Graph Convolution for Multimodal Information Extraction from Visually Rich Documents

Xiaojing Liu, Feiyu Gao, Qiong Zhang, Huasha Zhao
Alibaba Group
{huqiang.lxj,feiyu.gfy,qz.zhang,huasha.zhao}@alibaba-inc.com

Abstract

Visually rich documents (VRDs) are ubiquitous in daily business and life. Examples are purchase receipts, insurance policy documents, custom declaration forms and so on. In VRDs, visual and layout information is critical for document understanding, and texts in such documents cannot be serialized into the one-dimensional sequence without losing information. Classic information extraction models such as BiLSTM-CRF typically operate on text sequences and do not incorporate visual features. In this paper, we introduce a graph convolution based model to combine textual and visual information presented in VRDs. Graph embeddings are trained to summarize the context of a text segment in the document, and further combined with text embeddings for entity extraction. Extensive experiments have been conducted to show that our method outperforms BiLSTM-CRF baselines by significant margins, on two real-world datasets. Additionally, ablation studies are also performed to evaluate the effectiveness of each component of our model.

1 Introduction

Information Extraction (IE) is the process of extracting structured information from unstructured documents. IE is a classic and fundamental Natural Language Processing (NLP) task, and extensive research has been made in this area. Traditionally, IE research focuses on extracting entities and relationships from plain texts, where information is primarily expressed in the format of natural language text. However, a large amount of information remains untapped in VRDs.

VRDs present information in the form of both text and vision. The semantic structure of the document is not only determined by the text within it but also the visual features such as layout, tabular structure and font size of the document. Examples
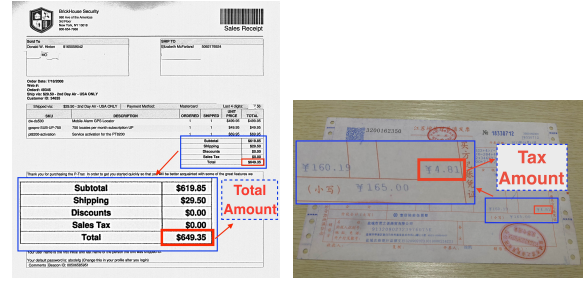
(a) Purchase receipt   (b) Value-added tax invoice

Figure 1: Examples of VRDs and example entities to extract.

of VRDs are purchase receipts, insurance policy documents, custom declaration forms and so on. Figure 1 shows example VRDs and example entities to extract.

VRDs can be represented as a graph of *text segments* (Figure 2), where each text segment is comprised of the position of the segment and the text within it. The position of the text segment is determined by the four coordinates that generate the bounding box of the text. There are other potentially useful visual features in VRDs, such as fonts and colors, which are complementary to the position of the text. They are out of the scope of this paper, and we leave them to future works.

The problem we address in this paper is to extract the values of pre-defined entities from VRDs. We propose a graph convolution based method to combine textual and visual information presented in VRDs. The graph embeddings produced by graph convolution summarize the context of a text segment in the document, which are further combined with text embeddings for entity extraction using a standard BiLSTM-CRF model. The following paragraphs summarize the challenges of the task and the contributions of our work.

1.1 Challenges

IE from VRDs is a challenging task, and the difficulties mainly arise from how to effectively incorporate visual cues from the document and the scalability of the task.

First, text alone is not adequate to represent the semantic meaning in VRDs, and the contexts of the texts are usually expressed in visual cues. For example, there might be multiple dates in the purchase receipts. However, it is up to the visual part of the model to distinguish between the Invoice Date, Transaction Date, and Due Date. Another example is the tax amount in value-added tax invoice, as shown in Figure 1(b). There are multiple "money" entities in the document, and there is a lack of any textual context to determine which one is tax amount. To extract the tax amount correctly, we have to leverage the (relative) position of the text segment and visual features of the document in general.

