

# Attention Where It Matters: Rethinking Visual Document Understanding with Selective Region Concentration

Haoyu Cao<sup>1\*</sup>, Changcun Bao<sup>1\*</sup>, Chaohu Liu<sup>1,2†</sup>, Huang Chen<sup>1</sup>, Kun Yin<sup>1</sup>, Hao Liu<sup>1</sup>,  
Yinsong Liu<sup>1</sup>, Deqiang Jiang<sup>1</sup>, Xing Sun<sup>1</sup>

Tencent YouTu Lab<sup>1</sup>, University of Science and Technology of China<sup>2</sup>

{rechyc, changcunbao, huaangchen, zhanyin, ivanhliu, jasonysliu, dqiangjiang}@tencent.com  
liuchaohu@mail.ustc.edu.cn, winfred.sun@gmail.com

## Abstract

We propose a novel end-to-end document understanding model called **SeRum** (SElective Region Understanding Model) for extracting meaningful information from document images, including document analysis, retrieval, and office automation. Unlike state-of-the-art approaches that rely on multi-stage technical schemes and are computationally expensive, **SeRum** converts document image understanding and recognition tasks into a local decoding process of the visual tokens of interest, using a content-aware token merge module. This mechanism enables the model to pay more attention to regions of interest generated by the query decoder, improving the model’s effectiveness and speeding up the decoding speed of the generative scheme. We also designed several pre-training tasks to enhance the understanding and local awareness of the model. Experimental results demonstrate that **SeRum** achieves state-of-the-art performance on document understanding tasks and competitive results on text spotting tasks. **SeRum** represents a substantial advancement towards enabling efficient and effective end-to-end document understanding.

## 1. Introduction

Understanding document images is a fundamental task that involves extracting meaningful information from them, such as document information extraction [51] or answering visual questions related to the document [34]. In today’s world, where the volume of digital documents is increasing exponentially, this task has become even more critical in various applications, including document analysis [6], document retrieval [33], and office robotic process automation (RPA) [1].

Current state-of-the-art approaches [15] rely on multi-stage technical schemes involving optical character recogni-

\*Equal contribution. †Work done during internship at YouTu Lab.

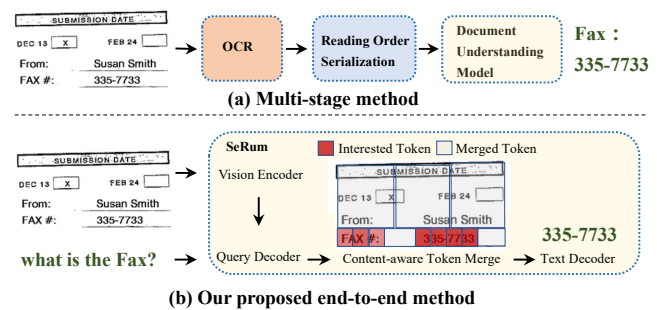


Figure 1. Comparison of multi-stage and **SeRum** end-to-end processing flows for visual document understanding. **SeRum**’s approach simplifies the pipeline by directly generating text output for *key visual tokens* from the image, eliminating the need for OCR and delivering highly efficient and effective document analysis. Best viewed in color.

tion (OCR) [42] and other modules [37] to extract key information, as shown in Figure 1. However, these approaches are suboptimal and computationally expensive, relying too much on prefacing modules such as accurate OCR recognition and document content ordering [18].

To address these challenges, we propose a novel end-to-end document understanding model with selective region concentration called **SeRum** (SElective Region Understanding Model). As shown in Figure 1, **SeRum** converts document image understanding and recognition tasks into a local decoding process of the visual tokens of interest, which includes a vision encoder, a query-text decoder, and a content-aware token merge module.

For document understanding tasks, the content to be extracted often takes up a small proportion of the whole document area but may change greatly in scale. Therefore, it is crucial to accurately identify the key area of interest first. **SeRum** extracts document image features using a vision Transformer-based encoder. We use a self-encoding Transformer query decoder inspired by MaskFormer [5], which decode the input query (question for tasks) and carries out

cross-attention mechanism with image features to form the embeddings of queries. Then, we obtain the area of interest mask through dot product with the up-sampled image features. Since the number of queries is greater than the number of text locations required, we use binary matching for pairing, following DETR [4].

The final sequence output is automatically generated by the text decoder through cross-attention with the encoded visual token. However, the presence of noise in long visual token sequence could adversely affect the decoding process. To address this issue, we propose a content-aware token merge mechanism that selects visual tokens associated with the query while merges the rest. This mechanism constrains attention to regions of interest generated by the query decoder, while simultaneously preserving global information and enhancing regional information of interest. The multi-query mechanism employed in our approach enables local generation of text, thereby resulting in shorter and more precise text content.

To further enhance the understanding and local awareness of the model, we design three pre-training tasks, including query to segmentation, text to segmentation and segmentation to text. In summary, we propose a novel end-to-end document understanding model called SeRum that improves the recognition ability of end-to-end models while achieving competitive results in word recognition. Our content-aware token merge mechanism limits the decoder’s attention to the local details of interest, improving the model’s effect and speeding up the decoding speed of the generative scheme. We believe that the SeRum model offers a valuable step towards efficient and effective end-to-end document understanding, with potential applications in various fields such as automatic document analysis, information extraction, text recognition and *etc.* Conclusively, our contributions are summarized into the three folds:

- We propose a novel end-to-end document understanding model called SeRum, which converts document image understanding and recognition tasks into a local decoding process of the interested visual tokens.
- We introduce a content-aware token merge mechanism that improves the model’s perception of image details and speeds up the decoding speed of the generative scheme.
- Experimental results on multiple public datasets show that our approach achieves state-of-the-art performance on document understanding tasks and competitive results on text spotting tasks.

