

# GraphRevisedIE: Multimodal Information Extraction with Graph-Revised Network

Panfeng Cao<sup>a</sup>, Jian Wu<sup>b</sup>

<sup>a</sup>*University of Michigan, Ann Arbor, 48109, MI, USA*

<sup>b</sup>*University of Science and Technology of China, Hefei, 230026, Anhui, PR China*

---

## Abstract

Key information extraction (KIE) from visually rich documents (VRD) has been a challenging task in document intelligence because of not only the complicated and diverse layouts of VRD that make the model hard to generalize but also the lack of methods to exploit the multimodal features in VRD. In this paper, we propose a light-weight model named GraphRevisedIE that effectively embeds multimodal features such as textual, visual, and layout features from VRD and leverages graph revision and graph convolution to enrich the multimodal embedding with global context. Extensive experiments on multiple real-world datasets show that GraphRevisedIE generalizes to documents of varied layouts and achieves comparable or better performance compared to previous KIE methods. We also publish a business license dataset that contains both real-life and synthesized documents to facilitate research of document KIE.

*Keywords:*

document information extraction, graph convolutional network, transformer

---

## 1. Introduction

Optical character recognition (OCR) is a technology to recognize the texts in the scanned documents, and KIE is the downstream task of OCR that extracts entity information from the texts. KIE is critical to applications such as document indexing, information archival, and information retrieval [1] because it can save significant amounts of time and resources. Deep learning based approaches have become the focus of modern research and achieved state-of-the-art (SOTA) results. However, it remains a challenge to effectively utilize the multimodal features in visually rich documents (VRD). As we can see in Figure 1, VRD can be a structured or unstructured document such as a receipt, ticket, business license, etc. There are varied formats, layouts, and contents in VRD, and multimodal information is critical to resolving the semantic ambiguity, which can occur when textual information alone is not enough to distinguish the entities. For example, in Figure 1(b), texts of both the month and train number are 03, but they are of different entity types. We can only distinguish them from the layout and visual information.

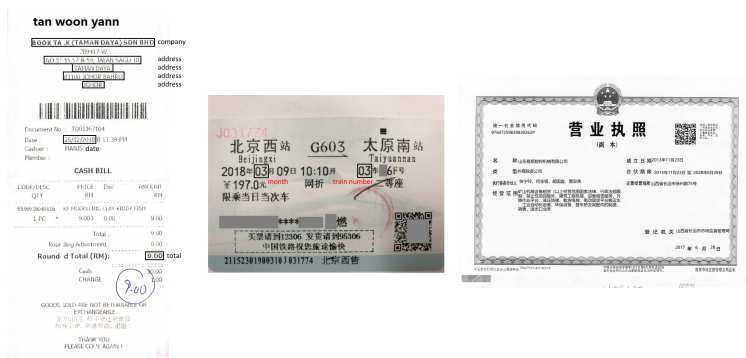


Figure 1: Example VRD of different layouts. (a) Key entities to be extracted are marked with red rectangles. (b) Same text O3 results in semantic ambiguities for different entities. (c) Example business license.

Traditionally, template or rule based KIE methods [2, 3] have been widely adopted in commercial applications. The interpretability of those methods makes the KIE program easy to be adopted and scaled for different scenarios. However, significant engineering efforts and domain-specific knowledge are needed to design the handcrafted rules and patterns for different entities. And those methods only support a limited number of document types and cannot deal with complicated and unstructured documents. Later KIE systems formalize the problem as a Named Entity Recognition (NER) task, which typically applies models to predict the beginning-inside-outside (BIO) tags of the tokens. [4, 5, 6] are based on bidirectional LSTM (BiLSTM) and add an additional convolutional neural network (CNN) layer to enrich the feature representation for entity extraction. [7] is a pure image based approach that introduces a fully convolutional encoder-decoder network based on the VGG architecture. The semantic features are encoded with the document layout to perform information extraction. Other methods [8, 9] use graph based long short-term memory (LSTM), which allows a varied number of incoming edges at each memory cell, to jointly learn the entities and relations extraction. While those methods have been proven effective, they do not make full use of the multiple modal features available in the VRD and cannot tackle semantic ambiguity.

Efficiently combining multimodal features in the VRD has become the focus of modern research. Graph based methods represent the document as a graph, with nodes representing segments and edges representing segment relations. Graph convolution is utilized to propagate the global context and enrich the feature embeddings. [10, 11] utilize predefined graphs to combine the textual and layout features. [12] uses a graph learning convolutional network (GLCN) [13] to dynamically learn the graph and generate a richer semantic representation of the segment. Lately, pre-training based methods such as LayoutLMv2 [14], StrucText [15] and BROS [16] have been proposed to deeply fuse multimodal features from large-scale pre-training datasets and achieved SOTA

performance in downstream KIE tasks. However, those models have a higher number of parameters and require larger datasets for effective pre-training. Furthermore, it’s also difficult to deploy and maintain the models in real-life settings due to the complicated multi-stage training paradigm.

In this paper, we propose a novel, lightweight framework named GraphRevisedIE to tackle the problem of multimodal feature embedding. Textual, visual, and layout features are jointly embedded in the model to address the semantic ambiguity. We design a graph module inspired by [17] to learn the graph representation for the document by graph revision and perform graph convolution to enrich the multimodal feature embedding with global context. The graph module also leverages the sparsification technique to learn the appropriate graph representation for the sparse document.

The main contributions of this paper are summarized as follows:

- In this paper, a novel framework named GraphRevisedIE is proposed to handle document KIE. Multimodal features in VRD are effectively embedded to cope with the semantic ambiguity.
- As far as our knowledge, GraphRevisedIE is the first graph based model that utilizes the graph revision technique in document KIE. The graph module can effectively learn document graphs and contextualize the multimodal feature embedding with global context.
- We publish a dataset that contains real-life and synthesized business licenses to facilitate the document KIE research.
- Extensive experiments on multiple public datasets show that GraphRevisedIE outperforms existing graph based models. The model also has comparable performance to pretrained models, while it has significantly fewer parameters and does not depend on large pre-training datasets <sup>1</sup>.