Template matching based algorithms (Chiticariu et al., 2013; Dengel and Klein, 2002; Schuster et al., 2013) utilize visual features of the document to extract entities; however, we argue that they are mostly not scalable for the task in real-world business settings. There are easily thousands of vendors on the market, and the templates of purchase receipts from each vendor are not the same. Thousands of templates need to be created and maintained in this single scenario. It requires substantial efforts to update the template and make sure it's not conflicting with the rest every time a new template comes in, and the process is error-prone. Besides, user uploaded pictures introduce another dimension of variance from the template. An example is shown in Figure 1(b). Value-added tax invoice is a nation-wide tax document, and the layout is fixed. However, pictures taken by users are usually distorted, often blurred and sometimes contain interfering objects in the image. A simple template-based system performs poorly in such a scenario, while sophisticated rules require significant engineering efforts for each scenario, which we believe is not scalable.

1.2 Contributions

In this paper, we present a novel method for IE from VRDs. The method first computes graph embeddings for each text segment in the document using graph convolution. The graph embeddings represent the context of the current text segment where the convolution operation combines both textual and visual features of the context. Then the graph embeddings are combined with text embeddings to feed into a standard BiLSTM for information extraction.

Extensive experiments have been conducted to show our method outperforms BiLSTM-CRF baselines by significant margins, on two real-world datasets. Additionally, ablation studies are also performed to evaluate the effectiveness of each component of our model. Furthermore, we also provide analysis and intuitions on why and how individual components work in our experiments.

2 Related Works

Our work is inspired by recent research in the area of graph convolution and information extraction.

2.1 Graph Convolution Network

Neural network architectures such as CNN and RNNs have demonstrated huge success on many artificial intelligence tasks where the underlying data has grid-like or sequential structure (Krizhevsky et al., 2012; Kim et al., 2016; Kim, 2014). Recently, there is a surge of interest in studying the neural network structure operating on graphs (Kipf and Welling, 2016; Hamilton et al., 2017), since much data in the real world is naturally represented as graphs. Many works attempt to generalize convolution on the graph structure. Some use a spectrum based approach where the learned model depends on the structure of the graph. As a result, the approach does not work well on dynamic graph structures. The others define convolution directly on the graph (Veličković et al., 2017; Hamilton et al., 2017; Xu et al., 2018; Johnson et al., 2018; Duvenaud et al., 2015). We follow the latter approach in our work to model the text segment graph of VRDs.

Different from existing works, this paper introduces explicit edge embeddings into the graph convolution network, which models the relationship between vertices directly. Similar to (Veličković et al., 2017), we apply self-attention (Vaswani et al., 2017) to define convolution on variable-sized neighbors, and the approach is computationally efficient since the operation is parallelizable across node pairs.

2.2 Information Extraction

Recently, significant progress has been made in information extraction from unstructured or semi-structured text. However, most works focus on plain text documents (Peng et al., 2017; Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016). For information extraction from VRDs, (Palm et al., 2017) which uses a recurrent neural network (RNN) to extract entities of interest from VRDs (invoices) is the closest to our work, but does not take visual features into account. Besides, some of the studies (d'Andecy et al., 2018; Medvet et al., 2011; Rusinol et al., 2013) in the area of document understanding deal with a similar problem to our work, and explore using visual features to aid text extraction from VRDs; however, approaches they proposed are based on a large amount of heuristic knowledge and human-designed features, as well as limited in known templates, which are not scalable in real-world business settings. We also acknowledge a concurrent work of (Katti et al., 2018), which models 2-D document using convolution networks. However, there are several key differences. Our neural network architecture is graph-based, and our model operates on text segments instead of characters as in (Katti et al., 2018).

Besides, information extraction based on the graph structure has been developed most recently. (Peng et al., 2017; Song et al., 2018) present a graph LSTM to capture various dependencies among the input words and (Wang et al., 2018) designs a novel graph schema to extract entities and relations jointly. However, their models are not concerned with visual information directly.

3 Model Architecture

This section describes the document model and the architecture of our proposed model. Our model first encodes each text segment in the document into graph embedding, using multiple layers of graph convolution. The embedding represents the information in the text segment given its visual and textual context. By visual context, we refer to the layout of the document and relative positions of the individual segment to other segments. Textual context is the aggregate of text information in the document overall; our model learns to assign higher weights on texts from neighbor segments. Then we combine the graph embeddings with text embeddings and apply a standard BiLSTM-CRF
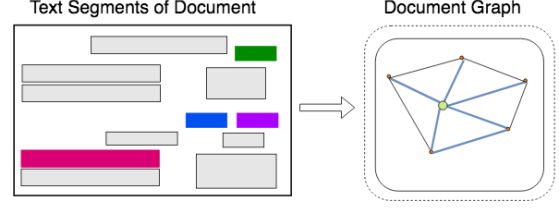

Figure 2: Document graph. Every node in the graph is fully connected to each other.

model for entity extraction.

