Integrating Coordinates with Context for Information Extraction in Document Images

Zhaohui Jiang†, Zheng Huang*†, Yunrui Lian‡, Jie Guo†, Weidong Qiu†
†*School of Electronic Information and Electrical*
Shanghai Jiao Tong University, Shanghai, China
‡*Xiamen No.1 High School, Fujian, China*
Email: {sophie_jiang, huang-zheng}@sjtu.edu.cn

***Abstract*—Information extraction from document collections is a fundamental and important step to understand, structure and analyze data. Many approaches with rules and deep learning based techniques have been applied on plain text, however, when it comes to document images, such demand still exists but becomes quite challenging without linguistic knowledge. In this paper, we propose an approach to extract required named entities (NEs) from document images by integrating the coordinate information from the detection and recognition stage into the contextual information of the BiLSTM-CRF model with an attention mechanism. We test this method on two real-world datasets. One is a Contract Dataset of Listed Companies, and the other is an Insurance Policy Dataset of our own. The result shows a combination of coordinates and context with attention leverages extraction in document images, opening up potential applications on such tasks.**

***Keywords*-BiLSTM-CRF, Coordinates and Context, Information Extraction, Named Entities, Scene Text Detection and Recognition**

I. Introduction

Information extraction has been studied for a long time as the fundamental step towards many other tasks [1] [2]. A lot of useful methods have been proposed, from traditional machine learning models [3] to neural networks like CNN and BiLSTM structure [4]. However, these studies are generally based on plain text and cannot be applied to document images directly.

Some approaches use optical character recognition (OCR) engines to recognize the characters or words in images, and then the conventional methods are used to process text results [5]. However, the features of the target information are often manually formulated, and the bigger problem is if just the text results are used, the layout and position information of the image will be lost, which is important for such tasks. There are also some structure-based context-aware methods to extract NEs [6] [7], especially from historical handwritten document images [8] [9]. These methods often use image segmentation technique and neural networks, but the way how to combine the information is not adequately considered.

*The corresponding author.

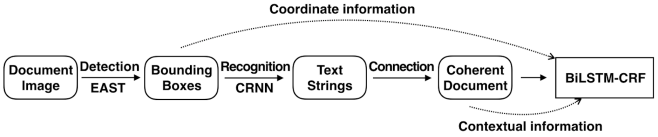

Figure 1. The pipeline of our method.

In addition to the OCR-based methods, some approaches directly do extraction without the recognition stage. As in [10], structural and positional features of the characters in the image are summarized and fed to a BiLSTM neural network. But the features are only suitable for a specific task. Moreover, the contextual information will be forgotten in such text-excluded way.

Based on the above considerations, in this work, we integrate the coordinate information with the contextual information to extract named entities from document images which are of some specific patterns, but not unified. Fig.1 shows the pipeline of our method. First, the scene text detection model EAST [11] is applied to the images, and according to the output bounding boxes of the text strings, we can easily obtain the coordinates (in pixels) of all the texts. We choose EAST because it is more general and efficient for various document images, including scanned ones and that of natural scenes. Then we use the text recognition model CRNN [12] to recognize the texts in the positioning areas and sort them into a coherent document, tagging with the BIOES labels [13] for named entities according to the annotation files. Finally, the combination of the coordinate and contextual information is input into one BiLSTM-CRF layer to extract the NEs. In this process, we have tried various combination methods and propose a new structure to separately encoding the two information and then combining them with modulated attention. The result shows our approach achieves good performance on two new datasets respectively about contracts and insurance policy in Chinese of this kind. The rest of this paper is organized as follows. First, we will introduce the datasets in Section II. Then, Section III describes the components and structures

Table I
DETAILED STATISTICS OF THE CONTRACT DATASET.