## 2. Related Works

Early research on entity extraction focuses on the exploitation of a single modal feature from the document. [18] utilizes BiLSTM to embed the textual feature. [8, 9] introduce the graph based LSTM that supports cross sentence semantic relation and entity extraction. [4, 5] utilize both LSTM and CNN to get better textual embedding to perform sequence labeling. [19] leverages plain-text semantic features from the document but does not exploit the layout and image features. [7, 20] use the image features to encode the semantic contents but leave the textual and layout features untapped. [10, 11] apply BiLSTM to embed the textual features and GCN to incorporate the layout features. Graph convolution is used not only to propagate the global contexts but also to generate

---

<sup>1</sup>Our code and business license dataset are publicly available at <https://github.com/AYSP/GraphRevisedIE>.

the node embedding. [10] depends on task-specific graphs and needs predefined data structures, which are hard to extend to other types of documents. [12] jointly embeds the textual and visual features and uses the absolute layout feature in the graph module to generate the node and edge embedding.

Pretrained transformer encoder based approaches achieved SOTA results by deeply fusing multimodal features in pretraining. LayoutLM [21] designs pre-training tasks that utilize the absolute 2D layout features and the textual features. Visual features are embedded in fine tuning. LayoutLMv2 [14] moves the visual feature embedding to pre-training and learns effective multimodal feature representation. However, those approaches depend on a large corpus and need significantly more parameters and time to train.

Compared to previous graph based methods [11, 12], GraphRevisedIE differs in several aspects. First, [11, 12] choose fully connected graph as the initial graph. Although they can dynamically update the edge weights during training, they do not support adding new edges due to the element-wise product in the attention function. If the edge weight is learned to be 0, it is removed and cannot be added back. When the document graph is highly sparse, the model eventually learns a suboptimal graph representation. Nevertheless, the graph module in our framework does not enforce a fully connected graph as the initial graph and it supports adding new edges as well as updating existing edge weights. Attention based graph convolution is performed to contextualize feature embedding with global context to facilitate final prediction. Furthermore, our graph module does not require a loss function and is seamlessly integrated with the downstream learning objective, which implicitly entices the graph representation learning. Finally, we rely on relative positional information to embed the layout feature instead of using absolute positional information, which can introduce spatial bias in the case of image twisting, shifting, and rotation. Relative positional information better captures the global invariant layout relations of entities and helps improve the model’s performance.

### 3. Model Architecture

Table 1 gives the notations used in the paper. Given  $D$  and  $I$ , we first use an open source OCR tool (e.g. Tesseract <sup>2</sup>) to recognize  $N$  segments that correspond to the nodes in the graph. The graph is represented by the weighted adjacency matrix  $A$ , in which the element is the edge weight. The model architecture is illustrated in Figure 2, which comprises three modules: a multimodal feature embedding module, a graph module, and a decoding module.

#### 3.1. Embedding

As shown in Figure 2, the multimodal feature embedding module has three branches, each embedding a single modal feature. First, for textual embedding  $TE$ , it includes all segment textual embeddings:

---

<sup>2</sup><https://tesseract-ocr.github.io/>

Symbol	Meaning
$D$	The document
$I$	The scanned document image of $D$
$T$	Text segments recognized by OCR in $D$
$N$	Number of text segments in $D$
$L$	Maximum text segment length of $T$
$t_i$	$i_{th}$ text segment
$c_j^i$	$j_{th}$ character of $t_i$
$B$	Bounding boxes of $T$
$b_i$	$i_{th}$ bounding box
$d$	Model dimension
$A$	Weighted adjacency matrix representing the graph
$S$	Similarity matrix of nodes in the graph
$v_i$	$i_{th}$ node corresponding to $t_i$
$a_{ij}$	Weight of the directed edge from $v_j$ to $v_i$
$d^n$	Embedding dimension in the graph module
$TE$	Textual embedding of $T$
$te^i$	Textual embedding of $t_i$
$VE$	Visual embedding for segments corresponding to $B$
$ve^i$	Visual embedding for the segment corresponding to $b_i$
$d^b$	Sinusoidal embedding dimension
$PE_{ij}$	Relative positional embedding of $b_i$ and $b_j$
$SE$	Segment embedding for segments corresponding to $B$
$HE$	Hidden embedding of $SE$
$DE$	Document embedding of $D$
$d_{tags}$	Number of predefined BIO tags
$Z_{ij}$	Probability of the $i_{th}$ character token being the $j_{th}$ tag
$T_{ij}$	Transition probability from the $i_{th}$ tag to the $j_{th}$ tag
$y_i$	$i_{th}$ predefined BIO tag
$Y_{DE}$	The set of all possible tag sequences for $DE$

Table 1: Notations.

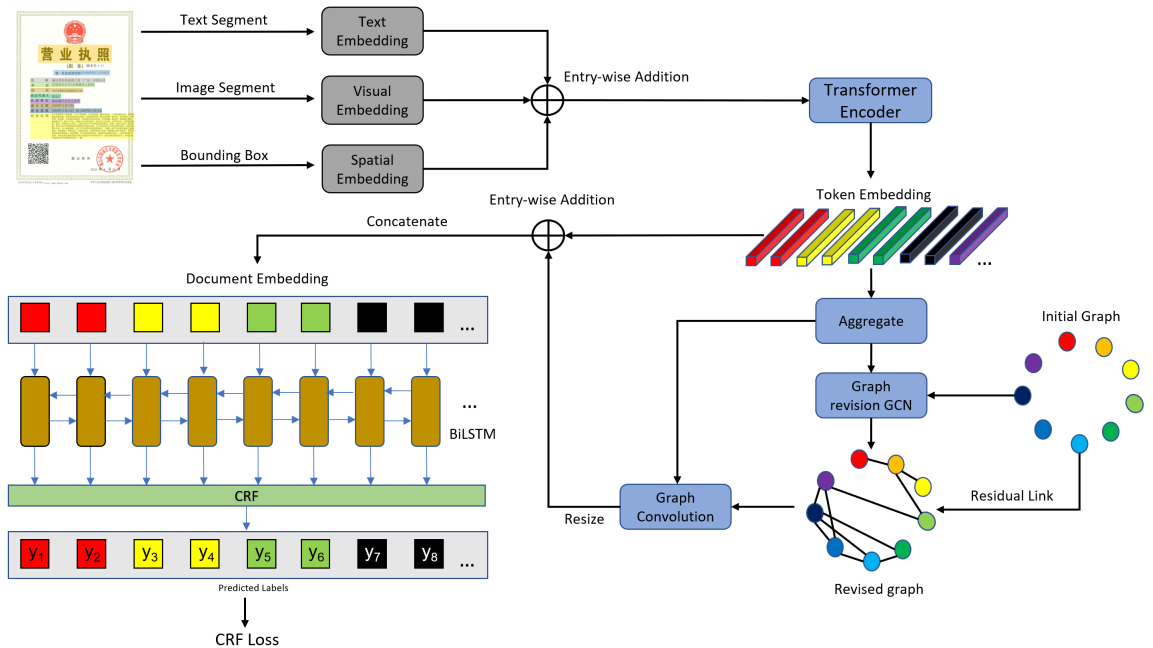


Figure 2: Overall diagram of the GraphRevisedIE framework. Note that for illustration purposes, we use the same color for all tokens in the same segment and different colors for tokens in different segments. The top section of the diagram demonstrates the process of multimodal feature fusion. The bottom right section explains the graph module for feature embedding enrichment. Self-connected edges are omitted. The bottom left section is the BiLSTM-CRF module that calculates the CRF loss and produces the final prediction.

