Attention-Based Graph Neural Network with Global Context Awareness for Document Understanding

Yuan Hua[1], Zheng Huang[1,2(✉)], Jie Guo[1], and Weidong Qiu[1]

[1] Shanghai Jiao Tong University, Shanghai, China
{isyuan.hua,huang-zheng,guojie,qiuwd}@sjtu.edu.cn
[2] Westone Cryptologic Research Center, Beijing, China

Abstract. Information extraction from documents such as receipts or invoices is a fundamental and crucial step for office automation. Many approaches focus on extracting entities and relationships from plain texts, however, when it comes to document images, such demand becomes quite challenging since visual and layout information are also of great significance to help tackle this problem. In this work, we propose the attention-based graph neural network to combine textual and visual information from document images. Moreover, the global node is introduced in our graph construction algorithm which is used as a virtual hub to collect the information from all the nodes and edges to help improve the performance. Extensive experiments on real-world datasets show that our method outperforms baseline methods by significant margins.

Keywords: Document understanding · Attention · Graph neural network

1 Introduction

Information Extraction [1,10,21] is a widely studied task of retrieving structured information from texts and many inspiring achievements have been made in this field. However, most of these works are generally focusing on extracting entities and relationships from plain texts which are not appropriate to apply directly on document understanding.

Document understanding is the process of automatically recognizing and extracting key texts from scanned unstructured documents and saving them as structured data. Document understanding was already introduced in a competition of ICDAR 2019, where the goal was to detect texts in documents and extract key texts from receipts and invoices. In this work, we focus on document understanding which is mainly about key information extraction from scanned unstructured documents. The following paragraphs summarize the challenges of the task and the contributions of our work.

M. Sun et al. (Eds.): CCL 2020, LNAI 12522, pp. 45–56, 2020.
https://doi.org/10.1007/978-3-030-63031-7_4

1.1 Challenges

Document understanding is a challenging task and there are little research works published in this topic so far. Although it seems that traditional named entity recognition networks or layout analysis networks are related to this topic, none of the existing research can fully address the problems faced by document understanding.

Firstly, context requires balance. The key cue of the entities usually appears in their neighbors and too much context will add noise and increase problem dimensionality making learning slower and more difficult. As shown in Fig. 1, in order to identify the label of *$11900*, the text *Total* on its left side is good enough for the model to recognize its tag correctly. Instead of increasing the recognition accuracy, too much context like *Tax*, *Subtotal* will lead the performance even worse. Appropriate context is very problem specific and we need to get this relationship by training.

Secondly, it is not adequate to represent the semantic meaning in documents by using text alone. For example, there can be multiple date related entities in one document such as *due date* and *purchase date*. It is difficult for the model to distinguish them only by textual information. Thus, more information like visual information or layout information also needs to be considered at the same time.

Thirdly, the positional cue is critical sometimes. An example is shown in the right side of Fig. 1. As for the entity *Vender Name*, it appears at the top of the document in most cases. The model will benefit from it if it can leverage this information.

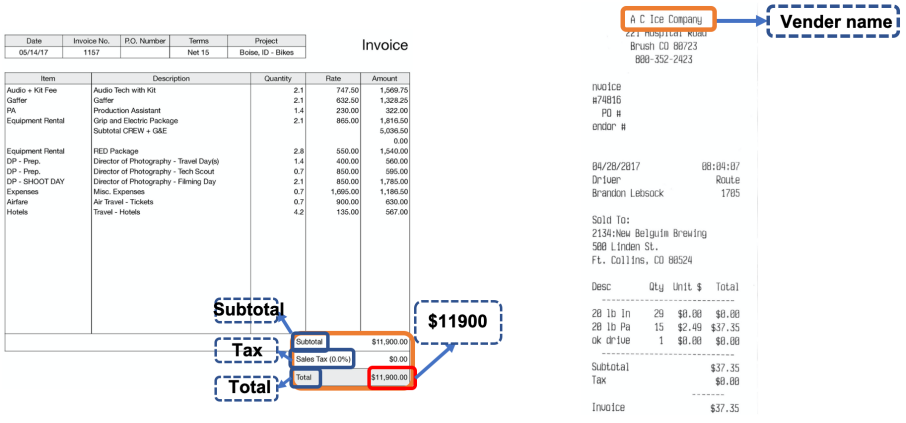

Fig. 1. Examples of Documents and example entities to extract.

