Tag, Copy or Predict: A Unified Weakly-Supervised Learning Framework for Visual Information Extraction using Sequences

Jiapeng Wang[1] , **Tianwei Wang**[1] , **Guozhi Tang**[1] , **Lianwen Jin**[1,3*] , **Weihong Ma**[1] ,
Kai Ding[2] and **Yichao Huang**[2]
[1]School of Electronic and Information Engineering, South China University of Technology, China
[2]IntSig Information Co., Ltd, Shanghai, China
[3]Guangdong Artificial Intelligence and Digital Economy Laboratory (Pazhou Lab), Guangzhou, China
scutjpwang@foxmail.com, wangtw@foxmail.com, eetanggz@mail.scut.edu.cn, eelwjin@scut.edu.cn,
eeweihong_ma@mail.scut.edu.cn, danny_ding@intsig.net, charlie_huang@intsig.net

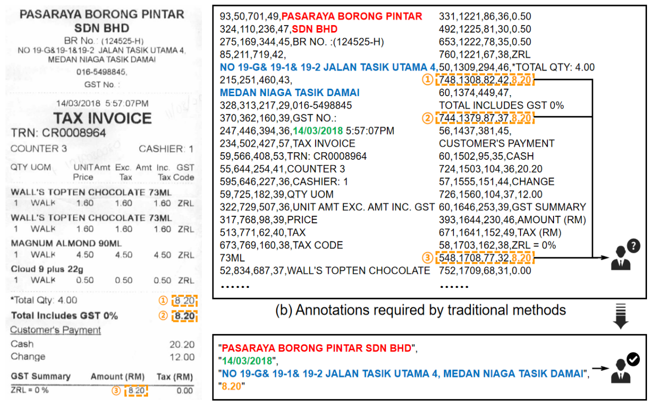
Abstract

Visual information extraction (VIE) has attracted increasing attention in recent years. The existing methods usually first organized optical character recognition (OCR) results into plain texts and then utilized token-level entity annotations as supervision to train a sequence tagging model. However, it expends great annotation costs and may be exposed to label confusion, and the OCR errors will also significantly affect the final performance. In this paper, we propose a unified weakly-supervised learning framework called TCPN (Tag, Copy or Predict Network), which introduces 1) an efficient encoder to simultaneously model the semantic and layout information in 2D OCR results; 2) a weakly-supervised training strategy that utilizes only key information sequences as supervision; and 3) a flexible and switchable decoder which contains two inference modes: one (*Copy or Predict* Mode) is to output key information sequences of different categories by copying a token from the input or predicting one in each time step, and the other (*Tag* Mode) is to directly tag the input sequence in a single forward pass. Our method shows new state-of-the-art performance on several public benchmarks, which fully proves its effectiveness.

1 Introduction

With the fast development of information interaction, document intelligent processing [Ferilli *et al.*, 2011] has attracted considerable attention. As an important part of it, visual information extraction (VIE) technique has been integrated into many real-world applications.

The existing VIE methods usually first organized text blocks (text bounding boxes and strings, which were provided by the ground truth or parsed by an OCR system) into plain texts according to the reading order and utilized effective encoding structures such as [Katti *et al.*, 2018; Liu *et al.*, 2019a; Xu *et al.*, 2020] to extract the most distinguishable representations for each input token from multi-sources. After this, a sequence tagging model like [Lample *et al.*, 2016] was trained with token-level category supervision.


PASARAYA BORONG PINTAR
SDN BHD
BR No.: (124525-H)
NO 19-G&19-1&19-2 JALAN TASIK UTAMA 4,
MEDAN NIAGA TASIK DAMAI
016-5498845,
GST No.:

14/03/2018 5:57:07PM
TAX INVOICE
TRN: CR0008964
COUNTER 3 CASHIER: 1

QTY	UOM	UNIT Price	Amt Exc. Tax	Amt Inc. Tax	GST Code
WALL'S TOPTEN CHOCOLATE 73ML					
1	WALK	1.60	1.60	1.60	ZRL
WALL'S TOPTEN CHOCOLATE 73ML					
1	WALK	1.60	1.60	1.60	ZRL
MAGNUM ALMOND 90ML					
1	WALK	4.50	4.50	4.50	ZRL
Cloud 9 plus 22g					
1	WALK	0.50	0.50	0.50	ZRL

*Total Qty: 4.00 ① 8.20
Total Includes GST 0% ② **8.20**
Customer's Payment
Cash 20.20
Change 12.00

GST Summary	Amount (RM)	Tax (RM)
ZRL = 0 %	③ 8.20	0.00

(a) A document image

```
93,50,701,49,PASARAYA BORONG PINTAR        331,1221,86,36,0.50
324,110,236,47,SDN BHD                     492,1225,81,30,0.50
275,169,344,45,BR NO. :(124525-H)          653,1222,78,35,0.50
85,211,719,42,                             760,1221,67,38,ZRL
NO 19-G& 19-1& 19-2 JALAN TASIK UTAMA 4,   50,1309,294,46,*TOTAL QTY: 4.00
215,251,460,43,                          ① 748,1308,82,42,8.20
MEDAN NIAGA TASIK DAMAI                    60,1374,449,47,
328,313,217,29,016-5498845                 TOTAL INCLUDES GST 0%
370,362,160,39,GST NO.:                  ② 744,1379,87,37,8.20
247,446,394,36,14/03/2018 5:57:07PM        56,1437,381,45,
234,502,427,57,TAX INVOICE                 CUSTOMER'S PAYMENT
59,566,408,53,TRN: CR0008964               60,1502,95,35,CASH
55,644,254,41,COUNTER 3                    724,1503,104,36,20.20
595,646,227,36,CASHIER: 1                  57,1555,151,44,CHANGE
59,725,182,39,QTY UOM                      726,1560,104,37,12.00
322,729,507,36,UNIT AMT EXC. AMT INC. GST  60,1646,253,39,GST SUMMARY
317,768,98,39,PRICE                        393,1644,230,46,AMOUNT (RM)
513,771,62,40,TAX                          671,1641,152,49,TAX (RM)
673,769,160,38,TAX CODE                    58,1703,162,38,ZRL = 0%
73ML                                     ③ 548,1708,77,32,8.20
52,834,687,37,WALL'S TOPTEN CHOCOLATE      752,1709,68,31,0.00
......                                     ......
```

(b) Annotations required by traditional methods

"PASARAYA BORONG PINTAR SDN BHD",
"14/03/2018",
"NO 19-G& 19-1& 19-2 JALAN TASIK UTAMA 4, MEDAN NIAGA TASIK DAMAI",
"8.20"

(c) Annotations required by our weakly-supervised method

Figure 1: Illustration of the annotations required by the traditional method and our weakly-supervised framework. Given (a) a document image, (b) traditional annotation scheme is to label the bounding box and string of each utterance, and further specific which category does each token/box belongs to. In contrast, (c) our method only requires each key information sequence. Annotation ambiguity of traditional annotation scheme is shown in the orange dotted bounding boxes. Different colors denote different entity categories.