## 2. Related Work

Document image understanding has been a significant topic in computer vision for many years and has at-

tracted considerable attention from the research community [6, 25]. Traditional rule-based methods [12, 40] and machine learning-based techniques [32, 39] often involve manual feature selection, which can be insufficient when dealing with complex layout documents. With the advent of deep learning technology, document information extraction methods have witnessed notable improvements in both effectiveness and robustness. In recent years, deep learning-based methods for document understanding can be categorized into three primary groups: OCR-dependent method with post-processing, OCR-dependent method without post-processing, and end-to-end method without OCR.

**OCR-dependent methods with post-processing** involve extracting textual information from document images using OCR engines and other auxiliary modules. In general, the OCR engine first detects and recognizes the text content of the document, then sorts the text according to the reading order, and finally marks each words in the way of sequence annotation through the document understanding model. In this way, the document understanding model focuses on the representation of document content and numerous significant works have arisen in this field. One such example is the LayoutLM family [45, 47, 46, 15], which employs multimodal networks to integrate visual and text features. GCN (Graph Convolution Network) is another technique utilized to model the intricate relationship between text and image [37, 26, 44, 48, 30, 31]. Additionally, Chargrid [21, 8] leverages layout information to extract entity relationships.

**OCR-dependent methods without post-processing** are designed to address the issue of token serialization errors in post-processing, and to mitigate OCR errors to a certain extent. For example, GMN [3] employs an optimally structured spatial encoder and a modality-attentive masking module to tackle documents of intricate structure which are hard to serialize. QGN [2] utilizes a query-based generation scheme to bolster the generation process even with OCR noise. Other works such as [38, 36, 41] encode the derived document information as a succession of tokens in XML format and output the XML tags delimiting the types of information.

**End-to-end methods without OCR** aim to further eliminate the dependence on the OCR recognition module, which exhibits faster reasoning speed and necessitates fewer parameters. These methods have recently gained attention due to their ability to achieve higher efficiency and effectiveness. For instance, Donut [22] and Dessurt [7] employ Swin Transformer [29] as an encoder for extracting image features, and BART-like Transformers [24] as a decoder for generating text sequences.

While the above solutions have addressed the document understanding dependency on OCR, they are all full graph encoding and global decoding. As such, they may not be

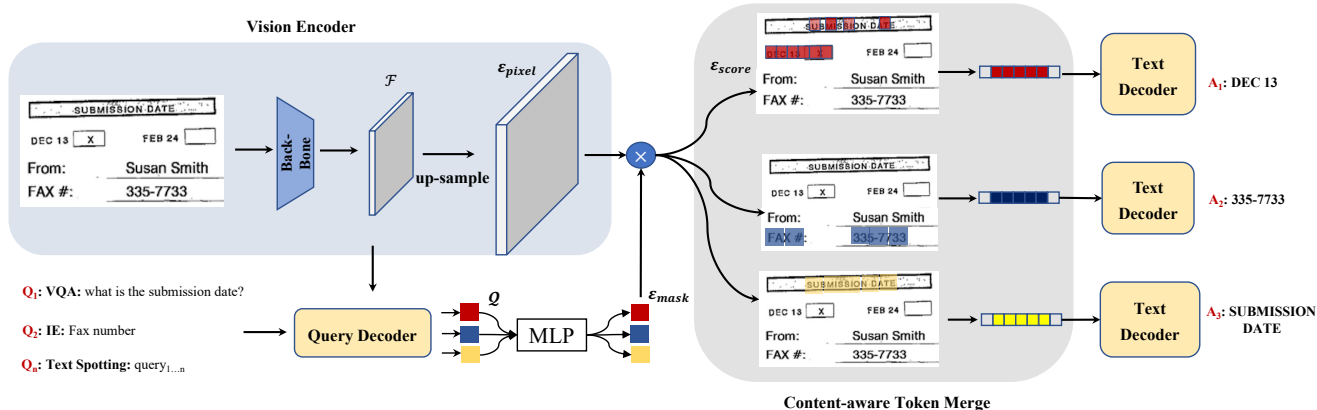


Figure 2. The architecture of our proposed SeRum. Given an input image and query, the vision encoder and query decoder encode the image and extract the region mask of interest, respectively. The resulting mask is then passed through the content-aware token merge module, which filters out the Top-K visual tokens of interest and merges the remaining ones. Finally, the resulting merged tokens are passed through the text decoder to obtain the document understanding result. Best viewed in color.

suitable for scenarios where OCR identification errors occur due to the inability to pay attention to the local details of the document, leading to low generation efficiency. The SeRum model proposed in this work offers a valuable step towards efficient and effective end-to-end document understanding and serves as a bridge for OCR recognition and document understanding.

## 3. Method

### 3.1. Overall Architecture

SeRum is an end-to-end architecture designed to tackle document understanding tasks by leveraging a process of decoding visual regions of interest. As shown in Figure 2, it consists of three main components: vision encoder, query-text decoder, and content-aware token merge modules. The vision encoder employs a visual transformers backbone to extract image features  $\mathcal{F}$  which are then upsampled to form high-resolution feature embeddings  $\mathcal{E}_{\text{pixel}}$ . The query decoder is a transformer decoder that encodes the input queries and attends to the image features. It produces  $N$  per-segment embeddings  $Q$  that independently generate  $N$  class predictions with  $N$  corresponding mask embeddings  $\mathcal{E}_{\text{mask}}$ . The model then predicts  $N$  possibly overlapping binary mask  $\mathcal{E}_{\text{score}}$  predictions via a dot product between vision embeddings  $\mathcal{E}_{\text{pixel}}$  and mask embeddings  $\mathcal{E}_{\text{mask}}$ , followed by a sigmoid activation. The content-aware token merge module merges irrelevant tokens, and via a text decoder, generates the final text token result. By using a transformer decoder to encode the input query, the model can effectively reason about the document’s semantic content and produce informative predictions. The content-aware token merge modules leverage the model’s attention to combine relevant tokens and improve the accuracy of the final predictions.

