

One-shot Text Field Labeling using Attention and Belief Propagation for Structure Information Extraction

Mengli Cheng, Minghui Qiu*, Xing Shi, Jun Huang, and Wei Lin

Alibaba Group

{mengli.cml,minghui.qmh,xing.shix,huangjun.hj,weilin.lw}@alibaba-inc.com

ABSTRACT

Structured information extraction from document images usually consists of three steps: text detection, text recognition, and text field labeling. While text detection and text recognition have been heavily studied and improved a lot in literature, *text field labeling* is less explored and still faces many challenges. Existing learning based methods for text labeling task usually require a large amount of labeled examples to train a specific model for each type of document. However, collecting large amounts of document images and labeling them is difficult and sometimes impossible due to privacy issues. Deploying separate models for each type of document also consumes a lot of resources. Facing these challenges, we explore one-shot learning for the text field labeling task. Existing one-shot learning methods for the task are mostly rule-based and have difficulty in labeling fields in crowded regions with few landmarks and fields consisting of multiple separate text regions. To alleviate these problems, we proposed a novel deep end-to-end trainable approach for one-shot text field labeling, which makes use of *attention* mechanism to transfer the layout information between document images. We further applied *conditional random field* on the transferred layout information for the refinement of field labeling. We collected and annotated a real-world one-shot field labeling dataset with a large variety of document types and conducted extensive experiments to examine the effectiveness of the proposed model. To stimulate research in this direction, the collected dataset and the one-shot model will be released¹.

CCS CONCEPTS

• **Applied computing** → **Document analysis**; *Optical character recognition*; • **Computing methodologies** → **Visual content-based indexing and retrieval**.

KEYWORDS

Text Field Labeling, Structure Information Extraction

*The corresponding author.

¹https://github.com/AlibabaPAI/one_shot_text_labeling

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '20, October 12–16, 2020, Seattle, WA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7988-5/20/10...\$15.00

<https://doi.org/10.1145/3394171.3413511>

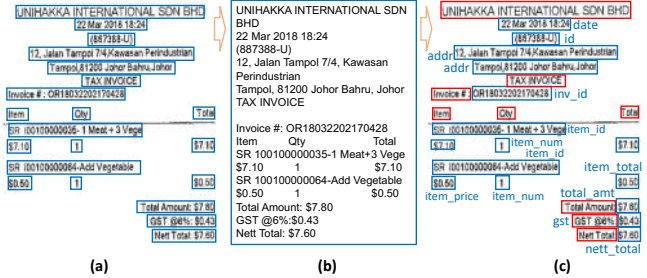


Figure 1: Structured information extraction workflow: (a) text detection, (b) text recognition, (c) text field labeling (red rectangles are static fields, while blue are dynamic fields.)

ACM Reference Format:

Mengli Cheng, Minghui Qiu, and et al. 2020. One-shot Text Field Labeling using Attention and Belief Propagation for Structure Information Extraction. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20)*, October 12–16, 2020, Seattle, WA, USA. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3394171.3413511>

1 INTRODUCTION

Structured information extraction, a task to extract structured information from document images, is widely used in many industry applications, such as online user authentication and automatic filling of forms. Fig.1 shows a sample workflow of structured information extraction from a receipt image. Text detection, text recognition, and text field labeling are the three key steps for structured information extraction[17, 23]. Text detection performance has been improved a lot due to the high performance of CNN[32]. Text recognition also achieves great progress using CTC model[20] or Seq2Seq model with attention[8]. With sufficient training data, it is possible to build a good text detection and recognition model for text with multiple languages in various scenes. However, the text field labeling problem (Fig.1c) is still less studied. The task aims to assign a label to each text region in a document so that text information could be extracted in structured formats.

Learning based methods [2, 10, 14, 22, 30] are shown to have good performance for text field labeling. They could automatically adapt to any type of layouts, but they usually require sufficient training data. As labeled examples are often hard to obtain, to alleviate this problem, we explored one-shot learning for the text field labeling task. Typical one-shot field labeling methods are rule based [3, 17], which are not robust enough [3] and have difficulty in extract field types consist of multiple text regions. Meanwhile, most existing one-shot methods are designed for image classification [16, 19]. Direct application of these methods on field labeling does not perform well. There are three reasons: first, text field labeling requires more

fine-grained image features which is known to be challenging[27]; second, text fields can be with high intra-type variance (e.g. addr in Fig.1c), third, text fields could have small inter-type variance (e.g. item_total and total_amt in Fig.1c).

To alleviate the problem, we proposed an attention based deep learning method for one-shot field labeling. Our method uses a landmark to field attention approach to transfer the relationship between landmarks and fields from support document to test document. To label fields in crowded regions lack of landmarks and to label fields consisting of multiple text regions, we proposed attention based belief propagation to make use of spatial relationships between text fields to refine field labeling.

Below we summarized our contributions.

- We are the first to propose a deep end-to-end trainable approach for one-shot text field labeling problem using landmarks, attention, and belief propagation.
- Due to the lack of one-shot datasets, we created a dataset consisting of 43 types of documents for training and testing, and also adapted SROIE [13] to generate a second dataset. These two datasets will be released to stimulate research in this direction.
- Extensive experiments show the effectiveness of the proposed method and the usefulness of different components. Results show our method has better performance than all the competitive baseline methods.

2 RELATED WORK

Our work is related to text field labeling and few-shot learning, and we summarized related works in these fields in the following.

2.1 Text field labeling.

Existing methods for the task are either learning- or rule- based. **Learning-based methods.** Learning-based methods treat field labeling as an image labeling problem. Early methods apply markov random fields[7] or conditional random fields[2] to solve the problem. More recent approaches tend to use CNN[14, 22, 30], GNN[10], and Bert[28] to improve performance. Despite their good performance, these methods require large amounts of training data which is hard to collect due to privacy reasons. Besides, these methods have to deploy a specific model for each type of document, which is labor-intensive and difficult to maintain.