However, the token-level category supervision expends great annotation costs and may be exposed to label ambiguity. Given a document image as shown in Figure 1 (a), the most widely used annotation scheme is to label the bounding box and string of each utterance, and further point out which category does each token/box belongs to, as shown in Figure 1 (b). In this way, a heuristic label assignment procedure is needed to train the aforementioned tagging model, of which the core idea is matching the detected boxes and recognized transcriptions with the given annotations and then assign label to each token/box of OCR results. However, this procedure may encounter problems from mainly two aspects. First, wrong recognition results will bring troubles to the matching operation, especially for key information sequences. Second,

*Corresponding author

the repeated contents will bring label ambiguities. As shown in Figure 1(a) and (b), three values with same content can be regarded as the answer of the key *Total Amount*. In most cases, it is hard to establish a uniform annotation specification to determine which one should be regarded as ground truth.

To address the aforementioned limitations, in this paper, we propose an end-to-end weakly-supervised learning framework, which can supervise the decoding process directly using the target key information sequences. The benefits it brings are mainly two-folds: first, it greatly saves the annotation costs, as shown in Figure 1 (c), and shortens the training process by skipping the matching between OCR results and the ground truth; second, our method solves the label ambiguity problem by automatically learning the alignment between OCR results and ground truth, which can adaptively distinguish the most likely one in the repeated contents. In addition, we also propose a flexible decoder, which is combined with our weakly-supervised training strategy and have two switchable modes – *Copy or Predict* Mode (TCPN-CP) and *Tag* Mode (TCPN-T), to balance the effectiveness and efficiency. In TCPN-CP, our decoder can generate key information sequences by *copying* a token from the input or *predicting* one in each time step, which can both retain novel contents in input and correct OCR errors. And in TCPN-T, our decoder can directly label each token's representations into a specific category in a single forward pass, which maintains the fast speed. It is notable that our decoder only needs to be trained once to work in different modes.

Besides, we propose an efficient encoder structure to simultaneously model the semantic and layout information in 2D OCR results. In our design, the semantic embeddings of different tokens in a document are re-organized into a vector matrix $I \in \mathcal{R}^{H\times W\times d}$ (here H, W and d are the height dimension, the width dimension and the number of channels, respectively), which we called **TextLattice**, according to the center points of token-level bounding boxes. After this, we adopt a modified light-weight ResNet [He *et al.*, 2016] combined with the U-Net [Ronneberger *et al.*, 2015] architecture to generate the high-level representations for the subsequent decoding. It is notable that the most relevant work of our encoding method was CharGrid [Katti *et al.*, 2018], which first used CNN to integrate semantic clues in a layout. However, it initialized an empty vector of the same size as the original document image, and then repeatedly filled each token's embedding at every pixel within its bounding box. This simple and direct approach may lead to the following limitations: 1) Larger tokens would be filled in more times, and it might result in the risk of category imbalance; 2) A pixel could completely represent a token, and repeated filling would waste extra cost; 3) The lack of concentration of information would make it difficult for the network to capture global clues. By comparison, our method greatly saves computing resources while maintains the space information of document. For example, for a receipt image in SROIE [Huang *et al.*, 2019] benchmark whose size is generally more than 1600×800, the side length of I after our processing is less than 100. In this way, both holistic and local clues can be captured, the relative position relationship between utterances are retained, and distance can be perceived in a more intuitive way through the receptive field.

Experiments on the two public SROIE and EPHOIE [Wang *et al.*, 2021] benchmarks demonstrate that our method shows competitive and even new state-of-the-art performance. Our main contributions can be summarized as follows:

- We propose an efficient 2D document representation called TextLattice, and the corresponding light-weight encoder structure.
- We propose a flexible decoder which has two inference modes - TCPN-CP for OCR error correction and TCPN-T for fast inference speed.
- We propose a weakly-supervised learning framework which guides decoding process directly using the key information sequences. This greatly saves the annotation costs and avoids label ambiguity.
- Our method achieves competitive performances even compared with the setting of token-level full supervision, which totally proves its superior.

2 Related Work

Early works of visual information extraction mainly utilized rule-based [Muslea and others, 1999] or template-based [Huffman, 1995] methods, which might tend to poor performance when the document layout was variable. With the development of deep learning, more advanced researches commonly extracted a feature sequence from the input plain text and used token-level supervision to train a sequence tagging model. [Lample *et al.*, 2016] first used a bidirectional long short-term memory [Hochreiter and Schmidhuber, 1997] (BiLSTM) network to model sequential information and a conditional random field (CRF) layer to decode the optimal classification path. Most of the follow-up works mainly focused on the feature encoding part: [Liu *et al.*, 2019a], GraphIE [Qian *et al.*, 2019] and PICK [Yu *et al.*, 2020] tried to use graph neural networks (GNNs) to extract node embeddings for better representation. LayoutLM [Xu *et al.*, 2020] embedded multi-source information into a common feature space, and utilized a BERT-like [Devlin *et al.*, 2019] model for feature fusion. TRIE [Zhang *et al.*, 2020] and VIES [Wang *et al.*, 2021] proposed the end-to-end VIE methods directly from image to key information, which introduced multi-head self-attention [Bahdanau *et al.*, 2015] to integrate multimodal clues during encoding.

As for the decoding part, EATEN [Guo *et al.*, 2019] first utilized sequence-level supervision to guide training. It generated feature maps directly from document image, and used several entity-aware attention-based decoders to iteratively parse the key information sequences. However, its efficiency could be significantly reduced as the number of entities increased, and it can only process simple and fixed layout since it had to overcome the difficulties of both OCR and VIE at the same time. When the given text blocks were accurate or directly the ground truth but the model still performed inference by step-by-step prediction, it might greatly slow down the speed and lead to the severe over-fitting problem of sequence generation due to the lack of corpus in VIE task.

