Received March 16, 2022, accepted March 30, 2022, date of publication April 4, 2022, date of current version April 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3164674

Tokengrid: Toward More Efficient Data Extraction From Unstructured Documents

ARSEN YEGHIAZARYAN[1], KHACHATUR KHECHOYAN[2], GRIGOR NALBANDYAN[3], AND SIPAN MURADYAN[4]

[1]Department of Theoretical Physics, Yerevan Physics Institute, Yerevan 0036, Armenia
[2]Faculty of Mathematics and Mechanics, Yerevan State University, Yerevan 0025, Armenia
[3]Department of Informatics, Technical University of Munich, Münich, 80333 Bavaria, Germany
[4]College of Science and Engineering, American University of Armenia, Yerevan 0019, Armenia

Corresponding author: Khachatur Khechoyan (khachatur.khechoyan@portmind.com)

ABSTRACT Key information extraction from unstructured documents is a practical problem in many industries. Machine learning models aimed at solving this problem should efficiently utilize textual, visual, and 2D spatial layout information of the document. Grid based approaches achieve this by representing the document as a 2D grid and feeding it to a fully convolutional encoder-decoder network that solves a semantic instance segmentation problem. We propose a new method for the instance detection branch of that network for the task of automatic information extraction from invoices. Our approach reduces this problem to 1D region detection. The proposed network has fewer parameters and a shorter inference times. Additionally, we suggest a new metric for evaluating the results.

INDEX TERMS Computer vision, data extraction, document handling, machine learning, object detection, semantic segmentation.

I. INTRODUCTION

The presence of various types of scanned business documents in industry and international trade remains significant. They often undergo a common procedure of extraction of key information from them. In many cases, it is still performed manually, which is laborious, slow, and error-prone.

The information in documents is generally organized into various layouts. Significant amount of information is conveyed with the 2D relative positioning of the words, as well as other visual cues such as grids, tables, text sizes, etc. For a good machine learning model aimed at automating the Key Information Extraction (KIE) task, it is crucial to make use of the textual, visual, and spatial information contained in the document as much as possible. The current work concentrates on building a strong machine learning model for invoices, but the method is applicable to other types of documents as well.

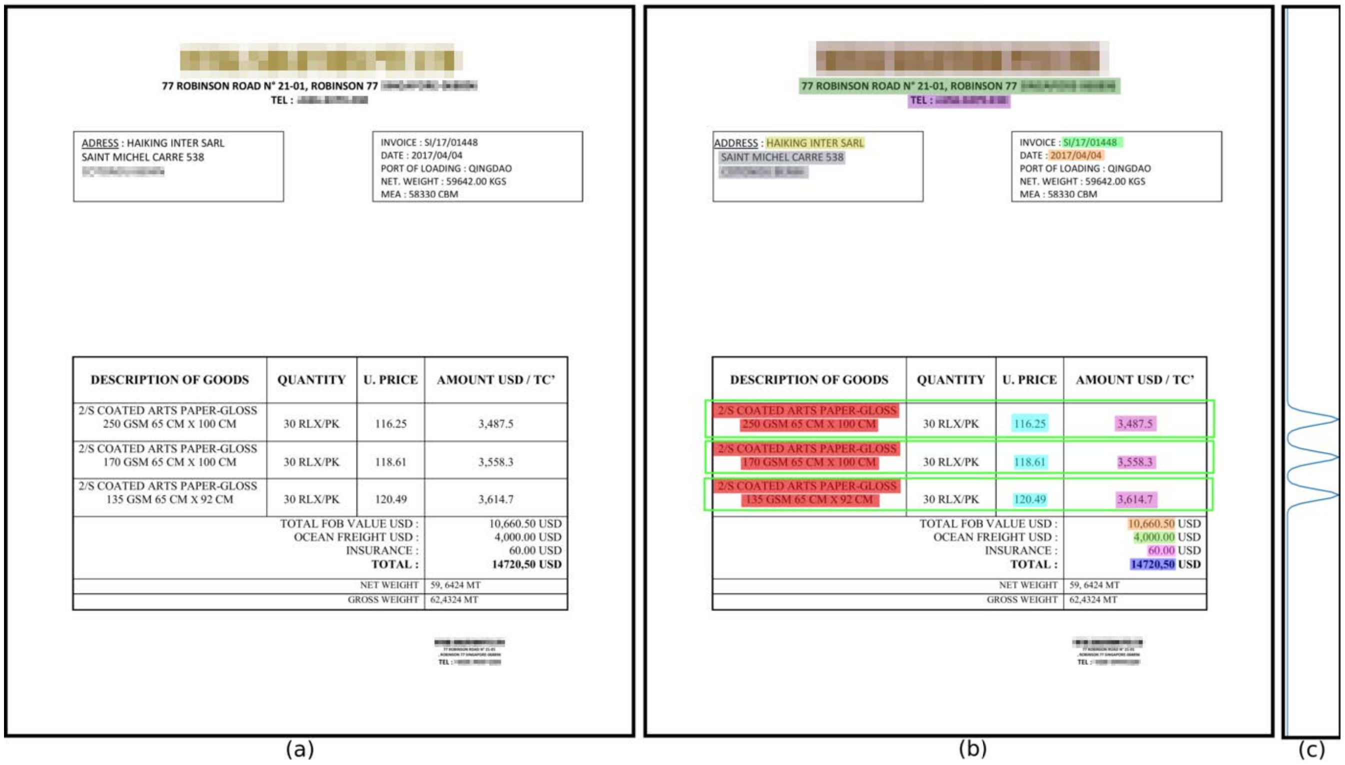
Invoices from different vendors can vary significantly in their layouts and the amount of information contained. In addition, the number of key fields in invoices is often significantly larger than in other documents, such as checks, passports, etc.

