detailed image. Due to some bad OCR results, redundant information, and irrelevant descriptions, we ignore these text blocks during the reordering process to guarantee that each text block's content is valid and necessary for comprehension. The layout of text blocks in DI includes horizontally text, multi-column text, ring, star and single key-value structural text, which implies different logical reading order. We communicate with real users including the visually impaired and the designers of the text images to understand how to comprehend the image only by the texts contained, then we induct and define the proper text order as all the text blocks from OCR are in the order of visual information acquisition, and keep the semantically related text blocks as close as possible in the ordering sequence. For our model, we assign 80% of the dataset for training, 15% for validation and 15% for the test.

4.2 Baselines

We compare the performance of our model with the following designed baselines.

Position-greedy (POS-Greedy) This method considers the position of the text blocks and under the row-major order to scan the OCR text blocks. It will select the nearest text block of the current one as its next linked one. Under the statistics, more than 98% detail images satisfy the rule that its first text block is relatively close to the top or left region, so we use it to decide the first block of the sequence.

Position-hierarchy (POS-Hier) This method considers the global minimum distance among all the pairs of OCR text blocks, then merge the pair into a new block iteratively and row-major order rules order the two text blocks.

Position-MLP (POS-MLP) This model only considers the geometrical feature with an MLP to predict the partial order of each pair. It solves the text block re-organization task according to the partial order pairs.

4.3 Evaluation metrics

Since it is a sequence order problem, we first use the **total order accuracy** of the detail image as the global sequence evaluation metric. We compare the ground truth sequence with our model's predict sequence by single block position matching, if there exist two blocks mismatching, the prediction of the detail image fails. The total order accuracy can be computed as the ratio of the number of detail images whose OCR text blocks are perfectly matched.

Other than the global sequence evaluation, we are inspired by the evaluation for discrete words in machine translation and apply the **BLEU** score for evaluating the local continuous coverage rate of the discrete OCR text blocks. Hence that we re-organize the text blocks from the input, it is meaningless to compute one block coverage (BLEU-1) as they always show the same value.

4.4 Results and Analysis

We first resize all the detailed images for 768×768 resolution as normalized input for feature extraction from the pretrained backbone, then we feed them into a two-layer graph convolution encoder for obtaining the graph embeddings, then the attention decoder will predict the sequence of the text blocks. We perform the last three models ten times within 300 epochs on NVIDIA Tesla P100 until convergence and choose the best one on the validation set. The main results are depicts in Table 1. As we can see, our proposed model GCN-PN and GCN-PN-Sinkhorn outperform among the baselines on global sequence prediction, which seems that the image feature from FCN is beneficial to predict more accurate re-organized sequence, because it is reasonable that the layout is related to the image feature and can help to infer the reading order. Meanwhile, the GCN encoder and PN decoder provide a more powerful order relation analysis than the rule-based method. Besides, adding Sinkhorn normalizing operation into the decoder is beneficial for total order prediction. It considers the total links among the text blocks and can weaken some potential wrong links that maybe only locally optimal.

Furthermore, we make a deep analysis on the local sub-sequence coverage. Intuitively, we use the BLEU score which is usually evaluated for machine translation tasks. Since we can consider each of the text blocks in the result sequence as a separate unit like word, we can compute the BLEU-2 and BLEU-4 for evaluating the coverage rate on 2 and 4 subsequent text blocks. Table 2 depicts the results. Hence, we use the NLTK package to compute the BLEU score, which adds a normalization to it and maps it into a value at $[0, 1]$ intervals. When the