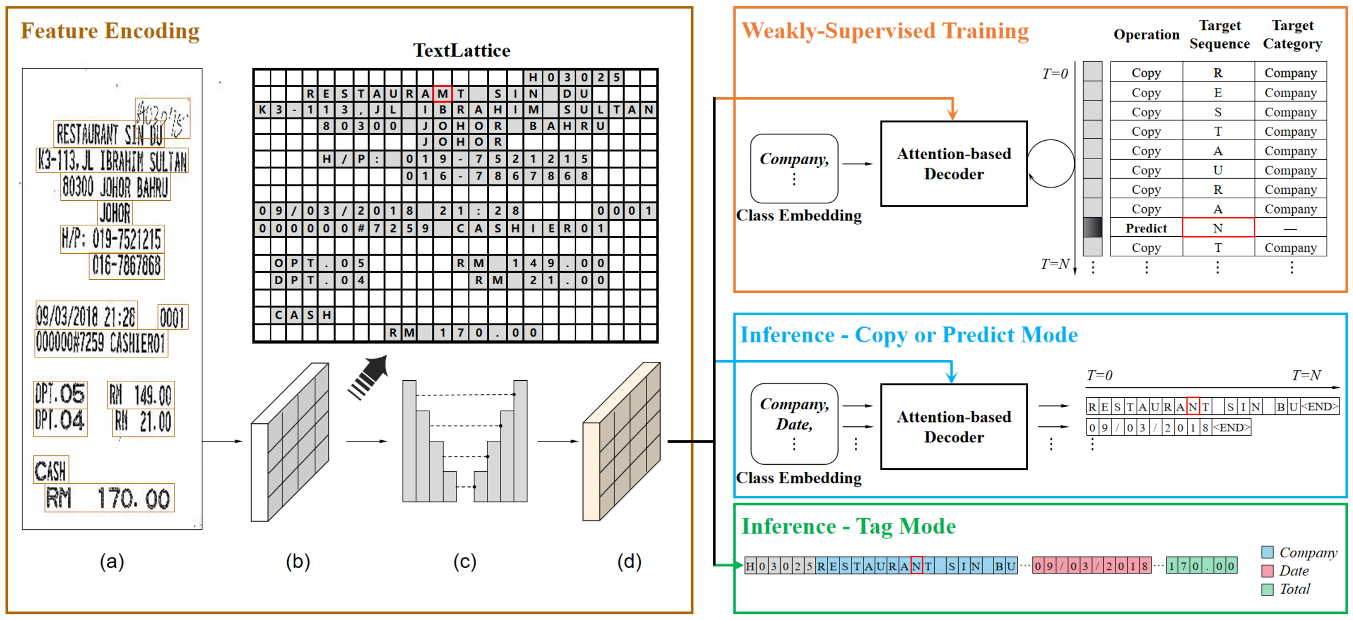


Figure 2: The overall framework of TCPN. (a) The input OCR results (the recognition result is ignored for visual clarity). (b) The corresponding TextLattice I. (c) The light-weight modified ResNet encoder combined with the U-Net architecture. (d) The result of feature encoding, which has the same size as I. The class embeddings of multiple entity categories and the corresponding target sequences are given to the attention-based decoder during training. In inference, different modes can be switched to depends on application requirements. The red bounding box shows a wrongly recognized character 'M'. In *Tag* Mode, it can be classified in a single forward pass; while in *Copy or Predict* Mode, it can be corrected as 'N'.

3 Method

Here we provide the details of the proposed TCPN. First we describe the approach of generating TextLattice, and how to encode higher-level features. Next we introduce details of our switchable decoder and weakly-supervised training strategy. Finally, we explain when and how to inference in different modes. Figure 2 gives an overview of our approach.

3.1 Document Representation

In this section, we introduce how to re-organize the OCR results into our 2D document representation - **TextLattice**. The whole process can be summarized as: 1) We first normalize y coordinates of detected bounding boxes B^u, sort B^u from top to bottom and left to right, and utilize heuristic rules to modify y coordinates to divide B^u into multiple rows; 2) Then, B^u is divided into token-level B^t according to lengths of the recognized strings S^u; 3) Next, the x coordinates of B^t are also normalized and modified to avoid information loss caused by overlapping; 4) Finally, we initialize an all-zero matrix $I \in \mathcal{R}^{H\times W\times d}$ where W and H are determined by the ranges of x and y coordinates of B^t, and fill in I according to the correspondence between token-level center points and d-dimensional token embeddings. The detailed procedure is shown in Appendix.

3.2 Feature Encoding

After acquiring I, we adopt ResNet [He *et al.*, 2016] as CNN encoder to capture more holistic features. The U-Net [Ronneberger *et al.*, 2015] structure is also combined to restore the down-sampled features to the same size as the input I and adaptively fuse both local and global clues extracted under diverse receptive fields. Vanilla ResNet adopts a 7×7 Conv2d as the `conv1` layer to capture association between local pixels in an RGB image. However, it may not be applicable in our scenario since the features of adjacent tokens also need to be separable, instead of high fusion of features in the first few layers. To this end, we replace `conv1` with a 3×3 Conv2d and remove the original `maxpool` layer. Thanks to the efficient design of TextLattice, both speed and superiority can be retained.

In order to further preserve the location clues, inspired by CoordConv [Liu *et al.*, 2018], two extra channels are concatenated to the incoming representation I, which contain horizontal and vertical relative location information in the layout of range from -1 to 1. The whole procedure of feature encoding can be formulated as:

$$\hat{I} = I + UNet(ResNet(I \oplus I_0)) \tag{1}$$

$$F = Indexing(\hat{I}, B^t) \tag{2}$$

Here, $\oplus$ is the concatenation operator, $I_0 \in \mathcal{R}^{H\times W\times 2}$ is the extra coordinate vector. Since the output of CNNs has the same size as I, we add them together as a residual connection. Finally, the features at the center points of token-level bounding boxes B^t are retrieved to form $F \in \mathcal{R}^{N\times d}$, where N is the number of tokens. We regard the rest pixels as useless and discard them directly for calculation efficiency.

3.3 Weakly-Supervised Training

As shown in Figure 2, the VIE task can be regarded as a set-to-sequence problem after feature encoding, since F is orderindependent. We introduce the class embedding $C \in \mathcal{R}^d$ to control the category of information parsed by the decoder, which is taken from a pre-defined trainable look-up table. Given C, our attention-based decoder takes it into account at each time step and iteratively predicts target sequence. Such novel design avoids class-specific decoders, alleviates the shortage of isolated class corpus, and decouples the serial correlation between different categories in the traditional sequence tagging model into parallel.

