Information Extraction System for Invoices and Receipts

Chapter *in* Lecture Notes in Computer Science · July 2023

DOI: 10.1007/978-981-99-4752-2_7

4 authors, including:

Qi Cao
University of Glasgow
97 PUBLICATIONS **510** CITATIONS
SEE PROFILE

Chee Kiat Seow
University of Glasgow
64 PUBLICATIONS **675** CITATIONS
SEE PROFILE

Peter Chunyu Yau
University of Glasgow
44 PUBLICATIONS **48** CITATIONS
SEE PROFILE

Information Extraction System for Invoices and Receipts

QiuXing Michelle Tan[1], Qi Cao[2(✉)], Chee Kiat Seow[2], and Peter Chunyu Yau[2]

[1] Computing Science, Singapore Institute of Technology - University of Glasgow, Singapore, Singapore
[2] School of Computing Science, University of Glasgow, Glasgow, Scotland, UK
qi.cao@glasgow.ac.uk

Abstract. Rapid growth in the digitization of documents, such as paper-based invoices or receipts, has alleviated the demand for methods to process information accurately and efficiently. However, it has become impractical for humans to extract the data manually, as it is labor-intensive and time-consuming. Digital documents contain various components such as tables, key-value pairs and figures. Existing optical character recognition (OCR) methods can recognize texts, but it is challenging to extract the key-value pairs in unformatted digital invoices or receipts. Hence, developing an information extraction system with intelligent algorithms would be beneficial, as it can increase the workflow efficiency for knowledge discovery and data recognition. In this paper, a pipeline of the information extraction system is proposed with intelligent computing and deep learning approaches for classifying key-value pairs first, followed by linking the key-value pairs. Two key-value pairing rules are developed in the proposed pipeline. Various experiments with intelligent algorithms are conducted to evaluate the performance of the pipeline of information extraction system.

Keywords: Information Extraction, Key-Value Pairs, Knowledge Discovery, Documents Digitization, Intelligent Algorithms

1 Introduction

On a daily basis, many organizations deal with many paper documents such as receipts or invoices [1]. Currently, most documents must be processed manually, which is time-consuming and expensive [2]. One of the most difficult aspects of invoice processing for logistics organizations is the time-consuming, intensive in-house procedure, requiring numerous manual works when extracting and keying data into various internal software systems. In addition, data captured from logistics invoices poses a particular difficulty because they are received in multiple formats. This is a significant issue, due to the intricacy of the invoices. Manually inspecting and correcting scanning errors greatly lengthens the correction and processing time [3].

Digital invoices contain various components such as tables, key-value pairs and figures. A key-value pair is made up of two connected data elements: a key which is a constant defining the data set; and a value which is a variable that is part of the set. A

fully formed key-value pair will be like Gross Weight: 123 kg, where “Gross Weight” is the key and “123 kg” is the value.

A corporation can benefit from digitising and extracting key information on a number of levels. Business owners can better track their processes, provide better customer services, increase employee productivity, and cut costs. Optical character recognition (OCR) also known as text recognition, is the process of extracting text information from scanned documents and images. Current OCR systems, such as Tesseract [4] and EasyOCR can recognize raw text in unformatted digital documents or images. But they are not capable of extracting information like key-value pairs from unformatted data. Key-value pairs are the most significant components in digital cargo invoices. Key-value pairs make raw texts more understandable.

The existing solutions, such as Amazon Web Services (AWS) Textract, provide such a service [5]. But the off-the-shelf solutions may not fit the specific domains, and lack flexibility to be customized or tailored for different requirements. Moreover, AWS Textract is a service hosting on the cloud platform. Users need to upload their digital invoices and receipts to the cloud, in order to get them processed. It raises possible privacy concerns as some companies might treat the data as private and business sensitive.

Hence, automation of extracting key-value pairs within unformatted digital invoices is proposed as it can significantly reduce manpower and costs while simultaneously ensuring the reliability of the data retrieved. Deep learning algorithms have been implemented in various applications with great success [6][7]. In this paper, an information extraction system with deep learning approaches that is capable of extracting key-value pairs will be presented to improve the overall performance. The proposed information extraction system would be beneficial as it would help companies achieve workflow efficiency, resource utilization and eliminate costly errors.