3.1 Document Modeling

We model each document as a graph of text segments (see Figure 2), where each text segment is comprised of the position of the segment and the text within it. The graph is comprised of nodes that represent text segments, and edges that represent visual dependencies, such as relative shapes and distance, between two nodes. Text segments are generated using an in-house Optical Character Recognition (OCR) system.

Mathematically, a document $\mathcal{D}$ is a tuple (T, E), where $T = \{t_1, t_2, \cdots, t_n\}, t_i \in \mathcal{T}$ is a set of n text boxes/nodes, $R = \{r_{i1}, r_{i2}, \cdots, r_{ij}\}, r_{ij} \in \mathcal{R}$ is a set of edges, and $E = T \times R \times T$ is a set of directed edges of the form (t_i, r_{ij}, t_j) where $t_i, t_j \in T$ and $r_{ij} \in R$. In our experiments, every node is connected to each other.

3.2 Feature Extraction

For node t_i, we calculate node embedding $\mathbf{t}_i$ using a single layer Bi-LSTM (Schuster and Paliwal, 1997) to extract features from the text content in the segment.

Edge embedding between node t_i and node t_j is defined as follows,

$$\mathbf{r}_{ij} = [x_{ij}, y_{ij}, \frac{w_i}{h_i}, \frac{h_j}{h_i}, \frac{w_j}{h_i}], \tag{1}$$

where x_{ij} and y_{ij} are horizontal and vertical distance between the two text boxes respectively, and w_i and h_i are the width and height of the corresponding text box. The third, fourth and fifth value of the embedding are the aspect ratio of node t_i, relative height, and width of node t_j respectively. Empirically, a visual distance between two segments is an important feature. For example, in general, the positions of relevant information are

Figure 3: Graph convolution of document graph. Convolution is defined on node-edge-node triplets (t_i, r_{ij}, t_j). Each layer produces new embeddings for both nodes and edges.

closer in one document, such as the key and value of an entity. Moreover, the shape of the text segment plays a critical role in representing semantic meanings. For example, the length of the text segment which has address information is usually longer than that of one which has a buyer name. Therefore, we use edge embedding to encode information regarding the visual distance between two segments, the shape of the source node, and the relative size of the destination node.

To summarize, node embedding encodes textual features, while edge embedding primarily represents visual features.

3.3 Graph Convolution

Graph convolution is applied to compute visual text embeddings of text segments in the graph, as shown in Figure 3. Different from existing works, we define convolution on the node-edge-node triplets (t_i, r_{ij}, t_j) instead of on the node alone. We compare the performances of the models using nodes only and node-edge-node triplets in Section 5.3. For node t_i, we extract features $\mathbf{h}_{ij}$ for each neighbour t_j using a multi-layer perceptron (MLP) network,

$$\mathbf{h}_{ij} = g(\mathbf{t}_i, \mathbf{r}_{ij}, \mathbf{t}_j) = \text{MLP}([\mathbf{t}_i \| \mathbf{r}_{ij} \| \mathbf{t}_j]), \quad (2)$$

where $\|$ is the concatenate operation. There are several benefits of using this triplet feature set. First, it combines visual features directly into the neighbor representation. Furthermore, the information of the current node is copied across the neighbors. As a result, the neighbor features can potentially learn where to attend given the current node.

In our model, graph convolution is defined based on the self-attention mechanism. The idea

is to compute the output hidden representation of each node by attending to its neighbors. In its most general form, each node can attend to all the other nodes, assuming a fully connected graph.