1.2 Contributions

In this work, we present a novel method that achieves the document understanding problem as a node classification task. The method first computes a text embedding and an image embedding for each text segment in the document. Then graph construction algorithm will use the coordinates of bounding boxes to generate a unique graph for each document. In order to leverage positional cue effectively, the global node is first proposed in document understanding field which represents the universal context of the current document. Finally, the graph attention network will combine textual information with visual information and the positional cue for information extraction.

The main contributions of this paper can be summarized as follows: 1) we propose a graph construction algorithm to generate a unique graph for each document and achieve the document understanding task as a graph node classification task; 2) the proposed model can capture global context information and local compositions effectively; 3) extensive experiments have been conducted on real-world datasets to show that our method has significant advantages over the baseline methods.

2 Related Works

Several rule-based document understanding systems were proposed in [2,3,14]. Laura et al. [2] presented a case for the importance of rule-based approaches to industry practitioners. SmartFix by Andreas et al. [3] employs specific configuration rules designed for each template. The study by Schuster et al. [14] offers a template matching based algorithm to solve the document understanding problem and plenty of templates have to be constructed and maintained to deal with different situations. However, rule-based methods rely heavily on the predefined templates or rules and are not scalable and flexible for most document understanding problems since documents in real life have no fixed layout. Furthermore, updating the templates or rules requires a lot of effort.

A recent study by Zhao et al. [20] proposed Convolutional Universal Text Information Extractor (CUTIE). CUTIE treats the document understanding task as an image semantic segmentation task. It applies convolutional neural networks on gridded texts where texts are semantical embeddings. However, this work only uses text-level features and doesn't involve image-level features.

Inspired by BERT [4], Xu et al. [18] proposed LayoutLM method. It applies BERT architecture for the pre-training of text and layout. Although LayoutLM uses image features in the pre-training stage and it performs well on several downstream tasks, the potential relationship between two text segments hasn't been taken into consideration. In addition, sufficient data and time are required to pre-train the model inefficiently.

Since graph neural networks [9,13,17] have shown great success in unstructured data tasks, more and more research works are focusing on using GNN to tackle the document understanding problem. Liu et al. [11] presented a GCN-based method for information extraction from document images. It is a work

attempting to extract key information with customized graph convolution model. However, prior knowledge and extensive human efforts are needed to predefine task-specific node and edge representations. One study by Yu et al. [19] explores the feature fusion of textual and visual embeddings by GNN. This work differs from ours because it still treats the document understanding task as the sequence tagging problem and uses a bi-directional LSTM model to extract entities which has already been proved to have limited ability to learn the relationship among distant words.

3 Proposed Method

This section demonstrates the architecture of our proposed model. To extract textual context, our model first encodes each text segment in the document by pre-trained BERT model as its corresponding text embeddding. Then using multiple layers of CNN to get its image embedding. The combination of these two types of embeddings will generate unique global node representation and various local node representations. These node representations contain both visual context and textual context and will be used as node input to the graph attention network. Our model transforms the document understanding task into a node classification problem by taking both local context and global context into account.

3.1 Feature Extraction

Figure 2 is the overall workflow of feature extraction. As shown in Fig. 2, we calculate node representations for both global nodes and local nodes where global nodes capture universal information and local nodes extract internal information. Different from the existing information extraction models that only use plain text features, we also use image features to obtain morphology information to our model.

Text Feature Extraction. We use pre-trained BERT model to generate text embeddings for capturing both global and local textual context. For a set of text segments in the document, we concatenate them by their coordinates from left to right and from top to bottom to generate a sequence. Given a sequence $seq_i = (w_1^{(i)}, w_2^{(i)}, ..., w_n^{(i)})$, text embeddings of a sequence seq_i are defined as follows

$$TE_{0:n}^{(i)} = BERT(w_{0:n}^{(i)}; \Theta_{BERT}) \tag{1}$$

where $w_{0:n}^{(i)} = [w_0^{(i)}, w_1^{(i)}, ..., w_n^{(i)}]$ denotes the input sequence padding with $w_0^{(i)} = [CLS]$. $[CLS]$ is a specific token to capture full sequence context which is introduced in [4]. $TE_{0:n}^{(i)} = [TE_0^{(i)}, TE_1^{(i)}, ..., TE_n^{(i)}] \in \mathbf{R}^{n*d_{model}}$ denotes the output sequence embeddings and d_{model} is the dimension of the model. $TE_k^{(i)}$ represents the k-th output of pre-trained BERT model for the i-th document.