When generating sequences, we need the model to be able to switch between copying tokens from input or directly predicting ones. The copying operation make the model be able to reproduce accurate information and retain novel words, while the predicting operation introduces the ability of correcting OCR errors. Inspired by [See *et al.*, 2017], which implemented a similar architecture for abstractive text summarization, our model recurrently generates the hidden state s_t by referring to the current input tokens x_t and the context vector in the previous step F^*_{t-1}:

$$e_i^t = W_e tanh(W_1 \mathbf{C} + W_2 F_i + W_3 s_t + W_4 \sum_{t'=1}^{t-1} \alpha_i^{t'} + b_1) \quad (3)$$

$$\alpha^t = Softmax(e^t) \quad (4)$$

$$F_t^* = \sum_i \alpha_i^t F_i \quad (5)$$

$$s_t = RNN(F_{t-1}^* \oplus x_t, s_{t-1}) \quad (6)$$

Here, α is the attention score where the historical accumulated values are also referenced during calculation. All Ws and bs are learnable parameters.

Then, the probability distribution of tokens in a fixed dictionary P_t^{pred} is calculated and a *copy score* p_t^{copy} is generated as a soft switch to choose between different operations in each time step t:

$$P_t^{pred} = Softmax(W_5(F_t^* \oplus s_t) + b_2) \quad (7)$$

$$p_t^{copy} = \sigma(W_6 F_t^* + W_7 s_t + W_8 x_t + b_3) \quad (8)$$

$$P_t(k^*) = p_t^{copy} \sum_{i:k_i=k^*} \alpha_i^t + (1 - p_t^{copy}) P_t^{pred}(k^*) \quad (9)$$

$$\mathcal{L}_t^S = -log(P_t(k^*)) \quad (10)$$

$P_t(k^*)$ is the probability score of token k^* in time step t, where k^* is the current target token. $\mathcal{L}_t^S$ is the sequence alignment loss function. In this way, our method acquires the ability to produce out-of-vocabulary (OOV) tokens, and can adaptively perform optimal operations.

As of now, our method can be seen as a sequence generation model trained with sequence-level supervision. However, it is notable that since the class embedding C of entity category c is given, when the model decides to copy a token k_i from the input at a step, k_i's feature vector in F should be also classified as c by a linear classifier. More generally speaking, our method can first learn the alignment relationship, and then train a classifier using the matched tokens. This novel idea enables our approach the ability of supervising the sequence tagging model. We adopt a linear layer to model the entity probability distribution, which can be formulated as:

$$P_{t,i^*}^c = Softmax\left(W_c F_{k_{i^*}} + b_c\right), \quad (11)$$

$$where \qquad i^* = \underset{i}{\operatorname{argmax}}(\alpha_i^t) \qquad and \qquad p_t^{copy} > 0.5. \quad (12)$$

$$\mathcal{L}_t^C = -log(P_{t,i^*}^c(c)) \quad (13)$$

It is worth noting that, equation (11) - (13) do not train the tokens which do not belong to any key information sequences. The neglect of negative samples may lead to severe defect that all input tokens will be classified as positive. Thus we construct an extra auxiliary loss function $\mathcal{L}_t^N$ for negative sample suppression:

$$P_{t,i}^c = Softmax\left(W_c F_{k_i} + b_c\right) \quad (14)$$

$$P_t(c) = \sum_i P_{t,i}^c(c) \quad (15)$$

$$\mathcal{L}_t^N = max(0,\ P_t(c)\ -\ Length(S_c)) \quad (16)$$

Here, $P_t(c)$ indicates the sum of the probabilities belong to entity category c of all input tokens, and $Length(S_c)$ is the length of current target sequence S_c. The main purpose of $\mathcal{L}_t^N$ is to limit the number of input tokens classified as c to be less than or equal to the actual number. This simple but effective design greatly improves the performance of the model in *Tag* Mode.

In summary, the final integrated loss function $\mathcal{L}_t$ is the weighted sum of multiple components mentioned above:

$$\mathcal{L}_t = \lambda_S \mathcal{L}_t^S + \lambda_C \mathcal{L}_t^C + \lambda_N \mathcal{L}_t^N \quad (17)$$

where λ_S, λ_C and λ_N are trade-off hyper-parameters.

3.4 Inference

In this section, we explain when and how to implement inference process in different modes. It is worth noting that, since class embeddings are sent into the decoder in the form of a batch, key information sequences of different categories of the same document can be generated under different modes according to the entity-specific semantic characteristics.

In most real-world scenarios, OCR results cannot be flawless. In this regard, users can switch our decoder to *Copy or Predict* Mode as described in equation (3) - (9) to supplement missing or wrong tokens. This mode is more suitable for sequences of categories with strong semantic relevance.

Thanks to the auto-alignment property of the proposed weakly-supervised training strategy, the decoder can also directly perform sequence tagging in a single forward pass in *Tag* Mode using equation (14). It prefers to extremely few OCR errors or categories of weak semantic correlation between adjacent contents.

4 Experiment

4.1 Implementation Details

We adopt ResNet-18[He *et al.*, 2016] as backbone in feature encoding, and use BiLSTM in attention mechanism of decoder. The number of channels d is set to 256, the hyper-parameters λ_S, λ_C and λ_N are all set to 1.0 in our experiments empirically. We set batch size as 4 and perform 450 training epochs. The learning rate is initialized as 1.0 with ADADELTA [Zeiler, 2012] optimization and decreased to a tenth for two times in 300 and 400 epochs.

4.2 Ablation Study

In this section, we evaluate the influences of components of the proposed TCPN on the public EPHOIE benchmark.

Encoding Architecture	**F1-Score**	**FPS**
BiLSTM in [Lample *et al.*, 2016]	96.16	103.84
Chargrid[Katti *et al.*, 2018]	96.23	5.54
GAT in [Liu *et al.*, 2019a]	96.37	88.65
BERT-like[Cui *et al.*, 2020]	97.19	62.23
TextLattice(Ours)	**98.06**	**112.11**

Table 1: Performance and speed comparison on EPHOIE dataset between different encoding architectures. FPS is tested on a GeForce GTX 1080 Ti.

Encoding Architecture	**F1-Score**
TextLattice(Ours)†	**98.06**
†- CoordConv[Liu *et al.*, 2018]	96.83
†- UNet[Ronneberger *et al.*, 2015]	92.37
†- Residual Connection	97.70

Table 2: Effects of different components in the proposed encoding architecture on EPHOIE dataset.