	Train	Test
File Number	893	223
Avg Character Length	1245	1283
Party A	893	223
Party B	893	223
Project Name	636	159
Contract Name	313	78
Contract Amount	893	223
Consortium Members	102	28
Total Entity	3730	934
Character Vocabulary	2583	

of our methods in detail. Section IV describes the training details, results, and comparison of our experiments. In the last section, Section V, we draw conclusions and discuss some future works.

II. DATASETS

A. Contract Dataset

This dataset is from Alibaba Tianchi Competition[1], whose task is information extraction from listed company announcements. Concretely, the aim is to detect the required named entities for further analyzing the market trend. There are three sub-datasets and we use the contract one since the length of the other two is up to tens of pages, which is not suitable for our task. This dataset has both the original PDF format documents and annotation files of target NEs. Considering that this is an industrial dataset and there will be some annotation problems, we first do some data cleaning and processing. Table I summarizes the data statistics after processing. The first two lines are the file number of training (test) set and the average character length of each set. The middle part shows all target entities and their numbers of occurrences in the dataset. The second last row is the total number of all the above entities and the last row shows the number of character vocabulary. Fig.2 is an example with the output bounding boxes and target NE labels drawn on it.

B. Insurance Policy Dataset

This dataset is collected by ourselves, which consists of camera-captured insurance policy images of four companies. All the annotation files are labeled by expert human annotators, containing the target NEs and their coordinates in the images. The images of every company are of a specific pattern. Although we can manually make some rules to extract the named entities, it is still a troublesome engineering work to make rules for every pattern of the images. Table II shows the data statistics, and the structure is the same as Table I.

[1]https://tianchi.aliyun.com/competition/information.htm?spm=5176.100067.5678.2.456c15e0OxKQSO&raceId=231659

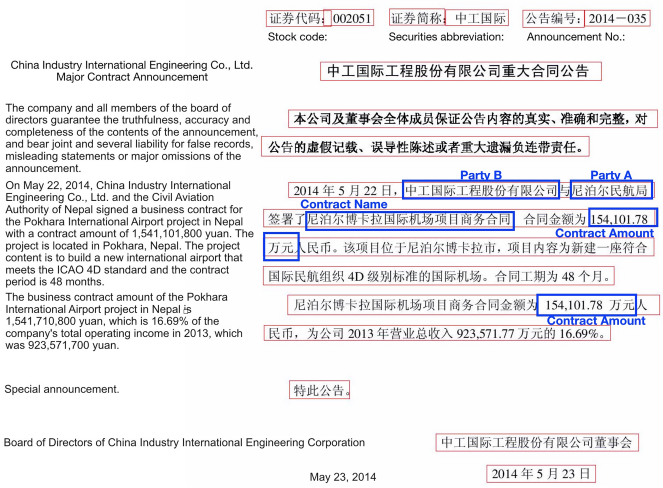

Figure 2. An example of the Contract Dataset.

Table II
DETAILED STATISTICS OF THE INSURANCE POLICY DATASET.

	Train	Test
File Number	493	132
Avg Character Length	1392	1398
Policyholder	466	114
Insured Name	491	121
Insurance Company	475	115
Policy Number	493	122
Premium	498	122
Total Entity	2423	594
Character Vocabulary	1712	

III. PROPOSED METHODS

In this section, we first introduce the components of our methods and then describe the way we represent the coordinate and the contextual information from previous components and the combination methods.

A. Components

EAST To obtain coordinate information, we use a scene text detection model EAST, which precisely means an efficient and accurate scene text detector. Unlike other models having multiple stages and components, EAST only consists of two stages: a Fully Convolutional Network (FCN) [14] and a Non-Maximum Suppression (NMS) [15] merging stage. The former FCN consists of a feature extractor stem and a feature-merging branch to directly produce text regions, which can be either rotated rectangles or quadrangles. Then these regions are sent to NMS to yield final bounding boxes. The results on several datasets show that EAST is not only much faster but also excellent in detection accuracy [11]. In this research, the model is trained on 2245 training images consisting of contract documents and bank notes collected by ourselves and the output is a rectangle region defined by the coordinates (in pixels) of four corners of every string.