To reduce the requirement of large amounts of labeled data, one-shot text field labeling problem is studied. For the one-shot text field labeling problem, it is assumed the query document and support respect to the same Manhattan layout. By same layout, it means that the spatial relationships between different types of fields keep unchanged for one-type of document, such as item_price is always on the right of the item_name, en_name is always under cn_name. But there are no constraints on the horizontal distance between item_name and item_price or vertical distance between en_name and cn_name. It also does not require the document boundaries of support and query image to be well aligned. Existing one-shot or few-shot text field labeling methods are generally rule-based.

Rule-based one-shot methods. Medvet[12] proposed a probabilistic text sequence extraction model, which assumes simple distributions and could only extract sequences of fixed length. Recent

methods tend to represent documents as attributed graphs and model it as a graph matching problem. The document graph described in [15] is actually a forest as each text region has at most one parent. A set of fields with fixed content is used to normalize the coordinates of query document and to establish correspondence between support and query document. Multiple field attributes and parent-child relationships between fields have to be specified, which is difficult for end users. Match against the forest is a top-down traverse of the forest. The match process is sensitive to missing fields: mismatch of a parent field has large impacts on all its descents.

Rusinol[17] introduced star graph, where each field is connected to all other fields. The spatial relationship between fields are modeled by polar coordinates. A weighted voting approach is used to determine the location of center field, where the weight is calculated by tf-idf stats[18]. Their approach eliminated the effort of specifying fields attributes and parent-child relationships. But the tf-idf stats are not robust enough because the satellite nodes are not pruned.

In addition, their voting approach is sensitive to vertical displacements. The study in [3] alleviated the problems by referring to static fields(keywords) in the same line. But inline keywords are not available for many of the fields: item_name and item_price have no inline keywords. The performance degradation problem[17] for fields in crowded regions with few static neighbors was still not solved. Hammami[5] proposed a subgraph isomorphism based method for field spotting, but their method could only work on colored forms. Existing methods generally use handcraft features and heuristics for field labeling, which are suboptimal. Moreover, all these methods have difficulty in extracting **multi-region fields** such as item_id and item_total(Fig.1c), which both falls into 2 separate text regions.

2.2 Few-shot learning

Recent progress on few-shot learning is generally based on meta-learning[19]. The meta-learning based methods can be grouped into three categories: metric learning, recurrent models and learning for fine-tuning. **Metric learning.** Koch[6] proposed to use siamese networks for few-shot image recognition. Prototypical networks[21] learn a metric space in which classes could be represented by their centers. Relation network[24] improves prototypical networks by learning a nonlinear similarity function. **Recurrent models.** Santoro[19] proposed a LRU based memory augmented network for few shot learning. Finn[4] improved by introducing attention mechanism into memory augmented network. While the augmented memory could generate context-aware encoding of images, the underlying LSTM network is not efficient and suffers from the order problem[26]. **Learning for fine-tuning.** MAML[4] proposed to meta-learn good initializations of network parameters for fine-tuning on target dataset. Ravi[16] improved by introducing LSTM based controller to optimize the gradient descent parameters. Despite good performance, the need to fine-tune increases deployment complexity and response time.

Existing one-shot methods generally work on image classification problem. Direct application of the methods to layout labeling leads to bad performance, which is due to three reasons: firstly, labeling text contents is a fine-grained classification problem, which

is challenging for existing few-shot methods[27]. Secondly, text contents could have high intra-class variance, such as `addr`(Fig.1c). Thirdly, text contents could also be similar across class, such as `item_total` and `total_amt`(Fig.1c), thus have small inter-class variance. In contrast, the spatial relationship between text fields is less variant for each type of documents. In this work, we proposed attention based approach to transfer the spatial relationship from support document to query document, which is then used for text field labeling. Our approach is end-to-end trainable and could model complex spatial relationship which is caused by multi-region fields.

3 METHODOLOGY

Like existing one-shot labeling methods, we represented documents as graphs. Let L to be the set of static text regions (called as **Landmark** later), F to be the set of dynamic text regions (called as **Field** later), $LF = \{(u, v) : u \in L \text{ and } v \in F\}$, $FF = \{(u, v) : u \in F \text{ and } v \in F\}$, Y be the labels of F , we can represent the document as a directed graph $G = \{L, F, LF, FF, Y\}$. We differentiated the two sets of text regions because the static ones could be located easily by keywords matching and be used as correspondence. Let G_s and G_q be the graphs defined on support document and query document, the one-shot text labeling problem is to determine labels Y_q of G_q given labels Y_s of G_s . Note that the text field labeling problem could not be directly solved by graph matching[9, 31] due to the existence of many to many matches of multi-region fields.

Fig.2 describes the overall picture of our model: first, attention is conducted on landmark to field edges(LFAttn) to transfer the spatial relationship between landmarks and fields from support document to query document and generate probabilistic distributions of fields; second, attention is conducted on field to field edges(FFAttn) to transfer the spatial relationship between fields and generate probabilistic distributions of field pairs; finally, belief propagation is applied on the two distributions to generate the final distributions of fields.

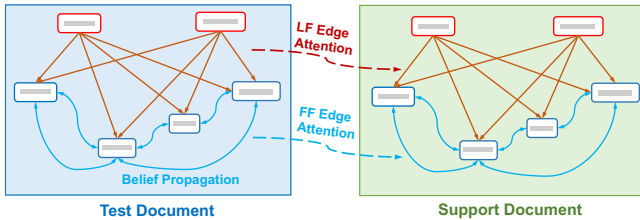


Figure 2: One-shot field labeling model (Landmarks are in red, Fields in blue), including 3 components: (1) attention on LF edges, (2) attention on FF edges, (3) belief propagation.

3.1 Document Graph Construction

We begin our method by describing how to setup the vertices and edges for both support and query document graphs.

Graph Vertices. For support document, both the static fields(L) and dynamic fields(F) on support document are specified by end users. Although there exists some automatic static regions(landmarks) detection methods [1, 17], they are not robust enough. In their methods, text regions of same content are taken as landmarks.

Although they reduce the effort to specify landmarks, they could cause ambiguities because fields of different types could have the same content.

For query document, the landmarks on test images can be identified as follows: first, we performed text detection[29] and text recognition[8]. Then we found out landmarks by matching their text contents with those in the support documents. Finally, we could calculate rough location of characters in landmarks using attention map, and thus the position of landmarks in test image. If a text line has extra text, the quadrilateral for extra text is calculated in the same way. The detection model and recognition model used in this paper are publicly available as a service².