Concretely the output embedding $\mathbf{t}'_i$ of the layer for node t_i is computed by,

$$\mathbf{t}'_i = \sigma\Big(\sum_{j \in \{1, \cdots, n\}} \alpha_{ij} \mathbf{h}_{ij}\Big), \quad (3)$$

where α_{ij} are the attention coefficients, and σ is an activation function. In our experiments, the attention mechanism is designed as the follows,

$$\alpha_{ij} = \frac{\exp(\text{LeakyRelu}(\mathbf{w}_a^T \mathbf{h}_{ij}))}{\sum_{j \in \{1, \cdots, n\}} \exp(\text{LeakyRelu}(\mathbf{w}_a^T \mathbf{h}_{ij}))}, \quad (4)$$

where $\mathbf{w}_a$ is a shared attention weight vector. We apply the LeakyRelu activation function to avoid the "dying Relu" problem and to increase the "contrast" of the attention coefficients potentially.

The edge embedding output of the graph convolution layer is defined as,

$$\mathbf{r}'_{ij} = \text{MLP}(\mathbf{h}_{ij}). \quad (5)$$

Outputs $\mathbf{t}'_i, \mathbf{r}'_{ij}$ are fed as inputs to the next layer of graph convolution (as computed in equation 2) or network modules for downstream tasks.

3.4 BiLSTM-CRF with Graph Embeddings

We combine graph embeddings with token embeddings and feed them into standard BiLSTM-CRF for entity extraction. As illustrated in Figure 4, visual text embedding generated from the graph convolution layers of the current text segment is
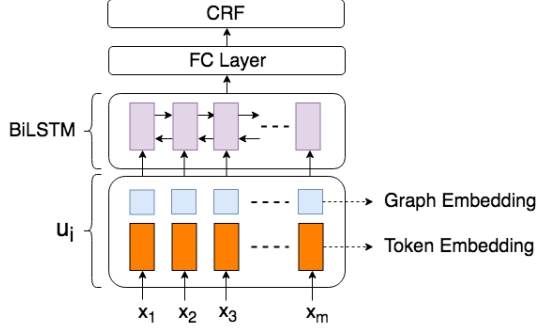
Figure 4: BiLSTM-CRF with graph embeddings.

concatenated to each token embedding of the input sequence; Intuitively, graph embedding adds contextual information to the input sequence.

Formally, assume for node t_i, the input token sequence of the text segment is $x_1, x_2, \cdots, x_m$, and the graph embedding of the node is $\mathbf{t}'_i$. The input embedding $\mathbf{u}_i$ is defined as,

$$\mathbf{u}_i = e(x_i) \| \mathbf{t}'_i \qquad (6)$$

where e is token embedding lookup function, and Word2Vec vectors are used as token embeddings in our experiments.

Then the input embeddings are fed into a BiLSTM network to be encoded, and the output is further passed to a fully connected network and then a CRF layer.

4 Model Supervision and Training

We build an annotation system to facilitate the labeling of the ground truth data. For each document, we label the values for each pre-defined entity, and their locations (bounding boxes). To generate training data, we first identify the text segment each entity belongs to, and then we label the text in the segment according to IOB tagging format (Sang and Veenstra, 1999). We assign label O to all tokens in empty text segments.

Since human annotated bounding boxes cannot match OCR detected box exactly, we apply a simple heuristic to determine which text segment an entity belongs to based on overlap area. A text segment is considered to contain an entity if $A_{overlap} / \min(A_{annotator}, A_{ocr})$ is bigger than a manually set threshold; Here $A_{annotator}, A_{ocr}, A_{overlap}$ are the area of the annotated box of the corresponding entity, the area of the ocr detected box and the area of the overlap between the two boxes respectively.

In our experiments, the graph convolution layers and BiLSTM-CRF extractors are trained jointly. Furthermore, to improve prediction accuracy, we add the segment classification task which classifies each text segment into a pre-defined tag as an auxiliary task and discuss the effect of multi-task learning in Section 5.4.3. We feed the graph embedding of each text segment into a sigmoid classifier to predict the tag. Since the parameters of the graph convolution layers are shared across the extraction task and segment classification task, we employ a multi-task learning approach for model training. In multi-task training, the goal is to optimize the weighted sum of the two losses. In our experiments, the weight is determined using a principled approach as described in (Kendall et al., 2017). The idea is to adjust each task's relative weight in the loss function by considering task-dependant uncertainties.

5 Experiments

We apply our model for information extraction from two real-world datasets. They are Value-Added Tax Invoices (VATI) and International Purchase Receipts (IPR).