Comparison between Different Encoding Structures

We organize OCR ground truth into plain texts and use different encoding structures to generate representations. Then a sequence tagging model is trained utilizing the official token-level category annotations. We mainly adopt the following models for comparison: **BiLSTM**: A bidirectional LSTM adopted in [Lample *et al.*, 2016]; **GAT**: A graph attention network (GAT) adopted in [Liu *et al.*, 2019a]; **BERT-like**: A BERT-like model similar to LayoutLM[Xu *et al.*, 2020]. Since vanilla LayoutLM is trained on English corpus, the pre-trained weights for Chinese in [Cui *et al.*, 2020] are loaded for fair comparison; **Chargrid**: The document modeling method introduced by Chargrid[Katti *et al.*, 2018].

The comparison results are given in Table 1. **BiLSTM** perceives sequential clues well, but it cannot effectively model location space in 1D form; **GAT** can adaptively fuse useful features using attention mechanism. However, the ability of capturing location clues highly depends on the way of feature embedding; **BERT-like** can perform forward calculation in parallel, and since the pretrained weights are loaded, it achieves satisfactory performance; **Chargrid** establishes input matrix using a more direct way, which means that both robustness and efficiency cannot be guaranteed. It is notable that **TextLattice(Ours)** achieves superior performance and maintains the fastest speed, which fully proves its efficiency. Our method has a more direct and sensitive perception of location clues than position embedding in **GAT** or **BERT-like**, and ensures a higher degree of information concentration than **Chargrid**. The fully parallel scheme also greatly contributes to the leading speed.

Effects of Components in TextLattice

We also conduct experiments to verify the effectiveness of different components in our encoding structure, such as CoordConv, the U-Net structure and the residual connection in equation (1). It can be seen in Table 2 that each design has a significant contribution to the final performance. Although CNN can capture the relative position relationship, **CoordConv** can further provide the global position clues relative to the whole layout, which brings higher discernibility; we also try to use **ResNet only** where all `stride` and the U-Net structure are removed to perform feature encoding. However, the performance decreases obviously, which indicates the importance of semantic feature fusion under different receptive fields; **Residual Connection** gives model the chance to directly receive token-level semantic embedding, which further improves the performance.

Method	**F1-Score**
Token-level Fully-Supervised	
[Lample *et al.*, 2016]	89.10
[Liu *et al.*, 2019a]	92.55
GraphIE[Qian *et al.*, 2019]	90.26
TRIE[Zhang *et al.*, 2020]	93.21
VIES[Wang *et al.*, 2021]	95.23
TextLattice(Ours)	**98.06**
Sequence-level Weakly-Supervised	
TCPN-T(Ours)	**97.59**

(a)

Method	**F1-Score**
Token-level Fully-Supervised	
[Lample *et al.*, 2016]	90.85
LayoutLM[Xu *et al.*, 2020]	95.24
[Liu *et al.*, 2019a]	95.10
PICK [Yu *et al.*, 2020]	96.12
TRIE [Zhang *et al.*, 2020]	96.18
VIES[Wang *et al.*, 2021]	96.12
TextLattice(Ours)	**96.54**
Sequence-level Weakly-Supervised	
TCPN-T(Ours)	**95.46**

(b)

Table 3: Performance comparison on (a) EPHOIE and (b) SROIE under ground truth setting. 'Token-level Fully-Supervised' indicates the token-level category annotation, and 'Sequence-level Weakly-Supervised' means the key information sequence annotation.

4.3 Comparison with the State-of-the-Arts

We compare our method with several state-of-the-arts on the SROIE and EPHOIE benchmarks. The following **Ground Truth Setting** indicates that the OCR ground truth is adopted, while **End-to-End Setting** indicates the OCR engine result.

Results under Ground Truth Setting

As shown in Table 3, our method exhibits superior performance on both SROIE and EPHOIE in the case of token-level full-supervision, which totally proves the effectiveness of our feature encoding method. Furthermore, the results under sequence-level weakly-supervised setting achieve competitive performance. This fully confirms the superiority of

Method	F1-Score
Rule-based Matching	
[Lample *et al.*, 2016]	71.95
[Liu *et al.*, 2019a]	75.07
TRIE[Zhang *et al.*, 2020]	80.31
VIES[Wang *et al.*, 2021]	83.81
Automatic Alignment	
TCPN-T(Ours, 112.11FPS)	**86.19**
TCPN-CP(Ours, 5.57FPS)	84.67

(a)

Method	F1-Score
Rule-based Matching	
NER [Ma and Hovy, 2016]	69.09
Chargrid [Katti *et al.*, 2018]	78.24
[Lample *et al.*, 2016]	78.60
[Liu *et al.*, 2019a]	80.76
TRIE [Zhang *et al.*, 2020]	82.06
VIES [Wang *et al.*, 2021]	91.07
Automatic Alignment	
TCPN-T(Ours, 88.16FPS)	91.21
TCPN-CP(Ours, 5.20FPS)	**91.93**

(b)

Table 4: Performance comparison on (a) EPHOIE and (b) SROIE under end-to-end setting. 'Rule-based Matching' indicates acquiring token-level label through traditional rule-based matching, and 'Automatic Alignment' means automatically learning the alignment using the key information sequences.

our learning strategy, which can model the correspondence between the input tokens and the output sequences.

Compared with SROIE, EPHOIE usually has less content and more token types, which reduces the difficulty of learning alignment. Relatively, since a receipt in SROIE often contains abundant tokens and the same token may appear repeatedly, which may lead to the alignment confusion, the gap between full- and weak-supervision is further widened.

Results under End-to-End Setting

We adopt BDN[Liu *et al.*, 2019b] as text detector and CRNN[Shi *et al.*, 2016] as text recognizer, and train them using the official annotations to get OCR results. It is worth noting that since there may inevitably exist errors in OCR results, all of our experiments under end-to-end setting are trained using our weakly-supervised manner, which avoids the matching process between OCR results and ground truth. The performances are shown in Table 4. Our method shows new state-of-the-art performance in every mode.

It can be inferred that an important basis for choosing TCPN-CP or TCPN-T mode is the richness of semantics and corresponding corpus. On SROIE, TCPN-CP obviously outperforms TCPN-T, which mainly benefits the ability to error correction; however, on EPHOIE, although both modes outperform counterparts, TCPN-T beats TCPN-CP by a large margin, where the main reason should be the diversity of Chinese tokens and the resulting lack of corpus.