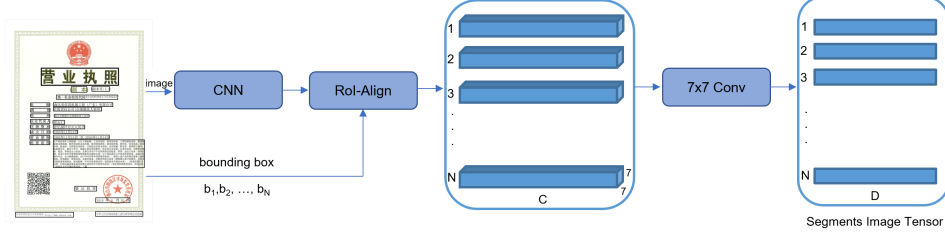


Figure 3: Illustration of generating the image embedding. Inputs are the raw image and bounding boxes of segments. RoI-Align is used to extract segment level features from the whole image feature produced by the CNN module. A convolution kernel is applied to transform the output dimension of RoI-Align to the model dimension.

$$TE = \text{Concat}(te^1, \dots, te^N) \in \mathbb{R}^{N \times L \times d} \quad (1)$$

$$te^i = \text{Concat}(\text{Emb}(c_1^i), \dots, \text{Emb}(c_L^i)) \in \mathbb{R}^{L \times d} \quad (2)$$

, where  $\text{Concat}$  is the concatenation operation and  $\text{Emb} : \mathbb{R} \rightarrow \mathbb{R}^d$  is the character token embedding function, e.g. one-hot embedding.

Then we use a CNN as the visual feature extractor to get the visual embedding. The visual features of the segment, such as font, size, and color, can help enrich the segment embedding. As presented in Figure 3, a CNN module is first used to get the global feature maps of the whole image, and then the local feature map of each bounding box is extracted from the global feature maps via RoIAlign[22]. Finally, we apply the convolution on the local feature map to generate the segment level visual embedding  $ve^i$ . Given  $I$  and  $B$ ,  $VE$  is calculated as follows:

$$VE = \text{Concat}(ve^1, \dots, ve^N) \in \mathbb{R}^{N \times d} \quad (3)$$

$$ve^i = \text{Conv}(\text{RoIAlign}(\text{CNN}(I), b_i)) \in \mathbb{R}^d \quad (4)$$

Within  $b_i$ , all characters share the same  $ve^i$  by design. The final form of  $VE$  after resizing is:

$$VE = \text{Concat}(ve^1, \dots, ve^N) \in \mathbb{R}^{N \times L \times d} \quad (5)$$

Finally, we embed the layout features. Inspired by the 1D positional embedding in Transformer [23], we designed the 2D relative positional embedding, which normalizes the spatial relations between segments. It's robust to the positional shifting caused by the raw image distortion and helps the model learn the inherent layout. Given  $T$  and  $B$ , we first normalize the coordinates so they fall between 0 and 100. Then for  $t_i, b_i$  and  $t_j, b_j$ , we calculate the relative positional

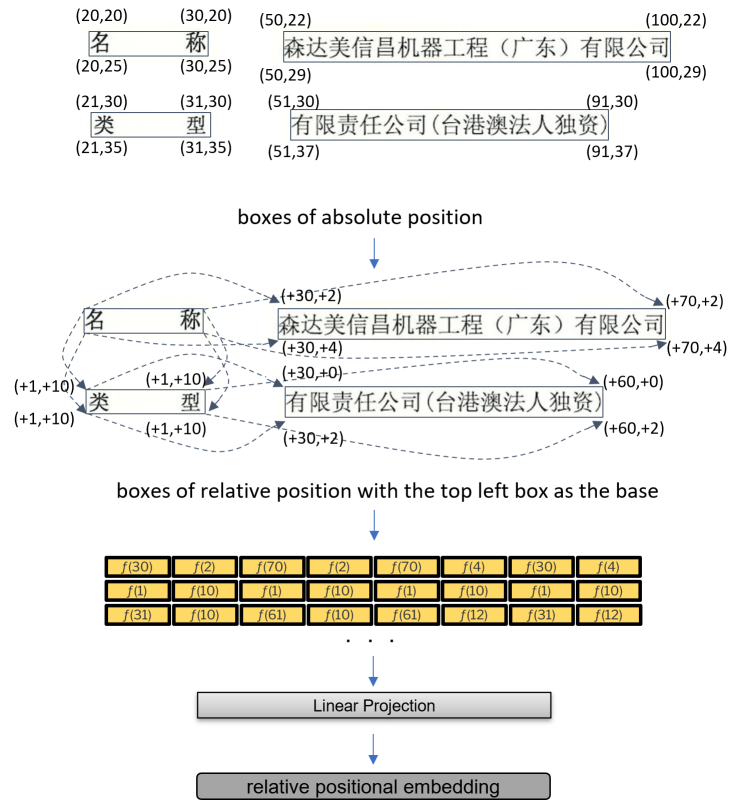


Figure 4: Process of generating the relative positional embedding. Relative positions are first embedded with the sinusoidal embedding function  $f$  and then go through a linear projection layer to get the final embedding.