The remaining parts of this paper are organized next. Section 2 presents the prior works in literature. Section 3 explains the pipeline of the proposed information extraction system. Section 4 analyses and discusses experiment results. Lastly, Section 5 concludes this paper.

2 Related Work

There are three different techniques of key-value pair extraction: regular expression, natural language processing (NLP) and layout detection. OCR affects the results when performing key-value pair extraction, as mistaken words might be wrongly classified. Hence, OCR plays a crucial role for key-value pair extraction. Vedant Kumar *et al*. [8] use Tesseract OCR on bill receipts images taken from a mobile phone to extract out the text information, with some image pre-processing such as binarization and removing of shadows, etc. OCR has been used for scanned documents to extract the text information in [9]. Similarly, they have pre-processed the scanned invoices by sharpening the images, threshing and binarization.

2.1 Key-Value Pair Extraction using Regular Expression

Regular expressions are patterns used to match character combinations in strings to find by text with colon. If the regular expression manages to find the word, this means that it is a key. A key-value searching system is developed with the open-source Tesseract OCR engine and post-processing techniques with regular expressions [10].

This method can be used to create a low-cost office automation system for invoice processing. It learns patterns in the dataset, collates all different types of patterns and put them into a pattern dictionary. The regular extraction pattern dictionary can be expanded over time to learn a wider variety of patterns, with more datasets being included and more patterns being added.

Using regular expressions to extract key-value pairs is efficient in finding specific patterns or text. However, when a new text or pattern is introduced, that would be an issue as the system does not understand and cannot extract them. Additionally, some keys and values may have a variety of different forms of patterns. Thus, there is a need to manually look through the dataset and update the patterns to extract the key-value pairs correctly. Furthermore, with many patterns declared, it degrades the readability and performance of the codes.

2.2 Key-Value Pair Extraction using NLP

Bidirectional Encoder Representations from Transformers (BERT) [11] utilizes Transformer, an attention mechanism that discovers contextual relationships between words in a text. Transformer has two independent working parts: an encoder that reads text inputs, and a decoder that generates a task prediction. BERT has two training tasks: Masked Language Model (MLM) and Next Sentence Prediction. Robustly Optimized BERT Pretraining Approach (Roberta) [12] is another approach for pretraining NLP with similar architecture to BERT, and training with bigger batch sizes and longer sequences. The BERT is extended to another model, StructBERT, by incorporating language structures into pre-training [13], and leveraging structural information in addition to the present masking method. Two structural objectives are added to model pre-training, focusing on inner-sentence and inter-sentence structures. It allows StructBERT to represent language structures explicitly to reconstruct the correct order of words and sentences for accurate predictions.

The NLP approach only takes in the text from the document and does not incorporate the position of the text. In the tasks of extracting key value pairs, the position of the text is useful. But StructBERT is able to understand the key-value pair structure using together with layout awareness algorithms to improve the accuracy.

2.3 Key-Value Pair Extraction using Layout Detection

LayoutLM [14] is reported to do labeling using positional information, text-based information, and image information. As an upgrade from LayoutLM, the LayoutLMv2 uses model architectures to pre-train text, layout and image in a multi-modal framework [15]. Unlike other Visually-rich Document Understanding (VrDU) approaches which aims to analyze scanned documents, LayoutLMv2 helps learn cross-modality

interaction and incorporates a spatially-aware self-attention mechanism into the Transformer design. It comprehends relative positioning relationships different text blocks. The performance of LayoutLMv2 is studied with multiple datasets, including open sourced FUNSD dataset which consists of different scanned forms [16].

Another approach, LAMBERT, is reported in [17] that tackles the challenges of comprehending documents where non-trivial layout affects the local semantics. It is a Layout-Aware Language Model, which combines NLP methods with layout understanding mechanisms. The LAMBERT uses the layout information of the document and trains it with the pretrained Roberta.

Influenced by LayoutLM, a pre-trained approach StructuralLM utilizes cells and document layouts from scanned documents [18]. It uses cell-level 2D-position embeddings to represent the layout information of cells. It introduces a cell position classification which attempts to predict a cell location and their semantic relationships.

3 Methodology

3.1 Proposed Pipeline of the Information Extraction System

The flow chart of the proposed information extraction system is shown in Fig. 1, which consists of two portions: the key-value pair classification and linking the key-value pairs. The key-value pair classification portion is to explore models to classify the text from OCR to “Question”, “Answer” and “Others”. The second portion is to link the key-value pairs with the use of layout spatial awareness like the bounding box position.

