Merge and Recognize: A Geometry and 2D Context Aware Graph Model for Named Entity Recognition from Visual Documents

Chuwei Luo[1], Yongpan Wang[1], Qi Zheng[1], Liangcheng Li[2], Feiyu Gao[1], Shiyu Zhang[1]
Alibaba Group[1]
College of Computer Science, Zhejiang University, No.38, Zheda Road[2]
chuwei.lcw@alibaba-inc.com, {yongpan,yongqi.zq}@taobao.com,
liangcheng_li@zju.edu.cn,
{feiyu.gfy,rickzhang.zsy}@alibaba-inc.com

Abstract

Named entity recognition (NER) from visual documents, such as invoices, receipts or business cards, is a critical task for visual document understanding. Most classical approaches use a sequence-based model (typically BiLSTM-CRF framework) without considering document structure. Recent work on graph-based model using graph convolutional networks to encode visual and textual features have achieved promising performance on the task. However, few attempts take geometry information of text segments (text in bounding box) in visual documents into account. Meanwhile, existing methods do not consider that related text segments which need to be merged to form a complete entity in many real-world situations. In this paper, we present GraphNEMR, a graph-based model that uses graph convolutional networks to jointly merge text segments and recognize named entities. By incorporating geometry information from visual documents into our model, richer 2D context information is generated to improve document representations. To merge text segments, we introduce a novel mechanism that captures both geometry information as well as semantic information based on pre-trained language model. Experimental results show that the proposed GraphNEMR model outperforms both sequence-based and graph-based SOTA methods significantly.

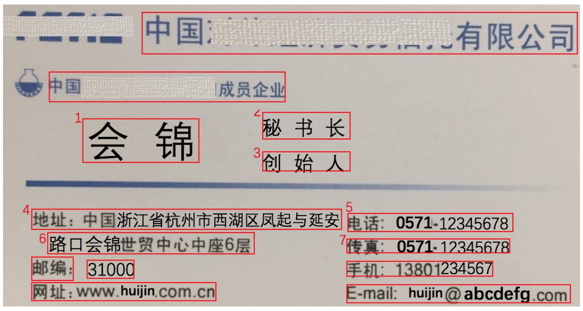
1 Introduction

With the rapid progresses in natural language processing and computer vision, visual documents become a mainstream media for expressing abundant information. Extracting the named entities from these visual documents is a critical step for further understanding. In the perspective of traditional natural language processing, the layout and the format information of documents is discarded. Only plain text that simply consists of sequential words do the researchers focus on and are necessary to be extracted entities. However, visual documents consist of many discrete text segments with a large variety of layouts and formats as Figure 1 shows. The combination of different text segments with different positions may represent different semantic information. Without the layout structure and the 2D semantic context information in it, the named entity recognition in visual documents could be much harder.

Although the named entity recognition in visual documents is a newly proposed under-researched task. Recently, in the attempt to make use of the structure information of visual documents, many works (Palm et al., 2017; Yang et al., 2017; Katti et al., 2018; Liu et al., 2018; Qian et al., 2019; Denk and Reisswig, 2019; Zhao et al., 2019) have designed NLP/CV/NLPCV based methods for visual-documents-related tasks. These approaches mostly focus on the coordinate of text segments to make features or learn embeddings. However, most of these methods do not consider the two problems:

1. Existing models often ignore the geometric information between text segments which is crucial for constructing 2D context for visual documents to extract named entities. It is hard to analyze the semantic meaning only through the plain text inside the bounding box and its coordinates.

Proceedings of the Graph-based Methods for Natural Language Processing (TextGraphs), pages 24–34
Barcelona, Spain (Online), December 13, 2020



[redacted] 中国[redacted]有限公司

中国[redacted]成员企业

会　锦

秘 书 长

创 始 人

地址：中国浙江省杭州市西湖区凤起与延安

路口会锦世贸中心中座6层

邮编：31000

网址：www.huijin.com.cn

电话：0571-12345678

传真：0571-12345678

手机：13801234567

E-mail：huijin@abcdefg.com

Figure 1: Example of a visual documents: visual business card. The red rectangular boxes are different text segments.

2. Because of the layout design, a complete named entity may be separated into several segments and cannot represent its full meaning. Meanwhile, some of text segments may lose the semantic information of their prefix ones and get incorrect tagging results. It is necessary and important to merge text segments into a complete named entity.