embedding between those two bounding boxes in the following equation:

$$\begin{aligned}
PE_{ij} = & \text{Concat}(f^{sinu}(x_i^{tl} - x_j^{tl}), f^{sinu}(y_i^{tl} - y_j^{tl}), \\
& f^{sinu}(x_i^{tr} - x_j^{tr}), f^{sinu}(y_i^{tr} - y_j^{tr}), \\
& f^{sinu}(x_i^{br} - x_j^{br}), f^{sinu}(y_i^{br} - y_j^{br}), \\
& f^{sinu}(x_i^{bl} - x_j^{bl}), f^{sinu}(y_i^{bl} - y_j^{bl})) \cdot W \\
& \in \mathbb{R}^d
\end{aligned} \tag{6}$$

, where  $f^{sinu} : \mathbb{R} \rightarrow \mathbb{R}^{d^b}$  is the sinusoidal embedding, which is used in [23] to help embed the relative positions of segments.  $W : \mathbb{R}^{8 \times d^b} \rightarrow \mathbb{R}^d$  is the linear projection matrix that maps from the sinusoidal embedding dimension to the model dimension. The process is also illustrated in Figure 4, where we use the top left box as the base box for comparison. It is worth noting that we embed all four vertices of the text segment, allowing the projection matrix to learn spatial features such as relative height, width, and distance. Since tokens in the same segment share the same bounding box coordinate, we resize and get the final relative positional embedding  $PE \in \mathbb{R}^{N \times L \times d}$ .

By now we have all the single modal embeddings, we calculate the merged multimodal embedding by performing element-wise addition of those embeddings and applying transformer encoding:

$$E = \text{transformer\_encoder}(TE + VE + PE) \in \mathbb{R}^{N \times L \times d} \tag{7}$$

### 3.2. Graph Module

The graph revised module propagates the non-local and non-sequential contexts among segments to enrich the segment embedding. Although similar, our graph module design differs from [17], which constructs a single large graph for the entire dataset to address the node classification problem. For the document KIE task, our graph module needs to learn the graph representation for all documents in the dataset. The initial graph of each document is represented by an identity matrix, and the graph module revises it to find the appropriate graph.

As is described in Figure 5, we perform two operations in this module, i.e. graph revision and attention based graph convolution. Given  $D$ , we first aggregate the character multimodal embeddings in the segments to produce the segment embedding  $SE \in \mathbb{R}^{N \times d}$ . With the initial weighted adjacency matrix  $A$ , we calculate the hidden embedding  $HE$  from  $SE$ :

$$HE = A \cdot \tanh(A \cdot SE \cdot W_1) \cdot W_2, \tag{8}$$

, where  $A \in \mathbb{R}^{N \times N}$ ,  $W_1 \in \mathbb{R}^{d \times d^n}$ ,  $W_2 \in \mathbb{R}^{d^n \times d}$ ,  $SE, HE \in \mathbb{R}^{N \times d}$  and  $\tanh$  is the activation function.

The similarity matrix  $S$  of segments is then derived from  $HE$ :

$$S = \text{Knn}(\text{Kernal}(HE, HE^T)) \in \mathbb{R}^{N \times N}, \tag{9}$$

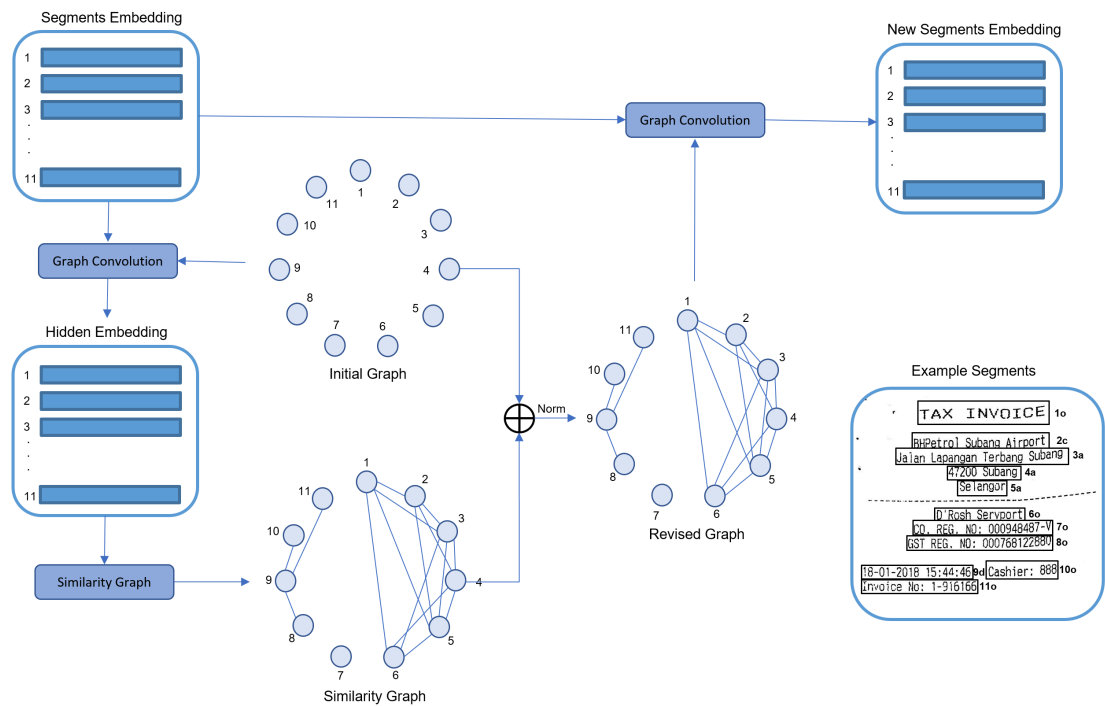


Figure 5: The graph module illustrated on an example SROIE receipt. In the bottom right, segments corresponding to the nodes are given with the indexes and labels (o: other, c: company, a: address, d: date). We use an identity matrix as the initial graph. For simplicity, self-connected edges are omitted. A new segment embedding is produced by graph convolution on the revised graph using the original segment embedding.

We use dot product as the kernel function following [17]. Since  $S$  is dense, the  $K$  nearest neighbor ( $Knn$ ) algorithm is applied to sparsify the graph and only keep the top  $K$  neighbors of each node. Unlike [17], where sparsification is mainly for memory and computation efficiency since important neighbor nodes are relatively constant, this sparsification process is necessary in our task because the entity in the document can be split into an unknown number of continuous multi-line segments.  $Knn$  helps identify important neighbors efficiently by removing unimportant edges. To obtain the revised adjacency matrix  $A'$ , we add  $S$  and  $A$  and normalize the result:

$$A' = Norm(A + S) \in \mathbb{R}^{N \times N} \quad (10)$$

With the element-wise addition operator  $+$ , new edges can be added and existing edges can be reweighted.