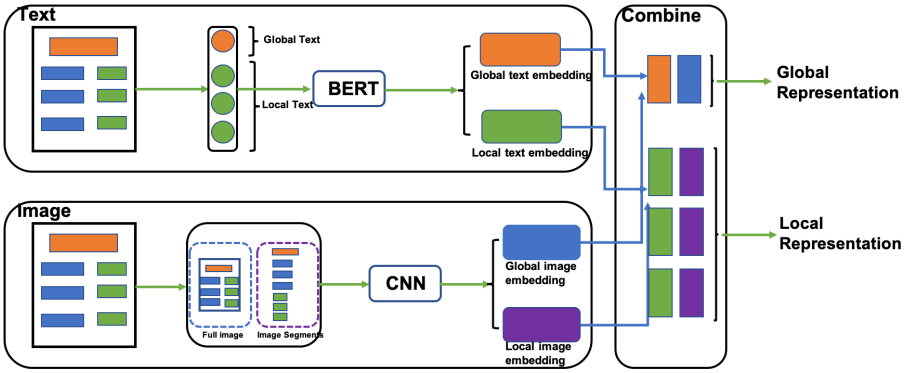

Fig. 2. Workflow of feature extraction.

Θ_{BERT} represents the parameters of pre-trained BERT model. Each text segment of a text sequence is encoded independently and we can get global text embedding and local text embedding simultaneously, defining them as

$$TE^{(i)}_{Global} = TE^{(i)}_{0} \tag{2}$$

$$TE^{(i)}_{Local} = [TE^{(i)}_{1}, TE^{(i)}_{2}, ..., TE^{(i)}_{n}] \tag{3}$$

Image Feature Extraction. For image embedding generation, we using CNN for catching both global and local visual information. Given a set of image segments cropped by bounding boxes $seg_i = (p^{(i)}_1, p^{(i)}_2, ..., p^{(i)}_n)$, image embeddings of segments seg_i are defined as follows

$$IE^{(i)}_{0:n} = CNN(p^{(i)}_{0:n}; \Theta_{CNN}) \tag{4}$$

where $p^{(i)}_{0:n} = [p^{(i)}_0, p^{(i)}_1, ..., p^{(i)}_n]$ denotes the input image segments appending with $p^{(i)}_0 = full\_image$. We use $p^{(i)}_0$ to capture global morphology information of the document image. $p^{(i)}_k \in \mathbf{R}^{H*W*3}$ represents k-th image segment of i-th document and H means height of the image, W means width of the image. $IE^{(i)}_{0:n} = [IE^{(i)}_0, IE^{(i)}_1, ..., IE^{(i)}_n] \in \mathbf{R}^{n*d_{model}}$ denotes the output image embeddings and d_{model} is the dimension of the model. In our work, we use classic ResNet model [6] as backbone to extract image features and a full connected layer is used to resize output to d_{model} dimension. $IE^{(i)}_k$ represents the k-th output of CNN model for the i-th document. Θ_{CNN} represents the parameters of CNN model. Each image segment is encoded independently and we can get global image embedding and local image embedding synchronously, defining them as

$$IE^{(i)}_{Global} = IE^{(i)}_{0} \tag{5}$$

$$IE^{(i)}_{Local} = [IE^{(i)}_{1}, IE^{(i)}_{2}, ..., IE^{(i)}_{n}] \tag{6}$$

Combination. After text feature extraction and image feature extraction, we can concatenate these features into a new representation RE, which will be used as node input to the graph neural network. $\oplus$ in the formula means concatenation operation.

$$RE^{(i)}_{Global} = TE^{(i)}_{0} \oplus IE^{(i)}_{0} \tag{7}$$

$$RE^{(i)}_{Local} = TE^{(i)}_{1:n} \oplus IE^{(i)}_{1:n} \tag{8}$$

3.2 Graph Construction

In order to capture relative positional information, we use the coordinates of bounding boxes to connect text segments. Inspired by Gui et al. [5], we propose the global node mechanism which is used as a virtual hub to capture long-range dependency and high-level features.