Table 1. Total order accuracy of these models on DI test data

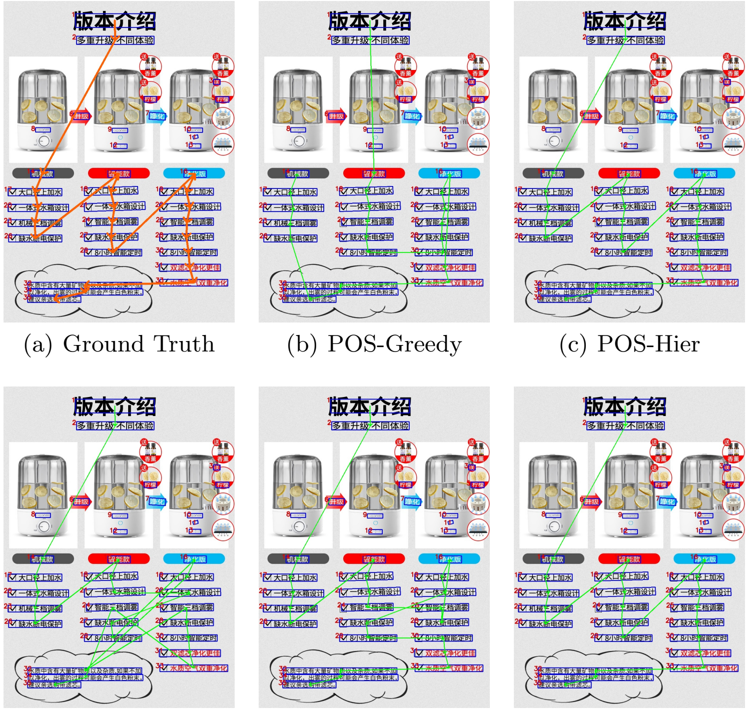
Method	Total Order Acc
POS-Greedy	0.41 ± 0.008
POS-Hier	0.70 ± 0.010
POS-MLP	0.75 ± 0.010
GCN-PN	0.79 ± 0.009
GCN-PN-Sinkhorn	**0.86 ± 0.005**

perfect matching happens, the value goes to 1, otherwise it goes to zero, and the large the value is, the higher the coverage rate is. From the table we can find that our GCN-PN-Sinkhorn model gets the highest coverage rate both on 2 and 4 subsequent text blocks, which implies the global order optimization also benefits the local order optimization. Hence that we also find the POS-Hier methods get high BLEU-2 score but low BLEU-4 score because this method merges two nearest blocks at each ordering step, it pays more attention to the 2-neighboring text blocks or text block groups. Furthermore, normal MLP may be more easily confused by some wrong sub links than attention-based GCN-PN models and are inferior to them on local evaluation. The greedy method shows the worst results on global and local evaluation because reading order on many complex layouts does not simply depend on position, such as multi-column, which has the rule that is reading the total column context one by one.

Table 2. The BLEU scores of these models on DI test data

Method	BLEU-2	BLEU-4
POS-Greedy	0.76	0.40
POS-Hier	0.89	0.66
POS-MLP	0.82	0.62
GCN-PN	0.90	0.71
GCN-PN-Sinkhorn	**0.92**	**0.74**

Fig 5 and Fig 6 show more details of the visual results. Fig 5 shows a multi-column structure example and we can find the POS-Hier (5(b)) and our GCN-PN-Sinkhorn model (5(f)) perform well as the ground truth, which also implies their ability for ordering local text blocks. Sinkhorn based model performs well than GCN-PN (5(e)) and POS-MLP because of the global optimizing to reduce the probability of some wrong links. Meanwhile, the Greedy method is easy to make a mistake and causes many inverse reading order links because it is highly sensitive to a variation on the text blocks' coordinates. Fig 6 shows a KV-table structure example and we find that POS-Hier (6(b)) cannot deal with this structure well because some keys in the table are more closed than keys to their



