A Robust Framework for One-Shot Key Information Extraction via Deep Partial Graph Matching

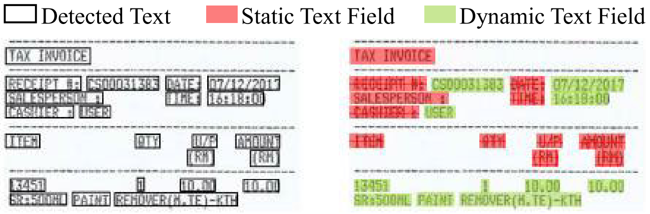
Minghong Yao, Zhiguang Liu, Liansheng Zhuang, *Member, IEEE*,
Liangwei Wang, and Houqiang Li, *Fellow, IEEE*

***Abstract*— Text field labelling plays a key role in Key Information Extraction (KIE) from structured document images. However, existing methods ignore the field drift and outlier problems, which limit their performance and make them less robust. This paper casts the text field labelling problem into a partial graph matching problem and proposes an end-to-end trainable framework called Deep Partial Graph Matching (dPGM) for the one-shot KIE task. It represents each document as a graph and estimates the correspondence between text fields from different documents by maximizing the graph similarity of different documents. Our framework obtains a strict one-to-one correspondence by adopting a combinatorial solver module with an extra one-to-(at most)-one mapping constraint to do the exact graph matching, which leads to the robustness of the field drift problem and the outlier problem. Finally, a large one-shot KIE dataset named DKIE is collected and annotated to promote research of the KIE task. This dataset will be released to the research and industry communities. Extensive experiments on both the public and our new DKIE datasets show that our method can achieve state-of-the-art performance and is more robust than existing methods.**

***Index Terms*— Document understanding, key information extraction, visual information extraction, graph matching.**

I. Introduction

INFORMATION extraction from structured documents has been an active topic in the research of Information Retrieval techniques. The traditional information retrieval task is usually based on pure text in documents. This task becomes more challenging when the system reads images of documents as input. Key Information Extraction (KIE) from structured document images aims to extract texts of a number of key fields from given document images and save texts to structured documents. In general, a typical KIE method consists of two key steps, including text detection and recognition and text field labelling, as shown in Fig. 1. A document image is firstly split into regions of interest by text detection modules. Then, texts in each region are obtained via text recognition modules. Finally, each text region is assigned a label by text field labelling modules so that people can extract texts of key information in structured formats according to their labels. Benefiting from the rapid advance of deep learning for document image understanding, both text detection and text recognition are well studied and have achieved great progress [1], [2], [3]. Text fields labelling modules are less explored. This paper focuses on solving the text field labelling problem where only one example image for each style is provided with labels that need to be extracted. Our settings are more challenging and practical than those in existing works [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19] that feed on a large number of samples for each document style.



(a) Text detection and recognition (b) Text field labelling

Fig. 1. The pipeline of extracting key information. (a) Text detection and recognition, (b) Text field labelling. Fields of a document consist of static and dynamic fields. Only dynamic fields have labels. Later, key information in dynamic fields can be extracted according to their predicted labels.

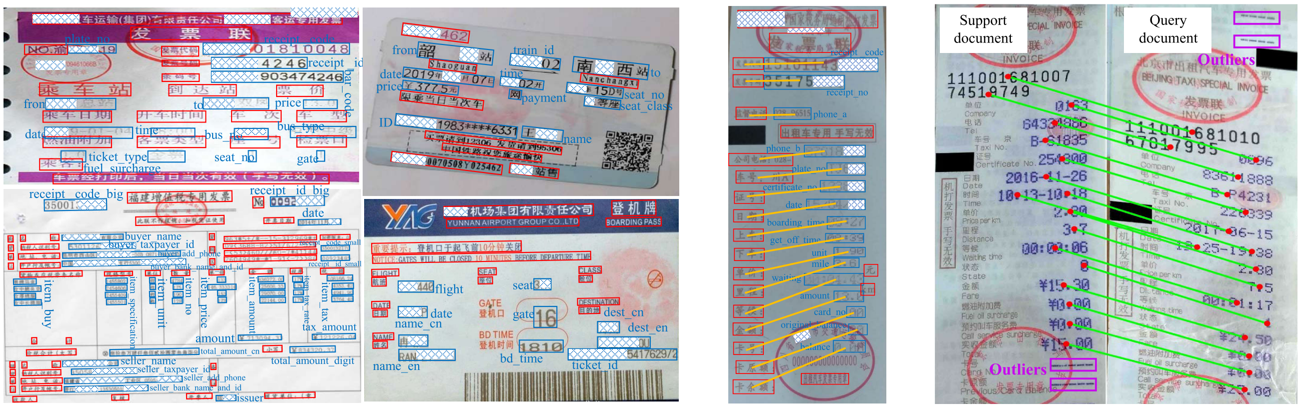
The past few decades have witnessed significant progress in key information extraction from document images, and many methods for the text field labelling problem have shown to have a good performance [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. However, it is still far from over. One main challenge faced by researchers is the diversity of the layouts of documents. Different types of documents have different layout styles. The number of styles of different layouts is overwhelming as shown in Fig. 2 (a). Other challenges of text field labelling include field drift and outlier problems as shown in Fig. 2 (b) and (c). Field drift problem often occurs when the printing paper of invoices slips during printing. In this case, some text fields drift into unexpected positions, which easily causes wrong labelling. The outlier problem may happen when documents are post-processed such as taking notes on them.

Manuscript received 10 October 2022; revised 3 August 2023; accepted 22 December 2023. Date of publication 29 January 2024; date of current version 2 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U20B2070, Grant 61976199, and Grant 61836011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiangqian Wu. *(Minghong Yao and Zhiguang Liu contributed equally to this work.) (Corresponding author: Liansheng Zhuang.)*

Minghong Yao, Liansheng Zhuang, and Houqiang Li are with the CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application Systems, University of Science and Technology of China, Hefei 230000, China (e-mail: mhyao1@mail.ustc.edu.cn; lszhuang@ustc.edu.cn; lihq@ustc.edu.cn).

Zhiguang Liu and Liangwei Wang are with the Noah's Ark Laboratory, Huawei Technologies, Shenzhen 518000, China (e-mail: liuzhiguang1@huawei.com; wangliangwei@huawei.com).

Digital Object Identifier 10.1109/TIP.2024.3357251



(a) Diversity of layouts (b) Drift problems (c) Outlier problems.

Fig. 2. Challenges of text field labelling. In (a)-(b), red boxes are static text fields, and their text stays the same for all documents of the same type. Blue boxes are dynamic text fields and their text is the key information to be extracted. (a) shows the diversity of layouts. (b) shows the field drift problem. The slop yellow lines indicate that the dynamic fields drift upwards. (c) shows the outlier problem. Purple boxes are outliers. As green lines show, graph matching can be used to solve the one-shot text field labelling task by transferring the labels from the support document to the query document.

In this case, some unseen text fields may appear in the query document, as shown in Fig. 2 (c), and disturb the labelling. These challenges may significantly degrade the performance of KIE methods. However, most existing methods focus on the layout diversity problem while ignoring the field drift and outlier problems, which limit their performance and make them less robust. Moreover, most existing methods require a large number of training data, which makes them difficult to use in practice because collecting and annotating enough training data are difficult and even impossible due to privacy issues. In many cases, users can only provide a few example document images labelled with the information that needs to be extracted.