Specifically, for the first problem, as illustrated in Figure 1, it is hard to tell whether "会锦"("HuiJin") in text segment 1 is a named entity only according to its own plain text. While according to some of the nearest text segments, "秘书长"("Secretary General") in text segment 2 and "创始人"("The Founder") in text segment 3, human can help infer that text segment 1 is a person entity.

For the second, as Figure 1 shows, neither text segment 4 nor text segment 6 is a complete named entity. However, the two segments (segment 4 and segment 6) together represent a complete location entity that is *6/F, Middle Block, Huijin World Trade Center, at the intersection of Fengqi Road and Yan'an Road, Xihu District, Hangzhou, Zhejiang, China*. Apparently, the ability to merge text segments into a complete entity is important for NER from visual document.

To solve the above two problems, in this paper, we propose GraphNEMR, a graph neural end-to-end joint model for named entity recognition and merging tokens into named entities from visual documents. GraphNEMR incorporates the geometric information with the semantics to automatically extract non-sequential context-aware hidden features for each text segments in the visual documents. Specifically, We regard a visual document as a graph structure and all text segments in it are the graph nodes. The geometric information is represented by the adjacency matrix of the graph. In each text segment, a BiLSTM structure is used to sequentially encode tokens to represent semantic features. Then the graph convolutional network (GCN) (Kipf and Welling, 2017) encoder integrates information between neighbor nodes to learn the final representation of each text segment. Then the representations are used as the inputs of the merge module we proposed to decide the relation between text segments. After GCN encoder and the merge module, a LSTM+CRF decoder is used to get the named entity tagging. Our main contributions of this paper are as follows:

- To address the visual text merge problem, we propose a general method that captures the geometry information and semantics information. To the best of our knowledge, our approach is the first work to merge tokens in different text segments into complete named entity in visual documents.

- We propose the 8-geometry neighbors relation for each text bounding boxes in visual documents to represent geometry information in merge layer. Meanwhile, we design a geometry-distance-related adjacency matrix for graph representation with GCN.

- We propose a loss called $Loss_{nsp/sop}$ for semantically supervising merging text segments. Furthermore, we can obtain the right prefix semantic information for each text segments via merging results.

Extensive experiments show that our model outperforms both the strong sequence-based baseline model (BiLSTM-CRF) and the SOTA graph-based model (GraphIE) in visual document named entity recognition task.

2 Related Work

Our model builds on recent research of information extraction in visual documents and nested named entity recognition.

Recently, there is a lot interest in the task of information extraction in visual documents. Most of these works combined approaches from NLP, CV and document analysis. Lample et al. (2016) propose model BiLSTM-CRF that is a strong and a wildly used baseline for NER. Many researchers apply BiLSTM-CRF directly in visual information extraction without structure information consideration. Palm et al. (2017) use the sequential recurrent neural network (RNN) to extract key-value information from invoices. Their work shows the ability of neural network approach for extracting information in visual documents. However, their RNN model also treats documents as sequential text. Yang et al. (2017) consider document information extraction as a pixel-wise segmentation task and applied a end-to-end multimodal network to in visual documents. Their experiments showed that the textual features help layout segmentation. Katti et al. (2018) try to preserves visual documents' 2D layout by incorporating coordinate of characters for information extraction from invoices. Zhao et al. (2019) also found that in documents key information extraction, spatial information plays intrinsic roles. Denk and Reisswig (2019) extended the work of (Katti et al., 2018) to incorporate contextualized embedding by BERT language model and Xu et al. (2019) propose a pre-trained LayoutLM with the text and coordinates of text segments as inputs. They showed the effectiveness of using a pre-trained language model to invoice information extraction. Based on graph convolution network (GCN), Qian et al. (2019) and Liu et al. (2018) introduced the graph-based model that integrate textual and position attribute (i.e., coordinate, font size) to do visual information extraction task. They show that graph-based model outperforms the sequence-based baseline BiLSTM-CRF and confirms the benefits of using layout structure in visual information extraction.

The task of nested NER (Finkel and Manning, 2009) focuses on recognizing entities that can be nested within each other. This can be considered as related problems to ours on how to merge text segments in visual document. Recently, a number of approaches have been proposed for nested NER (Ju et al., 2018; Wang and Lu, 2018; Fisher and Vlachos, 2019). Specifically, Fisher and Vlachos (2019) decomposes nested NER into two stages, that first merge tokens into entities and then do recognition. But compared with our merging tokens problems, their approaches applied on serialized 1-D text instead of visual documents.