### 3.2. Vision Encoder

The vision encoder module plays a crucial role in converting the input document image  $x \in \mathbb{R}^{H \times W \times C}$  into a feature map  $\mathcal{F} \in \mathbb{R}^{h \times w \times d}$  that can be processed by subsequent modules. The feature map is further serialized into a set of embeddings  $\{z_i \mid z_i \in \mathbb{R}^d, 1 \leq i \leq n\}$  where  $n$  represents the feature map size or the number of image patches, and  $d$  is the dimension of the latent vectors of the encoder. CNN-based models [13] or Transformer-based models such as the Swin Transformer [29] can be used as the encoder network. For this study, we utilize the Swin Transformer with modifications as it exhibits superior performance in our preliminary study on document parsing. The Swin Transformer divides the input image  $x$  into non-overlapping patches, and applies Swin Transformer blocks consisting of a shifted window-based multi-head self-attention module and a two-layer MLP to these patches. Patch merging layers are subsequently applied to the patch tokens at each stage, and the output of the final Swin Transformer block  $z$  is fed into the query-text decoder. We also upsample the feature map to a larger size,  $\mathcal{E}_{\text{pixel}} \in \mathbb{R}^{sh \times sw \times d}$  for fine-grained mask area acquisition, where  $s$  is the upsampled factor and we set  $s = 6$  as default. In addition to the Swin Transformer architecture, we introduce a learnable position embedding in the last layer of the vision encoder module, which enhances the model’s perception of location. The position embedding is added to the upsampled feature map:  $\mathcal{E}'_{\text{pixel}} = \mathcal{E}_{\text{pixel}} + \mathcal{P}$ , where  $\mathcal{E}'_{\text{pixel}}$  is the upsampled feature map with the position embedding,  $\mathcal{E}_{\text{pixel}}$  is the original upsampled feature map, and  $\mathcal{P}$  is the learnable position embedding.

### 3.3. Query-Text Decoder

The Query-Text decoder is an essential module in the field of visual contextual analysis. It comprises two sub-

modules, namely the query decoder and text decoder, which operate with shared weights.

**Query Decoder** utilizes the standard Transformer [43] to compute the per-segment embeddings  $\mathcal{Q} \in \mathbb{R}^{C_{\mathcal{Q}} \times N}$ ,  $\mathcal{E}_{\text{mask}} \in \mathbb{R}^{N \times d}$  encoding global information about each segment prediction from the image features  $\mathcal{F}$  and  $N$  learnable token embeddings or questions. Following the approach of [5], the decoder produces all predictions simultaneously.

**Text Decoder** is designed to decode the corresponding text token sequences for a variety of tasks, including Visual Question Answering (VQA), information extraction and text spotting. It generates text in an auto-regressive manner by attending to the query embedding, previously generated tokens, and the encoded visual features. The mathematical form of the text decoder is as follows:

$$\mathbf{h}_t = \text{Decoder}(\mathbf{h}_{t-1}, \mathcal{F}, \mathcal{Q}, \mathcal{T}_{t-1}) \quad (1)$$

Here,  $\mathbf{h}_t$  represents the hidden state of the decoder at time step  $t$ ,  $\mathcal{Q}$  represents the query embedding,  $\mathcal{T}_{t-1}$  represents the previous token embedding, and  $\mathcal{F}$  represents the visual feature map. In contrast, the query decoder is a self-encoding form, where each query attends to all other queries. The mathematical form is as follows:

$$\mathbf{h}_i = \text{Decoder}(\mathcal{F}, \mathcal{Q}_i, \mathcal{Q}_j \mid j \neq i) \quad (2)$$

Here,  $\mathbf{h}_i$  represents the hidden state of the decoder for query  $i$ , and  $\mathcal{Q}_j \mid j \neq i$  is the set of all other queries.

### 3.4. Content-aware Token Merge

Document visual understanding tasks differ from conventional visual tasks in that a amount of information is usually concentrated in a very small area, commonly comprising less than 5% of the entire image. Therefore, the visual tokens extracted by the vision module contain a large number of background areas, which increase the difficulty of decoding and reduce the decoding speed. To address this issue, we introduce a module called content-aware token merge that dynamically focuses on the more relevant foreground part of the visual tokens and reduces the focus on the background part of the tokens.

**Foreground Area:** As mentioned above, the correlation between query and visual tokens can be represented by  $\mathcal{E}_{\text{score}} \in \mathbb{R}^{N \times sh \times sw}$ . The higher the score, the stronger the correlation information of the visual token representing the location. We sort the  $\mathcal{E}_{\text{score}}$  and select the features corresponding to the top-K tokens with high scores to form a *foreground area sequence* for each query.  $\mathcal{E}_{\text{score}}^l$  denotes the average score associated with token  $l$ .

$$\mathbb{F}_f = [\mathbf{z}_l * \mathcal{E}_{\text{score}}^l \mid l = 1, \dots, K] \quad (3)$$

**Background Area:** The foreground area features represent the visual features of absolute dominance, while the

rest of the features are considered background area features. These features are typically global features or features used to assist in extrapolation. To preserve this part of the features and avoid interference with the foreground features caused by excessive features of this part, In our case, we project the background tokens using the basic attention mechanism onto the foreground tokens, resulting in a K-dimensional feature representation that captures the important information from both the foreground and background tokens. The formal formula for the attention mechanism is as follows:

$$\mathbb{F}_b = \text{softmax}(\mathbf{Q}\mathbf{f}_r^\top)(\mathbf{W}^v\mathbf{f}_r^\top)^\top \quad (4)$$

where  $\mathbb{F}_b \in \mathbb{R}^{K \times d}$  are merged background features,  $\mathbf{Q} \in \mathbb{R}^{K \times d}$  are foreground query features,  $\mathbf{f}_r \in \mathbb{R}^{(L-K) \times d}$  are primitive background features, and  $\mathbf{W}^v \in \mathbb{R}^{d \times d}$  is the projection weight. The softmax function is employed to normalize the dot product of the query and key matrices. Subsequently, the resulting weights are applied to the value matrix to derive the output. This process yields a new feature vector  $\mathbb{F}_b$ , which encapsulates the background contextual information. The visual context is represented by  $\mathbb{F} = \mathbb{F}_f + \mathbb{F}_b$ .