Graph Edges. We put an edge between each landmark and each field. In theory, we could also put an edge between every pair of fields, but we avoided this because the setting takes too much memory. In our method, edges are only added for pairs of fields which are visible to each other. Specially, each field A emits 36 rays from its center to surrounding fields, with the angle increases by 5 degrees to previous line. A pairwise connection is added between the field A and the first fields hit by the rays of A . Finally, loop edges are added to each field so that each field could propagate a message to itself. Detailed comparison of memory consumption for full edges and sparse edges is conducted in the FFAttn section.

3.2 LFAttn: Attention on Landmark to Field edges

We applied attention mechanism to transfer the spatial relationship between landmarks and fields from support documents to query documents. The landmark to field edge features (LF features) are simply the coordinates of landmark and field pairs normalized by their joint bounding box. Similar to prototypical networks [21], the LF features in support documents are first aggregated by type to form type centers. By comparing with type centers, the aggregation step speeds up computation and improves performance[21].

Let $f_L(i, j)$ be landmark to field features for F_i w.r.t a landmark L_j . In support document, the average LF edge features of fields with label k w.r.t landmark L_j is defined as:

$$c_{kj} = \frac{1}{|\{F_i | y_i = k\}|} \sum_{y_i=k} f_L(i, j) \quad (1)$$

Like relation network[24], we used non-linear similarity score functions at attention step: the features from test field and support field are concatenated, and then MLP is applied to generate attention scores:

$$Attn(f_L(i, j), c_{kj}) = MLP(f_L(i, j) \oplus c_{kj}) \quad (2)$$

The attention scores are a tensor of shape $|L| \times |F| \times K$, where K is the number of field types. Note that K could be a different number for different types of documents. We then applied reduce mean operation over the landmark dimension, which generates the final scores S of shape $|F| \times K$. Finally we applied softmax cross entropy loss on the scores S and the labels of fields.

In experiments we found that it is important to average LF edge features after the attention process. The reason is that the direction of LF features is more important than the magnitude, and adding up different LF features could lose important direction information.

²https://ocr.data.aliyun.com/experience/#/?first_tab=general

An intuitive example is that, assume the vector from landmarks a/b to field i are $LF_{ia} = (1, 1)$, $LF_{ib} = (-1, -1)$, then $LF_{ia} + LF_{ib} = (0, 0)$. The summary result (0,0) losses the original direction information.

3.3 FFAttn: Attention over Field to Field edges

It is natural to consider GNN based methods[10] to make use of the relationships between fields. We conducted initial experiments with GNNs(MPAttn) in the experiment section, but the improvement is not significant. The reason is the same as that for LFAttn: aggregation of messages before attention could cause information loss. Here, we proposed to apply pairwise Conditional Random Field (CRF) to utilize the spatial relationship between fields. We presented ablation studies between GNN and our method in the experiment section. We utilized attention mechanism to generate probabilistic distributions for FF edges. The edge features between fields (FF features) are generated in the same way as LF features and attention over the features is conducted similarly.

Let $FF_{k_1 k_2}$ be set of edges between fields with label k_1 and fields with label k_2 . Let $f_F(x_i, x_j)$ be the edge features between field i and field j. The FF edge features in support document image between fields with label k_1 and fields with label k_2 are defined as:

$$c_{k_1 k_2} = \frac{1}{|FF_{k_1 k_2}|} \sum_{y_i=k_1, y_j=k_2} f_F(i, j) \quad (3)$$

$$Attn(c_{k_1 k_2}, f_F(i, j)) = MLP(c_{k_1 k_2} \oplus f_F(i, j)).$$

Memory consumption reduction. If all pairwise edges are included, there could be $|F|^2$ edges between fields in each image, and the memory consumption will be $\alpha \times |F|^2 \times K^2$ (α is constant) at edge attention step, which is too heavy for modern GPUs. To avoid out of memory (OOM) issue, we considered only connections between fields that are visible to each other. As the test image and the support image are of the same layout, this approach will generate similar edge distributions, but memory consumption is reduced to $\alpha \times \beta \times |F| \times K^2$ (α and β are constants). On our test set, the average edge number is 207, and the average field number is 26, so β is approximately 8. In this case, using sparse edges helps to reduce more than 2/3 memory consumption.

3.4 BP: belief propagation

We considered BP for inference of pairwise CRF as it could be easily integrated into an end-to-end trainable pipeline. After FFAttn, a belief propagation matrix Q of dimension $|F| \times |F| \times K \times K$ is generated. Each cell $Q_{i,j,k,l}$ represents the joint probability of $C_i = k, C_j = l$. Then one standard BP step for all fields to field i could be described as $P_i = \prod_{j=0}^{N-1} Q_{ji} P_j$, where $Q_{ji} P_j$ is the message(or belief) passed from field j to i. To enable end-to-end train, BP is done for constant number of steps, while traditional BP ends until reaching the fix point or maximal steps. The message passing along each edge is carried out in parallel in each step. As BP works together with FFAttn, we coined the term **AttnBP** to represent them thereafter.

We presented the details of our one-shot field labeling model in Fig 3. The up part is about LFAttn, which encodes first-order information on fields. The bottom part is about FFAttn, which encodes

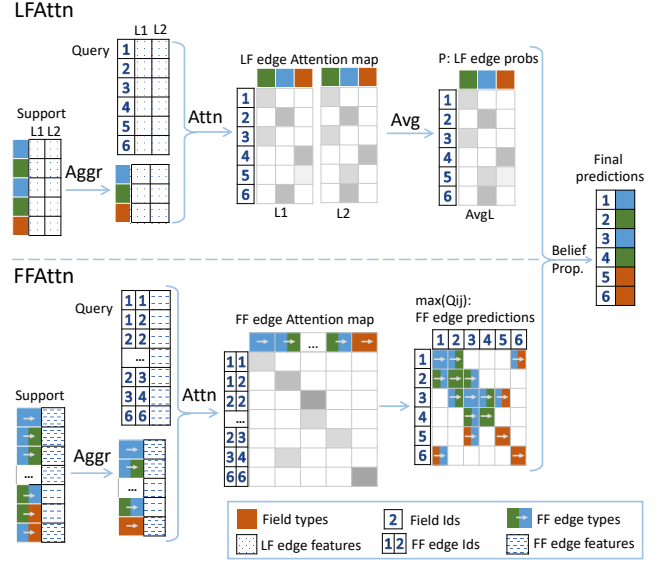


Figure 3: One-shot field labeling model details. L1/L2: landmarks. Aggr: aggregate features by field types or FF edge types. Attn: attention on edge features. Avg: average over the landmark dimension. The RGB colors represent three different field types. The degree of blackness in attention maps is proportional to similarity. Note that LFAttn generates wrong labels (predicted as green, actually red) for field 5, which is corrected by message passed from field 3.