Using attention-based graph convolution with  $A'$ , we compute the updated segment embedding  $SE'$ :

$$SE' = A' \cdot SE \cdot W_3 \in \mathbb{R}^{N \times d}, W_3 \in \mathbb{R}^{d \times d} \quad (11)$$

Compared with GLCN [13], our graph module does not have the graph representation learning loss, which simplifies the design and experiments. Note that although the graph module helps the model generalize on documents with varied and complex layouts, it is not indispensable when the document has a relatively fixed layout. We study the importance of the graph module in ablation study (§4.3.3) on various document datasets.

### 3.3. Decoding

We resize the segment embedding output by the graph module to  $N \times L \times d$  and add it to the multimodal character embedding to get the final embedding. As a result, the character embedding combines not only the textual, image, and layout features of its own segment but also the global context of neighboring segments. To begin decoding, we concatenate all character embeddings in the segments from left to right and from top to bottom to produce the document level embedding  $DE \in \mathbb{R}^{[N \cdot L] \times d}$ . The reason we do the concatenation is because if a sentence is broken down into several text segments, we can restore its original semantic structure. The document embedding series is passed to a BiLSTM model to encode the long short-term dependencies, and the prediction scores of BIO tags are calculated by:

$$Z = BiLSTM(DE) \cdot W_B \in \mathbb{R}^{[N \cdot L] \times d_{tags}} \quad (12)$$

, where  $W_B \in \mathbb{R}^{d \times d_{tags}}$  is the linear projection matrix mapping from the hidden dimension of BiLSTM to the output BIO tags dimension.  $Z$  is the scores matrix, in which  $Z_{ij}$  indicates the possibility of  $i_{th}$  token being the  $j_{th}$  tag. Finally, character level BIO tagging is performed via a CRF layer. CRF is particularly effective in NER tasks where token labels have strong interdependencies. The

Dataset	Type	Language	# Keys	# Images
SROIE	Receipt	English	4	Train 526, Val 100, Test 347
CORD	Receipt	English	30	Train 800, Val 100, Test 100
FUNSD	Form	English	4	Train 149, Val 0, Test 50
Train Ticket	Ticket	Chinese	8	Train 1749, Val 100, Test 80
Business License	License	Chinese	9	Train 1120, Val 100, Test 100

Table 2: Statistics of each dataset.

tagging decisions for the tokens are jointly considered for the document series. Given a sequence of predictions  $y = (y_1, y_2, \dots, y_n)$ , the score is defined as:

$$s(DE, y) = \sum_{i=0}^{N \cdot L} T_{y_i, y_{i+1}} + \sum_{i=1}^{N \cdot L} Z_{i, y_i} \quad (13)$$

, where  $T$  is the transition matrix of scores.  $T_{y_i, y_{i+1}}$  is the score of transitioning from  $y_i$  to  $y_{i+1}$ .  $y_0$  is the start tag and  $y_{N \cdot L + 1}$  is the end tag. The conditional probability of  $y$  given  $DE$  is calculated with the softmax operation:

$$p(y|DE) = \frac{e^{s(DE, y)}}{\sum_{\bar{y} \in Y_{DE}} e^{s(DE, \bar{y})}} \quad (14)$$

The loss function is the logarithm of the conditional probability:

$$Loss = -\ln(p(y|DE)) = -s(DE, y) + \ln \sum_{\bar{y} \in Y_{DE}} e^{s(DE, \bar{y})} \quad (15)$$

The optimal tag sequence is the one with the highest conditional probability:

$$y^* = \arg \max_{y \in Y_{DE}} p(y|DE) \quad (16)$$

We search the optimal tag sequence with dynamic programming.

## 4. Experiments

### 4.1. Datasets

Our model is evaluated on multiple real world public datasets: SROIE [24], CORD [25], FUNSD [26], Train Tickets[20] and Business Licenses.

**SROIE** dataset is used to extract entity information from scanned receipts. It contains 626 receipts for training and 347 receipts for testing. Each receipt has four entities for extraction: company, address, total, and date. The dataset has relatively complicated and varied layouts and is suitable to validate the generalizability of the model.

**CORD** dataset is for both entity extraction and entity linking. It has 800 scanned receipts for the training set, 100 for the validation set, and 100 for the test set. There are in total 4 categories in this dataset, which are further classified into 30 subclasses, such as menu name, total price, etc. We use this dataset to perform entity extraction.

**FUNSD** dataset consists of 199 forms annotated with 4 entity types: question, answer, header, and other. It supports both entity linking and entity

extraction tasks. The training set has 149 forms, and the test set has 50 forms. We utilize this dataset to evaluate the model’s performance on large documents.

**Train Tickets** dataset has a total of 2K real documents and 300K synthetic documents. The document image was taken in real-life settings with all the possible conditions, such as dim lighting, distortion, background noise, etc. Entities we need to extract from the train ticket are the ticket number, destination station, seat category, train number, starting station, date, ticket rates, and passenger name.

Since the dataset does not provide the OCR results, we used the dataset in [12], which sampled 400 real documents and 1530 synthetic documents from the original datasets and annotated them with bounding boxes and transcripts with OCR. [12] chose 320 real documents and all synthetic documents for training and 80 real documents for testing. The same setting is used by our model for fair comparison.

**Business Licenses** dataset contains 320 real documents and 500 synthetic documents. We collect the documents either online or by manually taking the photos in real-life settings. A business license contains nine fields: company name, company type, company start date, registration capital, legal person, operation dates, business scopes, company location, and social credit code. The content consists mainly of numbers and Chinese characters and has different layouts. Since the images are captured in real life, there is inevitable image distortion and background noise. We utilized OCR to extract the transcripts and manually labeled them with different entity types. For synthetic licenses, we first create the templates in variable layouts, then build our corpus for different entity types, and finally synthesize documents <sup>3</sup> with the templates and corpus.

#### 4.2. Experiment Settings

Our model is implemented in PyTorch and trained with a NVIDIA GTX 3060 GPU with 12GB memory. An Adam optimizer with a decaying learning rate is used. The learning rate is initially set to  $1e-4$  and decays by 0.1 every 50 epochs. The model dimension  $d$ , the graph module embedding dimension  $d^n$  and the BiLSTM hidden dimension  $d^b$  are all 512. The sinusoidal embedding dimension  $d^b$  is 1024.  $K$  is set to 4 for the  $Knn$  algorithm in the graph module. The dropout ratio is set to 0.1. Resnet50 [27] with default parameters is used as the image feature extractor. We use the default setting of the transformer encoder in the embedding module.