To adapt to varying size changes, we define the number of foreground tokens, denoted by  $K$ , as a function of the total number of tokens, represented by  $L$ , i.e.,  $K = \alpha L$ . During training, we sample  $\alpha$  from a uniform distribution that ranges from 0.02 to 1.0. Specifically,  $\alpha = 1.0$  indicates that no token merge occurs. During inference, the value of  $\alpha$  can be fixed based on the performance requirements.

### 3.5. Pre-training

In this study, we utilize multi-task pre-training to enhance the model’s position understanding and text generation capabilities. The pre-training process involves three subtasks: query to segmentation, text to segmentation, and segmentation to text.

**Query to Segmentation** aims to equip the model with text detection capability. We adopt the query generation approach used in DETR [4] but apply it to an instance segmentation task instead, as text generation is a highly dense prediction task. We utilize the vision encoder and query decoder to process a given image  $\mathbf{x}$  with  $N$  learnable token embeddings, resulting in  $N$  mask predictions  $\mathcal{E}_{\text{score}}$  for the text area, and we set  $N = 50$  as the default.

**Text to Segmentation** helps the model to comprehend the distinct positions of each text element, thus playing a crucial role in text-segmentation alignment. We feed the text snippet of the image to query decoder and turn this into an instance segmentation task as well. Since document images often contain a relatively low proportion of pixels with text, emphasizing the text region segmentation performance can enhance the quality of the decoder generated output.

**Segmentation to Text** aims to generate text from image segmentations in a manner similar to OCR. During this stage, the foreground and background features  $\mathbb{F}$  generated from the preceding layers is utilized as the visual context within the cross-attention mechanism of the text decoder, which auto-regressively generates tokens.

It is important to note that the three pre-training subtasks are performed simultaneously and with no particular hierarchical order. In other words, the training process includes data from all three subtasks in the same batch.

### 3.6. Training Strategy

**Loss Function.** SeRum is trained end-to-end with a loss function consisting of multiple parts. The first part is the Hungarian matching loss between queries and targets, which is used to handle the case where the number of queries is greater than the number of targets. Specifically, we conduct position matching at each layer of the query encoder and position and category matching at the last layer. The loss for this part is defined as:

$$\mathcal{L}_{match} = \sum_{i=1}^{N_e} L_{match}^{(i)} \quad (5)$$

where  $N_e$  is the number of query encoder layers, and  $L_{match}^{(i)}$  is the matching loss at layer  $i$ .

The second part of the loss is the autoregressive decoder loss for the text decoder, which is defined as:

$$\mathcal{L}_{decoder} = -\log P(y_{1:T} | \mathbf{h}_{1:T}, \mathbf{e}_1) \quad (6)$$

where  $y_{1:T}$  is the ground-truth text,  $\mathbf{h}_{1:T}$  is the hidden states of the decoder, and  $\mathbf{e}_1$  is the embedding of the start-of-sequence token.

To speed up the learning of areas of interest, we add a text constraint loss to the features of the image up-sampled above, where the mask of the loss constraint is changed to cover all text areas in the whole image. The loss for this part is defined as:

$$\mathcal{L}_{text} = \frac{1}{|\Omega|} \sum_{(i,j) \in \Omega} L_{text}(\mathcal{E}_{score}^{i,j}) \quad (7)$$

where  $\Omega$  is the set of all text areas in the image,  $L_{text}$  is the text constraint loss, and  $\mathcal{E}_{score}^{i,j}$  is the segmentation mask at position  $(i, j)$ .

The total loss is a weighted sum of the above loss parts:

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{match} + \lambda_2 \mathcal{L}_{decoder} + \lambda_3 \mathcal{L}_{text} \quad (8)$$

where  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are hyper-parameters controlling the weights of each loss part.

**More Details.** During pre-training, SeRum employs a binary matching mode between query and target and has

explicit supervision in the area of interest mask. However, during downstream task training, the model is constrained by the proportion of the area of interest, leading to an implicit supervision mode.

## 4. Experiments

We evaluate the proposed model on three tasks: document information extraction, document visual question answering, and text spotting. To assess the model’s performance, we compare it with state-of-the-art two-stage document information extraction methods as well as other end-to-end methods.

### 4.1. Tasks and Datasets

**Document Information Extraction (DIE)** is a process of extracting key-value pairs structured data from documents. We evaluate our model’s performance on three benchmark datasets commonly used to validate the efficiency of the models in DIE.

**Ticket** dataset [10] includes 300,000 composite images and 1,900 real images of Chinese train tickets. There are eight entities, such as start station, terminal station, time, price, etc., that need to be extracted.

**CORD** dataset [35] is a widely-used English benchmark for information extraction tasks. It comprises 800 training, 100 validation, and 100 test receipt images, with 30 distinct subclasses for extraction, such as menu name, menu num, menu price, etc. CORD is complex due to its multi-layered, nested structure, which requires the model not only to extract the relevant text but also to gain a deeper understanding of its structure.