5.1 Datasets Description

VATI consists of 3000 user-uploaded pictures and has 16 entities to exact. Example entities are the names of buyer/seller, date and tax amount. The invoices are in Chinese, and it has a fixed template since it is national standard invoice. However, there are many noises in the documents which include distracting objects in the image and skewed document orientation to name a few. IPR is a data set of 1500 scanned receipt documents in English which has 4 entities to exact (Invoice Number, Vendor Name, Payer Name and Total Amount). There exist 146 templates for the receipts. Variable templates introduce additional difficulties to the IPR dataset. For both datasets, we assign 70% of each dataset for training, 15% for validation and 15% for the test. The number of text segments varies per document from 100 to 300.

5.2 Baselines

We compare the performance of our system with two BiLSTM-CRF baselines. Baseline I applies BiLSTM-CRF to each text segment, where each text segment is an individual sentence. Baseline II applies the tagging model to the concatenated

Model	VATI	IPR
Baseline I	0.745	0.747
Baseline II	0.854	0.820
BiLSTM-CRF + GCN	0.873	0.836

Table 1: F_1 score. Performance comparisons.

Entities	Baseline I	Baseline II	Our model
Invoice #	0.952	0.961	0.975
Date	0.962	0.963	0.963
Price	0.527	0.910	0.943
Tax	0.584	0.902	0.924
Buyer	0.402	0.797	0.833
Seller	0.681	0.731	0.782

Table 2: F_1 score. Performance comparisons for individual entities from VATI dataset.

document. Text segments in a document are concatenated from left to right and from top to bottom according to (Palm et al., 2017). Baseline II incorporates a one-dimensional textual context to the model.

5.3 Results

We use the F_1 score to evaluate the performances of our model in all experiments. The main results are shown in Table 1. As we can see, our proposed model outperforms both baselines by significant margins. Capturing patterns from VRDs with one-dimensional text sequence is difficult. More specifically, we present performance comparisons of six entities from VATI dataset in Table 2. It can be seen that compared with two baselines, our model performs almost identical on "simple" entities which can be distinguished by the text segment's text feature alone (*i.e.*, Invoice Number and Date) where visual features and context information are not necessary. However, our proposed model clearly outperforms baselines on entities which can not be represented by text alone, such as Price, Tax, Buyer, and Seller.

To further examine the contributions made by each sub-component of the graph convolution network, we perform the following ablation studies. In each study, we exclude visual features (edge embeddings), textual features (node embeddings) and the use of attention mechanism respectively, to see their impacts on F_1 scores on both two datasets. As presented in Table 3, it can be seen that visual features play a critical role in the performance of our model; they lead to more than

Configurations/Datasets	VATI	IPR
Full model	0.873	0.836
w/o vis. features	0.808	0.775
w/o text features	0.871	0.817
w/o attention	0.872	0.821

Table 3: F_1 score. Ablation studies of individual component of graph convolution.

5% performance drop in both datasets. Intuitively, visual features provide more information about contexts of the text segments, so it improves the performance by discriminating between text segments with similar semantic meanings. Moreover, textual features make similar contributions. Furthermore, the attention mechanism shows more effectiveness on variable template datasets, which results in 1.5% performance gains. However, it makes no contribution to fixed layout datasets. We make further discussions on attention in the next section.

5.4 Discussions

5.4.1 Attention Analysis

To better understand how attention works in our model, we study the attention weights (on all other text segments) of each text segment in the document graph. Interestingly, for the VATI dataset, we find that attention weights are usually concentrated on a fixed text segment with strong textual features (the segment contains address information) regardless of the text segment studied. The reason behind that may be attention mechanism tries to find an anchor point of the document, and one anchor is enough as VATI documents all share the same template. Strong textual features help locate the anchor more accurately.

For variable layout documents, more attention is paid to nearby text segments, which reflects the local structure of the document. Specifically, attention weights of the left and upper segments are even higher. Furthermore, segments that contain slot keywords such as "address", "amount" and "name" receive higher attention weights.

5.4.2 The Number of Graph Convolution Layers

Here we evaluate the impact of different numbers of graph convolution layers. In theory, a higher number of layers encodes more information and therefore can model more complex relationships. We perform the experiments on selected entity

Entities	1 layer	2 layer	3 layer
Invoice #	0.959	0.975	0.964
Date	0.960	0.963	0.960
Price	0.931	0.943	0.931
Tax	0.915	0.924	0.917
Buyer	0.829	0.833	0.827
Seller	0.772	0.782	0.775

Table 4: F_1 score. Performance comparisons of different graph convolution layers for individual entities from VATI dataset.