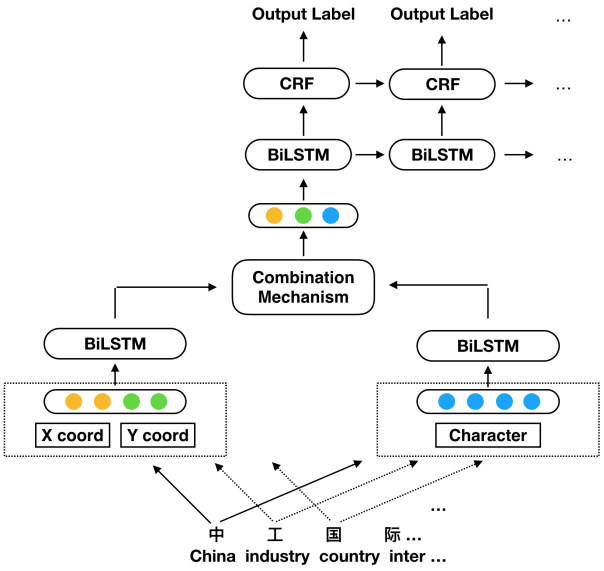

Figure 3. Main structure of combining information.

CRNN To obtain textual information, we first use a popular recognition model Convolutional Recurrent Neural Network (CRNN), which can recognize long text sequences. It includes a CNN feature extraction layer and a BiLSTM sequence feature extraction layer to perform end-to-end joint training. Then the final features go through a transcription layer to get the text sequence. The training set is the same one of the detection model.

BiLSTM-CRF BiLSTM-CRF is a widely adopted neural architecture for sequence labeling problems, including named entity recognition. It is a hierarchical model mainly consisting of one BiLSTM layer and one CRF layer. The BiLSTM layer can make effective use of contextual information for its time structure while the CRF layer can obtain a globally optimal tag sequence by considering the relationship between adjacent tags. In our method, the input of the last BiLSTM-CRF layer is a sequence of the combination results.

B. Main Structure

The main structure of our method is illustrated in Fig.3.

Character Features Since the datasets we use are composed of Chinese characters and the text results are from the recognition model, there are some problems of entity boundary and word segmentation. Recent research [16] [17] shows that character-based methods often outperform word-based methods for Chinese named entities, so we choose to use the character embeddings. For every character of a sequence with length T, $\mathbf{x}_t^{(char)}$ represents a 100-dimensional vector at the time t. Both the random embeddings and the pre-trained embeddings [18] on Chinese Wikipedia are tested, and the effects are almost the same in the tasks. In the next few sections, we just list the results of random character embeddings, whose values are initialized from the uniform distribution.

Coordinate Features The outputs of EAST are the x and y coordinates (in pixels) of four corners of the bounding boxes. In order to get the coordinate information of every character, for every box, we first calculate the average width of characters in it, then the relatively accurate x coordinate and y coordinate of every character can be obtained. For both the x and y coordinate feature, we use two methods to represent them. One is a random 50-dimensional vector, and the other is a vector calculated from the equations in [19]. That is, each dimension of the coordinate encoding corresponds to a sinusoid:

$$PE_{(pos,2i)} = \sin\left(\text{pos}/10000^{2i/d_{\text{coord}}}\right) \tag{1}$$

$$PE_{(pos,2i+1)} = \cos\left(\text{pos}/10000^{2i/d_{\text{coord}}}\right) \tag{2}$$

where pos is the coordinates in our research, i is the dimension and d_{coord} is the whole dimension of coordinate vectors. The final coordinate information of a character is a 100-dimensional vector represented as below:

$$\mathbf{x}_t^{(coord)} = [\mathbf{x}_t^{(xcoord)}; \mathbf{x}_t^{(ycoord)}] \tag{3}$$

During the training of the model, both the coordinate embeddings are continuously updated. The comparison will be shown in Table V.