**SROIE** dataset [16] consists of 973 scanned receipt images, divided into 626 for training and 347 for testing purposes. Its composition is simplistic, encompassing solely four key entities: company, total, date, and address. Nevertheless, due to the dense textual content present in the images, extracting relevant information is complicated, as some entities are spread across multiple lines.

We utilize two common metrics, F1 score [14, 47] and Tree Edit Distance (TED) [22, 50], to demonstrate the model’s performance on the datasets. F1 score is a harmonic mean of precision and recall of a classification model, where precision measures the ability of the model to correctly identify positive cases, and recall measures the ability of the model to identify all positive cases. A high F1 score indicates that the model has high accuracy and precision in classifying positive cases. Although F1 score is very intuitive, it is relatively strict and cannot accurately reflect the prediction accuracy at the character level. The TED is the minimum number of single-character editing operations required to convert one string to another. We calculate its score through  $\max(0, 1 - TED(pr, gt) / TED(\phi, gt))$ ,

where  $gt$ ,  $pr$ , and  $\phi$  stand for ground truth, predicted, and empty string.

**DocVQA** is a task that combines document understanding and visual question answering. The goal is to answer a question about a given document image. We evaluate our model on the DocVQA dataset [34], which contains 12,767 document images and 99,000 questions in total. The dataset is split into training, validation, and test sets, with 80%, 10%, and 10% of the data in each set, respectively. The questions are of various types, such as yes/no, counting, reasoning, and comparison, and require both text understanding and visual reasoning. The ANLS (Average Normalized Levenshtein Similarity) metric, which is an edit-distance-based metric, is used for evaluation.

## 4.2. Implementation Details

**Model Architecture.** Our proposed model, SeRum, utilizes Swin-B [29] as the visual encoder with slight modifications. Specifically, we set the layer numbers and window size as 2, 2, 14, 2 and 10 to extract features from the input image. Moreover, the input image resolution is set to  $1280 \times 960$ . To balance the trade-off between speed and accuracy, we use the first four layers of mBART as the decoder to generate the output text.

**Pre-training.** To improve the model’s performance, we pretrain SeRum on a large-scale synthetic dataset generated using the Synthetic Text Recognition (SynthText) [11], Synth90K [20], IIT-CDIP [23] datasets and multi-language synthetic dataset following Donut. The SynthText dataset is used to generate text instances with complex backgrounds, while the Synth90K dataset is used for text instances with simple backgrounds. The IIT-CDIP dataset comprises over 11M scanned images of English language documents. We use the multi-task pre-training strategy, as described in Section 3.5.

**Fine-tuning.** For the fine-tuning stage, we utilize the Adam optimizer with an initial learning rate of  $5 \times 10^{-5}$  for all datasets, and incorporate a learning rate decay factor of 0.1 after every 30 epochs to enhance the model’s performance. For the information extraction tasks, we set the batch size to 24, while for the DocVQA task, we set it to 8. To ensure optimal performance and convergence, we train the model for a maximum of 300 epochs. Additionally, we employ two generation mechanisms, SeRum-total and SeRum-prompt, for information extraction tasks. SeRum-total generates a complete token sequence of all key information using a predefined format, such as Donut, with the task name as queries. SeRum-Prompt uses the keys as queries and generates each information in parallel. We set the token keep ratio  $\alpha$  to 0.5 for SeRum-total generation mechanism and  $\alpha$  to 0.1 for SeRum-Prompt generation mechanism during test stage. Finally, to balance the loss function weights, we set  $\lambda_1 = \lambda_2 = \lambda_3$ .

## 4.3. Comparisons with Previous Approaches

**Document Information Extraction.** We present a comparative evaluation of our proposed method with several state-of-the-art approaches reported in recent years on three widely-used datasets, namely Ticket, CORD, and SROIE. Our comparison includes approaches that employ OCR as well as fully end-to-end methods. In the former, OCR is used to extract the text and position of the image, and the information is sorted and classified at the token level, after which the target information is determined based on the classification results. Approaches such as BERT [18], LayoutLMv2 [47], and SPADE [19] can produce satisfactory results when the OCR results are entirely accurate. We use the OCR engine API reported in Donut, including MS OCR and others, for consistency. In addition, we evaluate a generative method called WYVERN [17] that utilizes the encoder and decoder architecture of Transformer and requires OCR. End-to-end methods, which have a streamlined pipeline, are increasingly popular in both industry and academia. Donut [22] and Dessurt [7] are two prominent end-to-end methods, and we only study the results of Donut in this paper due to the underperformance of Dessurt.

Our proposed method achieves new state-of-the-art results on the three open benchmarks of document information extraction, as shown in Table 1. Our model excels in the Ticket dataset, achieving a score of over 99% and demonstrating near-complete success in addressing this task, outperforming the second best end-to-end method Donut by 5%. Moreover, our model exhibits robust character recognition and context understanding capabilities, as shown in Figure 3. Notably, SeRum is an end-to-end method that does not require an OCR module, making it more efficient for training and inference, and therefore suitable for industrial applications. In particular, on the SROIE dataset, the F1 score of SeRum exceeds that of the multi-stage method LayoutLMv2 by 24%.

**DocVQA.** We conduct an evaluation of our proposed SeRum model on the challenging DocVQA dataset and compare its performance against several state-of-the-art methods. The experimental results are summarized in Table 1. Our SeRum model demonstrates superior performance over the strong baseline of end-to-end methods and achieves competitive results with the multi-stage methods. In particular, in the ANLS\* dataset with handwriting text, end-to-end methods exhibit superiority over multi-stage methods due to their ability to jointly optimize text detection and recognition. We conduct further studies on this characteristic in the further discussion section and appendix.