Motivated by the above insights, this paper proposes a novel end-to-end trainable framework called deep Partial Graph Matching (denoted as dPGM) for the one-shot KIE task. The key idea is to represent each document as a graph and to cast the one-shot text field labelling task as a Partial Graph Matching (PGM) problem. The nodes of the graph represent text fields, and the edges represent some visibility relations between text fields. Since static text fields (whose contents are pre-defined and fixed) are easily located by keyword matching, our framework only estimates the correspondence between dynamic text fields in a test document and those in a support document. After obtaining the text field correspondence as shown in Fig. 2 (c), the labels in a support document will transfer to the dynamic text fields in a test document to accomplish the mission of text field labelling.

Compared with most existing learning-based methods [10], [13], [14], [15] which require a separate model for each type of document to gain promising performance, our dPGM framework allows to train on multiple datasets and inference on different types of documents with solely one model and thus is particularly suitable for the one-shot KIE task. Note here that, since the correspondence quality has a great impact on the final performance, this paper introduces a combinatorial solver module with an extra one-to-(at most)-one mapping constraint to get a strict one-to-one correspondence of dynamic text fields. Different from the existing method [21] which independently predicts the correspondence for each text field in a test document and often gets a sub-optimal solution, the solver module jointly estimates the correspondence for all dynamic text fields and thus obtains a globally optimal solution. Moreover, benefiting from the strict one-to-(at most)-one mapping constraint, the many-to-many correspondence cases caused by the drift fields and outlier fields are excluded. Therefore, our proposed method is robust to the field drift problem and the outlier problem.

Finally, to accelerate the research of the KIE problem, we collected and annotated a one-shot document KIE dataset named DKIE dataset, which consists of 2,500 document images captured by mobile phones in natural scenes. The dataset covers diverse types of document images, and many of them are highly difficult with spatial drift. We will release the data set to the research and industry communities. Extensive experiments on a public dataset [22] and our DKIE dataset show that our proposed framework is more robust and outperforms existing state-of-the-art methods.

In summary, the contributions of this paper are as follows:

- An end-to-end trainable framework based on partial graph matching is proposed for the one-shot KIE task. Different from existing dominant methods [10], [13], [14], [15] which require a separate model for each type of document, our framework allows training on multiple datasets and inference on different types of documents with solely one model.
- A combinatorial solver module with an extra one-to-(at most)-one mapping constraint is introduced to do exact graph matching. Benefiting from the strict one-to-one correspondence, our framework is robust to the field drift problem and the outlier problem.
- A large dataset named DKIE is collected and annotated to promote the research on the KIE problem. To our knowledge, the DKIE dataset is the largest available one-shot KIE dataset up to now.

II. Related Work

A. Key Information Extraction

There are many methods proposed to solve the text field labelling problem [4], [5], [6], [7], [8], [9], [10], [11], [21], [23], [24], [25], [26], [27], [28]. Early attempts are usually based on hand-crafted features for each layout style, such as regex and template matching [23], [24], [28], [29], [30], [31]. However, as these methods require much task-specific knowledge and human-designed rules, they are only limited to specific layouts. Additionally, they can not scale to other types of documents. To improve the generalization, many learning-based methods are proposed to automatically adapt to any type of layout by turning the KIE task into a Named Entity Recognition (NER) problem [11], [12], [32], [33], [34], [35]. They serialize all text in a document image into a sequence, then apply a sequential prediction model (such as the BiLSTM-CRF model [36]) to predict the label of each word using additional visual features such as position, font size and color. The main difference of these methods lies in their encoders.

To model the non-sequential relationship between text fields, graph neural network modules are introduced to refine both text fields' text and visual features before they are fed into a BiLSTM-CRF based module [6], [7], [8], [9], [10], [20]. Inspired by the progress of large pre-trained language models [37], [38], many general-purpose multimodal pre-training methods are proposed [13], [14], [15], [16], [17], [18], [19] and achieved promising results. They jointly models the text and layout information within a single framework using the transformer module.

Though learning-based methods have achieved promising results, they require a large amount of training or fine-tuning data to train separate models for each document type. However, collecting and annotating sufficient training data is time-consuming, and even impossible due to privacy. In practice, users can provide a few example document images labelled with the information that needs to be extracted.

B. Few-Shot KIE Learning

Few-shot KIE methods attract great attention of researchers due to their practicalities. Some few-shot learning KIE methods have been proposed in the past few years [21], [25], [27], [28], [30]. To generalize to unseen types of documents in the region of few-shot learning, one line of work [25], [27] is to employ the prior knowledge from a knowledge graph. By calling a cloud API of Google's knowledge graph, Tata et al. [27] used entity detectors to identify the possible labels of each text field. Instead of reusing prior knowledge about entities, Sunder et al. [25] proposed to build a small knowledge graph for each document, where the nodes in this knowledge graph are text fields and the relationships between nodes are predefined spatial relationships between them such as "Above-below". Later, a deductive learning module was adopted to synthesize reusable logic programs that can extract key information from unseen types of documents. The knowledge graph is either too expensive to construct or has privacy issues in scenarios where the contents of documents are confidential and entity detectors relying on cloud API are not acceptable. This limits the application of their methods.

Another line of work [21], [28], [30] focuses on the one-shot learning scenario. In this scenario, the users are asked to manually label all the fields to extract from a single document image (support document). Then all the unlabeled documents (query documents) of the same type can be processed automatically. Features that describe the spatial relationships between static and dynamic fields can be reused to help models generalize to unseen types of documents with the help of various handcraft features and heuristics [28], [30].

Instead of using handcraft features and heuristics [28], [30], Cheng et at. [21] proposed to use a multi-layer perceptron (MLP) to learn the similarity score between dynamic text fields in a support document and the ones in a query document. Experiments show that the MLP can take advantage of static text fields more efficiently. However, when the spatial relationships between static and dynamic text fields in support documents is different from the ones in query documents because of the field drift problem as shown in Fig. 2 (b), their method is not robust. Different from [21] who believed that the one-shot KIE cannot be solved by graph matching because of the multi-region fields, this paper uses a mapping constraint in the solver of PGM to relieve the field drift and outlier problems.

III. Our Model

This section presents a robust framework for one-shot KIE that can relieve the challenge of field drift and outlier problems. Sec III-A presents a brief overview of the proposed framework. Sec III-B describes how to construct graphs for documents. Sec III-C introduces how to calculate graph similarity. Sec III-D shows the estimation of graph correspondence.

A. Architecture Overview

Fig. 3 illustrates the overall framework, consisting of three stages: graph construction, similarity calculation, and correspondence estimation. Graph construction module constructs graphs for support and query document, G_s and G_q. The graph similarity calculation module uses several MLPs to calculate the nodes' and edges' similarity. The correspondence estimation module uses a combinatorial solver that solves a PGM problem to estimate the correspondence between G_s and G_q based on the nodes and edges similarity. At last, labels in a support document are transferred into a query document according to the correspondence between them.

B. Graph Construction Module

Take the graph of a support document G_s as an example, all nodes are the dynamic text fields, and the edges reflect the visibility between nodes. In this paper, $\{F_s\}$ represents the set of nodes in G_s and $|F_s|$ represents the number of nodes. Similarly, $\{FF_s\}$ represents the set of edges and $|FF_s|$ represents the number of edges. The meaning of $|F_q|$ and $|FF_q|$ are similar. The following paragraphs introduce precise definitions or important details of nodes and edges.