As we can see, 2D layout documents features is crucial for most existing work on visual documents information extraction. These models however simply equipped position features (i.e., absolute/relative coordinate) but ignore the geometry of neighbourhood and geometry distance information. Inevitably, simply combining position features with neural models may help little with 2D semantic context, as the layout of the different document varies a lot. And in many cases, text segments should be merged to represent a complete named entity. Thus we are thus motivated to look into the relative geometry of neighbourhood, exploring how to integrate geometry distance to build better 2D semantic context of each text segments and researching on how to merge text segments into complete named entity in visual documents.

3 Proposed Model

3.1 Overview

Text segments (characters in text segments and the bounding box position coordinates of the text segments) of a visual document are acquired by an OCR engine. Mathematically, let a visual document be $D = (t_0, t_1, ..., t_n)$, where t_i stands for a text segments and n is the number of text segments in the visual document D. An overview of our proposed model is illustrated in Figure 2. Firstly, we model a visual document D as a weighted fully connected graph by a graph convolution network encoder into geometry&2D context aware hidden representations, where each text segments t_i is the node of the graph. Secondly, given these hidden representations, a *merge layer* is applied to infer text segments merge decision. Lastly, we combine the graph hidden representations with merge information to reconstruct the

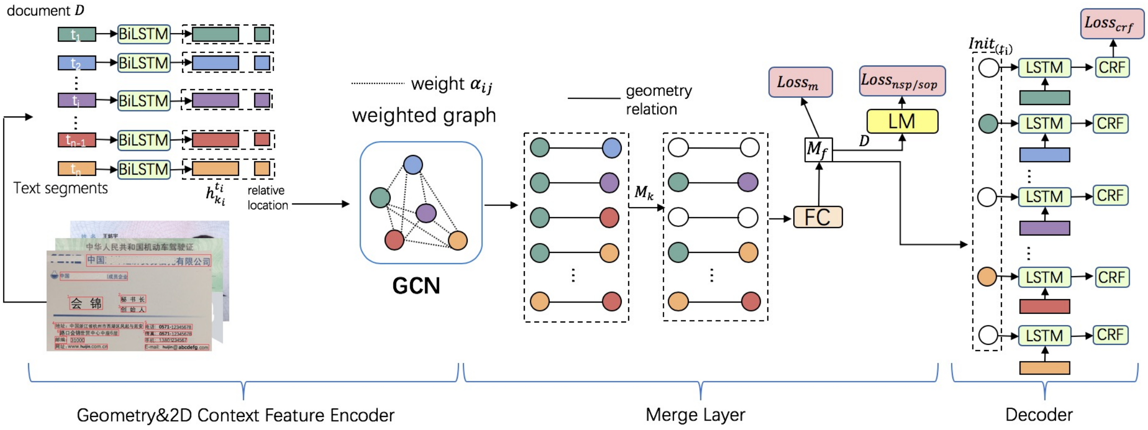


Figure 2: GraphNEMR Architecture

sequential front state of text segments and apply LSTM+CRF for named entity recognition.

3.2 Geometry&2D Context Feature Encoder

Given a text segment t_i, let the text in t_i be $S_{t_i} = (w_1^i, w_2^i, ..., w_{k_i}^i)$ where k_i stands for the length of S_{t_i} and w_j^i is the j-th character in sequence S_{t_i}.We first use Bi-LSTM to calculate sequential text embeddings:

$$h_{1:k_i}^{(t_i)} = BiLSTM(S_{t_i}) \tag{1}$$

where $h_{1:k_i}^{(t_i)}$ denotes the hidden states. We then use the last hidden states $h_{k_i}^{(t_i)}$ to represent the text sequence. To encode basic 2D information, relative coordinates and relative text segment size are concatenated to text embeddings. So the hidden representation node t_i is defined as follows,

$$e_{(t_i)} = [h_{k_i}^{(t_i)}, \frac{x^{(t_i)} - x_{min}}{s}, \frac{y^{(t_i)} - y_{min}}{s}, \frac{w^{(t_i)}}{s}, \frac{h^{(t_i)}}{s}] \tag{2}$$

where $x^{(t_i)}$ and $y^{(t_i)}$ are x-coordinate and y-coordinate of text segments respectively, x_{min} and y_{min} are the circumscribed square's minimum xy-coordinates of all text segments' bounding boxes, s is the side length of the circumscribed square.