## 4.4. Ablation Study

In the ablation experiments, we conduct a thorough analysis of the effectiveness of each of our contributions, including pre-training, prompt generation, and content-aware

	OCR	#Params	CORD [35]		Ticket [10]		SROIE [16]		DocVQA [34]	
			F1	Acc.	F1	Acc.	F1	Acc.	ANLS	ANLS*
SPADE [19]	✓	93M + $\alpha^\ddagger$	74.0	75.8	14.9	29.4	-	-	-	-
WYVERN [17]	✓	106M + $\alpha^\ddagger$	43.3	46.9	41.8	54.8	-	-	-	-
BERT [18]	✓	86M + $\alpha^\ddagger$	73.0	65.5	74.3	82.4	-	-	63.5	-
LayoutLMv2 [47]	✓	179M + $\alpha^\ddagger$	78.9	82.4	87.2	90.1	61.0	91.1	<b>78.1</b>	67.3
Donut [22]		143M	84.1	90.9	94.1	98.7	83.2	92.8	67.5	72.1
<b>SeRum-total</b>		136M	80.5	85.8	97.9	99.6	85.6	92.8	-	-
<b>SeRum-prompt</b>		136M	<b>84.9</b>	<b>91.5</b>	<b>99.2</b>	<b>99.8</b>	<b>85.8</b>	<b>95.4</b>	71.9	<b>77.9</b>

Table 1. **Performances on various document understanding tasks.** The field-level F1 scores and tree-edit-distance-based accuracies are reported. **SeRum-prompt** shows competitive in different tasks. Parameters for vocabulary are omitted for fair comparisons among multi-lingual models.  $^\ddagger$ # parameters represents that OCR needs to be used.

method	F1
query to segmentation	59.3
+ text to segmentation	82.5
+ segmentation to text	85.8

Table 2. Effect of pre-training method on the SROIE dataset.

Token keep ratio $\alpha$	F1	Text decoder latency(ms)
2%	72.5	194
5%	83.2	198
10%	85.8	209
20%	84.8	225
30%	84.9	231
50%	84.9	234
100%	84.9	306

Table 3. Effect of token keep ratio on the SROIE dataset. The latency measures on a T4 GPU.

token merge module.

**Impact of Pre-training.** The role of pre-training in augmenting the performance of large-scale models is widely recognized in the research community. In this paper, we investigate the effect of three pre-training tasks on the model’s performance in the context of the SROIE dataset. We present the experimental results in Table 2, which demonstrate the efficacy of pre-training in enhancing the information extraction performance of the model.

**Impact of Generation Manners.** As an end-to-end generative model, SeRum enables complete extraction of essential information by generating a string sequence that includes all relevant keys, referred to as SeRum-total. To achieve this, we utilize a Donut-like method [22] to serialize a json style string, with each key-value pair represented as  $\langle s\_key \rangle value \langle e\_key \rangle$ , where both  $\langle s\_key \rangle$  and  $\langle e\_key \rangle$  are added to the tokenizer as special tokens. The complete json is encoded into a single sequence, which is then decoded to extract all the necessary information in one pass.

Table 1 displays the performance differences between SeRum-total and SeRum-prompt in the information extraction task. Our results indicate that the one-pass extraction method of SeRum-total shows varying degrees of degradation compared to the prompt-based approach, particularly on the CORD dataset. This degradation may be attributed to the fact that our pre-training is a pseudo-OCR task, limiting the model’s ability to comprehend complex structures. Nonetheless, parsing the CORD dataset into a dictionary query leads to a significant enhancement of the model’s performance. Moreover, the substantial improvement observed in the Ticket and SROIE datasets further validates the effective-

ness of prompt generation.

**Impact of Content-aware Token Merge.** Table 3 presents the impact of the content-aware token merge mechanism on the recognition accuracy and decoding speed of SeRum. We observe that with an increasing token keep ratio, the recognition accuracy of SeRum improves gradually, peaking at a merge ratio of 10%. However, surpassing this ratio causes a decline in recognition accuracy. Regarding decoding speed, as the token keep ratio decreases, the decoding speed of SeRum improves gradually. These findings suggest that the content-aware token merge mechanism in SeRum is capable of enhancing both recognition accuracy and decoding speed. Furthermore, the ratio can be dynamically adjusted to achieve varying speeds during inference.

#### 4.5. Further discussion

**Case Study.** In order to further investigate the effectiveness and problems of the scheme, we further analyzed the differences with the SOTA method on multiple data sets, as well as the accuracy of the selected region mask. As shown in Figure 3, the benefit of SeRum model is a higher accuracy of word recognition by focusing on important relevant details. Especially in some confusing scenes, it is often better to combine the contextual information of the text. See the appendix for more details and cases.

**Generalization to Other Tasks.** SeRum’s ability to



	SROIE: date	CORD: menu.nm	Ticket: start station	DocVQA: What is the title of the document?
Image				
Text Mask				
Selective Region Mask				
Result of SeRum	05/02/2018	TICKET CP	肇庆东站	FINANCIAL HIGHLIGHTS

Figure 3. Visualization of SeRum on various datasets. The figure depicts five row from top to bottom, showcasing the task and query, original image, text regions of pre-training model, selected region of interest obtained by the query, and corresponding text result SeRum extracted. The results showed that SeRum effectively captured the regions of interest and decoded them correctly. Best viewed in color.



Figure 4. Results of text spotting. SeRum demonstrates significant advantages in handling documents with complex formats, including WordArt, curved text and etc. Best viewed in color.

generalize to other tasks was assessed by testing its performance on text spotting, as illustrated in Figure 4. Text spotting involves detecting and recognizing text in natural images, such as street views or signs. In our evaluation, we used the widely-used CTW-1500 [49] dataset, and the experimental results are summarized in Table 4. The experimental results show that SeRum attains competitive performance across multiple tasks, emphasizing its efficacy in diverse applications. We suggest that this approach could be promising for various document understanding tasks.

method	F1
TextDragon [9]	39.7
ABCNet [27]	45.2
SPTS v2 [28]	<b>63.6</b>
SeRum	41.8

Table 4. Performances on CTW-1500 text spotting datasets.