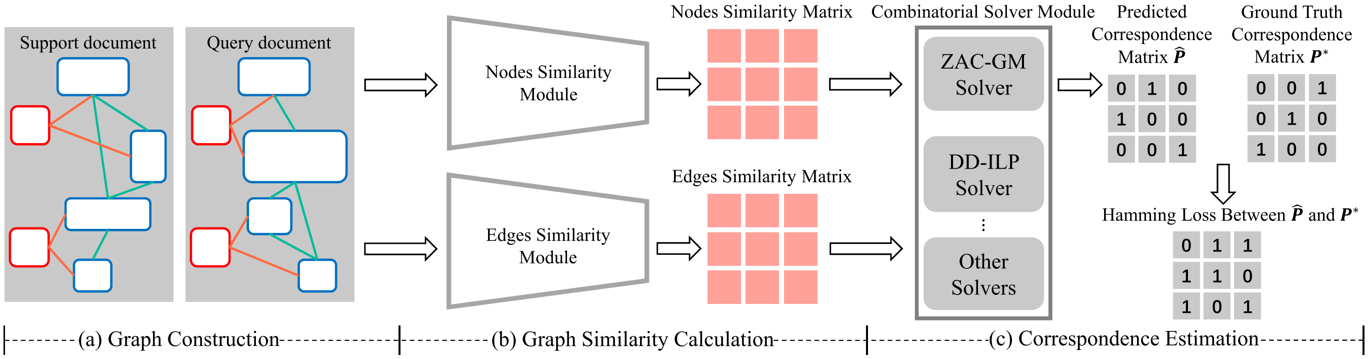


Fig. 3. Overview of the proposed model. In step (a), we build the graphs. In step (b), we feed different features into separate Multi-layer Perceptrons (MLP), and their outputs are nodes and edges similarity matrices. In step (c), the nodes and edges similarity matrices are fed into a combinatorial solver to estimate the correspondence between two graphs.

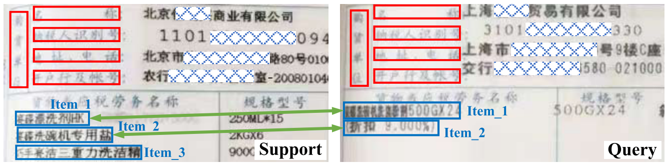


Fig. 4. Multi-region fields. Red boxes are static text fields and blue boxes are multi-region dynamic fields. When a piece of text is too long to fit into one line, multi-region fields appear.

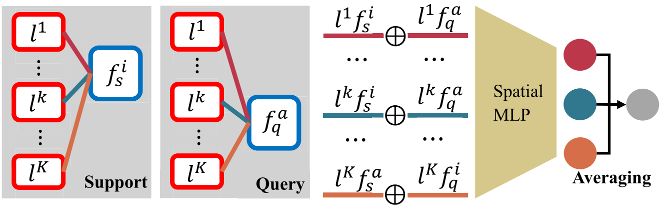

l^1
l^k
l^K
f_s^i
Support
l^1
l^k
l^K
f_q^a
Query
$l^1 f_s^i \oplus l^1 f_q^a$
$l^k f_s^i \oplus l^k f_q^a$
$l^K f_s^a \oplus l^K f_q^i$
Spatial MLP
Averaging

Fig. 5. Spatial similarity calculation.

Nodes: The i-th node in G_s is the i-th dynamic text field, and it is denoted as f_s^i. A node f_s^i has a set of node features and a label y_s^i. The set of node features consists of three types of features: a node aspect feature, a text feature, and a set of spatial features. The node aspect feature is the concatenation of the height and width of the bounding box of f_s^i. i.e., it is a two-dimensional feature. The text feature is the average word embedding of the words in f_s^i. The set of spatial features consists of all line segments that connect f_s^i and all static text fields. Specifically, let the k-th static text field be denoted as l^k, then the coordinates difference between l^k and f_s^i is taken as one of the spatial features and is denoted as $l^k f_s^i$. Assume that the number of static text fields in a support document is K, then the set of spatial features is denoted as $\{l^k f_s^i\}^{k=1,\cdots,K}$.

To cast the one-shot text field labelling task into a PGM problem, the labels annotated in a support document should be different from each other, i.e., $y_s^i \neq y_s^j$ if $i \neq j$. When the support document has multi-region fields, which usually share the same label, number suffixes are appended to the original labels, as shown in Fig. 4. After labeling the query documents, the number suffixes can be removed to restore the original labels. A good support document should contain as many dynamic text fields as possible to cover the possible multi-region fields in the query documents.

Edges: For a pair of nodes, say f_s^i and f_s^j, if the line segment connecting them is not blocked by another dynamic text field, then there will be an edge between them. Each edge has two features: an edge direction feature and an edge aspect feature. The direction feature is the coordinates difference between f_s^i and f_s^j and is denoted as $f_s^i f_s^j$. The edge aspect feature is the concatenation of two node aspect features, i.e., the node aspect feature of f_s^i and f_s^j is concatenated to form a four-dimensional feature.

C. Graph Similarity Calculation Module

The graph similarity between support and query documents includes node similarity and edge similarity.

Nodes Similarity: The node similarity between f_s^i and f_q^a is calculated based on their node features. Since each node has three types of node features, as stated in the subsection III-B, three different MLPs are adopted to calculate the spatial, aspect, and text similarity separately. Fig. 5 shows the calculation of spatial similarity. If a static text field l^k exists in both support and query documents, then l^k serves as a "landmark" so that an MLP can determine whether f_s^i and f_q^a are spatially similar by directly comparing the spatial relationship between l^k and f_s^i and the one between l^k and f_q^a. Put this idea in a formal way, two line segments $l^k f_s^i$ and $l^k f_q^a$ will be concatenated and then be fed into an MLP to calculate the spatial similarity between f_s^i and f_q^a w.r.t. l^k:

$$Sim_{spatial}(f_s^i, f_q^a, l^k) = MLP_{spatial}(l^k f_s^i \oplus l^k f_q^a), \qquad (1)$$

where "$\oplus$" denotes the concatenation operation and $l^k f_s^i \oplus l^k f_q^a$ is a four-dimensional vector. By averaging across all static text fields, the spatial similarity between f_s^i and f_q^a is calculated as:

$$Sim_{spatial}^{ia} = \frac{1}{|K|} \sum_{k=1}^{|K|} Sim_{spatial}(f_s^i, f_q^a, l^k), \qquad (2)$$

where $Sim_{spatial}^{ia}$ denotes the spatial similarity.

The aspect and text similarities between f_s^i and f_q^a are similar to the equation (1) but with separate MLPs. Let Sim_{aspect}^{ia} and Sim_{text}^{ia} to represent the aspect and text similarities between f_s^i and f_q^a:

$$A_i^a = \frac{1}{3}(Sim_{spatial}^{ia} + Sim_{aspect}^{ia} + Sim_{text}^{ia}), \tag{3}$$

where A_i^a represents the final node similarity between f_s^i and f_q^a. The nodes similarity matrix consists of A_i^a and its shape is $|F_s| \times |F_q|$.