The whole document is converted into a directed graph, as shown in Fig. 3, where each node represents a text segment and the connection between two nodes can be treated as an edge. Given a set of text segments inside a document, first of all, we need to merge these text segments into different lines based on their bounding boxes' coordinates. To be more specific, if the overlap of the two text segments on the vertical axis exceeds 60%, the two text segments are considered to belong to the same line. In order to capture layout information, we build connection for each text segment in the same line. In addition, an extra connection is built between current text segment and every text segments in its previous line.

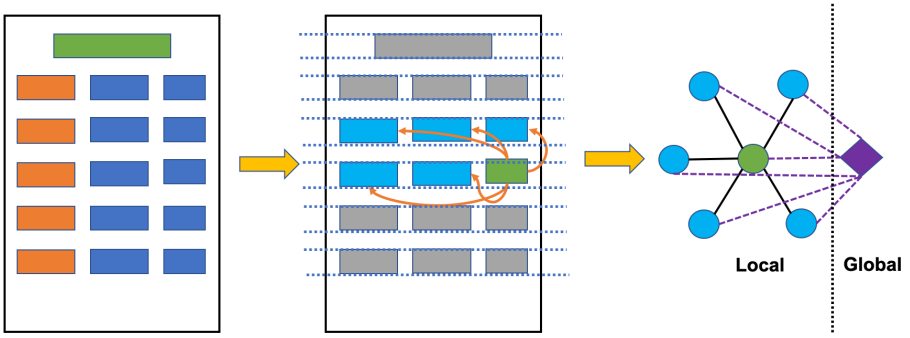

Fig. 3. Illustration of graph construction.

To capture global information, we add a global node to connect each local node. The global node is used as a virtual hub to collect universal information from all the nodes inside the graph. Since all internal nodes are connected with global node which means every two non adjacent nodes are two-hop neighbors, universal information can be distributed to these local nodes through such connections.

3.3 Recurrent-Based Aggregate and Update

Attention-based graph neural network [17] is applied to fuse multiple information in the graph, as shown in Fig. 4. In our model, graph convolution is defined based on the self-attention mechanism and aggregation and update of global node and local node are treated equally.

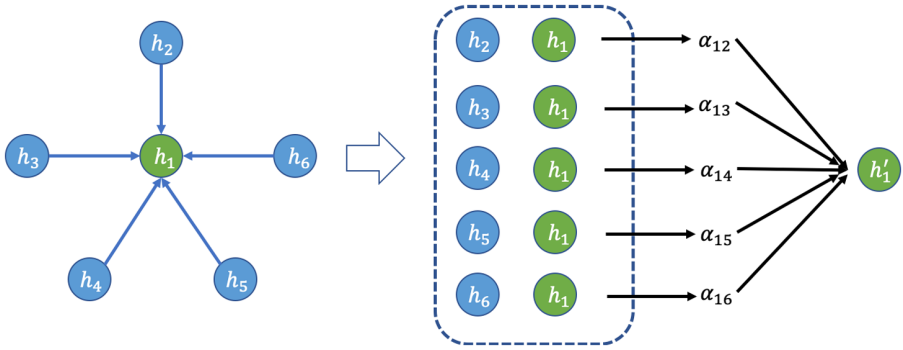

Fig. 4. Aggregation in Graph Neural Network.

Given a node v_i and its hidden state h_i which is initialized by RE, the output embedding of node v_i can be calculated by self-attention mechanism as the follows

$$\boldsymbol{h}'_i = \sigma(\sum_{j \in N_i} \alpha_{ij} W \boldsymbol{h}_j) \tag{9}$$

where $\boldsymbol{h}'_i$ is the aggregation and update of $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$ is the hidden state of node v_i's neighbour v_j. σ is an activation function and α_{ij} is the attention coefficient which indicates the importance of node j's features to node i. The coefficients computed by the attention mechanism can be expressed as:

$$\alpha_{ij} = \frac{exp(LeakyReLU(V^T[Wh_i \oplus Wh_j]))}{\sum_{k \in N_i} exp(LeakyReLU(V^T[Wh_i \oplus Wh_k]))} \tag{10}$$

where W and V are trainable parameters. We apply the LeakyReLU nonlinearity (with negative input slope $\alpha = 0.2$) to avoid the "dying ReLU" problem.