second-order information between fields. These two kinds of information are passed to BP module to generate the final probability distributions of fields. Note that, FFAttn could improve the results when field labels could not be fully determined by landmarks. For example, LFAttn generates wrong labels for field 5, which is corrected by message passed from field 3. The up part can be trained alone or jointly with BP. The bottom part and BP can be also trained alone by replacing the output of up part with uniform label distributions.

4 EXPERIMENTS

4.1 Datasets

Due to the lack of one-shot field labeling datasets, we collected and annotated a new dataset and adapted an existing dataset to form another dataset. Both datasets will be released.

DocLL-oneshot. We created a new dataset consisting of 1307 documents of 43 different types. The dataset is split into two parts, training set with 31 types and testing with 12 types. The train set could be further grouped into 3 big categories: 15 types of receipts, 8 types of certificates and 8 types of administrative documents. The test set could also be grouped into the same big categories: 8 types of receipts, 2 types of administrative documents, and 2 types of certificates. Fig.7 shows sample images from the test set. The dataset is challenging because the document images are taken by smart phones instead of scanned, thus uneven illumination, 3D distortions[11], and noises are quite common.

SROIE-oneshot. Furthermore, we adapted the SROIE[13] dataset to create an additional dataset. We grouped the receipts by shop names, and kept groups of more than 10 receipts, ended up with 10 kinds of receipts. The SROIE-oneshot will be used as an additional test set. Although the images are scanned and have less distortions, the dataset is still challenging because they also includes lots of **multi-region** fields(Fig.8), and different field types are interleaved, such as *i_name*, *i_price*, *i_unit*, *i_code*, *i_no* in Fig.8(a).

Groundtruth Generation. The two datasets used in this work are labeled by two annotators separately. After that, we crossed check the labels and fix conflicts. After labeling, each image is rotated so that the average angle of all texts is zero. Finally, the unlabeled text lines are detected by the IncepText detector[29] and labeled as background classes.

Train. We used Tensorflow to implement our models, all the models are trained on 1080Ti with 10G memory. For all experiments, we trained with SGD using a batch size of 8 and 20000 training iterations. The momentum of SGD is 0.9, and initial learning rate is 0.01, and decayed by 0.1 after every 5000 iterations. Note that if BP is used, more GPU memory will be consumed. The training process takes around 5.5 hours which is around 2 times higher than the one without BP. For each batch, we random sampled 8 types of templates, and then random sampled a pair of images from each type of templates. The whole training process takes 2.5/5.5 hours depending on whether BP is used. If BP is used, we placed its ops and gradients on a separate gpu, which can be easily implemented using tf.device api. For inference, all modules of the model are placed on a single gpu.

Evaluation. For each query document Q , its support set S includes all other documents of the same class. The accuracy of Q is defined as the mean accuracy of pairs in $\{(Q, Q') | Q' \in S\}$, which is then averaged by type. The final accuracy is defined as the mean accuracy of all types.

4.2 Overall results

In this section, we compared our method with existing rule-based, image feature based, and landmark feature based methods. The existing one-shot methods using text image features are based on VGG like networks[6], but modern CNN networks with residual connections are more accurate and efficient. Hence in this work, a Resnet-18 variant that has better performance is used for feature extraction. We also extended existing rule-based methods(IncSt[17], HybridIncSt[3]) in order to handle **multi-region** types of fields: we did not vote for the most probable text region for each type, we voted for the most probable type for each text region instead.

The results are presented in Table 1. First, existing few-shot methods on image features have lower performance than rule-based methods. The reason is that image features of different types are not easy to separate as intra-type variance is large compared with inter-type variance for many field types. In contrast, the spatial relationship between fields are relatively stable, thus their intra-type variance is small compared with inter-type variance. Our LFAtn method improves the rule-based methods, as the learned features maximize the gap between different types and generate more robust classifiers than handcraft rules. Besides, in contrast to rule-based methods that extract different field types separately, our methods

Method	Dist	DocLL-oneshot	SROIE-oneshot
* Rule-based			
IncSt[17]	-	0.777	0.764
HybridIncSt[3]	-	0.793	0.783
* Image features			
Siamese	Euclidean	0.414	0.257
MatchNet	Euclidean	0.665	0.593
ProtoNet	Euclidean	0.638	0.550
RelNet	Learned	0.627	0.566
* LF Features			
Siamese	Cosine	0.328	0.206
MatchNet	Cosine [†]	0.941	0.969
ProtoNet	Cosine	0.938	0.960
Ours	Learned	0.939	0.965
* LF Features + FF Features			
MatchNet	Cosine [†]	0.942	0.970
ProtoNet	Cosine	0.948	0.973
Ours	Learned	0.964	0.985

Table 1: Comparison with state of art one-shot methods. [†] means we added linear transform to Cosine distance.

Methods	All_L	AttnBP	AttnBP	
			+All_L	+Det_L
DocLL-oneshot	0.936	0.934	0.964	0.961
SROIE-oneshot	0.960	0.962	0.985	0.984

Table 2: The importance of landmarks and BP. All_L: LFAtn use all landmarks, Det_L: LFAtn use landmarks detected in preprocessing step.

extract all fields types at once. We could then take into consideration of spatial constraints between fields by making use of AttnBP, which further improves the results by $\sim 3\%$ (from 0.939 to 0.964). Second, for MatchNet and ProtoNet with landmark features, we used cosine similarity instead of euclidean similarity. The reason is that LF features do not conform to exponential distribution assumed by euclidean distance[21]. The variance in magnitude could be large due to the difference in image sizes, while the variance in angles is relatively small. Last, the FF feature distribution is more complex than LF feature distribution due to the existence of **multi-region** field types. When FF features is used, ProtoNet has better performance than MatchNet as it poses attention on the type center which is more stable [21]. Our method uses learned nonlinear similarity function, which is shown to improve the ProtoNet.