#### 4.3. Experiment results

Since the model is character based, each character in the segment is labeled with the entity type that maximizes the conditional probability of the document series. The label of the segment is decided by the majority of the character labels. For example, the decoded BIO tags of **01/18** is *B-date, I-date, O*,

---

<sup>3</sup>Our code for synthesizing the business licenses is publicly available at <https://github.com/AYSP/Business-Licenses>.

Dataset	Precision		Recall		F1	
	Baseline	GraphRevisedIE	Baseline	GraphRevisedIE	Baseline	GraphRevisedIE
SROIE	96.79	<b>96.80</b>	95.46	<b>96.04</b>	96.12	<b>96.42</b>
CORD	91.75	<b>93.91</b>	93.26	<b>94.61</b>	92.50	<b>94.26</b>
Train Ticket	98.75	<b>99.07</b>	98.45	<b>98.76</b>	98.60	<b>98.91</b>
Business License	99.05	<b>99.37</b>	99.21	<b>99.37</b>	99.13	<b>99.37</b>

Table 3: Comparison of the baseline model to GraphRevisedIE on four datasets. GraphRevisedIE outperforms the baseline model.

Menu	Model		Total	Model	
	Baseline	GraphRevisedIE		Baseline	GraphRevisedIE
unitprice	86.96	<b>96.40</b>	emoneyprice	28.57	<b>40</b>
num	76.19	<b>95.24</b>	total_price	93.72	<b>96.15</b>
sub_cnt	90.32	<b>93.75</b>	menutype_cnt	54.55	<b>60</b>
sub_price	<b>80</b>	77.78	menuqty_cnt	81.97	<b>84.38</b>
discountprice	52.63	<b>58.82</b>	creditcardprice	<b>88.89</b>	85

Table 4: Entity level F1 score comparison on the CORD dataset between the baseline method and GraphRevisedIE. We select the menu and total entity types as examples for easier explanation.

*B-date* and *I-date*. While the third character is mislabeled as *O*, the word level prediction is still *date*, which is decided by the majority of the predicted character labels.

#### 4.3.1. Baseline

We chose PICK[12] as the baseline method because both PICK and GraphRevisedIE are graph based, and PICK has been proven effective in document KIE. We compare PICK with our model on datasets including SROIE, CORD, train tickets, and business licenses. We evaluate the model’s performance using the entity-level F1 score.

#### 4.3.2. Results

As shown in Table 3, the baseline method achieves competitive performance on the datasets. GraphRevisedIE still outperforms the baseline with small improvements. In comparison to the baseline, the relative positional embedding in GraphRevisedIE is critical in allowing the model to learn the document layout quickly and efficiently. For datasets with relatively fixed layouts, such as train tickets, GraphRevisedIE achieves 0.3% improvements on the F1 score, although the baseline almost achieves a full score. For the CORD dataset, GraphRevisedIE improves the F1 score by about 1.7%. Finally, for the SROIE and business license datasets with variable document layouts, GraphRevisedIE still has a 0.2–0.3% improvement on the F1 score compared to the baseline.

Table 4 illustrates the entity level comparison between the baseline and GraphRevisedIE on the CORD dataset. Due to the limited space and the large number of different entity types in this dataset, we only selected a subset of entity types for easy elaboration. There are entity types with rich relative positional features. For example, menu.unitprice is usually on the right of menu.num and is in the rightmost column, while menu.num is usually on the left

Model	Modality	Pretrained	# Params	SROIE			CORD			FUNSD		
				P	R	F	P	R	F	P	R	F
BERT	T	✓	340M	90.99	90.99	90.99	88.33	91.07	89.68	54.69	67.10	60.26
RoBERTa	T	✓	355M	91.07	91.07	91.07	-	-	-	66.48	66.48	66.48
UniLMv2	T	✓	340M	94.59	94.59	94.59	89.87	91.98	90.92	65.61	72.54	68.90
LayoutLM	T+L	✓	343M	94.38	94.38	94.38	94.37	95.08	94.72	76.77	81.95	79.27
LayoutLMv2	T+L+V	✓	426M	96.25	96.25	96.25	94.53	95.39	94.95	80.29	85.39	82.76
PICK	T+L+V	✗	-	96.79	95.46	96.12	91.75	93.26	92.50	-	-	-
GraphRevisedIE	T+L+V	✗	68M	96.80	96.04	96.42	93.91	94.61	94.26	76.67	80.22	78.41

Table 5: We compare the model’s performance with other models, including the large version of the pretrained models and PICK.

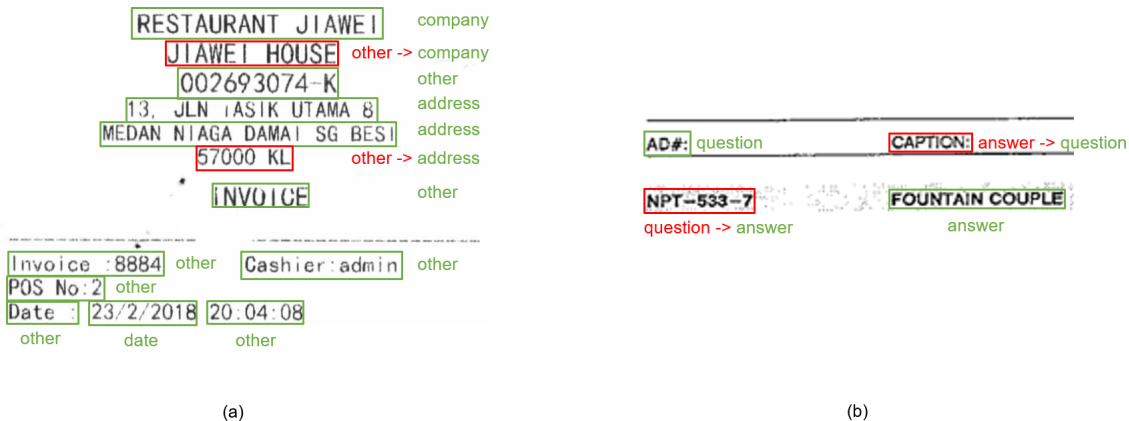


Figure 6: Example predictions made by GraphRevisedIE on (a) SROIE receipt and (b) FUNSD form. (a) **JIAWEI HOUSE** is incorrectly labeled as **other** when it should be part of the company, and **57000 KL** is incorrectly labeled as **other** when it should be part of the address. (b) The question answer pairs should be in the column direction, while GraphRevisedIE bases the prediction on the row direction. The performance of GraphRevisedIE can be further improved by using more powerful textual embedding to deal with isolated semantic context.

of menu.nm and is in the leftmost column. The relative positional information can be effectively captured by the relative positional embedding as illustrated in Figure 4. GraphRevisedIE outperforms the baseline by a large margin on those entity types.