(a) Ground Truth (b) POS-Greedy (c) POS-Hier

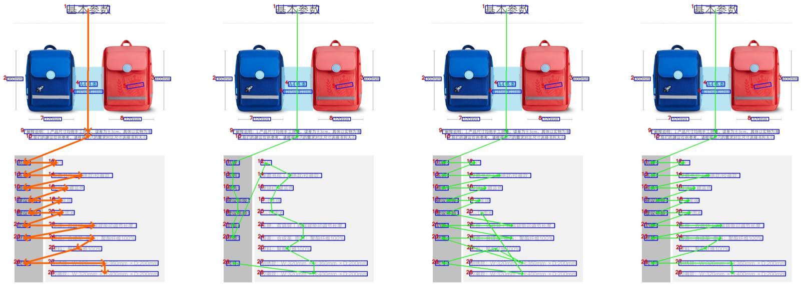
(d) POS-MLP (e) GCN-PN (f) GCN-PN-Sinkhorn

Fig. 5. An example of visualized reading order results. (a) is the ground truth order with orange arrow lines and (b)-(f) are the results of the methods with green arrow lines indicating the reading order.

values, resulting in a wrong merge operation. POS-MLP (6(c)) can order part of the former text blocks but failed at the latter ones, which implies the shortage of long order sequences. Our two model shows the same good results (6(d)) because the encoder-decoder structure can keep and use more global layout information to order the sequence.

4.5 Real user experience

We also design a real user experience in which the real blind people will participate in our test and check the predicted text block sequence that can be comprehended fluently. In this test, we use our model to generate the text block sequence from 113 detail images as a test group and use the untreated text block sequence (ordered by the reading scheme from top to bottom and left to right) as a control group. Meanwhile, three blind people who all receive compulsory education and often participate in online shopping are invited to our experiment. Their task is to hear both of the sequences and decide which one is better



(a) Ground Truth (b) POS-Hier (c) POS-MLP (d) Ours

Fig. 6. A KV-table structure example of visualized reading order results for analyzing row-major locality. (a) is the ground truth order with orange arrow lines and (b)-(d) are the results of the methods with green arrow lines indicating the reading order.

to comprehend. There is no other comprehension assistance during the experiment, and three of them do not know the corresponding model of the sequence beforehand. It takes them a week to complete the task and submit their choices and feedbacks. The result shows that all the subjects believe that our model outperforms more than 70% detailed images to help them comprehend well.

5 Conclusion

In this paper, we focus on the OCR text reordering problems. An end-to-end re-organization sequence learning structure is first proposed in the e-commerce scene. With a pretrained text detection network FCN, we extract the image feature and incorporate it with the geometric feature to build a weighted directed graph structure. Then a graph convolution encoder with a self-attention mechanism is considered to obtain the graph embeddings. Then a pointer-based attention decoder with a Sinkhorn global normalization is applied to predict the permutation. Our model outperforms the baselines both on global and local evaluations and will help get a more accurate and thorough comprehension of detailed images, especially for the visually impaired.

6 Acknowledgement

This work is supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies, The National Key R&D Program of China (No. 2018YFC2002603, 2018YFB1403202), Zhejiang Provincial Natural Science Foundation of China (No. LZ13F020001), the National Natural Science Foundation of China (No. 61972349, 61173185, 61173186) and the National Key Technology R&D Program of China (No. 2012BAI34B01, 2014BAK15B02).