In a nutshell, with the incorporation of attention (LFAtn) and belief propagation (AttnBP), our method achieves the state-of-the-art results on all the datasets.

4.3 The importance of landmarks and BP

Here we conducted model ablation studies to examine the importance of landmarks and belief propagation (BP). As shown by Table 2, the results on both test datasets are consistent. A comparison of

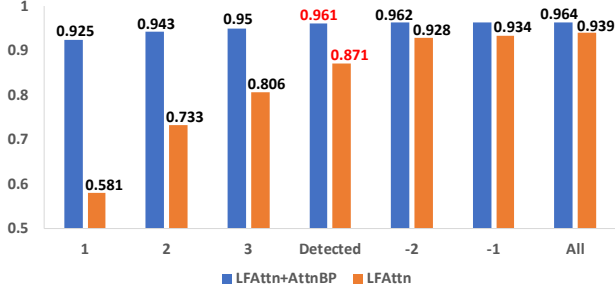


Figure 4: Document field labeling accuracy vs number of landmarks. All:use all landmarks, -1/-2:random drop 1/2 landmarks, detected:using landmarks from preprocessing step, 1/2/3: only keep 1/2/3 landmarks.

Methods	DocLL-oneshot	SROIE-oneshot
AvgAttn	0.774	0.702
LFAtn	0.939	0.965
LFAtn + MPAttn	0.946	0.978
LFAtn + GAT	0.948	0.979
LFAtn + AttnBP	0.964	0.985

Table 3: Comparison of various attention methods.

using All_L+AttnBP and AttnBP shows that landmarks contribute to an increase of 3% of accuracy on average. For belief propagation, two message passing steps are used as more steps leads to no more improvement. The introducing of attention based belief propagation module greatly improves the labeling accuracy. It also shows that AttnBP itself can achieve very good performance.

We further examine the impacts of the number of landmarks on labeling accuracy in Fig.4. As expected, the labeling accuracy grows as the number of landmarks increases. The overall accuracy is good if dropping less than 3 landmarks (see -1,-2 in x-axis). Our existing landmark detection method misses 1.7 landmarks in average, but with a large variance (the maximal number of landmarks missed is 16), so the final performance is worse than randomly dropping 2 landmarks. The reason is that our text detection and recognition models are not specifically trained for documents, hence maybe not very precise in landmark detection. If we collected more documents to finetune those models, this issue can be alleviated. Another way to alleviate this is to use end-to-end training with AttnBP. As in Fig.4, the accuracy of LFAtn+AttnBP is less affected by the number of landmarks because AttnBP introduces second-order information. Thus, even some of the landmarks are not detected, the accuracy is almost the same. Moreover, AttnBP could handle complex table layout which is composed of interleaved field types in crowded regions where fewer landmarks are available. More details are presented in the case-study section.

4.4 The effectiveness of attention

Our model achieves high accuracy on a variety of test documents which are not seen in training phase. It shows that layout information transferred by attention mechanism is effective. We further

tested two alternative designs to verify the effectiveness of the design of attention module:

Average before Attention (AvgAttn). The average operation over the landmark dimension is moved to the front of attention operation. As in Table 3, the accuracy is far behind that of LFAtn. We also found reduce-max has no improvement. Without attention, aggregation over LF features mixes up the information from all landmarks and leads to less distinguishable features.

Message Passing before Attention (MPAttn). Following the graph matching network [9], we considered message passing before field feature attention, which is defined as:

$$f(i) = \sum_{j=1}^{|F|} MLP(f_F(i, j) \oplus Pad(P_i) \oplus Pad(P_j))$$

$$c_k = \frac{1}{|\{F_i | y_i = k\}|} \sum_{y_i=k} f(i) \quad (4)$$

where P_i is from LFAtn, and Pad is to pad to the max number of types (48 in experiments). The field feature $f(i)$ of support document is built in the same way except that P_i is the one hot encoding of y_i . After message passing, attention of $f(i)$ over c_k is conducted in the same way as LFAtn. Finally, residual structure is applied to improve performance; otherwise, the performance is worse than LFAtn. As shown in Table 3, the improvement is less than AttnBP. We also tried graph attention networks(GAT)[25], the improvement is also inferior to our methods. With attention ahead of message passing, the message types can be determined first, and messages passed between fields are more informative.

4.5 Case study

In this section, we presented a case study on document d0 - one of the most challenging documents in our dataset. The accuracies for d0 are: HybridIncSt 0.72, RelNetImg 0.77, LFAtn 0.84, and LFAtn+AttnBP 0.93. Our LFAtn+AttnBP method performs consistently better than all the other methods.

To dive into the details of our model performance, we presented the confusion matrix of HybridIncSt, RelNetImg, LFAtn, and LFAtn+AttnBP for document d0 in Fig.6. Rule-based method (Fig.6a) has difficulty in classifying m:change and h:pay(□), as well as k:discount and p:total_price(□), as they are distant from landmarks. The document is challenging for RelNetImg (Fig.6b) in two aspects: first, the text contents of g:addr, b:buy_desc, d:buy_name fields are highly variant, thus many of them are misclassified as a:background(□); second, fields of h:pay, p:total_price, and m:change(□) have similar content. The document is also challenging for LFAtn(Fig.6b) because there are lots of multi-region fields in the middle but there are few landmarks in the middle: the confusion between d:buy_name, o:date, and q:card is high(□), so as the confusion between i:buy_price and n:time(□). A comparison of the confusion matrix(Fig.6c, Fig.6d) shows that LFAtn+AttnBP clears the above confusions.