To begin the pipeline of the proposed system, data collection is performed by taking images of invoices or receipts, with a quality check if the images are not blurry. After that, the invoices will be annotated into the FUNSD format [16] using an open source tool, Banksy [19], where the Named Entity Recognition (NER), Named Entity Linking (NEL), and a box region on the image will be outputted. The NER will be the labels into “Question”, “Answer” or “Others”. The NEL is the task to link the “Question” text to the “Answer” text. Lastly, the bounding box is drawn for the region of texts.

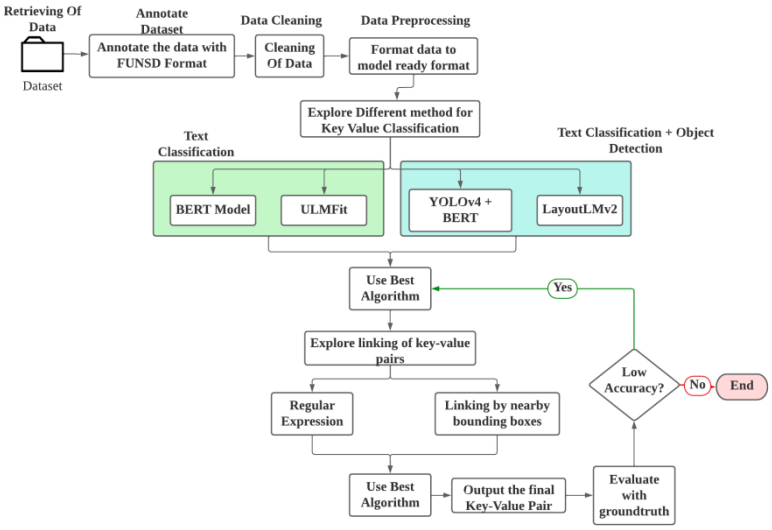

Fig. 1. Flow chart for the pipeline of proposed information extraction system.

The next step is to perform data cleaning and data preprocessing to format the data into a model-ready form with splitting the dataset into the train set (80%) and the test set (20%). For the key-value pair classification step in the pipeline of information extraction system, different intelligent algorithms are evaluated with the best performing model being chosen. For the linking of key-value pairs step, both channels of the regular expression and pairing by nearby bounding boxes are evaluated in the pipeline, where the better performing channel is selected to output the final key-value pair.

The output is then compared with the ground truth. If the performance of the output from the proposed information extraction pipeline is not satisfied, it will go through another iteration to improve the accuracy with different model hyperparameters. The iterative process keeps going till the satisfied accuracy achieved, with the optimal model selected by the proposed pipeline.

3.2 Algorithms Evaluated in the Pipeline

1) Key-value Pair Classification Step of the Pipeline. Two types of techniques are in the pipeline: NLP methods, and NLP methods combined with layout spatial awareness.

NLP methods are able to understand context-sensitive human languages. By using NLP, it can effectively extract data from text-based documents [18]. For the NLP methods, both BERT model and Universal Language Model (ULMFit) model are explored in the pipeline. The BERT model analyzes the left and right sides of a word to infer its context. Additionally, The BERT model uses MLM, which covers or masks a word in a sentence. MLM enables or enforces bidirectional learning from a text by requiring BERT to predict the word on either side of the covered word [20]. Applying it to our constructed cargo invoice dataset, the model could understand the key-value pairs. For example, the term "Product" could be next to the word "Number". The ULMFit model

is trained using a general-domain corpus to capture overall language properties at several layers and then learns task-specific features [21]. It uses 3-layer Weight-Dropped Long Short-Term Memory Networks (LSTM).

As the other technique, NLP with layout spatial awareness, the pipeline will evaluate two algorithms. The first algorithm is YOLOv4 combined with BERT. This algorithm is able to understand the position of the text as well as using BERT to understand the text embeddings. The second algorithm is LayoutLMv2; LayoutLMv2 not only considers the text and layout information, but also integrates the new text-image alignment and text image matching tasks, which help to learn cross-modality interaction [15].