References

1. Baek, Y., Lee, B., Han, D., Yun, S., Lee, H.: Character region awareness for text detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9365–9374 (2019)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Binmakhashen, G.M., Mahmoud, S.A.: Document layout analysis: A comprehensive survey. ACM Computing Surveys (CSUR) **52**(6), 1–36 (2019)
4. Bissacco, A., Cummins, M., Netzer, Y., Neven, H.: Photoocr: Reading text in uncontrolled conditions. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 785–792 (2013)
5. Busta, M., Neumann, L., Matas, J.: Deep textspotter: An end-to-end trainable scene text localization and recognition framework. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2204–2212 (2017)
6. Chakraborty, A., Paranjape, B., Kakarla, S., Ganguly, N.: Stop clickbait: Detecting and preventing clickbaits in online news media. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). pp. 9–16. IEEE (2016)
7. Cheng, J., Lapata, M.: Neural summarization by extracting sentences and words. arXiv preprint arXiv:1603.07252 (2016)
8. Dai, Y., Huang, Z., Gao, Y., Xu, Y., Chen, K., Guo, J., Qiu, W.: Fused text segmentation networks for multi-oriented scene text detection. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 3604–3609. IEEE (2018)
9. Eriguchi, A., Hashimoto, K., Tsuruoka, Y.: Tree-to-sequence attentional neural machine translation. arXiv preprint arXiv:1603.06075 (2016)
10. Filippova, K., Alfonseca, E., Colmenares, C.A., Kaiser, L., Vinyals, O.: Sentence compression by deletion with lstms. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. pp. 360–368 (2015)
11. Freeman, H., Garder, L.: Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. IEEE Transactions on Electronic Computers (2), 118–127 (1964)
12. Graves, A., Wayne, G., Danihelka, I.: Neural turing machines. arXiv preprint arXiv:1410.5401 (2014)
13. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. International Journal of Computer Vision **116**(1), 1–20 (2016)
14. Khare, V., Shivakumara, P., Raveendran, P., Blumenstein, M.: A blind deconvolution model for scene text detection and recognition in video. Pattern Recognition **54**, 128–148 (2016)
15. Kool, W., van Hoof, H., Welling, M.: Attention, learn to solve routing problems! arXiv preprint arXiv:1803.08475 (2018)
16. Liu, X., Gao, F., Zhang, Q., Zhao, H.: Graph convolution for multimodal information extraction from visually rich documents. arXiv preprint arXiv:1903.11279 (2019)
17. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
18. Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. In: European Conference on Computer Vision. pp. 69–84. Springer (2016)

19. Pomeranz, D., Shemesh, M., Ben-Shahar, O.: A fully automated greedy square jigsaw puzzle solver. In: CVPR 2011. pp. 9–16. IEEE (2011)
20. Rong, X., Yi, C., Tian, Y.: Unambiguous text localization and retrieval for cluttered scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5494–5502 (2017)
21. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science (1985)
22. Santa Cruz, R., Fernando, B., Cherian, A., Gould, S.: Deeppermnet: Visual permutation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3949–3957 (2017)
23. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. arXiv preprint arXiv:1704.04368 (2017)
24. Sholomon, D., David, O., Netanyahu, N.S.: A genetic algorithm-based solver for very large jigsaw puzzles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1767–1774 (2013)
25. Sinkhorn, R.: A relationship between arbitrary positive matrices and doubly stochastic matrices. The annals of mathematical statistics **35**(2), 876–879 (1964)
26. Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics **21**(2), 343–348 (1967)
27. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. pp. 3104–3112 (2014)
28. Vinyals, O., Bengio, S., Kudlur, M.: Order matters: Sequence to sequence for sets. arXiv preprint arXiv:1511.06391 (2015)
29. Vinyals, O., Fortunato, M., Jaitly, N.: Pointer networks. In: Advances in Neural Information Processing Systems. pp. 2692–2700 (2015)
30. Weston, J., Bordes, A., Chopra, S., Rush, A.M., van Merriënboer, B., Joulin, A., Mikolov, T.: Towards ai-complete question answering: A set of prerequisite toy tasks. arXiv preprint arXiv:1502.05698 (2015)
31. Xu, K., Wu, L., Wang, Z., Feng, Y., Witbrock, M., Sheinin, V.: Graph2seq: Graph to sequence learning with attention-based neural networks. arXiv preprint arXiv:1804.00823 (2018)
32. Yin, F., Wu, Y.C., Zhang, X.Y., Liu, C.L.: Scene text recognition with sliding convolutional character models. arXiv preprint arXiv:1709.01727 (2017)
33. You, Y., Jia, W., Liu, T., Yang, W.: Improving abstractive document summarization with salient information modeling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2132–2141 (2019)
34. Zhu, Y., Yao, C., Bai, X.: Scene text detection and recognition: Recent advances and future trends. Frontiers of Computer Science **10**(1), 19–36 (2016)