Table 5 presents the model performance on the public SROIE, CORD, and FUNSD datasets. Performance metrics of the pretrained models are obtained from their original papers. Particularly, we compare our model with the large version of the pretrained models. On the SROIE dataset, GraphRevisedIE achieves the highest F1 score even with fewer parameters and without pre-training. On the CORD dataset, GraphRevisedIE achieves comparable performance with the pretrained models, which proves its generalization ability on small datasets with varied layouts. Since the CORD dataset has many more entity labels than SROIE, we speculate that richer textual embedding can help the model learn the label semantics better and reduce the ambiguity. Therefore, pre-training methods achieve the best performance. Despite its effectiveness on

	Train Tickets	Business Licenses
Full Model	98.9	99.3
w/o spatial features	↓0.6	↓0.5
w/o textual features	↓0.7	↓5.4
w/o image features	↓0.3	↓0.2
w/o graph module	↓0.4	↓0.7

Table 6: Ablation study on train tickets and business licenses to evaluate the importance of each component in the framework.

small documents, GraphRevisedIE does not perform well on large and complicated documents such as the FUNSD form. A large document usually contains richer semantic information than a small document. Since our model only leverages the vanilla one-hot textual embedding, it is hard to embed the semantic features effectively, especially when the semantic context is split into multiple segments as shown in Figure 6. This observation is aligned with the findings in the CORD dataset. Other limitations of GraphRevisedIE include the requirement of some initial experiments to determine the optimal  $K$  for the graph module and post-processing to generate the word level predictions since our model is character based.

#### 4.3.3. Ablation Study

In this section, we first do an ablation study on the train ticket and business license datasets to evaluate the importance of components in the model. As is shown in Table 6, one observation is that removing the textual feature does not reduce the F1 score much on the train ticket dataset. The model can still rely on the layout and visual features to achieve good performance. For the business licenses dataset, the F1 score decreases by 5.4% when the textual feature is removed. This is aligned with the fact that the business license has more semantic information than the train ticket. Intuitively, visual features also contribute to the model’s performance. Information such as font, color, size, background, etc. can only be captured by the visual encoder, which is important to reduce ambiguity. Last but not least, since the graph module enriches the character embedding with global context, removing the graph module causes decreased F1 scores on both datasets.

We perform another ablation study to evaluate the impact of  $K$ , the number of neighbors used in the  $Knn$  algorithm in the graph module, on the CORD and train ticket datasets. From Figure 7, gradually increasing  $K$  from 1 results in better performance, and then the performance starts to decrease after some point. For the CORD dataset, the best F1 score is achieved when  $K$  equals 4. Further increasing  $K$  only brings in more noise from distant segments and reduces the performance. For the train tickets, similarly, the best F1 score is achieved when  $K$  is 2. This study demonstrates that a sparse graph, i.e.,  $K$  is small, better captures the document graph than a dense graph, i.e.,  $K$  is large, on the CORD and train ticket datasets and results in optimal performance.



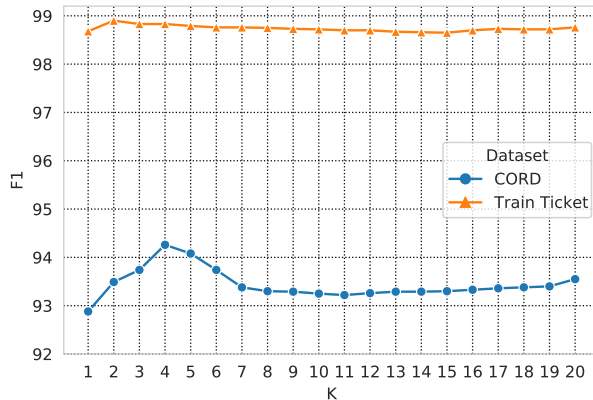


Figure 7: Experiments on the CORD and Train Ticket datasets to evaluate model performance under different  $K$  in the  $Knn$  algorithm.

## 5. Conclusion

In this paper, we propose a novel framework named GraphRevisedIE, which can effectively combine multimodal features from VRD, to perform the KIE task. We integrate with a graph module to model the underlying document graph, which is used to propagate the global context among segments to enrich the character embedding. GraphRevisedIE is has been proven to achieve good performance on multiple public datasets, and it is able to generalize over small documents with varied layouts. It’s worth mentioning GraphRevisedIE does not perform well on large documents. Replacing character level embedding with pretrained word level embedding to utilize more semantic features can possibly improve the model’s performance. Besides, although GraphRevisedIE provides flexibility to customize the number of neighbors in the graph module for different datasets, it adds more manual effort and could be automated. Finally, we set the kernel function to be the dot product in the graph module, while more options could be explored in future work.

## References

- [1] K. Jung, K. In Kim, A. K. Jain, Text information extraction in images and video: a survey, *Pattern Recognition* 37 (2004) 977–997. URL: <https://www.sciencedirect.com/science/article/pii/S0031320303004175>. doi:<https://doi.org/10.1016/j.patcog.2003.10.012>.
- [2] A. Dengel, B. Klein, Smartfix: A requirements-driven system for document analysis and understanding, in: *Proceedings of the 5th International Workshop on Document Analysis Systems V, DAS '02*, Springer-Verlag, Berlin,