Then, a graph convolution is applied to capture 2D context and geometry features from input embeddings that contain text and position information. Intuitively, from the perspective of 2D context, the closer the distance between text segments, the stronger the relevant information they represents. Different from exist gcn-based methods that use mean/max aggregation, to better build a 2D context with geometry information considered, we utilize the distance between text segments as a *weighted aggregation information*. For node t_i, our model retrieves new node features as follows,

$$g_{t_i}^{l+1} = ReLU(\sum_{t_j \in D} \alpha_{ij}(W^{l+1} g_{t_j}^{l} + b^{l+1})) \tag{3}$$

where $g_{t_j}^l \in R^f$ denotes the hidden feature of node t_j at layer l, W^{l+1} and b^{l+1} are learnable weights. Our model aggregates information from the neighbors of each node by the weight α_{ij} that is denoted as

$$\alpha_{ij} = \frac{(s - d_{ij})}{s} \tag{4}$$

where the $d_{ij} = |t_i, t_j|$ is the geometry distance between t_i and t_j. Intuitively, the closer the distance between t_i and t_j, the greater the value of α_{ij}. Thus, by using this weighted aggregation, closer relevant information between text segments can be encoded.

Since our GCN layer propagates information between nodes by every connected nodes with distance-based weight to construct 2D context. And the node embedding consist of semantics and geometry information. After we do graph convolution by L layers, each node t_i can capture both 2D context and geometry information. So, given document D, after getting each node hidden representations by our encoder, we get tensor D_g of shape $[n, f]$,

$$D_g = [g^L_{t_0}, g^L_{t_1}, ..., g^L_{t_n}] \tag{5}$$

3.3 Merge Layer

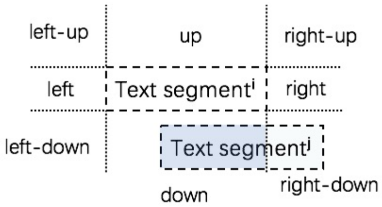


Figure 3: 8-geometry neighbors relation from Text segment^i (t_i) to Text segment^i (t_j)

The merge layer is responsible for merging text segments into a complete entity. Intuitively, one text segment can only be merged with its nearest text segments of 8-geometry neighborhoods. We obtain relative position between two text segments by 8-geometry neighbors. Given a text segment t_i with its bounding box area p^{t_i}, its 8-geometry neighbors areas are defined as a set $P^{t_i} = \{p^{ti}_{left-up}, p^{ti}_{up}, p^{ti}_{right-up}, p^{ti}_{right}, p^{ti}_{right-down}, p^{ti}_{down}, p^{ti}_{left-down}, p^{ti}_{left}\}$ where represent the left-up, up, right-up, right, right-down, down, left-down and left area of t_i respectively in visual document D. Given another text segment t_j with its bounding box area p^{t_j}, then the geometry position relation $p_{t_i:t_j}$ can be denoted by a 9-dim-one-hot vector where the first eight dimensions stand for 8-geometry neighbors and the last dimension represent the self-area of given text segments. Notice that p^{t_j} may intersect with more than one area in P^{t_i}, we choose the area which has the largest intersection with p^{t_j}. For example in Figure 3, the size of the intersecting area between text segments t_j and p^{ti}_{down} is larger than the others, the relation from t_i to t_j is *down*. Following encoder, by tiling and expanding dim on D_g with $p_{t_i:t_j}$ being added, we have

$$\mathbf{D_M} = \begin{bmatrix} [m_{00}] & [m_{01}] & \cdots & [m_{0n}] \\ [m_{10}] & [m_{11}] & \cdots & [m_{1n}] \\ \vdots & \vdots & \ddots & \vdots \\ [m_{n0}] & [m_{n1}] & \cdots & [m_{nn}] \end{bmatrix} = [[m_{ij}]] \tag{6}$$