Edges Similarity: The edge similarity between $f_s^i f_s^j$ and $f_q^a f_q^b$ is calculated based on their edge features and is denoted as A_{ij}^{ab}. Similar to the calculation of A_i^a, A_{ij}^{ab} is calculated as:

$$\begin{aligned} A_{ij}^{ab} = \frac{1}{2}(&MLP(f_s^i f_{s_{dir}}^j \oplus f_q^a f_{q_{dir}}^b) \\ &+ MLP(f_s^i f_{s_{asp}}^j \oplus f_q^a f_{q_{asp}}^b)) \end{aligned} \tag{4}$$

The shape of the edges similarity matrix is $|FF_s| \times |FF_q|$.

D. Correspondence Estimation Module

The correspondence between G_s and G_q is obtained by solving a constrained discrete optimization problem introduced by the PGM problem [39]. The objective of this discrete optimization problem is to find the best correspondence between G_s and G_q so that their graph similarity is maximized:

$$\max_P \quad F(P) = \sum_{i=1}^{|F_s|}\sum_{a=1}^{|F_q|} P_{ia} A_i^a P_{ia} + \sum_{ij\in FF_s}^{|FF_s|}\sum_{ab\in FF_q}^{|FF_q|} P_{ia} A_{ij}^{ab} P_{jb}, \tag{5}$$

$$\text{s.t.} \quad P \in \left\{ P\mathbf{1} \le \mathbf{1},\, P^\top \mathbf{1} \le \mathbf{1},\, P \in \{0,1\}^{|F_s|\times|F_q|} \right\}. \tag{6}$$

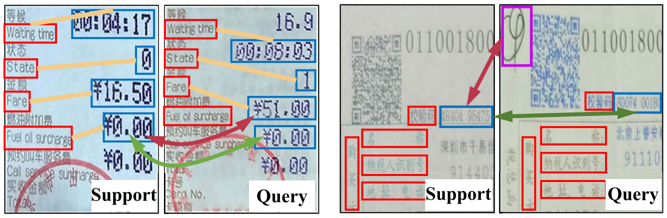
In equation (5) and (6), P represents the correspondence matrix between G_s and G_q, i.e., P_{ia} is 1 if a support field f_s^i is matched with a query field f_q^a, 0 otherwise. $F(P)$ denotes the graph similarity given P. $\mathbf{1}$ is a column-wise vector whose elements are all one. The first inequality in equation (6) forbids a feasible correspondence matrix P to match multiple support fields with one query field and vice versa by the second inequality. Both inequalities allow part of support and query fields to match with no fields if the outliers appeared in the documents. The equation (6) is also called the one-to-(at most)-one mapping constraints.

Solving Partial Graph Matching Problem: To solve the equation (5) and (6), the solver will relax all binary elements of P into decimals between 0 and 1 so that the feasible domain becomes continuous and convex. Let $\mathcal{P}$ represent the new feasible domain. Within the new feasible domain $\mathcal{P}$, a gradient-based method, called the Frank-Wolfe method [40], was adopted to maximize the graph similarity $F(P)$ by performing the following iterations:

$$\tilde{P}^{(t+1)} \in \arg\max_{P\in\mathcal{P}} \left\langle \nabla F\left(\hat{P}^{(t)}\right), P \right\rangle, \tag{7}$$

$$\hat{P}^{(t+1)} = \hat{P}^{(t)} + \alpha^{(t)}\left(\tilde{P}^{(t+1)} - \hat{P}^{(t)}\right), \tag{8}$$

where $\hat{P}^{(t)}$ is the approximate solution to (5) at the (t)-th iteration and its elements are decimals between 0 and 1. $\nabla F\left(\hat{P}^{(t)}\right)$ is the gradient of $F\left(\hat{P}^{(t)}\right)$ at $\hat{P}^{(t)}$ and $\alpha^{(t)}$ is the step size obtained by exact or inexact line search [41]. The sub-problem described in the equation (7) can be solved by the Hungarian algorithm [42] because the optimal solution $\tilde{P}^{(t+1)}$ is always an extreme point of $\mathcal{P}$. The elements in $\hat{P}^{(t)}$ can be interpreted as the possibilities of two fields that are matched with each other. Therefore, if one column (row) of $\hat{P}$ is close to a zero vector, then the corresponding query (support) field is not likely to match with any support (query) fields and this query field should be identified to be an outlier. After removing all columns (rows) of possible outliers from $\hat{P}^{(t)}$, the solver will pick out the true correspondence between the rest fields so that the total possibilities of picked pairs are maximized without violating the constraints in the equation (6). This picking process is a linear programming problem that can be solved efficiently by the Hungarian algorithm again. After picking, the continuous $\hat{P}^{(t)}$ becomes binary again and was output as the predicted graph correspondence. More details on solving the PGM problem can be found in the literature in [43]. Inspired by [44], we used a third-party library called DD-ILP [45], to solve the graph matching problem without considering the outliers. The ZAC-GM solver in [43] is implemented to handle the outliers.



(a) Field drift problem

(b) Outlier problem

Fig. 6. Drift field and outliers lead to many-to-one mappings. In both (a) and (b), a support field is mapped to two query fields as indicated by the red and green lines. Red lines indicate wrong mappings. Yellow lines in (a) indicate that query fields drift downwards. In the query of (b), the purple boxes contain an outlier that reads ④. Please zoom in to see the text in static fields.

Importance of the Mapping Constraints: The constraints described in the equation (6) can relieve the field drift and outlier problems. In Fig. 6 (a), the support field "¥0.00" is very spatially similar to the query field "¥51.00". Please observe their spatial relationships to the static field "Fuel oil surcharge" to check this claim. Therefore, the spatial MLP will assign a high similarity score to them which will lead to a correspondence between them later. However, if the mapping constraint is applied, the wrong mapping (the red line) between "¥0.00" and "¥51.00" will be replaced by the correct mapping between "¥0.00" and "¥0.00" (the green line).

Training objective: To improve the performance of our framework, this paper integrates the graph similarity calculation module and the correspondence estimation module into an end-to-end trainable framework. Denote the ground truth correspondence matrix as P^*, whose elements are binary. The hamming loss between the predicted correspondence matrix $\hat{P}$ and P^* is:

$$\ell_H = \frac{1}{|F_s| * |F_q|} \sum_{i=1}^{|F_s|}\sum_{a=1}^{|F_q|} XOR\left(\hat{P}_{ia}, P_{ia}^*\right), \tag{9}$$

where $XOR(\cdot,\cdot)$ represents the "exclusive or" operation. What's more, inspired by [43], another auxiliary loss, called

TABLE I

Labels, Types, and the Number of Images for Each Dataset

Dataset	# Images	# Public/New Dataset	#Types
CORD	Train 800, Test 100	Public Dataset	Receipts
SROIE	Train 626, Test 276	Public Dataset	Receipts
DKIE	Train 2000, Test 500	New Dataset	Forms

ranking loss, was designed to enlarge the similarity score difference between correct and wrong node pairs during training:

$$\ell_R = \sum_{i,a} \sum_{j \neq i, b=a} \sum_{j=i, b \neq a} \frac{P^*_{ia} * \min\{\epsilon^+, \beta * (A^a_i - A^b_j)\}}{|F_s| * |F_q|}, \tag{10}$$

where A^a_i represents the node similarity score of the correct field pair. β is 1 when $A^a_i > A^b_j$, -1 otherwise.