- Heidelberg, 2002, p. 433–444. URL: <https://dl.acm.org/doi/abs/10.5555/647798.736679>.
- [3] D. Schuster, K. Muthmann, D. Esser, A. Schill, M. Berger, C. Weidling, K. Aliyev, A. Hofmeier, Intellix – end-user trained information extraction for document archiving, in: 2013 12th International Conference on Document Analysis and Recognition, 2013, pp. 101–105. URL: <https://ieeexplore.ieee.org/abstract/document/6628593>. doi:10.1109/ICDAR.2013.28.
- [4] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Berlin, Germany, 2016, pp. 1064–1074. URL: <https://aclanthology.org/P16-1101>. doi:10.18653/v1/P16-1101.
- [5] J. P. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, Transactions of the Association for Computational Linguistics 4 (2016) 357–370. URL: <https://aclanthology.org/Q16-1026>. doi:10.1162/tacl\_a\_00104.
- [6] J. I. Toledo, M. Carbonell, A. Fornés, J. Lladós, Information extraction from historical handwritten document images with a context-aware neural model, Pattern Recognition 86 (2019) 27–36. URL: <https://www.sciencedirect.com/science/article/pii/S0031320318303145>. doi:<https://doi.org/10.1016/j.patcog.2018.08.020>.
- [7] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, J. B. Faddoul, Chargrid: Towards understanding 2D documents, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4459–4469. URL: <https://aclanthology.org/D18-1476>. doi:10.18653/v1/D18-1476.
- [8] N. Peng, H. Poon, C. Quirk, K. Toutanova, W.-t. Yih, Cross-sentence n-ary relation extraction with graph LSTMs, Transactions of the Association for Computational Linguistics 5 (2017) 101–115. URL: <https://aclanthology.org/Q17-1008>. doi:10.1162/tacl\_a\_00049.
- [9] L. Song, Y. Zhang, Z. Wang, D. Gildea, N-ary relation extraction using graph-state LSTM, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 2226–2235. URL: <https://aclanthology.org/D18-1246>. doi:10.18653/v1/D18-1246.
- [10] Y. Qian, E. Santus, Z. Jin, J. Guo, R. Barzilay, GraphIE: A graph-based framework for information extraction, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short

- Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 751–761. URL: <https://aclanthology.org/N19-1082>. doi:10.18653/v1/N19-1082.
- [11] X. Liu, F. Gao, Q. Zhang, H. Zhao, Graph convolution for multimodal information extraction from visually rich documents, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 32–39. URL: <https://aclanthology.org/N19-2005>. doi:10.18653/v1/N19-2005.
- [12] W. Yu, N. Lu, X. Qi, P. Gong, R. Xiao, Pick: Processing key information extraction from documents using improved graph learning-convolutional networks, in: 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 4363–4370. URL: <https://ieeexplore.ieee.org/abstract/document/9412927>. doi:10.1109/ICPR48806.2021.9412927.
- [13] B. Jiang, Z. Zhang, D. Lin, J. Tang, B. Luo, Semi-supervised learning with graph learning-convolutional networks, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 11305–11312. URL: <https://ieeexplore.ieee.org/abstract/document/8953909>. doi:10.1109/CVPR.2019.01157.
- [14] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florencio, C. Zhang, W. Che, M. Zhang, L. Zhou, LayoutLMv2: Multi-modal pre-training for visually-rich document understanding, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 2579–2591. URL: <https://aclanthology.org/2021.acl-long.201>. doi:10.18653/v1/2021.acl-long.201.
- [15] Y. Li, Y. Qian, Y. Yu, X. Qin, C. Zhang, Y. Liu, K. Yao, J. Han, J. Liu, E. Ding, Structext: Structured text understanding with multi-modal transformers, in: Proceedings of the 29th ACM International Conference on Multimedia, MM '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 1912–1920. URL: <https://doi.org/10.1145/3474085.3475345>. doi:10.1145/3474085.3475345.
- [16] T. Hong, D. Kim, M. Ji, W. Hwang, D. Nam, S. Park, Bros: a pre-trained language model for understanding texts in document (2020). URL: <https://openreview.net/forum?id=punMXQEsPr0>.
- [17] D. Yu, R. Zhang, Z. Jiang, Y. Wu, Y. Yang, Graph-revised convolutional network, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part III, Springer-Verlag, Berlin, Heidelberg,

- 2020, p. 378–393. URL: [https://doi.org/10.1007/978-3-030-67664-3\\_23](https://doi.org/10.1007/978-3-030-67664-3_23). doi:10.1007/978-3-030-67664-3\_23.
- [18] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 260–270. URL: <https://aclanthology.org/N16-1030>. doi:10.18653/v1/N16-1030.
- [19] T. Gui, Y. Zou, Q. Zhang, M. Peng, J. Fu, Z. Wei, X. Huang, A lexicon-based graph neural network for Chinese NER, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1040–1050. URL: <https://aclanthology.org/D19-1096>. doi:10.18653/v1/D19-1096.
- [20] H. Guo, X. Qin, J. Liu, J. Han, J. Liu, E. Ding, Eaten: Entity-aware attention for single shot visual text extraction, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 254–259. URL: <https://ieeexplore.ieee.org/abstract/document/8978053>. doi:10.1109/ICDAR.2019.00049.
- [21] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, M. Zhou, Layoutlm: Pre-training of text and layout for document image understanding, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery; Data Mining, KDD '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 1192–1200. URL: <https://doi.org/10.1145/3394486.3403172>. doi:10.1145/3394486.3403172.
- [22] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2020) 386–397. URL: <https://ieeexplore.ieee.org/abstract/document/8372616>. doi:10.1109/TPAMI.2018.2844175.
- [23] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, I. Polosukhin, Attention is all you need, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 30, Curran Associates, Inc., 2017. URL: <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- [24] Z. Huang, K. Chen, J. He, X. Bai, D. Karatzas, S. Lu, C. V. Jawahar, Icdar2019 competition on scanned receipt ocr and information extraction, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 1516–1520. URL: <https://ieeexplore.ieee.org/abstract/document/8977955>. doi:10.1109/ICDAR.2019.00244.

- [25] S. Park, S. Shin, B. Lee, J. Lee, J. Surh, M. Seo, H. Lee, Cord: a consolidated receipt dataset for post-ocr parsing, in: Workshop on Document Intelligence at NeurIPS 2019, 2019.
- [26] G. Jaume, H. Kemal Ekenel, J.-P. Thiran, Funsd: A dataset for form understanding in noisy scanned documents, in: 2019 International Conference on Document Analysis and Recognition Workshops (ICDARW), volume 2, 2019, pp. 1–6. URL: <https://ieeexplore.ieee.org/abstract/document/8892998>. doi:10.1109/ICDARW.2019.10029.
- [27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. URL: [https://openaccess.thecvf.com/content\\_cvpr\\_2016/html/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html).