$$m_{ij} = [g^L_{t_i}, p_{t_i:t_j}, g^L_{t_j}] \tag{7}$$

where $\mathbf{D_M} \in R^{n \times n \times (2f+9)}$ denotes all nodes pair features, $m_{ij} \in R^{2f+9}$ is the concatenation of $g^L_{t_i}$, $p_{t_i:t_j}$ and $g^L_{t_j}$. Then, a fully-connected network with sigmoid activation function is applied to learn a merge matrix M^f as

$$M^f = FC(D_M) \tag{8}$$

where $M_f \in R^{n \times n}$ represents whether two text segments should be merged and which text segment is the front segment. Here, the merge decisions are trained using cross entropy (CE) loss:

$$Loss_m = CE(M^f \cdot M^k, M^f_{label}) \tag{9}$$

where M^f_{label} is the label of M^f with binary value of 0 or 1. M^k is also a binary matrix where $M^k[i,j] == 1$ means that t_j is one of the top k nearest text segments from one of t_i's 8-geometry neighbors. By doing $\cdot$ (dot product operation) between M^f and M^k, only the top k nearest text segments in each 8-geometry neighbors can be merged. During inference, for example, if $M^f[i,j] == 1$, it means that text segment t_i should be merged with t_j and t_i is in front of t_j. $M^f[i,j] == 0$ means t_i and t_j should not be merged.

To leverage sequential language semantics, inspired by the next sentence prediction (NSP) training in BERT (Devlin et al., 2019) and sentence-order prediction (SOP) training in ALBERT (Lan et al., 2019), we propose a loss called $Loss_{nsp/sop}$ for semantically supervising merging text segments as follows,

$$Loss_{nsp/sop} = -LM_{nsp/sop}(D \times M^f) \tag{10}$$

where $LM_{nsp/sop}$ is the pre-trained language model that use NSP or SOP training. × represents matrix multiplication. The language model's parameters are fixed during training. By doing matrix multiplication between D and M^f, we can get the pair that our model hope to be merged in equation (8). The BERT model then get the pairwise input and by maximizing $Loss_{nsp/sop}$, the parameters of our model will be upgraded to make the merge decision in language model's perspective.

3.4 NER Decoder

The last NER Decoder is for named entity tagging. The structure is a standard LSTM+CRF. But different from previous works that use LSTM+CRF for tagging, we utilize the front text segment information that we get from *Merge Layer* as an initial state $Init_{(t_i)}$ for LSTM+CRF, for every node in D,

$$h^{t_i}_{lstm} = LSTM(u^{(t_i)}_{1:k_i}, Init_{(t_i)}) \tag{11}$$

$$u^{(t_i)}_{1:k_i} = [h^{t_i}_1 | g^L_{t_i}, h^{t_i}_2 | g^L_{t_i}, ..., h^{t_i}_{k_i} | g^L_{t_i}] \tag{12}$$

where $h^{t_i}_{lstm}$ is the hidden state of LSTM.| is the concatenate operation. $Init_{(t_i)}$ is the initial state for the LSTM that is denoted as follows,

$$Init_{(t_i)} = \begin{cases} g^L_{t_j} & \text{if}(M^f[i,j] == 1) \\ \\ 0 & \text{otherwise}, \end{cases} \tag{13}$$

where we can easily get the front text segment by doing matrix multiplication between D_g and M^f to get $Init_{(t_i)}$.

Then, a conditional random fields (CRF) is applied to perform entity tagging,

$$Loss_{crf} = CRF(h^{t_i}_{lstm}) \tag{14}$$

Finally, the objective function to be optimized is as follows,

$$Loss = Loss_m + Loss_{nsp/sop} + Loss_{crf} \tag{15}$$

In this way, the geometry information and 2D context is encoded into to our model's hidden layer, and with the *Merge Layer* and the last decoder, our model perform merge and recognize in visual documents.

4 Experiments

We first introduce the datasets for evaluating our proposed model. Then we describe baselines we compared with, the evaluation metrics and briefly explain our implementation details. Next, we show the results for two datasets. Finally, we demonstrate the improved effect of our model via ablation study.

4.1 Dataset

The aim of ICDAR 2019 SROIE task3[1] is to extract different kinds of text of several keys which are *company, address, date and total* from given receipts. The SROIE dataset consists of 1,000 scanned receipt images. Since the annotation of this task is incomplete and not well applied to our problem, we relabeled all the named entities in this dataset and get the text and corresponding bounding boxes

[1] https://rrc.cvc.uab.es/?ch=13&com=evaluation&task=3

according to the ground truth OCR annotations of SROIE. We build the dataset as SROIE-VNER. Our goal is to extract all the named entities (location, organization, date) in SROIE's receipt images. For our relabeled SROIE-VNER dataset, we split the dataset in 70% for training, 30% for testing.