Let ρ denotes the weight of ℓ_R. We minimize the this loss:

$$\ell = \ell_H + \rho * \ell_R. \tag{11}$$

IV. Experiments

A. Datasets

Table I lists three KIE benchmark datasets: CORD [46], SROIE [22] and DKIE. The other public datasets, such as FUNSD [47] and EPHOIE [48] are not suitable for the one-shot KIE task because many testing samples do not have corresponding support documents. Moreover, we create a new dataset, DKIE, to promote the research looking into the one-shot KIE task, especially with regard to the problems of drifted fields and outliers.

B. Implementation Details

Training Details: We compared our model with 8 different KIE models. Five of them are supervised-learning-based models [10], [13], [14], [15], [37] and three of them are one-shot-learning based models [21], [49], [50]. The backbones of all one-shot-learning-based models are multilayer perceptrons, and the number of parameters is 2k for all backbones to ensure a fair comparison between the proposed method and the other methods. We reimplemented these models because their original codes were not available. The backbones of all supervised-learning-based models are transformers. We cite their performance on the public datasets and the number of parameters from the literature.

Testing Details: In the literature on supervised-learning-based methods, precision, recall, and F1 score were adopted while the labeling accuracy was reported in the literature on the one-shot-learning-based method. Therefore, all of them are reported on public datasets. Labeling accuracy is reported on the DKIE dataset.

C. Performance on the Public Dataset

TABLE II

Comparison With Supervised-Learning-Based Methods on the CORD Datasets. Precision, Recall, and F1 Scores Are Reported

Methods	Precision	Recall	F1	Params
Supervised-learning-based. Performance cited from the literature.				
BERT [37]	88.33	91.07	89.68	-
LayoutLM-Base [13]	94.37	95.08	94.72	113M
LayoutLM-Large [13]	94.32	95.54	94.93	343M
LayoutLMv2-Base [14]	94.53	95.39	94.95	200M
LayoutLMv2-Large [14]	95.65	96.37	96.01	426M
One-shot-learning-based. Reimplemented code tested on the CORD data.				
MatchNet [49]	62.74	58.79	60.70	2k
ProtoNet [50]	62.61	59.07	60.80	2k
LF-BP [21]	88.11	84.32	86.15	2k
Ours (DD-ILP)	91.96	92.8	92.38	2k
Ours (ZAC-GM)	93.89	94.07	93.98	2k

TABLE III

Comparison With Supervised-Learning-Based Methods on the SROIE Datasets. Precision, Recall and F1 Score Are Reported

Methods	Precision	Recall	F1	Params
PICK [10]	96.79	95.46	96.12	-
LayoutLM-Base [13]	94.38	94.38	94.38	113M
LayoutLM-Large [13]	95.24	95.24	95.24	343M
LayoutLMv2-Base [14]	96.25	96.25	96.25	200M
LayoutLMv2-Large [14]	**99.04**	96.61	97.81	426M
StrucTexT [15]	95.84	98.52	96.88	107M
Ours (DD-ILP)	97.37	98.6	97.98	2k
Ours (ZAC-GM)	98.96	**98.68**	**98.82**	2k

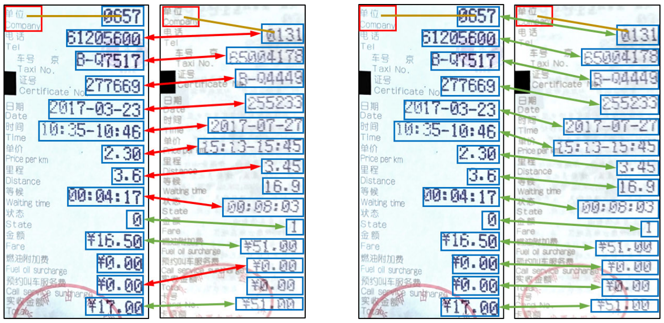
TABLE IV

Comparison With One-Shot Learning Methods on the SROIE Datasets. Entity-Level Labelling Accuracy Is Reported

Types	MatchNet [49]	ProtoNet [50]	LF-BP [21]	Ours (DD-ILP)	Ours (ZAC-GM)
T0	80.4	81.1	97.3	**100**	**100**
T1	65.4	69.1	70.6	**100**	**100**
T2	89.0	89.0	90.7	**100**	**100**
T3	86.6	87.5	91.7	**100**	**100**
T4	72.2	74.0	84.4	**95.9**	**95.9**
T5	84.0	84.4	**99.2**	**99.2**	**99.2**
T6	71.7	74.0	74.0	**96.6**	**96.6**
T7	96.8	97.9	97.9	97.9	**98.9**
Average	78.5	80.2	88.2	98.5	**98.7**

Table II summarizes the results of the CORD dataset. Our proposed method (referred to as “Ours (ZAC-GM)”) performed well, with a precision of 93.89, recall of 94.07, and an F1-score of 93.98, all comparable to the LayoutLM-based models. The LayoutLM-Based model outperforms our method due to the nature of the CORD dataset, where each image typically contains only 1 to 3 static text fields. As a result, one-shot learning-based methods struggle to effectively distinguish the relative positions of different dynamic text fields with respect to static text fields. This limitation leads to significant errors in our method’s performance. In ablation experiments, we observed a considerable decline in model performance when the number of static text fields was low.

In Table III, our model achieved comparable results against supervised-learning-based methods. Our model size is significantly smaller than the supervised-learning-based methods. Therefore, our model can be deployed to memory-limited devices, such as mobile phones, more easily. Notice that our model relies on the existence of support documents so that the labels can be transferred from support to query documents



(a) LF-BP

(b) Ours

Fig. 7. LF-BP (left) and our model's (right) predictions of documents containing drifted fields in the T0 dataset. In both (a) and (b), the support document is on the left and the query document is on the right. The yellow line in the query document is slop indicating downwards drifted text fields. Red arrows are the wrong correspondences while green ones are correct.

while the supervised-learning-based methods do not have such requirements. This is a trade-off between model size and the existence of support documents.

In Table IV, our model achieved the best performance in all types of documents when compared with the other one-shot learning-based methods. What's more, our model converged after 5 iterations of training data while the rest models converged after 12 iterations. There are two reasons that explain why our model converged more quickly. First, our model uses hamming loss to calculate the gradients, while the rest models use cross-entropy loss. Second, the combinatorial solvers in our model are not sensitive to the subtle change of affinity matrices, which are the outputs of MLP modules. Therefore, when the affinity matrices are approximately correct, combinatorial solvers can find the correct mapping between support and query fields.

D. Performance on the DKIE Dataset

In Table V, our model outperformed the other one-shot learning-based methods on all test types. Despite that the supervised-learning-based models consumed more features (spatial+visual+text versus only spatial), our model achieved the best results on all test types but the T0, T2, and T7 types. When our model consumes more features, the performance of our model on T0 and T2 types will increase in the Ablation Study section IV-E.

To investigate the performance of our model on samples containing drifted fields and outliers separately, we further split each type of document into 2 parts. There are "drifted" documents in which some fields have drifted so badly that even humans need to check each field very carefully to judge the labels. There are also documents containing "outliers". A small number of documents contain both drifted fields and outliers. They are included in "drifted" and "outliers" parts at the same time.