Per Document Type Results. As shown in Table 4, our approach has the best performance for almost all types of document. It can also be seen that receipts are usually more challenging than administrative documents and certificates, because receipt layouts are more complex and have fewer landmarks. Rule-based methods

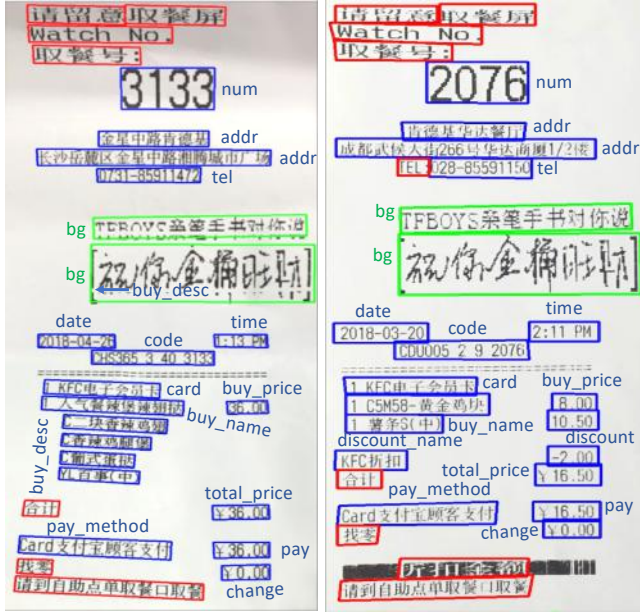


Figure 5: Sample d0 from one-shot dataset.

perform good for fields that are close to landmarks(d3, d6, d7); but they have low performance on documents of complex(d1, d2). They also perform bad on d4, d5 because they are **sensitive to background(bg) fields**: many background fields are labeled incorrectly as issue_* and valid_* because the position of issue_* and valid_* w.r.t to landmarks could shift. If the background fields are dropped, their performance improves a lot (d4:0.997, d5:0.954). As the percentages of background fields are relatively large(17% for DocLL-oneshot test dataset and 16% for SROIE-oneshot dataset), we further evaluated the impact of background fields on labeling accuracy by dropping all the background fields: Table 5 indicates that rule based methods are sensitive to background fields, while our method is almost not affected by this issue, thus our method is more robust.

5 FEW-SHOT FIELD LABELING

Previous studies on image classification problem report that few-shot learning methods improve over their corresponding one-shot versions. We extended our one-shot approach to few-shot to further improve the performance.

Due to the difference in scales of different images, taking the average of LF and FF features over the support set S does not lead to good results. Hence following matching networks, we took the average of the one-shot probabilities of $|S|$ query-support pairs as the output of few-shot methods. Following the existing studies, we considered 5-shot layout labeling, i.e. $|S| = 5$.

As expected, all five-shot methods except Siamese Network perform better than one-shot methods (Table 6). Our LFAtn+AttnBP model consistently outperforms all other methods and outperforms image features based methods by a large margin. As the improvement of five-shot LFAtn+AttnBP over the one-shot version is relatively small (1.4%), we took a close examination of per document type improvement and found that five-shot improves more over

Methods	d1	d2	d3	d4	d5	d6	d7
* Rule based							
IncSt	0.83	0.72	0.91	0.69	0.75	0.96	0.91
HybridIncSt	0.84	0.73	0.92	0.71	0.80	0.96	0.99
* Image features							
MatchNet	0.52	0.65	0.66	0.76	0.68	0.65	0.84
ProtoNet	0.51	0.64	0.58	0.71	0.65	0.65	0.77
RelNet	0.50	0.64	0.58	0.67	0.62	0.67	0.83
* LF Features							
MatchNet	0.91	0.90	0.98	1.00	0.97	0.97	1.00
ProtoNet	0.90	0.88	0.98	1.00	0.99	0.98	0.99
Ours	0.91	0.88	0.98	1.00	1.00	0.98	1.00
* LF Features + FF features							
MatchNet	0.91	0.90	0.99	1.00	0.98	0.97	1.00
ProtoNet	0.90	0.94	0.99	0.98	1.00	0.96	0.99
Ours	0.93	0.95	0.99	1.00	1.00	0.97	1.00

Table 4: Per-type labeling accuracy of DocLL-oneshot test documents.

Dataset	IncSt	HybridIncSt	LFAttn	LFAttn + AttnBP
DocLL-oneshot	0.883	0.891	0.944	0.971
Incre	0.106	0.098	0.005	0.007
SROIE-oneshot	0.862	0.866	0.964	0.985
Incre	0.098	0.083	-0.001	0.000

Table 5: The impact of background fields(bg fields) on labeling accuracy. Incre is the increase in accuracy if background fields is dropped.

Methods	Dist.	DocLL-5-shot	SROIE-5-shot
* Rule based			
IncSt	-	0.776(-0.001)	0.779(+0.015)
HybridIncSt	-	0.806(+0.013)	0.797(+0.014)
* Image Features			
Siamese	Euclidean	0.435 (+0.031)	0.266 (+0.009)
MatchNet	Euclidean	0.724 (+0.059)	0.651 (+0.058)
ProtoNet	Euclidean	0.732 (+0.094)	0.651 (+0.101)
RelNet	Learned	0.711 (+0.088)	0.663 (+0.097)
* Landmark Features			
Siamese	Cosine	0.314 (-0.014)	0.229 (-0.023)
MatchNet	Cosine†	0.962 (+0.021)	0.983 (+0.014)
ProtoNet	Cosine	0.963 (+0.025)	0.975 (+0.015)
LFAttn	Learned	0.961 (+0.022)	0.979 (+0.014)
* Landmark Features + AttnBP			
MatchNet	Cosine†	0.952 (+0.038)	0.978 (+0.024)
ProtoNet	Cosine	0.971 (+0.023)	0.986 (+0.013)
LFAttn	Learned	0.978 (+0.014)	0.991 (+0.006)

Table 6: Comparison with the state of art five-shot methods. The numbers in () are improvements from one-shot methods. † means we added linear transform to Cosine distance.