The BCD dataset consists of 13,498 real-world business card images that is much larger than SROIE-VNER dataset. The collection is provided by user-uploading. We get the text and corresponding bounding boxes with Alibaba's OCR API[2]. Each character in text is manually labelled with B/I/E named entity tagging. Our goal is to extract all named entities (person, organization, location) in business card images. Business card styles of different companies are usually different and they are in large layout variability. So the layout of the images in the BCD dataset is more diverse than the SROIE-VNER dataset. 80% images in BCD dataset are used as training data. The left 20% images in BCD dataset are used for testing.

4.2 Baselines and Evaluation Metrics

We implement BiLSTM-CRF as a sequential tagger baseline as many researchers do in invoice/receipt images information extraction. According to (Palm et al., 2017) and (Liu et al., 2018), text segments in a document are concatenated from left to right and from top to bottom. And then they apply BiLSTM-CRF model to the concatenated document. We also compared our model to a graph-based tagging model GraphIE (Qian et al., 2019) which is probably the SOTA graph-based model in visual information extraction.

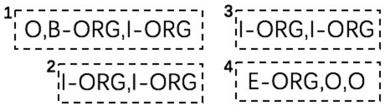


Figure 4: Example text segments in a visual document

The evaluation metrics are the standard named entity recognition precision, recall and F1 score. However, even if the tags of each text segments are completely correct, it cannot achieve extracting the complete entity correctly due to the order of the text segments. The traditional NER CoNLL evaluation method can not cover this problem. For example in Figure 4, assuming that entity tags in text segments 1-4 are correct. It is difficult for humans to determine whether the order (1,2,3,4) or the order (1,3,2,4) is right. And without the right order, we can not extract the right complete named entity. To address this problem, we evaluate the precision, recall and F1 score on complete entities recognition.

4.3 Implementation Details

We calculate the distance between text segments and determine whether the two regions intersect by the Shapely Python package[3]. For LSTM in our model, the dimension of hidden state is 256. The 300-dimensional pre-trained fasttext English word embeddings are used in SROIE-VNER experiments and 300-dimensional pre-trained fasttext Chinese character embeddings are used in BCD experiments. We use an one-layer GCN that is the same with GraphIE and the hidden size is 256. For language model for supervision, we utilize the sentence-order prediction (SOP) of ALBERT.

4.4 Results

Table 1 shows the precision, recall, and F1 score of the tagging results in SROIE-VNER dataset and BCD dataset for BiLSTM-CRF, GraphIE, and GraphNEMR. In SROIE-VNER dataset, both graph-based model GraphIE and our GraphNEMR have over 5.7% improvement compared to the sequential-based BiLSTM-CRF model. But from named entity character tagging results, GraphNEMR* dose not have significant improvement over GraphIE. In BCD dataset, which has a large diversity of layouts, GraphNEMR further surpasses GraphIE by 5.34% and yields 10.60% improvement over BiLSTM-CRF.

[2] `https://duguang.aliyun.com/`

[3] `https://shapely.readthedocs.io/en/stable/manual.html`

dataset	Entity	BiLSTM-CRF			GraphIE			GraphNEMR*		
		P	R	F1	P	R	F1	P	R	F1
SROIE-VNER	LOC	86.75	86.18	86.46	91.42	94.43	**92.90**	89.88	94.43	92.10
	ORG	75.71	75.71	75.71	80.95	86.29	**83.54**	84.38	82.23	83.29
	DATE	94.15	84.73	89.19	87.18	89.47	88.31	90.96	90.00	**90.48**
	ALL	85.37	82.08	83.69	87.70	91.18	89.40	88.81	90.28	**89.54**
BCD	PER	85.35	88.72	87.00	83.66	94.14	88.59	94.32	93.71	**94.02**
	LOC	80.67	85.47	83.00	84.42	95.59	89.66	93.84	94.12	**93.98**
	ORG	69.75	70.93	70.34	74.10	79.58	76.74	83.17	82.46	**82.81**
	ALL	76.92	79.60	78.24	79.58	87.82	83.50	89.05	88.63	**88.84**