Method	F1-Score
Rule-based Matching	
[Lample *et al.*, 2016]	78.79
[Liu *et al.*, 2019a]	80.92
TCPN-T(Ours)	**82.15**
Automatic Alignment	
TCPN-T(Ours)	84.37
TCPN-CP(Ours)	**89.08**

Table 5: Performance comparison on Business License Dataset under end-to-end setting.

Results on A Business License Dataset

In order to further explore the effectiveness of our framework in real-world applications, we collect an in-house dataset of business license. It contains 2331 photos taken by mobile phone or camera with real user needs, and most of images are inclined, distorted or the brightness changes dramatically. We randomly select 1863 images for training and 468 images for testing, and there are 13 types of entities to be extracted. Furthermore, the OCR results are generated by our off-the-shelf engines, which definitely contains OCR errors due to the poor image quality.

The detailed results are shown in Table 5. Our end-to-end weakly-supervised learning framework outperforms traditional rule-based matching method by a large margin, which can also greatly reduce the annotation cost. Compared with TCPN-T, the implicit semantic relevance learned by TCPN-CP can further boost the final performance by correcting OCR errors. Some qualitative results are shown in Appendix.

5 Conclusion

In this paper, we propose a unified weakly-supervised learning framework called TCPN for visual information extraction, which introduces an efficient encoder, a novel training strategy and a switchable decoder. Our method shows significant gain on EPHOIE dataset and competitive performance on SROIE dataset, which fully verifies its effectiveness.

Visual information extraction task is in the cross domain of natural language processing and computer vision, and our approach aims to alleviate the over-reliance on complete annotations and the negative effects caused by OCR errors. For future research, we will explore the potential of our framework through large-scale unsupervised data. In this way, the generalization of encoder, the alignment capability of decoder and the performance of our TCPN-CP can be further improved.

Acknowledgments

This research is supported in part by NSFC (Grant No.: 61936003, 61771199), GD-NSF (no.2017A030312006).

Appendix

A TextLattice Generation

Algorithm 1 shows the detailed procedure of our TextLattice Generation Algorithm introduced in Section 3.1.

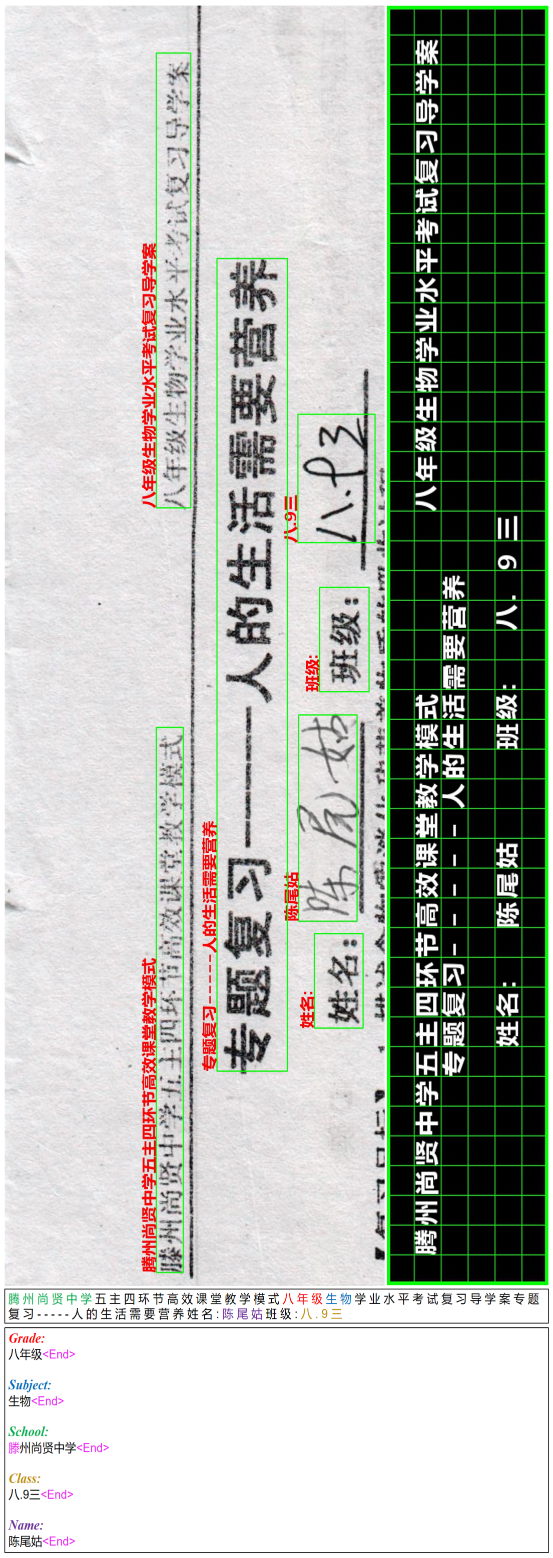
B Qualitative Results

Some qualitative results are shown in Figure 3, 4 and 5.

Algorithm 1 TextLattice Generation Algorithm

Input: Utterance-level detected bounding boxes B^u and corresponding recognized strings S^u
Parameter: Normalization threshold R^t and ratio R^r.
Output: TextLattice I

```
1:  Calculate $R^h$ = Average height of $B^u$.
2:  Normalize $y$ coordinates of $B^u$ using $R^h$.
3:  Sort $B^u$ from top to bottom, and then from left to right.
4:  Let $Y$ = unique values of $y$ coordinates of center points
    of $B^u$ sorted from small to large, a list $\widetilde{Y} = [Y[1], ]$.
5:  for $i$ from 2 to Count($Y$) do
6:    if $(Y[i] - Y[i-1]) \leq R^t$ then
7:      $dY = 1$.
8:    else
9:      $dY = max(1, (Y[i] - Y[i-1])/R^r)$.
10:   end if
11:   Append $\widetilde{Y}[-1] + dY$ into $\widetilde{Y}$.
12: end for
13: Modify $y$ coordinates of center points of $B^u$
    corresponding to $Y \to \widetilde{Y}$.
14: Split utterance-level boxes $B^u$ into token-level boxes $B^t$
    according to lengths of $S^u$.
15: Calculate $R^w$ = Average width of $B^t$.
16: Normalize $x$ coordinates of $B^t$ using $R^w$.
17: for $\hat{y}$ in $Y$ do
18:   Let $\hat{X}$ = $x$ coordinates of boxes $B^{t\prime}$ whose $y$
      coordinates of center points equal to $\hat{y}$.
19:   Sort $\hat{X}$ from small to large, let $\widetilde{X} = [\hat{X}[1], ]$
20:   for $i$ from 2 to Count($\hat{X}$) do
21:     if $(\hat{X}[i] - \hat{X}[i-1]) \leq R^t$ then
22:       $dX = 1$.
23:     else
24:       $dX = max(1, (\hat{X}[i] - \hat{X}[i-1])/R^r)$.
25:     end if
26:     Append $\widetilde{X}[-1] + dX$ into $\widetilde{X}$.
27:   end for
28:   Modify $x$ coordinates of center points of $B^{t\prime}$
      corresponding to $\hat{X} \to \widetilde{X}$.
29: end for
30: Combine $S^u$ in the same order as $B^u$ to get $S^t$.
31: Let $x_{min}, x_{max}, y_{min}, y_{max}$ = minimal and max values
    of $x$ and $y$ coordinates of center points of $B^t$, initialize
    an all-zero matrix $I \in \mathcal{R}^{(y_{max}-y_{min})\times(x_{max}-x_{min})\times d}$.
    Subtract $x_{min}, y_{min}$ from the coordinates of $B^t$.
32: Fill in $I$ according to the correspondence between $B^t$
    and $d$-dimensional token embeddings of $S^t$.
33: return $I$.
```