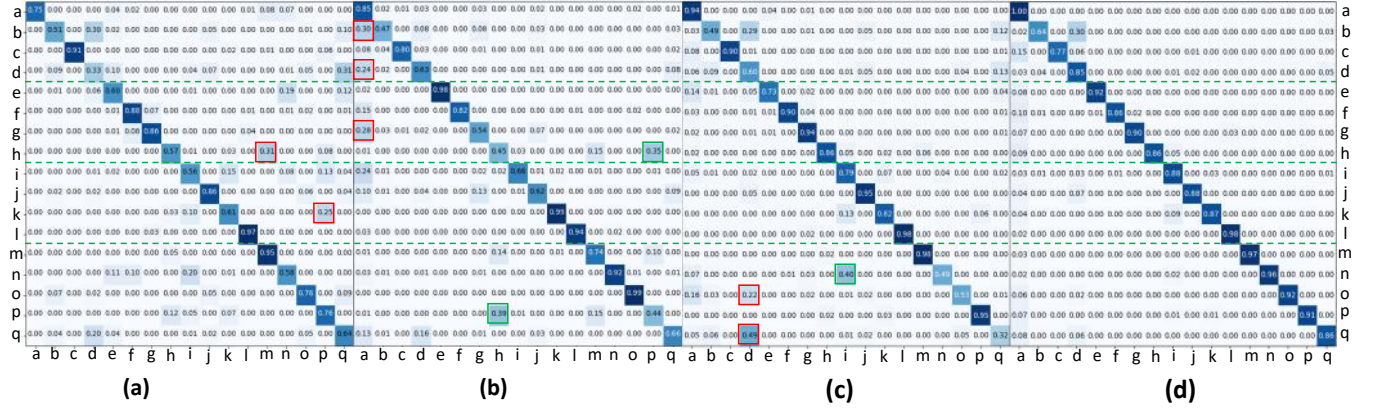


Figure 6: Confusion matrix for document d0 (best viewed in color). x axis: prediction, y axis: groundtruth.(a) HybridIncSt, (b) RelNetImg, (c) LFAtn, (d) LFAtn+AttnBP. Field label alias: a:background, b:buy_desc, c:pay_type, d:buy_name, e:code, f:tel, g:addr, h:pay, i:buy_price, j:discount_name, k:discount, l:num, m:change, n:time, o:date, p:total_price, q:card.

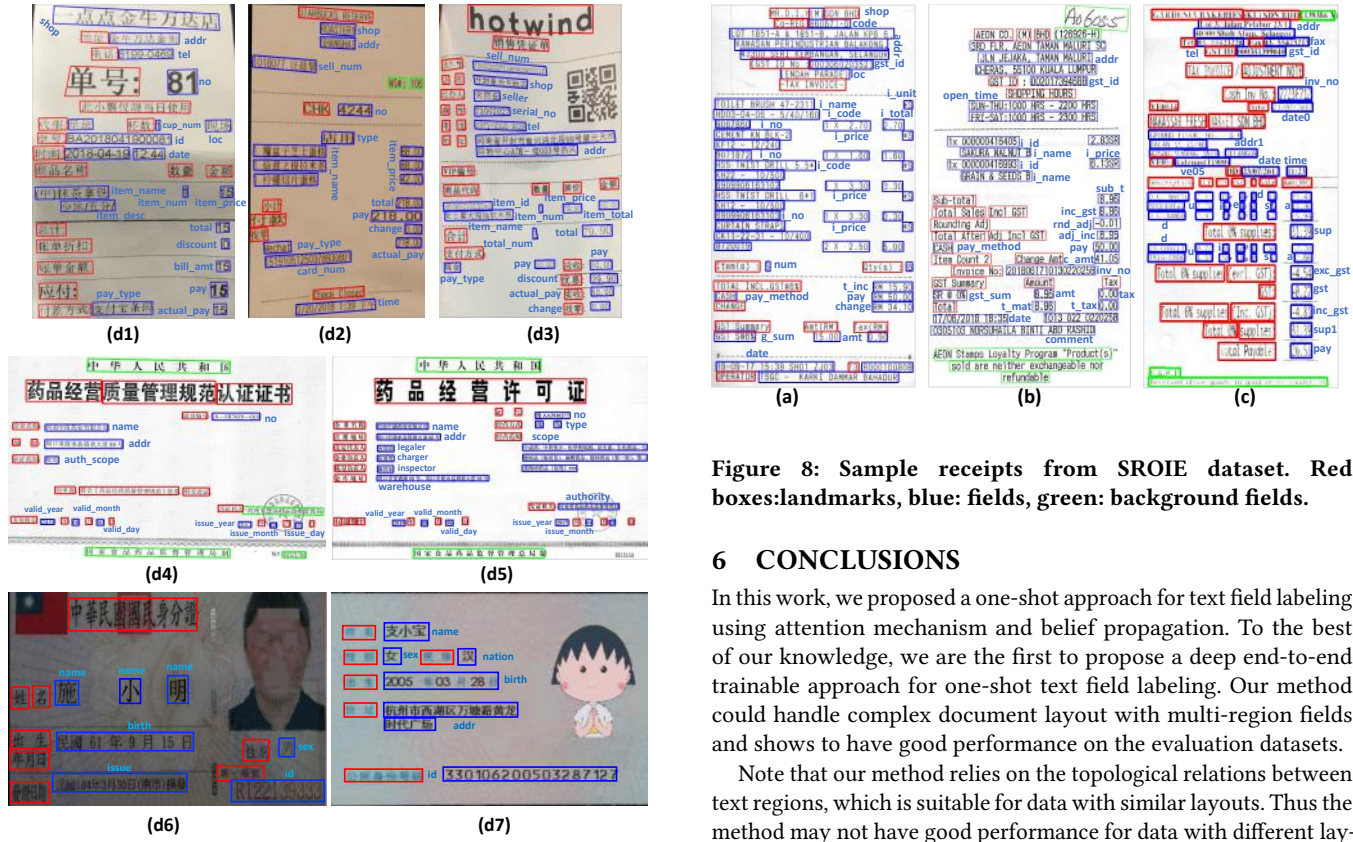


Figure 7: Sample receipts (d1,d2,d3), administrative docs(d4, d5), and certificates(d6, d7). Red boxes:landmarks, blue: fields, green: background fields.

the challenging types: 2.6% over 6 types of hardest documents, 0.3% for the rest 6 types whose accuracy is close to 1(99.2%).