Table 1: Visual documents named entity character tagging results. For a fair comparison, here we use GraphNEMR* that removes the pre-trained language model $Loss_{nsp/sop}$ supervision in equantion (10).

dataset	Entity	BiLSTM-CRF			GraphIE			GraphNEMR*		
		P	R	F1	P	R	F1	P	R	F1
SROIE-VNER	LOC	86.75	86.18	86.46	55.33	42.03	47.77	90.53	94.43	**92.44**
	ORG	75.71	75.71	75.71	80.95	86.29	**83.54**	84.38	82.23	83.29
	DATE	94.15	84.73	89.19	88.54	89.47	89.01	90.96	90.00	**90.48**
	ALL	85.37	82.08	83.69	72.08	64.71	68.19	89.14	90.28	**89.71**
BCD	PER	85.35	88.72	87.00	83.11	92.07	87.36	94.28	93.68	**93.98**
	LOC	80.67	85.47	83.00	74.53	84.05	79.00	93.64	93.86	**93.75**
	ORG	69.70	70.89	70.29	72.70	77.02	74.80	82.23	80.73	**81.48**
	ALL	76.90	79.58	78.22	75.95	82.86	79.25	88.58	87.75	**88.16**

Table 2: Visual documents NER results, evaluating on complete entities recognition.

Table 2 presents the comparisons of our model with the sequence-based model and the graph-based model on complete named entities recognition. Intuitively, this is a more suitable evaluation method for visual documents related tasks. Since GraphIE model doesn't have the ability to merge text segments, on this evaluating method, the LOC result of GraphIE has dropped significantly in SROIE-VNER dataset. Sequence-based model BiLSTM-CRF merges the text from left-to-right and from up-to-down and our model GraphNEMR* learns to merge. GraphNEMR* significantly outperforms the sequence-based model BiLSTM-CRF by 6.02% and the graph-based model GraphIE by 21.52% separately. In BCD dataset, GraphNEMR also obtains significant improvements over BiLSTM-CRF by 9.94% and GraphIE by 8.91%.

4.5 Analysis

From the dataset perspective, the layout of SROIE-VNER dataset is relatively simple. But many text segments need to be merged. The BCD dataset is quite different from SROIE-VNER. Since the style of each business card image is quite different, the BCD dataset has large layout varieties and also many text segments that need to be merged.

Since the sequence-based BiLSTM-CRF concatenate text segments in a document based on a common order which many layouts follow in real-world situations. So in SROIE-VNER dataset that is in a relatively simple layout, no matter what evaluation methods, the sequence-based BiLSTM-CRF can achieve relatively stable and comparable results. But the order that is from left-to-right and from top-to-bottom may not be guaranteed. The sequence order of entities like *location* and *organization* that often appear as multiple lines or multiple text segments are broken by such concatenations. So in BCD dataset that is in large varieties layouts, the performance of the sequence-based model suffers from significant performance degradation. Entities that usually have a short text length and in most cases are in left-to-right order in single text segment, i.e. *person*, are not influenced by concatenations and can keep a relatively stable performance in different layout varieties.

GraphNEMR	P	R	F1
k=1	86.57	85.49	86.03
k=2	88.33	89.21	88.77
k=3	87.70	90.10	**88.88**
k=$+\infty$	86.62	88.61	87.60

Table 3: Results of different *k* nearest text segments from 8-geometry neighbors. $+\infty$ means using all text segments from the document.

Model	F1
GraphNEMR-FULL	88.88
- relative location	88.49($\downarrow$ 0.39)
- merge with geometry	87.48($\downarrow$ **1.4**)
- $Loss_{nsp/sop}$ supervise	88.16($\downarrow$ 0.72)

Table 4: Ablation study ("-" means removing the sub-component from GraphNEMR.

In BCD dataset, visual documents have diverse layouts and many text segments need to be merged. Because graph-based models take the visual layout into account, both GraphIE and GraphNEMR* achieve better results on visual documents with diverse layout changes than sequence-based model BiLSTM-CRF. So under these circumstances, the gap between our proposed GraphNEMR* and GraphIE mainly comes from the geometry features. While the visual documents in SROIE-VNER have simple layouts and a lot of text segments need to be merged, both GraphNEMR* and GraphIE have a better performance than BiLSTM-CRF in character tagging results. Since GraphIE doesn't have the ability of mergence and BiLSTM-CRF uses a naive merging strategy, GraphNEMR* performs much better than other models.