## 5. Conclusion

In conclusion, we have presented a novel query-based approach for image-based document understanding that does not rely on OCR, but instead uses a visual encoder and a text decoder. Our approach encodes the prompt and extracts the relevant token regions through a context-aware token merge mechanism between the prompt and image features, and with a decoder to generate the local document understanding results.

Our method achieves competitive performance on several public datasets for document information extraction and text spotting, and is almost comparable to the two-stage solution while being more efficient and enabling parallel generation based on the prompt. The proposed model has the potential to be applied in real-world scenarios where traditional OCR-based approaches are not feasible or efficient.



## References

- [1] Bernhard Axmann and Harmoko Harmoko. Robotic process automation: An overview and comparison to other technology in industry 4.0. In *10th International Conference on Advanced Computer Information Technologies, ACIT 2020, Deggendorf, Germany, September 16-18, 2020*, pages 559–562. IEEE, 2020.
- [2] Haoyu Cao, Xin Li, Jiefeng Ma, Deqiang Jiang, Antai Guo, Yiqing Hu, Hao Liu, Yinsong Liu, and Bo Ren. Query-driven generative network for document information extraction in the wild. In João Magalhães, Alberto Del Bimbo, Shin’ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni, editors, *MM ’22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 4261–4271. ACM, 2022.
- [3] Haoyu Cao, Jiefeng Ma, Antai Guo, Yiqing Hu, Hao Liu, Deqiang Jiang, Yinsong Liu, and Bo Ren. GMN: generative multi-modal network for practical document information extraction. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3768–3778. Association for Computational Linguistics, 2022.
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2020.
- [5] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. In Marc’Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17864–17875, 2021.
- [6] Lei Cui, Yiheng Xu, Tengchao Lv, and Furu Wei. Document AI: benchmarks, models and applications. *CoRR*, abs/2111.08609, 2021.
- [7] Brian L. Davis, Bryan S. Morse, Brian L. Price, Chris Tensmeyer, Curtis Wigington, and Vlad I. Morariu. End-to-end document recognition and understanding with dessert. In Leonid Karlinsky, Tomer Michaeli, and Ko Nishino, editors, *Computer Vision - ECCV 2022 Workshops - Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part IV*, volume 13804 of *Lecture Notes in Computer Science*, pages 280–296. Springer, 2022.
- [8] Timo I. Denk and Christian Reisswig. Bertgrid: Contextualized embedding for 2d document representation and understanding. *CoRR*, abs/1909.04948, 2019.
- [9] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 9075–9084. IEEE, 2019.
- [10] He Guo, Xiameng Qin, Jiaming Liu, Junyu Han, Jingtuo Liu, and Errui Ding. EATEN: entity-aware attention for single shot visual text extraction. In *2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019*, pages 254–259. IEEE, 2019.
- [11] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2315–2324. IEEE Computer Society, 2016.
- [12] Jaekyu Ha, R.M. Haralick, and I.T. Phillips. Recursive x-y cut using bounding boxes of connected components. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 952–955 vol.2, 1995.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016.
- [14] Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. BROS: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelfth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10767–10775. AAAI Press, 2022.
- [15] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document AI with unified text and image masking. In João Magalhães, Alberto Del Bimbo, Shin’ichi Satoh, Nicu Sebe, Xavier Alameda-Pineda, Qin Jin, Vincent Oria, and Laura Toni, editors, *MM ’22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 4083–4091. ACM, 2022.
- [16] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and C. V. Jawahar. ICDAR2019 competition on scanned receipt OCR and information extraction. *CoRR*, abs/2103.10213, 2021.
- [17] Alyssa Hwang, William R. Frey, and Kathleen R. McKeown. Towards augmenting lexical resources for slang and african american english. In Marcos Zampieri, Preslav Nakov, Nikola Ljubesic, Jörg Tiedemann, and Yves Scherrer, editors, *Proceedings of the 7th Workshop on NLP for Similar Languages, Varieties and Dialects, VarDial@COLING 2020, Barcelona, Spain (Online), December 13, 2020*, pages 160–172. International Committee on Computational Linguistics (ICCL), 2020.
- [18] Wonseok Hwang, Seonghyeon Kim, Minjoon Seo, Jinyeong Yim, Seunghyun Park, Sungrae Park, Junyeop Lee, Bado Lee, and Hwalsuk Lee. Post-ocr parsing: building simple