2) Linking of Key-value Pair Step in the Pipeline. Both regular expression and Pairing via Nearby Bounding Box methods are incorporated. The regular expression is efficient in finding specific patterns or text which is common in invoices and receipts, for example, "Product No.: 123". The patterns commonly seen for keys and values are with colons. For the regular expression algorithm, an analysis will be done to see the common key-value pairs and their patterns to extract the key and values pairing. The workflows of the regular expression algorithm are as follow:

- Find all key-value pairs.
- Calculate Levenshtein distance with given identifiers to see which one is the most likely identifier.
- Return key-value pair with the lowest normalized Levenshtein distance.

The Pairing via Nearby Bounding Box algorithm pairs the key and value by finding the nearest bounding box according to the pairing rules. It combines the word level text into sentence level to make sense of the full question and answer, based on the bounding box position and the labels. There are two pairing rules in the proposed pipeline.

The flow of how the first key-value pairing rule works is shown in Fig. 2. With the words of the key and values, it first checks if “Question”/ “Answer” has a right neighbor. Then the pairing rule checks whether the *y* coordinates between the neighbors are the same. Next, it checks whether the coordinates of the right side of one bounding box (bbox) are identical to the left side of the other bounding box. The pairing results are returned.

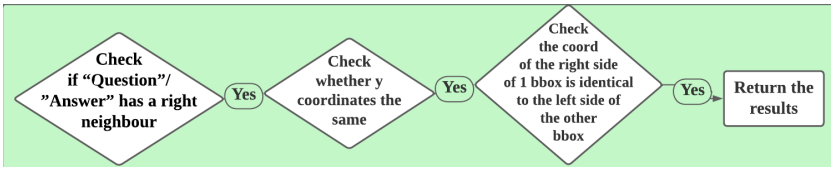

Fig. 2. Flowchart of the first key-value pairing rule.

After performing the first key-value pairing rule, the derived results are split into successfully paired outputs and unsuccessfully paired outputs. The unsuccessfully paired results will be taken into the second key-value pairing rule to pair the questions and answers, whose flowchart is shown in Fig. 3. Firstly, it checks whether the center

of both bounding boxes is within the range of the other bbox in the *y* direction. Then it checks whether both bounding boxes overlap in the *x* direction or the gap in the *x* direction less than 50% of the width of the smaller bounding box. If yes, the key and value will be paired successfully.

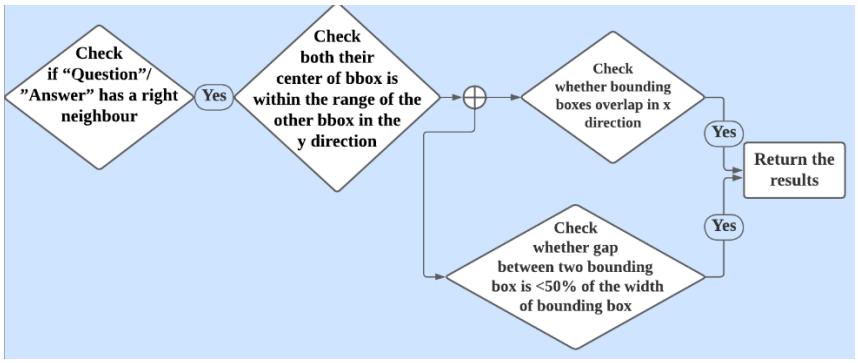

Fig. 3. Flowchart of the second key-value pairing rule.

The final output of this step consists of information including the labels, bounding boxes, text, left_neightbour, right_neighbour, the width, the height, the variable of pair_with, and the center attribute.

3.3 Datasets

The datasets used for experiments are the FUNSD dataset [16], and the Cargo Invoices dataset which is constructed in this research with cargo invoices images obtained from the warehouses. Table 1 lists the number of training and testing data in these datasets.

Table 1. Number of data in these two datasets.

Dataset	Training	Testing	Total Data
FUNSD [16]	149	50	199
Cargo Invoices Dataset	484	121	605

4 Experiment Results and Analysis

4.1 Evaluation Metrics

The evaluation metrics used in the ICDAR 2019 Robust Reading Challenge on Scanned Receipts OCR and Information Extraction [22] are employed to evaluate the experiment performances in this paper: the Precision, Recall and F1 Score. The extracted text will be compared to the ground truth for each test image. If the submitted content and category of the extracted text matches the ground truth, it is marked as correct; otherwise, it is labelled as inaccurate. Furthermore, the algorithms will also be further evaluated by the classification report and confusion matrix.