We also evaluate the impact of different numbers of *k* nearest text segments from 8-geometry neighbors. In theory, a text segment wouldn't be merged with a long-distance text segment. And a text segment will not only be merged with its nearest neighbors. Table 3 presents the results of using different *k* nearest text segments from 8-geometry neighbors. As we can see, using all text segments as candidates do not help the merging task and only setting the nearest neighbors as merging candidates will hurt the model performance.

4.6 Ablation Study

To better understand the contributions of each sub-component of GraphNEMR, we perform ablation studies in BCD dataset. Table 4 presents the results. In each study, we exclude the relative location, geometry information in merge layer and the use of the pre-trained ALBERT language model as $Loss_{nsp/sop}$ to supervise sentence order respectively. As we can see that the geometry information plays a more important role than others. The semantic $Loss_{nsp/sop}$ is also very helpful for recognize complete named entities. Intuitively, with the 8-geometry neighbors information considered, a richer 2D context and layout information is provided to better merge and recognize named entities.

5 Conclusions

In this paper, we propose GraphNEMR, a graph-based model that uses graph convolutional networks to jointly merge text segments and recognize named entities from visual documents. We model the visual document as a graph and incorporate geometry information into graph convolution network to build richer 2D context. To address the problem that text segments need to be merged into a complete entity, we combine geometry features with semantic features for learning to merge and recognize named entities. We evaluate our model on relabeled SROIE-VNER dataset and a real-world BCD dataset. Results show that our model outperforms sequence-based model (BiLSTM-CRF) and graph-based model (GraphIE) for named entity recognition from visual documents.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Timo I Denk and Christian Reisswig. 2019. Bertgrid: Contextualized embedding for 2d document representation and understanding.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 141–150. Association for Computational Linguistics.

Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested ner. In *ACL*.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418.

Rinon Gal, Nimrod Morag, and Roy Shilkrot. 2018. Visual-linguistic methods for receipt field recognition. In *Asian Conference on Computer Vision*, pages 542–557. Springer.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459.

Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards understanding 2d documents. In *EMNLP*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Sai Chandra Kosaraju, Mohammed Masum, Nelson Zange Tsaku, Pritesh Patel, Tanju Bayramoglu, Girish Modgil, and Mingon Kang. 2019. Dot-net: Document layout classification using texture-based cnn. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1029–1034. IEEE.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Anh Duc Le, Dung Van Pham, and Tuan Anh Nguyen. 2019. Deep learning approach for receipt recognition. In *International Conference on Future Data and Security Engineering*, pages 705–712. Springer.

Xiaojing Liu, Feiyu Gao, Qiong Zhang, and Huasha Zhao. 2018. Graph convolution for multimodal information extraction from visually rich documents. In *NAACL*.

Devashish Lohani, A Belaïd, and Yolande Belaïd. 2018. An invoice reading system using a graph convolutional network. In *Asian Conference on Computer Vision*, pages 144–158. Springer.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *AAAI*.

Rasmus Berg Palm, Ole Winther, and Florian Laws. 2017. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 406–413. IEEE.

Rasmus Berg Palm, Florian Laws, and Ole Winther. 2018. Attend, copy, parse-end-to-end information extraction from documents. In *arXiv preprint arXiv:1812.07248*.

Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. 2019. Graphie: A graph-based framework for information extraction. In *NAACL*.

Rizlene Raoui-Outach, Cecile Million-Rousseau, Alexandre Benoit, and Patrick Lambert. 2017. Deep learning for automatic sale receipt understanding. In *2017 Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *EMNLP*.

Bailin Wang and Wei Lu. 2019. Combining spans into entities: A neural two-stage approach for recognizing discontiguous entities.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2019. Layoutlm: Pre-training of text and layout for document image understanding. *arXiv preprint arXiv:1912.13318*.

Xiao Yang, Ersin Yumer, Paul Asente, Mike Kraley, Daniel Kifer, and C Lee Giles. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5315–5324.

Xiaohui Zhao, Zhuo Wu, and Xiaoguang Wang. 2019. Cutie: Learning to understand documents with convolutional universal text information extractor. *arXiv preprint arXiv:1903.12363*.