腾州尚贤中学五主四环节高效课堂教学模式
八年级生物学业水平考试复习导学案
专题复习-----人的生活需要营养
八.9三
班级:
陈尾姑
姓名:

腾州尚贤中学五主四环节高效课堂教学模式八年级生物学业水平考试复习导学案专题复习-----人的生活需要营养姓名:陈尾姑班级:八.9三

Grade:
八年级<End>

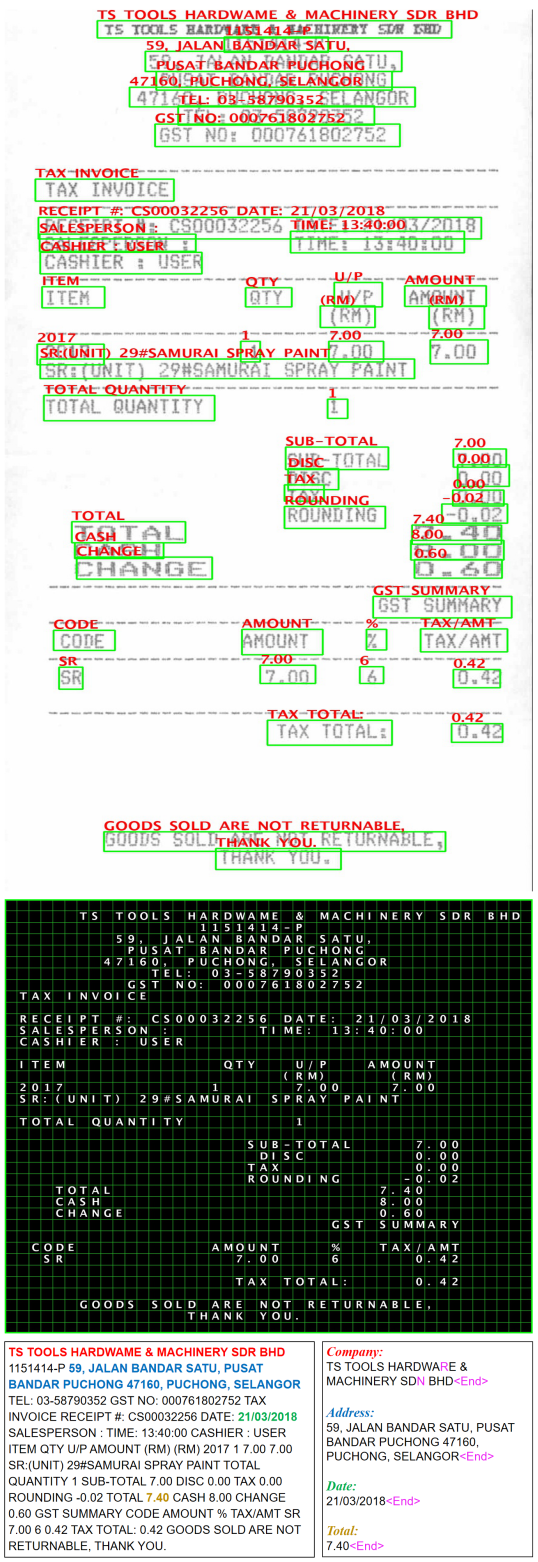
Subject:
生物<End>

School:
滕州尚贤中学<End>

Class:
八.9三<End>

Name:
陈尾姑<End>

Figure 3: Qualitative results of TCPN on EPHOIE dataset. From top to bottom and left to right are the OCR results, the TextLattice I, and the final predictions of TCPN-T and TCPN-CP. In TextLattice I, each green box represents a pixel. Pink tokens in TCPN-CP are generated by *predicting*, while the rest is produced by *copying*. Other different colors denote different entities.



TS TOOLS HARDWAME & MACHINERY SDR BHD 1151414-P 59, JALAN BANDAR SATU, PUSAT BANDAR PUCHONG 47160, PUCHONG, SELANGOR TEL: 03-58790352 GST NO: 000761802752 TAX INVOICE RECEIPT #: CS00032256 DATE: 21/03/2018 SALESPERSON : TIME: 13:40:00 CASHIER : USER ITEM QTY U/P AMOUNT (RM) (RM) 2017 1 7.00 7.00 SR:(UNIT) 29#SAMURAI SPRAY PAINT TOTAL QUANTITY 1 SUB-TOTAL 7.00 DISC 0.00 TAX 0.00 ROUNDING -0.02 TOTAL 7.40 CASH 8.00 CHANGE 0.60 GST SUMMARY CODE AMOUNT % TAX/AMT SR 7.00 6 0.42 TAX TOTAL: 0.42 GOODS SOLD ARE NOT RETURNABLE, THANK YOU.