The values of the precision, recall, F1 score, and support are the four most critical headers for classification results to pay attention to in the classification reports. The value of precision refers to the ability of a classifier to avoid labelling a negative instance as positive [23]. The recall is the ability of a classifier to find all positive instances. The F1 score is a weighted harmonic mean of precision and recall. The support is the number of actual class occurrences in the specified dataset.

4.2 Evaluations for Classification Models in the Pipeline

The dataset is split into 80% training and 20% testing. Four classification models in the pipeline are ULMFit, BERT, YOLOv4 combined with BERT and LayoutLMv2, each of which classifies texts into “Question”, “Answer” and “Others”.

Table 2. ULMFit Results with Cargo Invoice Dataset.

Class	Precision	Recall	F1 Score	Support
Question	99%	98%	98%	882
Answer	88%	96%	92%	882
Other	93%	80%	86%	568
Accuracy		93%		2332

1) Results of the ULMFit Model. The results of precision, recall, F1 score and support for the ULMFit model with the Cargo Invoice dataset are shown in Table 2. Overall, the performance of this model is good with 93% accuracy. However, the model does not perform well for the "Other" class, compared to the other two classes. It might be because there are fewer "Other" labels compared to those of "Question" and "Answer".

2) Results of the BERT Model. The experiment results of the BERT classification model with the Cargo Invoice dataset are shown in Table 3. It achieves 88% accuracy. This model can perform well on the “Other” class for the values of precision, recall, and F1 score, but not do well on the “Answer” class.

Table 3. BERT Results with Cargo Invoice Dataset.

Class	Precision	Recall	F1 Score	Support
Question	85%	91%	88%	2128
Answer	83%	76%	79%	1387
Other	96%	95%	95%	1493
Accuracy		88%		5008

3) Results of YOLOv4 Combined with BERT Model. The classification results of the YOLOv4 combined with BERT model are shown in Table 4. It shows that the overall performance is worse than other classification models. For “Other” label, it performs the worst. It might be due to the imbalanced dataset whereby “Other” has the least amount of data comparatively. The results in this experiment show that the combination of YOLOv4 and BERT algorithms does not enhance the performance, compared with the BERT model alone.

Table 4. Results of YOLOV4 Combined with BERT Model.

	Precision	Recall	F1 Score	Support
Answer	52%	42%	46%	882
Other	46%	39%	40%	568
Question	49%	40%	44%	882
Accuracy	40.27%			2332

4) Results of the LayoutLMv2 Model. Different from the other three models, the labeling methods of the LayoutLMv2 model follow the BIOES tagging, where B means beginning; I mean in the middle; O means others; E means the ending; and S means a single word representing a full sequence [24]. For example, "Product No.", will be split into "Product" which will be B-QUESTION and "No." is "E-QUESTION".

The experiment results of the LayoutLMv2 classification model are shown in Table 5. It is observed that the LayoutLMv2 model achieves an accuracy of 96%, which is the highest out of other three classification models. All the labels achieve more than 90% for the precision, recall and F1 score. It shows that the LayoutLMv2 model can predict well and distinguish each label properly.

Table 5. LayoutLMv2 Results with Cargo Invoice Dataset.

	Precision	Recall	F1 Score	Support
B-ANSWER	93%	94%	94%	331
B-QUESTION	98%	98%	98%	507
E-ANSWER	90%	96%	93%	331
E-QUESTION	98%	98%	98%	507
I-ANSWER	96%	98%	97%	915
I-QUESTION	96%	97%	97%	104
O	97%	93%	94%	1387
S-ANSWER	95%	97%	96%	502
S-QUESTION	96%	98%	97%	375
Accuracy	96%			4959

5) Experiment Results with the FUNSD Dataset. The classification results of the experiments are also performed on the FUNSD dataset to compare the performances of these four classification models, shown in Table 6.

Table 6. Classification Results with the FUNSD Dataset.

Model	Precision	Recall	F1 Score
ULMFit	77%	76%	76%
BERT	55%	67%	60%
YOLO combined with BERT	49%	40%	44%
LayoutLMV2	80%	85%	83%

It has shown that LayoutLMv2 model performs the best out of other three models on both datasets. This has been demonstrated that integrating the spatial-aware self-attention mechanism into the Transformer architecture in the LayoutLMv2 model has fully allowed the model to understand the relative positional relationship among different text blocks. On the other hand, YOLOv4 combined with the BERT model does not

perform well. Even though it is trained with the text along with the bounding box of the texts, it is not able to find the relative positional relationship accurately compared to the LayoutLMv2 model. Thus, the LayoutLMv2 model will be the model chosen by the proposed pipeline process for the key-value pair classification tasks.