Since the coordinates are all discrete numbers in pixels and there must be some deviations during the bounding box detection stage, if we use the accurate coordinates of every character, there will be too many coordinate features, bringing some negative effects. Based on this consideration, we try to create a list of buckets at a certain distance. If the coordinate falls into one bucket, then it is rounding to one of the boundary numbers of this bucket. That means the character at a similar position in the document image will have the same x and y coordinate embeddings. For example, if the x coordinate of one character is 129 (in pixels) and the buckets are from 0 to 5000 at the interval of 100 in pixels, then the x coordinate will fall into the [100,200] pixel bucket and be regulated to 100. We try different intervals from 50 to 300 at the step of 50 and find 200 is the best in our tasks (where the resolution is 300 PPI). All the representation methods are the same as described above.

Combination After obtaining the embeddings of characters and coordinates, we try various methods to combine them. Although every combination method can bring improvements to the result of the only text-based way, the way encoding the character and coordinate information separately with modulated attention combination is the best. Here we use $\mathbf{x}_t^{(comb)}$ to represent the input sequence of the last BiLSTM-CRF layer and the combination methods are listed as followed:

- **(baseline)** $\mathbf{x}_t^{(char)}$ **only**: only takes the character embedding as the input without any coordinate information. That is:

$$\mathbf{x}_t^{(comb)} = \mathbf{x}_t^{(char)} \tag{4}$$

- **(sum/cat)** $\mathbf{x}_t^{(char)} + \mathbf{x}_t^{(coord)}$: since the embedding of every character and its coordinate information are of the same dimension, we can directly sum them together as a whole embedding. That is:

$$\mathbf{x}_t^{(comb)} = \mathbf{x}_t^{(char)} + \mathbf{x}_t^{(coord)} \tag{5}$$

or we can concatenate them:

$$\mathbf{x}_t^{(comb)} = [\mathbf{x}_t^{(char)}; \mathbf{x}_t^{(coord)}] \tag{6}$$

- **(sum/cat)** $\mathbf{h}_t^{(char)} + \mathbf{h}_t^{(coord)}$: another BiLSTM layer is used to encode the embeddings. In recent work by Kitaev and Klein [20], instead of combining content and position information at first as usual, they choose to explicitly separate the two information with split self-attention model in the parsing task and the results increase. Inspired by this idea, after obtaining the character and coordinate embeddings, we use two separate BiLSTM encoders to get the corresponding output representations $\mathbf{h}_t^{(char)}$ and $\mathbf{h}_t^{(coord)}$, which respectively contains the contextual information and the global coordinate information of every character. Similarly, we can simply sum them up or concatenate them:

$$\mathbf{x}_t^{(comb)} = \mathbf{h}_t^{(char)} + \mathbf{h}_t^{(coord)} \tag{7}$$

$$\mathbf{x}_t^{(comb)} = [\mathbf{h}_t^{(char)}; \mathbf{h}_t^{(coord)}] \tag{8}$$

- **(proposed)** $\mathbf{h}_t^{(char)}$ **+** $\mathbf{h}_t^{(coord)}$ **(with attention)**: inspired by the attention mechanism in translation [21] and the modality attention combining text features and visual features of social media NER task [22] [23], instead of directly concatenating or adding the two information, we use a modulated attention module, which adaptively emphasizes or attenuates different information as a whole at each step t, thus a better combination of the contextual and coordinate information can be obtained and help for the last BiLSTM-CRF layer to extract the named entities. The processing formulas are as followed:

$$\mathbf{a}_t^{(char)} = \sigma\left(\mathbf{W}_h^{(char)} \cdot \mathbf{h}_t^{(char)} + \mathbf{b}_h^{(char)}\right) \tag{9}$$