1) Performance on "Drifted" Documents: In Table VI, the performance of our model dropped moderately while the rest models failed on the "drifted" data. Our model significantly outperformed the rest models across all types. There are no "drifted" documents in T1, T4 and T7 types. Take the T0 type for example, we found that the fields in this type are arranged vertically. If one of the fields in the head part of a document drifted downwards, all the fields below it would then also drift downwards. Typical samples of the T0 type can be found in Figure 2 (b) and Figure 7. Both the online demo[3] released by [21] and our reimplemented LF-BP model achieve low accuracy on the drifted fields. Typical mistakes made by the LF-BP model are shown in Figure 7.

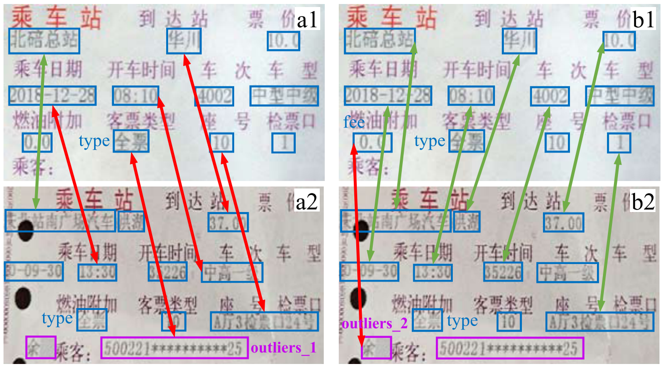


Fig. 8. Documents containing horizontally drifted fields and outliers in the d3 dataset. (a1) and (a2) show the prediction of the LF-BP model. (b1) and (b2) show the prediction of our model with the ZAC-GM solver. Purple boxes are outliers. Red lines indicate the wrong mapping between fields. Both models failed in this example.

2) Performance on "Outliers" Documents: Table VII shows that our model, when using the ZAC-GM solver, is the only one that succeeds across all datasets. When our model uses the DD-ILP solver, it cannot handle documents that contain outliers. We found that DD-ILP aims to solve the graph matching problem and requires the support and query documents to have the same number of fields. This is not true in documents that contain outliers. However, the ZAC-GM solver [43] is reimplemented and employed to pick out the outliers, our model can handle the drifted fields and outliers at the same time to some extent.

Some documents in the T0, T3, and T6 datasets contain drifted fields and outliers at the same time. Not only did the rest models fail on these documents, but also the performance of our model dropped by a relatively large margin. Figure 8 shows such samples in the T3 type. When the outliers are close to the drifted fields, they are hard to distinguish from each other solely based on their spatial features. For example, the LF-BP model maps the "type" field in (a1) to the "outliers_2" field in (a2). Our model also maps the "fee" field in (b1) to the "outliers_1" field in (b2). The positions of these outliers are so close to other drifted query fields that the models may confuse them with the situation of multi-region fields. This indicates that the similarity between fields should be measured using more diverse features such as the width and height of bounding boxes of fields or the text embedding in fields.

[1] LayoutLMv2-Large model can not handle chinese characters in DKIE.

[2] StrucTexT-base model performs poorly due to insufficient training data.

[3] https://ocr.data.aliyun.com/experience#/?first_tab=general

TABLE V

Comparison With Supervised-Learning-Based and One-Shot Learning-Based Methods on the DKIE Datasets. Entity-Level Labelling Accuracy Is Reported. From T0 to T8, Each One Represents a Type of Document

Methods	Input features	T0	T1	T2	T3	T4	T5	T6	T7	T8
Supervised-learning-based. Original code tested on the DKIE dataset.										
PICK [10]	spatial+visual+text	97.9	97.8	98.3	95.8	95.7	96.6	92.3	94.9	92.2
LayoutLM-Base [13]		97.3	97.0	98.1	94.6	95.0	96.1	91.8	94.3	92.5
LayoutLM-Large [13]		97.7	97.5	98.4	94.8	95.6	96.3	92.1	94.7	92.8
LayoutLMv2-Base [14][1]		**98.4**	98.6	**99.2**	95.3	98.8	99.2	94.1	**97.8**	96.2
StrucTexT-base [15][2]		95.7	96.3	97.3	93.9	93.5	95.1	92.3	92.6	85.4
One-shot-learning based. Reimplemented code tested on the DKIE dataset.										
MatchNet [49]	only spatial	78.9	97.3	83.2	94.1	91.9	96.1	86.8	95.5	92.8
ProtoNet [50]		78.6	97.5	83.3	94.1	92.1	96.4	86.7	95.8	93.1
LF-BP [21]		80.4	98.2	84.1	94.4	92.5	97.3	87.2	95.8	93.8
Ours (DD-ILP)		93.6	98.5	84.7	96.0	97.1	98.4	94.4	96.1	97.2
Ours (ZAC-GM)		95.7	**99**	85.2	**98.5**	**99.5**	**100**	**96.4**	97.2	**98**

TABLE VI

Comparison With One-Shot Learning Methods on the "drifted" DKIE Datasets. Entity-Level Labelling Accuracy Is Reported

Types	MatchNet [49]	ProtoNet [50]	LF-BP [21]	Ours (DD-ILP)	Ours (ZAC-GM)
T0	56.2	58.1	60.0	**97.0**	96.3
T2	66.7	67.0	68.6	**71.1**	**71.1**
T3	39.7	41.3	42.4	90.0	**93.9**
T5	79.1	79.3	80.9	**100**	**100**
T6	72.8	74.3	75.6	**96.0**	**96.0**
T8	63.7	62.5	65.2	**96.2**	**96.2**

TABLE VII

Comparison With One-Shot Learning Methods on the "outliers" DKIE Datasets. Entity-Level Labelling Accuracy Is Reported

Types	MatchNet [49]	ProtoNet [50]	LF-BP [21]	Ours (DD-ILP)	Ours (ZAC-GM)
T0	61.9	62.0	64.4	89.0	**91.2**
T1	95.3	95.8	97.9	97.2	**98.7**
T2	89.4	89.2	**90.0**	85.2	**90.0**
T3	80.0	79.2	81.3	79.1	**100**
T4	90.3	91.2	93.1	86.3	**98.0**
T5	94.5	95.2	97.0	96.2	**100**
T6	68.2	68.3	70.0	88.0	**95.8**
T7	89.4	90.7	91.2	91.3	**96.0**
T8	88.5	88.9	90.0	95.8	**97.0**

TABLE VIII

Testing the Impact of Different Features Using the T0 Type. Each Column Corresponds to One Type of Taxi Receipt in One Province. "AVG" Means Average Accuracy of All Documents

Different Features	SC	BJ	AH	JS	CQ	AVG
Spatial	**98.8**	91.2	81.8	92.8	**99.1**	93.6
Spatial+Aspect	**98.8**	85.3	94.4	94.0	97.2	93.2
Spatial+Edge	97.2	87.9	95.8	91.6	**99.1**	93.2
Aspect+Edge	91.6	82.2	74.1	**98.2**	90.0	87.1
Spa+Asp+Edg	98.2	88.5	**97.2**	94.0	**99.1**	94.2
Sp+As+Ed+Text	96.9	**93.3**	95.8	94.6	**99.1**	**95.1**

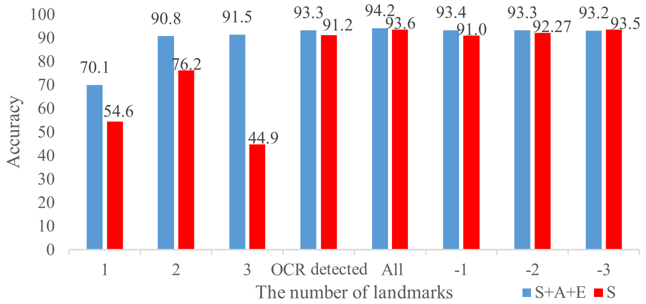


Fig. 9. Labelling accuracy versus the number of static text fields. "S", "A", and "E" represent the spatial, aspect, and edge features.