4.3 Evaluation for Linking of Key-value Pair Algorithms in the Pipeline

Some analysis is done on the dataset to understand the key-value patterns and how they are paired to enhance the extracting process. There are various formats of the keys, such as the short forms. Hence, a list of the different forms of keys is created with some examples of various forms shown in Table 7.

After finding out different formats of the keys from the dataset, the next step is understanding their values pattern from these common keys. Some examples of the patterns of values are shown in Table 8.

After performing the key-value pairing, the results are evaluated by comparing the derived final key-value pairs with the ground-truth. The experiment result comparisons are shown in Table 9.

Table 7. Some Examples of Different Formats of Keys.

Word	Different Formats of Keys
Gross Weight	G/W G/Weight Gross wt Gross wght Gross wght(kg)
Net Weight	Net WT Net wght Net weight(kg) Net wt(kg) Net wght(kg) N/W
Dimension	Dim Dims Dim(mm) DIM (CM) Dimensions(cm)

Table 8. Pattern of Values.

Key	Values	Patterns
Gross Weight	G/W 2134 23Kg/g 88,500 KG	For gross weight the pattern is always an integer/float followed by the metrics (KG/lb/g). Sometimes G/W would be infront of the integer/float.
Net Weight	270 lb 88,500 KG 3.840 KG	For Net weight the pattern is similar to Gross weight. The pattern is always an integer/float followed by the metrics (KG/lb/g).
PO	PO-IMA-21007 PO2000004328	For PO the pattern is PO followed by letters then integers or just integers.
Dimension	228.00 × 148.00 × 165.00 CM 89.76 × 58.26 × 64.96 INCH 3815 × 2150 × 2550 mm	For Dimension, the pattern is integer/float X integer/float X integer/float

Table 9. Results for Key Value Pairing in the pipeline.

Algorithm	Precision	Recall	F1 Score
Regular Expression	63%	60%	66%
Pairing via Nearby Bounding Box	73%	72%	70%

It is observed from the results that the second algorithm, Pairing via Nearby Bounding Box, has done a better job with a precision of 73%, recall of 72% and F1 score of 70%, than those of the regular expression algorithm. The reason for the regular expression not performing well might be because there are limited patterns in the system.

Even though the algorithm of Pairing via Nearby Bounding Box performs better, it can still be further improved, as on some occasions the questions and answers are very far apart in a horizontal direction. Furthermore, there are a few mistakes that are made by the OCR results which causes the classification models to misclassify some labels and hence messes up the key-value pairs.

5 Conclusion

In this paper, an end-to-end pipeline of information extraction system for extracting key-value pairs from invoices or receipts is presented. The proposed system uses deep learning and intelligent computing approaches for knowledge discovery, classifying key-value pairs, and linking the key-value pairs. Its performances are evaluated.

First, a few deep learning approaches are employed to explore key-value label classification and linking key-value pairs. Experiments are conducted to evaluate and compare the performances of each model for key-value pairs in the pipeline. It is observed from the results that the LayoutLMv2 model performs the best. It shows that the LayoutLMv2 architecture of layout spatial awareness and words embedding improve the results of standard text classification.

Afterwards, experiments for linking key-value pairs are conducted for two methods for linking the key-value pairs in the pipeline: the regular expression and a unique method by linking the key-value pairs by finding the nearby bounding boxes. Evaluated the performance, the algorithm of linking by Pairing via Nearby Bounding Box performs better with a precision of 73%, recall of 72% and F1-score of 70%.

Several recommendations can be made for future improvements to the overall pipeline. For the key-value label classification, the LayoutLMv2 [15] model has recently introduced a new version known as LayoutLMv3 [25], which may be implemented to investigate its efficacy in enhancing the key-value label classification results compared to the current model in the pipeline of information extraction system.

For linking key-value pairs, most key-value pairs are typically located beside each other on the same line horizontally. However, there are some cases where the values are below the keys vertically. Currently, this proposed pipeline of information extraction system has not explored that and can only find nearby bounding boxes horizontally. Hence, to further improve the system, it needs to be able to get nearby key-value pairs vertically.