$$\mathbf{a_t}^{(coord)} = \sigma\left(\mathbf{W}_h^{(coord)} \cdot \mathbf{h}_t^{(coord)} + \mathbf{b}_h^{(coord)}\right) \tag{10}$$

$$\alpha_t^{(k)} = \mathrm{softmax}\left(\mathbf{a}_t^{(char)}, \mathbf{a}_t^{(coord)}\right) \tag{11}$$

$$k \in \{char, coord\} \tag{12}$$

$$\mathbf{x}_t^{(comb)} = \sum_{k \in \{char, coord\}} \alpha_t^{(k)} \mathbf{h}_t^{(k)} \tag{13}$$

here the $\mathbf{h}_t^{(char)}$ and $\mathbf{h}_t^{(coord)}$ are outputs of two separate BiLSTM layers and of the same dimension, if not, a projection layer can be added. The projections are parameter matrices $\mathbf{W}_h \in \mathbb{R}^{d_\mathrm{h} \times d_\mathrm{h}}$ and $\mathbf{b}_h \in \mathbb{R}^{d_\mathrm{h}}$. σ is the sigmoid function.

Table III
HYPER-PARAMETER SETTINGS.

Parameters	**Value**
Char embed size	100
Xcoord embed size	50
Ycoord embed size	50
LSTM hidden	100
LSTM layer	1
Coord buckets interval	200 pix
Parameters	**Value**
Char dropout	0.5
Coord dropout	0.1
Batch size	10
Optimizer	Adam
Initial learning rate	0.015
Decay rate	0.05

Table IV
EVALUATION OF DIERENT COMPONENTS ON INSURANCE POLICY DATASET. VALUES BETWEEN 0-100%.

Element		char baseline $\mathbf{x}^{(char)}$	rand coord (sum) $\mathbf{x}^{(char)}$, $\mathbf{x}^{(coord)}$	rand coord (cat) $\mathbf{x}^{(char)}$, $\mathbf{x}^{(coord)}$
Policyholder	P	**100.00**	**100.00**	**100.00**
	R	96.49	**98.25**	96.49
	F_1	98.21	**99.12**	98.21
Insured Name	P	99.20	**99.21**	98.41
	R	96.88	**98.44**	96.88
	F_1	98.02	**98.82**	97.64
InsurCompany	P	**100.00**	**100.00**	**100.00**
	R	**100.00**	**100.00**	**100.00**
	F_1	**100.00**	**100.00**	**100.00**
Policy Number	P	98.36	**99.17**	**99.17**
	R	**99.17**	**99.17**	**99.17**
	F_1	98.77	**99.17**	**99.17**
Premium	P	98.39	**100.00**	**100.00**
	R	**100.00**	**100.00**	**100.00**
	F_1	99.19	**100.00**	**100.00**
Macro-average	P	99.16	**99.66**	99.49
	R	98.50	**99.17**	98.50
	F_1	98.83	**99.42**	98.99

IV. EXPERIMENTS

A. Training Details

We use Pytorch framework to implement our experiments on a GTX 1080Ti GPU and Table III illustrates the training hyper-parameters of all the experiments on different datasets, where the learning rate is calculated by: lr = initial lr / (1 + step * decay rate).

B. Evaluation

We measure the performance of each method in terms of $precision(P)$, $recall(R)$, and F_1score as the NER task. Since the focus of our method is on information extraction from document images, the correctness and completeness of a whole target named entity are more important than one token of it. Based on this consideration, the methods

Table V
EVALUATION OF DIERENT COMPONENTS ON CONTRACT DATASET. VALUES BETWEEN 0-100%.