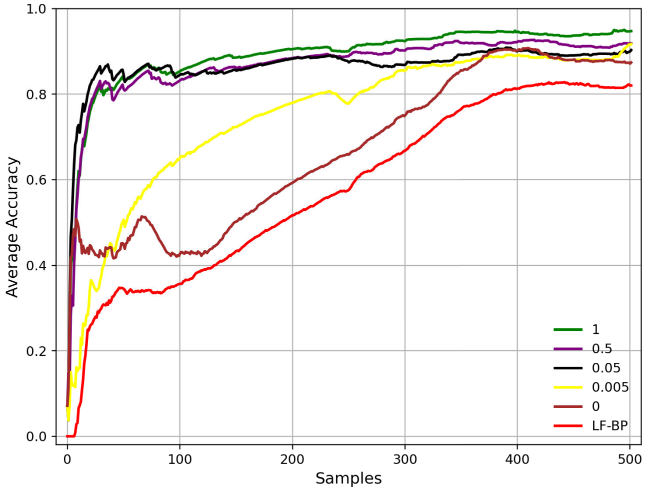


Fig. 10. The accuracy of our models (ZAC-GM) that are trained with different ranking loss weights. Better viewed in color.

E. Ablation Study

1) Different Features: Additional MLP modules are designed to incorporate more diverse features and the benefits of this practice are tested across different types of documents in Table VIII.

The first line of Table VIII shows that our model achieves good performance solely based on the spatial features on most types of documents except for the "AH" type. The last line of Table VIII shows that if all possible features ("Spatial+Aspect+Text+Edge") are used, the accuracy of our model on all types reached 90%. We believe the proposed four features are complementary to each other.

2) Static Text Fields: The impacts of the number of static text fields on the accuracy of our model are further evaluated in Figure 9. The overall accuracy is good when less than three static text fields (see −1, −2, −3 in the x-axis) are dropped. When multiple features are used, the labelling accuracy grows as the number of static text fields increases. This is not true when only the spatial features are used. This also proves that these features are complementary to each other.

3) Ranking Loss: The effectiveness of ranking loss is also evaluated by changing the weight of ranking loss. Figure 10 shows that our ranking loss can help to accelerate the training process. When our model does not employ the ranking loss, it can still outperform the LF-BP model. By increasing the weight of ranking loss, our model converged much more quickly and the accuracy also increased. When the weight of the ranking loss is too large, the performance of our model drops. When the ranking loss is applied to two solvers, their accuracy improves 1% across different testing styles in the T0 type of documents.

V. Conclusion

This paper proposed a robust deep Partial Graph Matching (dPGM) framework for the one-shot KIE task. The proposed framework represents each document as a graph. It solves the text field labelling problem by estimating the correspondence between text fields in a test document and those in a support document. It enables the learning of graph similarity and correspondence estimation in an end-to-end trainable framework. To get a strict one-to-one correspondence, a combinatorial solver module with an extra one-to-(at most)-one mapping constraint is introduced to do exact graph matching, which results in the robustness of the field drift problem and the outlier problem. To promote the research of the KIE task, a large one-shot KIE dataset called DKIE is collected and annotated, which contains 2,500 document images and diverse types of document images. Extensive experiments on the public dataset and the new DKIE dataset demonstrate the effectiveness and robustness of the proposed method.

References

[1] P. Dai, Y. Li, H. Zhang, J. Li, and X. Cao, "Accurate scene text detection via scale-aware data augmentation and shape similarity constraint," *IEEE Trans. Multimedia*, vol. 24, pp. 1883–1895, 2022.

[2] M. Zhao, W. Feng, F. Yin, X.-Y. Zhang, and C.-L. Liu, "Mixed-supervised scene text detection with expectation-maximization algorithm," *IEEE Trans. Image Process.*, vol. 31, pp. 5513–5528, 2022.

[3] Z. Wan, M. He, H. Chen, X. Bai, and C. Yao, "TextScanner: Reading characters in order for robust scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 7, 2020, pp. 12120–12127.

[4] A. R. Katti et al., "Chargrid: Towards understanding 2D documents," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4459–4469.

[5] T. I. Denk and C. Reisswig, "BERTgrid: Contextualized embedding for 2D document representation and understanding," 2019, *arXiv:1909.04948*.

[6] Y. Qian, E. Santus, Z. Jin, J. Guo, and R. Barzilay, "GraphIE: A graph-based framework for information extraction," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, vol. 1. Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 751–761. [Online]. Available: https://www.aclweb.org/anthology/N19-1082

[7] X. Liu, F. Gao, Q. Zhang, and H. Zhao, "Graph convolution for multimodal information extraction from visually rich documents," in *Proc. Conf. North*, 2019, pp. 32–39.

[8] M. Wei, Y. He, and Q. Zhang, "Robust layout-aware IE for visually rich documents with pre-trained language models," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 2367–2376.

[9] B. P. Majumder, N. Potti, S. Tata, J. B. Wendt, Q. Zhao, and M. Najork, "Representation learning for information extraction from form-like documents," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 6495–6504.

[10] W. Yu, N. Lu, X. Qi, P. Gong, and R. Xiao, "PICK: Processing key information extraction from documents using improved graph learning-convolutional networks," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4363–4370.

[11] P. Zhang et al., "TRIE: End-to-end text reading and information extraction for document understanding," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1413–1422.

[12] Z. Cheng et al., "TRIE++: Towards end-to-end information extraction from visually rich documents," 2022, *arXiv:2207.06744*.

[13] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "LayoutLM: Pre-training of text and layout for document image understanding," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1192–1200.

[14] Y. Xu et al., "LayoutLMv2: Multi-modal pre-training for visually-rich document understanding," 2020, *arXiv:2012.14740*.

[15] Y. Li et al., "StrucTexT: Structured text understanding with multi-modal transformers," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 1912–1920.

[16] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, and R. Manmatha, "DocFormer: End-to-end transformer for document understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 993–1003.

[17] P. Li et al., "SelfDoc: Self-supervised document representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5652–5660.

[18] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei, "LayoutLMv3: Pre-training for document AI with unified text and image masking," 2022, *arXiv:2204.08387*.

[19] T. Hong, D. Kim, M. Ji, W. Hwang, D. Nam, and S. Park, "BROS: A pre-trained language model focusing on text and layout for better key information extraction from documents," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 10, 2022, pp. 10767–10775.

[20] G. Tang et al., "MatchVIE: Exploiting match relevancy between entities for visual information extraction," 2021, *arXiv:2106.12940*.

[21] M. Cheng, M. Qiu, X. Shi, J. Huang, and W. Lin, "One-shot text field labeling using attention and belief propagation for structure information extraction," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 340–348.

[22] Z. Huang et al., "ICDAR2019 competition on scanned receipt OCR and information extraction," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1516–1520.

[23] L. Chiticariu, Y. Li, and F. Reiss, "Rule-based information extraction is dead! long live rule-based information extraction systems!" in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 827–832.