Similarly to Vaswani et al. [16], we also employ multi-head attention to improve the performance of our model. K attention mechanisms execute independently and their features are concatenated in the end. The final representation is as the follows and $\oplus$ in the formula means concatenation operation:

$$\boldsymbol{h}'_i = \mathop{\Big\Vert}_{k=1}^{K} \sigma(\sum_{j \in N_i} \alpha_{ij}^k W^k \boldsymbol{h}_j) \tag{11}$$

3.4 Decoding and Information Extraction

A conditional random field (CRF) is used to generate a family of conditional probability for the sequence. Given the sequence of final node states $h_{1:n}^{final} = [h_1^{final}, h_2^{final}, ..., h_n^{final}]$, and the probability of a label sequence $\hat{y} = [\hat{l}_1, \hat{l}_2, ..., \hat{l}_n]$ can be defined as the follows

$$p(\hat{y}|s) = \frac{exp(\sum_{i=1}^{n} W_{(l_{i-1},l_i)} h_i^{final} + b_{(l_{i-1},l_i)})}{\sum_{y' \in Y(s)} exp(\sum_{i=1}^{n} W_{(l'_{i-1},l'_i)} h_i^{final} + b_{(l'_{i-1},l'_i)})} \tag{12}$$

where W and b are the weight and bias parameters and $Y(s)$ is the set of all arbitrary label sequences.

Our model parameters of whole networks are jointly trained by minimizing the following loss function as:

$$L = -\sum_{i=1}^{N} log(p(y_i|s_i)) \tag{13}$$

Decoding of CRF layer is to search the output sequence y^* having the highest conditional probability for testing.

$$y^* = \underset{y \in Y(s)}{argmax}\, p(y|s) \tag{14}$$

Viterbi algorithm is used to calculate the above equations, which can improve algorithm operation efficiency.

4 Experiments

We use Pytorch framework to implement our experiments on a GTX 1080Ti GPU and apply our model for information extraction from two real-world datasets.

4.1 Datasets

We conduct experiments on two document understanding datasets. **(1) Contract Dataset:** Contract Dataset is a dataset from Alibaba Tianchi Competition. The dataset contains six types of named entities: Party A, Party B, Project Name, Contract Name, Contract Amount and Consortium Members. This dataset has both the original PDF format documents and annotation files of target named entities. The train set consists of 893 contracts and test set consists of 223 contracts. **(2) SROIE:** SROIE is composed of scanned receipt images and is annotated with 4 types of named entities: Company, Address, Date and Total. The train set consists of 627 receipt images and test set consists of 347 receipt images.

4.2 Implementation Details

We use the Adam [8] as the optimizer, with a learning rate of 3e-6 for all datasets. We employ the Dropout [15] with a rate of 0.5 for node aggregation and update. In the feature extraction part, the text feature extractor is pre-trained BERT model and the hyper-parameter of BERT used in our paper is same as [4]. The dimension of text embedding is 512. The image feature extractor is ResNet-50 model and the hyper-parameter of ResNet-50 used in our paper is same as [6]. We add a full connected layer after ResNet-50 to resize the output dimension to 512. Then the combination of text embeddings and image embeddings is applied as the input of the graph neural network. We apply 3 graph attention layers with 24 multi-heads and the dimension of hidden state is 1024. The standard F1 score is used as evaluation metrics.

4.3 Evaluation

We compare the performance of our model with Bi-LSTM-CRF [7] and BERT-CRF [4]. Bi-LSTM-CRF uses Bi-LSTM architecture to extract text information and a CRF layer to get tags. BERT-CRF applies BERT model as backbone to replace Bi-LSTM model and also a CRF layer after to extract entities. The input text sequence is generated by text segments concatenated from left to right and from top to bottom according to [12].

Table 1. F1-score performance comparisons from contract dataset.

Entities	Bi-LSTM-CRF	BERT-CRF	Our model
Party A	72.2	75.3	79.1
Party B	83.5	84.2	88.4
Project Name	65.6	68.3	74.8
Contract Name	69.2	71.5	80.2
Contract Amount	86.3	89.8	92.3
Consortium Members	45.2	46.1	54.6
Macro Average	70.3	72.5	**78.2**

4.4 Result

We report our experimental results in this section. Table 1 lists the F1 score of each entity of contract dataset. Macro-averages in the last row of the table are the averages of the corresponding columns, indicating the overall performance of each method on all entity types. In the contract scenario, as can be seen from Table 1, our model outperforms Bi-LSTM-CRF by 12% in F1 score and leads to a 8.00% increment of F1 score over BERT-CRF model. Moreover, our model

Table 2. F1-score performance comparisons from SROIE dataset.