Element		char baseline $\mathbf{x}^{(char)}$	rand coord (sum) $\mathbf{x}^{(char)}$, $\mathbf{x}^{(coord)}$	rand coord (cat) $\mathbf{x}^{(char)}$, $\mathbf{x}^{(coord)}$	separate lstm (sum) $\mathbf{h}^{(char)}$, $\mathbf{h}^{(coord)}$	separate lstm (cat) $\mathbf{h}^{(char)}$, $\mathbf{h}^{(coord)}$	separate lstm (attention) $\mathbf{h}^{(char)}$, $\mathbf{h}^{(randcoord)}$	separate lstm (attention) $\mathbf{h}^{(char)}$, $\mathbf{h}^{(sinecoord)}$
Party A	P	75.00	70.64	73.10	78.14	69.04	**80.10**	77.25
	R	58.51	68.88	59.75	68.46	68.46	**69.71**	67.63
	F_1	65.73	69.75	65.75	72.98	68.75	**76.18**	72.12
Party B	P	82.73	79.60	78.09	80.83	79.51	85.91	**88.32**
	R	79.48	82.53	85.59	84.72	84.72	**86.90**	82.53
	F_1	81.07	81.04	81.67	82.73	82.03	**86.40**	85.33
Project Name	P	63.10	65.97	69.84	67.07	69.86	72.19	**72.30**
	R	30.64	54.91	50.87	**64.74**	58.96	63.01	61.85
	F_1	41.25	59.94	58.86	65.88	63.95	**67.28**	66.67
Contract Name	P	65.79	58.75	66.67	66.30	69.23	**75.68**	65.22
	R	61.73	58.02	69.14	75.31	**77.78**	69.14	74.07
	F_1	63.69	58.39	67.88	70.52	**73.26**	72.26	69.36
Contract Amount	P	88.62	83.96	76.09	86.17	88.24	**88.35**	83.82
	R	83.99	87.25	88.24	87.58	89.22	91.50	**93.14**
	F_1	86.24	85.58	81.72	86.87	88.78	**89.89**	88.24
Consortium Members	P	26.32	29.41	41.67	40.00	33.33	46.15	**47.06**
	R	29.41	29.41	29.41	35.29	41.18	35.29	**47.06**
	F_1	27.78	29.41	34.48	37.50	36.84	40.00	**47.06**
Macro-average	P	66.93	64.72	67.57	69.75	68.20	**74.73**	72.33
	R	57.29	63.50	63.83	69.35	70.05	69.26	**71.05**
	F_1	61.73	64.10	65.65	69.55	69.11	**71.89**	71.68

are evaluated for their ability to identify entire contract or insurance policy elements instead of per token. That means for each element type such as contracting parties, the strictest evaluation will count as true positives only the predicted elements that match exactly every character of gold ones, and similarly for false positives and false negatives.

C. Results

According to the different combination mechanisms mentioned in section III-B, the results of the two datasets are shown in Table IV and Table V.

Insurance Policy Dataset: Table IV lists the results of every element and the best result per element type is shown in bold font. The last line of macro-averages are the averages of the corresponding columns, indicating the overall performance of each method on all element types. Although the char-based baseline gives a relatively good result of 98.83% macro-averaged F_1, both the simple concatenating and adding way lead to improvement, especially to the $recall(R)$ rates, which shows the coordinate (position) information is helpful to such task of extracting named entities from document images. Through the comparison of the third column and the fourth column, it seems that the summing-based way performs better than concatenating-based way for combining the coordinate and contextual information in this dataset. Since the results are good enough, we do not try more complex methods on this dataset.

Contract Dataset: Table V lists the results and macro-averages of the Contract Dataset and the best results are also shown in bold font. As shown in the last line of the table, when coordinate information is available, the model performance greatly improves over the char baseline, showing the combination of coordinate and contextual information is helpful for such extraction task. By comparing the second column with the third and fourth columns, we can see the macro-averaged F_1 increases by 3 to 4 percent with the coordinate information of the simplest combining method. As shown in the fifth and sixth columns, with the structure of two separate BiLSTM layers encoding the different information, the macro-averaged F_1 is up from 61.73% to 69.55% and the results of this structure are much better than the simplest combining way. There seems little difference between the summing-based way and the concatenating-based way. Moreover, it can be seen from the last two columns that the modulated attention between the hidden representations further improves the extraction performance to 71.89%, which indicates this mechanism can maximize the information gain, and a proper combination between the coordinate and contextual information can receive better performance. The random coordinate embedding and the sinusoidal coordinate encoding produce nearly identical results.