[24] D. Schuster et al., "Intellix—End-user trained information extraction for document archiving," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 101–105.

[25] V. Sunder, A. Srinivasan, L. Vig, G. Shroff, and R. Rahul, "One-shot information extraction from document images using neuro-deductive program synthesis," 2019, *arXiv:1906.02427*.

[26] N. Or and S. Urbach, "Few-shot learning for structured information extraction from form-like documents using a diff algorithm," in *Proc. Document Intell. Workshop KDD*, 2021.

[27] S. Tata, N. Potti, J. B. Wendt, L. B. Costa, M. Najork, and B. Gunel, "Glean: Structured extractions from templatic documents," in *Proc. VLDB Endowment*, 2021, pp. 997–1005.

[28] M. Rusiñol, T. Benkhelfallah, and V. P. dAndecy, "Field extraction from administrative documents by incremental structural templates," in *Proc. 12th Int. Conf. Document Anal. Recognit.*, Aug. 2013, pp. 1100–1104.

[29] E. Medvet, A. Bartoli, and G. Davanzo, "A probabilistic approach to printed document understanding," *Int. J. Document Anal. Recognit. (IJDAR)*, vol. 14, no. 4, pp. 335–347, 2011.

[30] V. P. d'Andecy, E. Hartmann, and M. Rusiñol, "Field extraction by hybrid incremental and *a-priori* structural templates," in *Proc. 13th IAPR Int. Workshop Document Anal. Syst. (DAS)*, Apr. 2018, pp. 251–256.

[31] R. B. Palm, F. Laws, and O. Winther, "Attend, copy, parse end-to-end information extraction from documents," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 329–336.

[32] R. B. Palm, O. Winther, and F. Laws, "CloudScan–A configuration-free invoice analysis system using recurrent neural networks," in *Proc. 14th IAPR Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Nov. 2017, pp. 406–413.

[33] C. Sage, A. Aussem, H. Elghazel, V. Eglin, and J. Espinas, "Recurrent neural network approach for table field extraction in Bus. Documents," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 1308–1313.

[34] H. Guo, X. Qin, J. Liu, J. Han, J. Liu, and E. Ding, "EATEN: Entity-aware attention for single shot visual text extraction," in *Proc. Int. Conf. Document Anal. Recognit. (ICDAR)*, Sep. 2019, pp. 254–259.

[35] J. Wang et al., "Tag, copy or predict: A unified weakly-supervised learning framework for visual information extraction using sequences," 2021, *arXiv:2106.10681*.

[36] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," 2015, *arXiv:1508.01991*.

[37] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.

[38] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 2978–2988. [Online]. Available: https://www.aclweb.org/anthology/P19-1285

[39] R. E. Burkard, E. Cela, P. M. Pardalos, and L. S. Pitsoulis, "The quadratic assignment problem," in *Handbook of Combinatorial Optimization*. Cham, Switzerland: Springer, 1998, pp. 1713–1809.

[40] S. Lacoste-Julien and M. Jaggi, "On the global linear convergence of frank-wolfe optimization variants," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 496–504.

[41] A. A. Goldstein, "On steepest descent," *J. Soc. Ind. Appl. Math., Ser. A, Control*, vol. 3, no. 1, pp. 147–151, 1965.

[42] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics (NRL)*, vol. 52, no. 1, pp. 7–21, Feb. 2005.

[43] F. Wang, N. Xue, J.-G. Yu, and G.-S. Xia, "Zero-assignment constraint for graph matching with outliers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3030–3039.

[44] M. Vlastelica, A. Paulus, V. Musil, G. Martius, and M. Rolínek, "Differentiation of blackbox combinatorial solvers," 2019, *arXiv:1912.02175*.

[45] P. Swoboda, J. Kuske, and B. Savchynskyy, "A dual ascent framework for Lagrangean decomposition of combinatorial problems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4950–4960.

[46] J. Wang et al., "Towards robust visual information extraction in real world: New dataset and novel solution," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, 2021, pp. 2738–2745.

[47] G. Jaume, H. K. Ekenel, and J.-P. Thiran, "FUNSD: A dataset for form understanding in noisy scanned documents," in *Proc. Int. Conf. Document Anal. Recognit. Workshops (ICDARW)*, vol. 2, Sep. 2019, pp. 1–6.

[48] S. Park et al., "CORD: A consolidated receipt dataset for post-OCR parsing," in *Proc. Workshop Document Intell. (NeurIPS)*, 2019.

[49] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg, "MatchNet: Unifying feature and metric learning for patch-based matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3279–3286.

[50] J. Snell and etc., "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4080–4090.

Minghong Yao received the B.E. degree in thermal power dynamic engineering from Hohai University in 2018. He is currently pursuing the Ph.D. degree with the School of Cyber Science and Technology, University of Science and Technology of China. His research interests include natural language processing and deep learning. His current research work is focused on document understanding.

Zhiguang Liu received the B.Sc. and M.Sc. degrees from the Harbin University of Science and Technology, Harbin, China, in 2009 and 2012, respectively, and the Ph.D. degree from the 3D Motion Capture Laboratory, Department of Computer Science, City University of Hong Kong, in 2016. He was a Postdoctoral Researcher at INRIA, France. He is currently a Senior Researcher with the Noah's Ark Laboratory, Huawei Technologies. His current research interests include multimodal document understanding, scene text recognition, and layout generation.

Liansheng Zhuang (Member, IEEE) received the bachelor's and Ph.D. degrees from the University of Science and Technology of China (USTC), Hefei, China, in 2001 and 2006, respectively. He was a Vendor Researcher with the Visual Computing Group, Microsoft Research, Beijing, China. From 2012 to 2013, he was a Visiting Research Scientist with the Department of EECS, University of California at Berkeley, Berkeley, CA, USA. He is currently an Associate Professor with the School of Information Science and Technology, USTC. In 2011, he was nominated to join the STARTRACKER Project of Microsoft Research of Asia (MSRA). His main research interests include computer vision and machine learning. He is a member of ACM and CCF.

Liangwei Wang received the M.Phil. degree from the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, in 2001. He is currently a Technician Expert at the Noah's Ark Laboratory, Huawei Technologies. He is also leading the Industry Vision Research Group. His research interests include text recognition, document analysis, industry vision inspection, and robotic vision. He is a reviewer of the Huawei Computer Vision Technology Committee.

Houqiang Li (Fellow, IEEE) received the B.S., M.Eng., and Ph.D. degrees in electronic engineering from the University of Science and Technology of China (USTC), Hefei, China, in 1992, 1997, and 2000, respectively.

He is currently a Professor with the Department of Electronic Engineering and Information Science, USTC. He has authored and coauthored over 200 papers in journals and conferences. His research interests include multimedia search, image/video analysis, and video coding and communication. He is the Winner of the National Science Fund (NSFC) for Distinguished Young Scientists, the Distinguished Professor of the Changjiang Scholars Program of China, and the Leading Scientist of the Ten Thousand Talent Program of China. He was a recipient of the Best Paper Awards from VCIP 2012, ICIMCS 2012, and ACM MUM 2011, the National Natural Science Award of China (Second Class) in 2015, and the National Technological Invention Award of China (Second Class) in 2019. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2010 to 2013, the TPC Co-Chair for VCIP 2010, and the General Co-Chair for ICME 2021.