Entities	Bi-LSTM-CRF	BERT-CRF	Our model
Company	85.1	86.8	93.5
Address	88.3	89.1	94.6
Date	94.2	96.2	97.3
Total	83.5	84.7	92.1
Macro Average	87.8	89.2	**94.4**

outperforms the two baseline models in all entities. Further analysis shows that our model makes great improvements in those entities like *Contract Name* and *Project Name*. These entities have conspicuous layout features and morphological features which can't be captured by text alone models.

Furthermore, as shown in Table 2, our model shows significant improvement over the baseline methods on SROIE dataset. Compared with the existing Bi-LSTM-CRF model and BERT-CRF model, our model gives the best results by a large margin. These results suggest that, compared to previous text alone methods, our model is able to extract more information from the document to learn a more expressive representation through graph convolutions.

4.5 Ablation Studies

To study the contribution of each component in our model, we conduct ablation experiments on both two datasets and display the results in Table 3. In each study, we exclude visual features and the use of global node respectively, to see their impacts on F1 scores on both two datasets.

Table 3. Ablation studies of individual component.

Configurations	Contract dataset	SROIE dataset
Full model	78.2	94.4
W/o visual feature	75.3	90.1
W/o global node	76.7	92.3

As described in Table 3, when we remove visual features, the result drops to the F1 score of 75.3 on contract dataset and 90.1 on SROIE dataset. This indicates that visual features can play an important role in addressing the issue of ambiguously extracting key information. Furthermore, the results show that the model's performance is degraded if the global node is removed, indicating that global connections are useful in the graph structure.

5 Conclusions and Future Works

This paper studies the problem of document understanding. In this work, we present a novel method that takes global context into account to refine the graph architecture on the complex documents. The explanatory experiments suggest that our proposed model is capable of extracting more information from documents to learn a more expressive representation through attention-based graph convolutions. We hope that our research will serve as a base for future studies on document understanding. Furthermore, we intend to extend our model to other document related tasks, such as document classification or document clustering.

Acknowledgements. This work was supported by The National Key Research and Development Program of China under grant 2017YFB0802704 and 2017YFB0802202.

References

1. Akbik, A., Bergmann, T., Vollgraf, R.: Pooled contextualized embeddings for named entity recognition. 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), pp. 724–728 (2019)
2. Chiticariu, L., Li, Y., Reiss, F.: Rule-based information extraction is dead! long live rule-based information extraction systems! In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 827–832 (2013)
3. Dengel, A.R., Klein, B.: *smartFIX*: a requirements-driven system for document analysis and understanding. In: Lopresti, D., Hu, J., Kashi, R. (eds.) DAS 2002. LNCS, vol. 2423, pp. 433–444. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45869-7_47
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Gui, T., et al.: 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), pp. 1039–1049 (2019)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
7. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991 (2015)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
9. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
10. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360 (2016)
11. Liu, X., Gao, F., Zhang, Q., Zhao, H.: Graph convolution for multimodal information extraction from visually rich documents. arXiv preprint arXiv:1903.11279 (2019)

12. Palm, R.B., Winther, O., Laws, F.: Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 406–413. IEEE (2017)
13. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2008)
14. Schuster, D., et al.: Intellix-end-user trained information extraction for document archiving. In: 2013 12th International Conference on Document Analysis and Recognition, pp. 101–105. IEEE (2013)
15. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
16. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 5998–6008 (2017)
17. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
18. Xu, Y., Li, M., Cui, L., Huang, S., Wei, F., Zhou, M.: Layoutlm: pre-training of text and layout for document image understanding. arXiv preprint arXiv:1912.13318 (2019)
19. Yu, W., Lu, N., Qi, X., Gong, P., Xiao, R.: Pick: processing key information extraction from documents using improved graph learning-convolutional networks. arXiv preprint arXiv:2004.07464 (2020)
20. Zhao, X., Niu, E., Wu, Z., Wang, X.: Cutie: learning to understand documents with convolutional universal text information extractor. arXiv preprint arXiv:1903.12363 (2019)
21. Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P., Xu, B.: Joint extraction of entities and relations based on a novel tagging scheme. arXiv preprint arXiv:1706.05075 (2017)