V. CONCLUSIONS AND FUTURE WORK

We propose a method combining the coordinate and contextual information for named entity extraction from document images which may have specific patterns, but not unified. We demonstrate that the coordinate information can be quite helpful in such tasks compared to the text-only baseline, without any complicated manual rules. In addition, separate encoding layers and a modulated attention mechanism are proposed to combine different information better, which achieves the best performance on a Contract Dataset. Although we only test on the Chinese datasets now, this method is also suitable for other languages. For future work, we plan to find more effective coordinate expressions of the target information like a relative positional encoding and test on more different kinds of datasets, extending this

method to other information extraction tasks like relation extraction.

ACKNOWLEDGMENT

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

REFERENCES

[1] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. Strassel, and R. M. Weischedel, "The automatic content extraction (ace) program-tasks, data, and evaluation." in *LREC*, vol. 2, 2004, p. 1.

[2] K. Jung, K. I. Kim, and A. K. Jain, "Text information extraction in images and video: a survey," *Pattern Recognition*, vol. 37, no. 5, pp. 977–997, 2004.

[3] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *Advances in neural information processing systems*, 2005, pp. 1185–1192.

[4] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," 2016.

[5] G. Zhu, T. J. Bethea, and V. Krishna, "Extracting relevant named entities for automated expense reimbursement," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 1004–1012.

[6] C. Pitou12 and J. Diatta, "Textual information extraction in document images guided by a concept lattice," 2016.

[7] H. Bouressace and J. Csirik, "Recognition of the logical structure of arabic newspaper pages," in *International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining*. Springer, 2018, pp. 251–258.

[8] A. Fornés, V. Romero, A. Baró, J. I. Toledo, J. A. Sánchez, E. Vidal, and J. Lladós, "Icdar2017 competition on information extraction in historical handwritten records," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1. IEEE, 2017, pp. 1389–1394.

[9] J. I. Toledo, M. Carbonell, A. Fornés, and J. Lladós, "Information extraction from historical handwritten document images with a context-aware neural model," *Pattern Recognition*, vol. 86, pp. 27–36, 2019.

[10] C. Adak, B. B. Chaudhuri, and M. Blumenstein, "Named entity recognition from unstructured handwritten document images," in *Document Analysis Systems (DAS), 2016 12th IAPR Workshop on*. IEEE, 2016, pp. 375–380.

[11] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proc. CVPR*, 2017, pp. 2642–2651.

[12] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[13] L. Ratinov and D. Roth, "Design challenges and misconceptions in named entity recognition," in *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, 2009, pp. 147–155.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[15] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3. IEEE, 2006, pp. 850–855.

[16] H. Li, M. Hagiwara, Q. Li, and H. Ji, "Comparison of the impact of word segmentation on name tagging for chinese and japanese." in *LREC*, 2014, pp. 2532–2536.

[17] J. He and H. Wang, "Chinese named entity recognition and word segmentation based on character," in *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*, 2008.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *International Conference on Neural Information Processing Systems*, 2013, pp. 3111–3119.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[20] N. Kitaev and D. Klein, "Constituency parsing with a self-attentive encoder," *arXiv preprint arXiv:1805.01052*, 2018.

[21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[22] D. Lu, L. Neves, V. Carvalho, N. Zhang, and H. Ji, "Visual attention model for name tagging in multimodal social media," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2018, pp. 1990–1999.

[23] S. Moon, L. Neves, and V. Carvalho, "Multimodal named entity recognition for short social media posts," *arXiv preprint arXiv:1802.07862*, 2018.