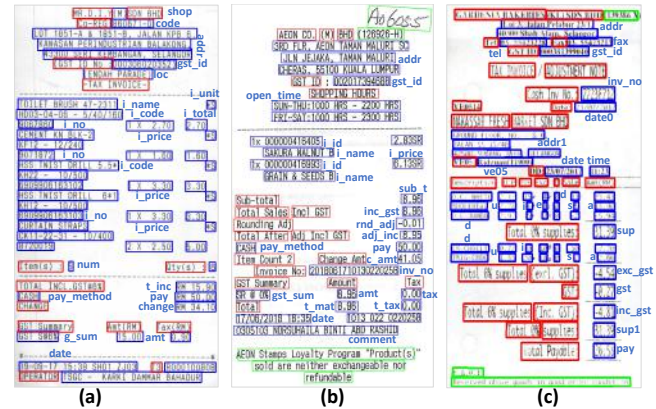


Figure 8: Sample receipts from SROIE dataset. Red boxes:landmarks, blue: fields, green: background fields.

6 CONCLUSIONS

In this work, we proposed a one-shot approach for text field labeling using attention mechanism and belief propagation. To the best of our knowledge, we are the first to propose a deep end-to-end trainable approach for one-shot text field labeling. Our method could handle complex document layout with multi-region fields and shows to have good performance on the evaluation datasets.

Note that our method relies on the topological relations between text regions, which is suitable for data with similar layouts. Thus the method may not have good performance for data with different layouts. We seek to handle such variations to improve the robustness of our method in future. Meanwhile, our method can be extended to zero-shot scenarios as well, we leave it as another future work.

7 ACKNOWLEDGMENTS

This work was partially funded by China Postdoctoral Science Foundation (No. 2019M652038). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

REFERENCES

- [1] David Aldavert, Marçal Rusiñol, and Ricardo Toledo. 2017. Automatic static/variable content separation in administrative document images. In *International Conference on Document Analysis and Recognition*, Vol. 1.
- [2] Santanu Chaudhury, Megha Jindal, and Sumantra Dutta Roy. 2009. Model-guided segmentation and layout labelling of document images using a hierarchical conditional random field. In *ICPRAI*. Springer.
- [3] Vincent Poulain d'Andecy, Emmanuel Hartmann, and Marçal Rusiñol. 2018. Field extraction by hybrid incremental and a-priori structural templates. In *International Workshop on Document Analysis Systems*.
- [4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*.
- [5] Maroua Hammami, Pierre Hérault, et al. 2015. One-shot field spotting on colored forms using subgraph isomorphism. In *International Conference on Document Analysis and Recognition*.
- [6] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *International Conference on Machine Learning workshop*, Vol. 2.
- [7] Sunil Kumar, Rajat Gupta, et al. 2007. Text extraction and document image segmentation using matched wavelets and MRF model. *IEEE Transactions on Image Processing* (2007).
- [8] Chen-Yu Lee and Simon Osindero. 2016. Recursive recurrent nets with attention modeling for ocr in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [9] Yujia Li, Chenjie Gu, et al. 2019. Graph Matching Networks for Learning the Similarity of Graph Structured Objects. *International Conference on Machine Learning* (2019).
- [10] Xiaojing Liu, Feiyu Gao, et al. 2019. Graph Convolution for Multimodal Information Extraction from Visually Rich Documents. *The North American Chapter of the Association for Computational Linguistics* (2019).
- [11] Ke Ma, Zhixin Shu, et al. 2018. Docunet: document image unwarping via a stacked U-Net. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [12] Eric Medvet, Alberto Bartoli, and Giorgio Davanzo. [n.d.]. A probabilistic approach to printed document understanding. *International Journal on Document Analysis and Recognition* 14, 4 ([n. d.]).
- [13] International Conference on Document Analysis and Recognition. 2019. *SROIE, ICADR*, <https://rrc.cvc.uab.es/?ch=13&com=introduction>, 2019.
- [14] Rasmus Berg Palm, Florian Laws, and Ole Winther. 2018. Attend, Copy, Parse-End-to-end information extraction from documents. *arXiv preprint arXiv:1812.07248* (2018).
- [15] Claudio Antonio Peanho, Henrique Stagni, et al. [n.d.]. Semantic information extraction from images of complex documents. *Applied Intelligence* 37, 4 ([n. d.]).
- [16] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. *International Conference on Learning Representations* (2017).
- [17] Marçal Rusiñol, Tayeb Benkhelfallah, et al. 2013. Field extraction from administrative documents by incremental structural templates. In *International Conference on Document Analysis and Recognition*.
- [18] Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management* 24, 5 (1988), 513–523.
- [19] Adam Santoro, Sergey Bartunov, et al. 2016. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning*.
- [20] Baoguang Shi, Xiang Bai, and Cong Yao. 2016. 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* (2016).
- [21] Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Neural Information Processing Systems*.
- [22] Carlos Soto and Shinjae Yoo. 2019. Visual Detection with Context for Document Layout Analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3455–3461.
- [23] Vishal Sunder, Ashwin Srinivasan, et al. 2019. One-shot Information Extraction from Document Images using Neuro-Deductive Program Synthesis. *International Joint Conference on Artificial Intelligence workshops* (2019).
- [24] Flood Sung, Yongxin Yang, et al. 2018. Learning to compare: Relation network for few-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *International Conference on Learning Representations* (2018).
- [26] Oriol Vinyals, Samy Bengio, and Manjunath Kudlur. 2015. Order matters: Sequence to sequence for sets. *International Conference on Learning Representations* (2015).
- [27] Oriol Vinyals, Charles Blundell, et al. 2016. Matching networks for one shot learning. In *Neural Information Processing Systems*.
- [28] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2019. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. [arXiv:1912.13318 \[cs.CL\]](https://arxiv.org/abs/1912.13318)
- [29] Qiangpeng Yang, Mengli Cheng, et al. 2018. Inceptext: A new inception-text module with deformable psroi pooling for multi-oriented scene text detection. *International Joint Conference on Artificial Intelligence* (2018).
- [30] Xiao Yang, Ersin Yumer, et al. 2017. Learning to extract semantic structure from documents using multimodal fully convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.
- [31] Andrei Zănfir and Cristian Sminchisescu. 2018. Deep learning of graph matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2684–2693.
- [32] Xinyu Zhou, Cong Yao, et al. 2017. EAST: an efficient and accurate scene text detector. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5551–5560.