Acknowledgment

The first author would like to thank her intern supervisor Mr. Eric Tan of Infocomm Media Development Authority (IMDA) Singapore, for his guidance and dedicated supports in the project.

References

1. Turner, R.: The myth of the paperless office. New Library World, 104(3), 120-121 (2003).
2. Klein, B., Agne, S., Dengel, A.: Results of a study on invoice-reading systems in Germany. In: Int. Workshop on Document Analysis Systems, pp. 451–462 (2004).
3. Document Recognizer to modernize information processing, https://crossmasters.com/en/blog/document-recognizer-to-modernize-information-processing/, last accessed 2022/7/16.
4. Kay, A.: Tesseract: an open-source optical character recognition engine. Linux Journal (2007).
5. Amazon Web Services: Form Data (Key-Value Pairs), https://docs.aws.amazon.com/textract/latest/dg/how-it-works-kvp.html, last accessed 2022/6/15.
6. Qing, Y., Zeng, Y., Cao, Q., Huang, G.-B.: End-to-end novel visual categories learning via auxiliary self-supervision. Neural Networks, 139, 24-32 (2021).
7. Xu, D., Li, Z., Cao, Q.: Object-based illumination transferring and rendering for applications of mixed reality. Visual Computer, 38(12), 4251-4265 (2022).
8. Kumar, V. Kaware, P. Singh, P., et al.: Extraction of information from bill receipts using optical character recognition. Int. Conf. on Smart Electronics and Communication, pp. 72-77 (2020).
9. Kamisetty, V. S. R., Chidvilas, B. S., Revathy, S., et al.: Digitization of data from invoice using OCR. In: 6th Int. Conf. on Computing Methodologies and Communication (2022).
10. Kaló, Á. Z., Sipos, M. L.: Key-value pair searching system via Tesseract OCR and post processing. In: 19th World Symp. on Applied Machine Intelligence and Informatics (2021).
11. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. North American Chapter of the Association for Computational Linguistics (2019).
12. Liu, Y., Ott, M., Goyal, N., et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 (2019).
13. Wang, W., Bi, B., Yan, M., et al.: StructBERT: Incorporating language structures into pretraining for deep language understanding. In: Int. Conf. on Learning Representations (2020).
14. Xu, Y., Li, M., Cui, L., et al.: LayoutLM: Pre-training of text and layout for document image understanding. In: 26th ACM Int. Conf. on Knowledge Discovery & Data Mining (2020).
15. Xu, Y., Xu, Y., Lv, T., et al.: LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In: 59th Annual Meeting of the Association for Computational Linguistics and 11th Int. Joint Conf. on Natural Language Processing (2021).
16. Jaume, G. Ekenel H. K., Thiran, J.: FUNSD: A dataset for form understanding in noisy scanned documents. In: Int. Conf. on Document Analysis and Recognition Workshops, pp. 1-6 (2019).
17. Garncarek, L., Powalski, R., Stanisławek, T., et al.: LAMBERT: Layout-aware language modeling for information extraction. In: Int. Conf. on Document Analysis and Recognition (2021).

18. Li, C., Bi, B., Yan, M., et al.: StructuralLM: Structural pre-training for form understanding. In: 59th Annual Meeting of the Association for Computational Linguistics and 11th Int. Joint Conf. on Natural Language Processing (2021).
19. Banksy Annotation Tool, https://github.com/AboutGoods/Banksy-annotation-tool, last accessed 2022/6/18.
20. Muller, B.: BERT 101 state of the art NLP model explained. https://huggingface.co/blog/bert-101, last accessed 2022/6/21.
21. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: 56th Annual Meeting of the Association for Computational Linguistics (2018).
22. ICDAR 2019 robust reading challenge on scanned receipts OCR and information extraction, https://rrc.cvc.uab.es/?ch=13&com=tasks, last accessed 2022/6/13.
23. Agrawal, S.: Metrics to evaluate your classification model to take the right decisions, https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/, last accessed 2022/6/21.
24. Johansen, B.: Named-entity recognition for norwegian. In: 22nd Nordic Conference on Computational Linguistics (2019).
25. Huang, Y., Lv, T., Cui, L., Lu, Y., Wei, F.: LayoutLMv3: Pre-training for document AI with unified text and image masking. arXiv:2204.08387 (2022).