Model	VATI	IPR
BiLSTM-CRF + GCN	0.873	0.836
+ Multi-task	0.881	0.849

Table 5: F_1 score. Effectiveness of multi-task learning approach.

types of the VATI dataset, and the results are presented in Table 4. As we can see, additional layers in the network do not help simple tasks. By simple task, we mean that task achieves high accuracy with single layer graph convolution. However, more layers indeed improve the performances of more difficult tasks. As shown in the table, the optimal number of layers is two for our task, as three layers overfit the model. Ideally, the number of graph convolution layers used should be adaptive to the specific task, of which we leave the study to future works.

5.4.3 Multi-Task Learning

As shown in Table 5, our task benefits from the segment classification task and multi-task learning method in both datasets. The two tasks in our experiments are complementary, and compared with the single task model, the multi-task learning model may have better generalization performance by adopting more information. Furthermore, we find that the multi-task learning helps the training converge much faster.

6 Conclusions and Future Works

This paper studies the problem of entity extraction from VRDs. A graph convolution architecture is proposed to encode text embeddings given visually rich context. BiLSTM-CRF is applied to extract the final results. We manually annotated two real-world datasets of VRDs, and perform comprehensive experiments and analysis. Our system outperforms BiLSTM baselines and presents a

novel method for IE from VRDs. Furthermore, we plan to extend the graph convolution framework to other tasks in VRDs, such as document classification.

Acknowledgments

References

Laura Chiticariu, Yunyao Li, and Frederick R Reiss. 2013. 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*, pages 827–832.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Vincent Poulain d'Andecy, Emmanuel Hartmann, and Marçal Rusiñol. 2018. Field extraction by hybrid incremental and a-priori structural templates. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 251–256. IEEE.

Andreas R Dengel and Bertin Klein. 2002. smartfix: A requirements-driven system for document analysis and understanding. In *International Workshop on Document Analysis Systems*, pages 433–444. Springer.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. 2015. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image generation from scene graphs. *arXiv preprint*.

Anoop Raveendra Katti, Christian Reisswig, Cordula Guder, Sebastian Brarda, Steffen Bickel, Johannes Höhne, and Jean Baptiste Faddoul. 2018. Chargrid: Towards understanding 2d documents. *arXiv preprint arXiv:1809.08799*.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2017. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *arXiv preprint arXiv:1705.07115*, 3.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Eric Medvet, Alberto Bartoli, and Giorgio Davanzo. 2011. A probabilistic approach to printed document understanding. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(4):335–347.

Rasmus Berg Palm, Ole Winther, and Florian Laws. 2017. Cloudscan-a configuration-free invoice analysis system using recurrent neural networks. *arXiv preprint arXiv:1708.07403*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *arXiv preprint arXiv:1708.03743*.

Marçal Rusinol, Tayeb Benkhelfallah, and Vincent Poulain dAndecy. 2013. Field extraction from administrative documents by incremental structural templates. In *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*, pages 1100–1104. IEEE.

Erik F Sang and Jorn Veenstra. 1999. Representing text chunks. In *Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics*, pages 173–179. Association for Computational Linguistics.

Daniel Schuster, Klemens Muthmann, Daniel Esser, Alexander Schill, Michael Berger, Christoph Weidling, Kamil Aliyev, and Andreas Hofmeier. 2013. Intellix–end-user trained information extraction for document archiving. In *2013 12th International Conference on Document Analysis and Recognition*, pages 101–105. IEEE.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, 45(11):2673.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph state lstm. *arXiv preprint arXiv:1808.09101*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Shaolei Wang, Yue Zhang, Wanxiang Che, and Ting Liu. 2018. Joint extraction of entities and relations based on a novel graph scheme. In *IJCAI*, pages 4461–4467.

Kun Xu, Lingfei Wu, Zhiguo Wang, and Vadim Sheinin. 2018. Graph2seq: Graph to sequence learning with attention-based neural networks. *arXiv preprint arXiv:1804.00823*.