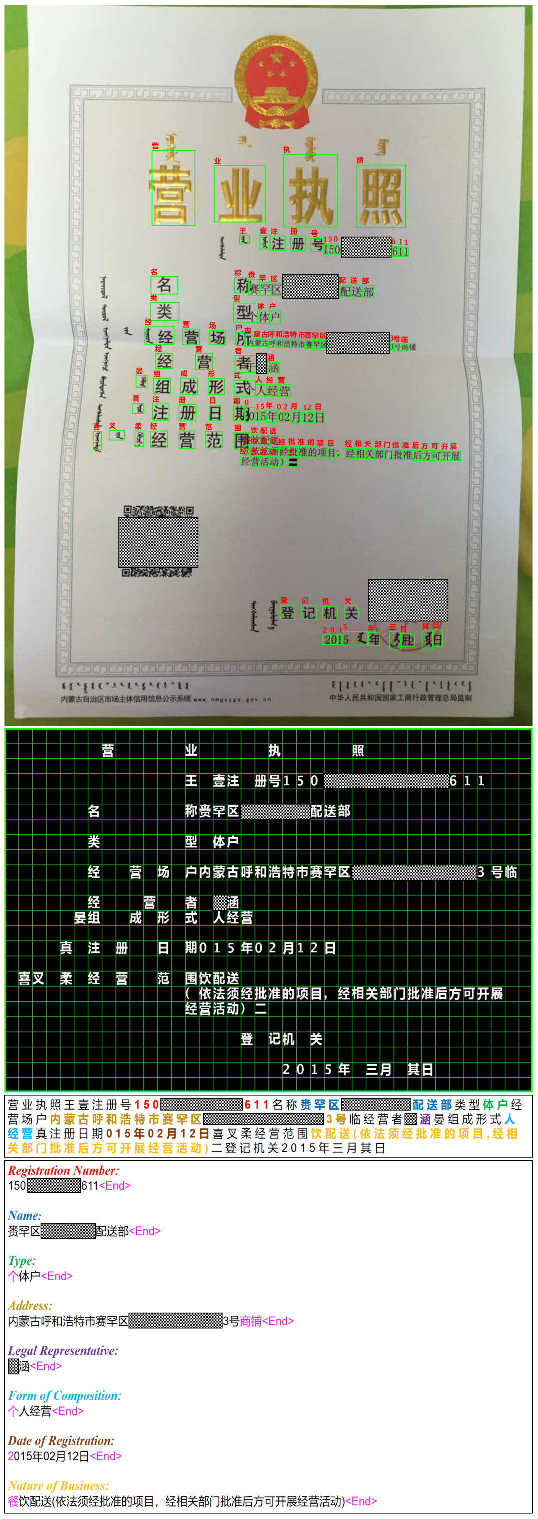
Company:
TS TOOLS HARDWARE & MACHINERY SDN BHD<End>

Address:
59, JALAN BANDAR SATU, PUSAT BANDAR PUCHONG 47160, PUCHONG, SELANGOR<End>

Date:
21/03/2018<End>

Total:
7.40<End>

Figure 4: Qualitative results of TCPN on SROIE dataset.



营业执照王壹注册号150[redacted]611名称赉罕区[redacted]配送部类型体户经营场户内蒙古呼和浩特市赛罕区[redacted]3号临经营者[redacted]涵晏组成形式人经营真注册日期015年02月12日喜叉柔经营范围饮配送(依法须经批准的项目,经相关部门批准后方可开展经营活动)二登记机关2015年三月其日

Registration Number:
150[redacted]611<End>

Name:
赉罕区[redacted]配送部<End>

Type:
个体户<End>

Address:
内蒙古呼和浩特市赛罕区[redacted]3号商铺<End>

Legal Representative:
[redacted]涵<End>

Form of Composition:
个人经营<End>

Date of Registration:
2015年02月12日<End>

Nature of Business:
餐饮配送(依法须经批准的项目，经相关部门批准后方可开展经营活动)<End>

Figure 5: Qualitative results of TCPN on the in-house Business License dataset. For the purpose of privacy protection, part of the correctly parsed information is *masked*.

References

[Bahdanau *et al.*, 2015] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[Cui *et al.*, 2020] Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for Chinese natural language processing. In *EMNLP: Findings*, pages 657–668, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, 2019.

[Ferilli *et al.*, 2011] Stefano Ferilli, Floriana Esposito, Teresa MA Basile, Domenico Redavid, and Incoronata Villani. Dominus plus-document management intelligent universal system (plus). In *Italian Research Conference on Digital Libraries*, pages 123–126, 2011.

[Guo *et al.*, 2019] He Guo, Xiameng Qin, Jiaming Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Eaten: Entity-aware attention for single shot visual text extraction. In *ICDAR*, pages 254–259, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[Hochreiter and Schmidhuber, 1997] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Huang *et al.*, 2019] Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. Icdar2019 competition on scanned receipt ocr and information extraction. In *ICDAR*, pages 1516–1520, 2019.

[Huffman, 1995] Scott B Huffman. Learning information extraction patterns from examples. In *IJCAI*, pages 246–260, 1995.

[Katti *et al.*, 2018] Anoop R Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. Chargrid: Towards understanding 2d documents. In *EMNLP*, pages 4459–4469, 2018.

[Lample *et al.*, 2016] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *NAACL-HLT*, pages 260–270, 2016.

[Liu *et al.*, 2018] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the coordconv solution. In *Advances in neural information processing systems*, pages 9605–9616, 2018.

[Liu *et al.*, 2019a] Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. Graph convolution for multimodal information extraction from visually rich documents. In *NAACL-HLT*, pages 32–39, 2019.

[Liu *et al.*, 2019b] Yuliang Liu, Sheng Zhang, Lianwen Jin, Lele Xie, Yaqiang Wu, and Zhepeng Wang. Omnidirectional scene text detection with sequential-free box discretization. In *IJCAI*, pages 3052–3058, 2019.

[Ma and Hovy, 2016] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, pages 1064–1074, 2016.

[Muslea and others, 1999] Ion Muslea et al. Extraction patterns for information extraction tasks: A survey. In *The AAAI-99 workshop on machine learning for information extraction*, volume 2, 1999.

[Qian *et al.*, 2019] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. GraphIE: A graph-based framework for information extraction. In *NAACL-HLT*, pages 751–761, 2019.

[Ronneberger *et al.*, 2015] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[See *et al.*, 2017] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *ACL (Volume 1: Long Papers)*, pages 1073–1083, 2017.

[Shi *et al.*, 2016] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *TPAMI*, 39(11):2298–2304, 2016.

[Wang *et al.*, 2021] Jiapeng Wang, Chongyu Liu, Lianwen Jin, Guozhi Tang, Jiaxin Zhang, Shuaitao Zhang, Qianying Wang, Yaqiang Wu, and Mingxiang Cai. Towards robust visual information extraction in real world: New dataset and novel solution. In *AAAI*, 2021.

[Xu *et al.*, 2020] 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 *ACM-SIGKDD*, pages 1192–1200, 2020.

[Yu *et al.*, 2020] 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 *ICPR*, 2020.

[Zeiler, 2012] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[Zhang *et al.*, 2020] 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 *ACM-MM*, 2020.