- and robust parser via bio tagging. In *Workshop on Document Intelligence at NeurIPS 2019*, 2019.
- [19] Wonseok Hwang, Jinyeong Yim, Seunghyun Park, Sohee Yang, and Minjoon Seo. Spatial dependency parsing for semi-structured document information extraction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 330–343. Association for Computational Linguistics, 2021.
- [20] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *CoRR*, abs/1406.2227, 2014.
- [21] Anoop R. Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. Chargrid: Towards understanding 2d documents. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun’ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4459–4469. Association for Computational Linguistics, 2018.
- [22] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoon Yun, Dongyoon Han, and Seunghyun Park. Ocr-free document understanding transformer. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXVIII*, volume 13688 of *Lecture Notes in Computer Science*, pages 498–517. Springer, 2022.
- [23] David D. Lewis, Gady Agam, Shlomo Argamon, Ophir Frieder, David A. Grossman, and Jefferson Heard. Building a test collection for complex document information processing. In Efthimis N. Efthimiadis, Susan T. Dumais, David Hawking, and Kalervo Järvelin, editors, *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, August 6-11, 2006*, pages 665–666. ACM, 2006.
- [24] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics, 2020.
- [25] Xin Li, Yan Zheng, Yiqing Hu, Haoyu Cao, Yunfei Wu, Deqiang Jiang, Yinsong Liu, and Bo Ren. Relational representation learning in visually-rich documents. In *Proceedings of the 30th ACM International Conference on Multimedia, MM ’22*, page 4614–4624, New York, NY, USA, 2022. Association for Computing Machinery.
- [26] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. Graph convolution for multimodal information extraction from visually rich documents. In Anastassia Loukina, Michelle Morales, and Rohit Kumar, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 2 (Industry Papers)*, pages 32–39. Association for Computational Linguistics, 2019.
- [27] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 9806–9815. Computer Vision Foundation / IEEE, 2020.
- [28] Yuliang Liu, Jiaxin Zhang, Dezhi Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, et al. Spts v2: single-point scene text spotting. *arXiv preprint arXiv:2301.01635*, 2023.
- [29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 9992–10002. IEEE, 2021.
- [30] Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. Openceres: When open information extraction meets the semi-structured web. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3047–3056. Association for Computational Linguistics, 2019.
- [31] Colin Lockard, Prashant Shiralkar, Xin Luna Dong, and Hannaneh Hajishirzi. Zeroshotceres: Zero-shot relation extraction from semi-structured webpages. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8105–8117. Association for Computational Linguistics, 2020.
- [32] Simone Marinai, Marco Gori, and Giovanni Soda. Artificial neural networks for document analysis and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(1):23–35, 2005.
- [33] Yosi Mass and Haggai Roitman. Ad-hoc document retrieval using weak-supervision with BERT and GPT2. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 4191–4197. Association for Computational Linguistics, 2020.
- [34] Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for VQA on document images. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pages 2199–2208. IEEE, 2021.

- [35] Seunghyun Park, Seung Shin, Bado Lee, Junyeop Lee, Jaehung Surh, Minjoon Seo, and Hwalsuk Lee. Cord: A consolidated receipt dataset for post-ocr parsing. 2019.
- [36] Rafal Powalski, Lukasz Borchmann, Dawid Jurkiewicz, Tomasz Dwojak, Michal Pietruszka, and Gabriela Palka. Going full-tilt boogie on document understanding with text-image-layout transformer. In Josep Lladós, Daniel Lopresti, and Seiichi Uchida, editors, *16th International Conference on Document Analysis and Recognition, ICDAR 2021, Lausanne, Switzerland, September 5-10, 2021, Proceedings, Part II*, volume 12822 of *Lecture Notes in Computer Science*, pages 732–747. Springer, 2021.
- [37] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. Graphie: A graph-based framework for information extraction. In Jill Burstein, Christy Doran, and Tamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 751–761. Association for Computational Linguistics, 2019.
- [38] Clément Sage, Alex Aussem, Véronique Eglin, Haytham Elghazel, and Jérémy Espinas. End-to-end extraction of structured information from business documents with pointer-generator networks. In Priyanka Agrawal, Zornitsa Kozareva, Julia Kreutzer, Gerasimos Lampouras, André F. T. Martins, Sujith Ravi, and Andreas Vlachos, editors, *Proceedings of the Fourth Workshop on Structured Prediction for NLP@EMNLP 2020, Online, November 20, 2020*, pages 43–52. Association for Computational Linguistics, 2020.
- [39] M. Shilman, P. Liang, and P. Viola. Learning nongenerative grammatical models for document analysis. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 962–969 Vol. 2, 2005.
- [40] Anikó Simon, Jean-Christophe Pret, and A. Peter Johnson. A fast algorithm for bottom-up document layout analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(3):273–277, 1997.
- [41] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. *CoRR*, abs/2212.02623, 2022.
- [42] Daniel van Strien, Kaspar Beelen, Mariona Coll Ardanuy, Kasra Hosseini, Barbara McGillivray, and Giovanni Colavizza. Assessing the impact of OCR quality on downstream NLP tasks. In Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik, editors, *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 1, Valletta, Malta, February 22-24, 2020*, pages 484–496. SCITEPRESS, 2020.
- [43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.
- [44] Mengxi Wei, Yifan He, and Qiong Zhang. Robust layout-aware IE for visually rich documents with pre-trained language models. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2367–2376. ACM, 2020.
- [45] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In Rajesh Gupta, Yan Liu, Jiliang Tang, and B. Aditya Prakash, editors, *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM, 2020.
- [46] Yiheng Xu, Tengchao Lv, Lei Cui, Guoxin Wang, Yijuan Lu, Dinei Florêncio, Cha Zhang, and Furu Wei. Layoutlm: Multimodal pre-training for multilingual visually-rich document understanding. *CoRR*, abs/2104.08836, 2021.
- [47] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei A. F. Florêncio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2579–2591. Association for Computational Linguistics, 2021.
- [48] Wenwen Yu, Ning Lu, Xianbiao Qi, Ping Gong, and Rong Xiao. PICK: processing key information extraction from documents using improved graph learning-convolutional networks. In *25th International Conference on Pattern Recognition, ICPR 2020, Virtual Event / Milan, Italy, January 10-15, 2021*, pages 4363–4370. IEEE, 2020.
- [49] Tai-Ling Yuan, Zhe Zhu, Kun Xu, Cheng-Jun Li, Tai-Jiang Mu, and Shi-Min Hu. A large chinese text dataset in the wild. *J. Comput. Sci. Technol.*, 34(3):509–521, 2019.
- [50] Kaizhong Zhang and Dennis E. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Comput.*, 18(6):1245–1262, 1989.
- [51] Peng Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. TRIE: end-to-end text reading and information extraction for document understanding. In Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann, editors, *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020*, pages 1413–1422. ACM, 2020.