The associate editor coordinating the review of this manuscript and approving it for publication was Massimo Cafaro.

In machine learning, the scanned document parsing problem is generally decoupled into two sequential phases, Optical Character Recognition (OCR) and KIE. Alternatively, in [2] an end-to-end trainable framework was proposed to handle the two phases in one pass. However, we employ the common two-phase approach, and in this work we concentrate only on the second phase KIE problem.

Initial attempts to solve the KIE task with deep learning were based on the Named Entity Recognition (NER) methods of Natural Language Processing (NLP). For example, in [3], the document was serialized into a 1D text sequence and fed to a recurrent neural network (RNN), which was trained to predict the IOB tags [4] of the input sequence. It is evident that such serialization discards important 2D layout information. To mitigate this, the 1D text in [3] was augmented with several manually engineered features that incorporate the spatial information of the document.

Graph Convolutional Networks (GCN) were employed in [5] to encode the textual and spatial components of a document. The text segments in the document were represented as graph nodes, and the spatial relationships were encoded in the edges. After passing through several layers of graph convolution, the text segment representations were fed to a BiLSTM-CRF [6] module to predict the IOB tags.


[redacted]
77 ROBINSON ROAD N° 21-01, ROBINSON 77 [redacted]
TEL : [redacted]

ADRESS : HAIKING INTER SARL
SAINT MICHEL CARRE 538
[redacted]

INVOICE : SI/17/01448
DATE : 2017/04/04
PORT OF LOADING : QINGDAO
NET. WEIGHT : 59642.00 KGS
MEA : 58330 CBM

DESCRIPTION OF GOODS	QUANTITY	U. PRICE	AMOUNT USD / TC'
2/S COATED ARTS PAPER-GLOSS 250 GSM 65 CM X 100 CM	30 RLX/PK	116.25	3,487.5
2/S COATED ARTS PAPER-GLOSS 170 GSM 65 CM X 100 CM	30 RLX/PK	118.61	3,558.3
2/S COATED ARTS PAPER-GLOSS 135 GSM 65 CM X 92 CM	30 RLX/PK	120.49	3,614.7
		TOTAL FOB VALUE USD :	10,660.50 USD
		OCEAN FREIGHT USD :	4,000.00 USD
		INSURANCE :	60.00 USD
		TOTAL :	**14720,50 USD**
		NET WEIGHT	59, 6424 MT
		GROSS WEIGHT	62,4324 MT

[redacted]
TEL : [redacted]

FIGURE 1. **(a) A sample invoice, (b) Segmentation ground truth of the invoice colored with different colors for each tag and Line Item detection ground truth annotated with rectangles, (c) Line Item heatmaps generated from rectangles, the heatmaps have high values at the center of the boxes and fade out towards the box boundaries. Some information in the invoice is blurred on purpose.**

However, both [5] and [3] lack the capacity to efficiently utilize the visual modality of the data.

Grid based methods [1], [7], [9] exploit the textual and spatial information of a document by building a grid in which pixels are encoded using character- or token-level embeddings. This grid is then fed to a convolutional encoder-decoder network. The visual modality can be utilized by concatenating the RGB channels of the document to the input grid (e.g. [1]), or by passing the image through a separate encoder (e.g. [9]). One-hot character encoding was used in [1], while in [9] static word embeddings were utilized. In [7] contextualized token embeddings taken from BERT [14] model were used. The BERT model in this case was pre-trained independently on a large unsupervised invoice dataset.

Another approach for utilizing visual, textual, and spatial information of documents is based on transformer [19] architecture. Methods like [16]–[18] allow cross-modal end-to-end interaction in the unsupervised pre-training and downstream task fine-tuning stages. To achieve this, [16], [17] modified the regular self-attention layer of the transformer network to include visual features as well, while [18] implemented early and late fusions of BERT and CNN features.

In the case of invoices, in addition to token classification, Line Item detection is also necessary (see Figure 1). This is similar to the semantic instance segmentation problem, which is a well-known task in computer vision. In this problem a separate convolutional branch is generally added for instance detection alongside the segmentation branch, as in [25], [26]. Grid based methods use fully convolutional architectures and can include an instance detection branch in the most natural manner, compared to other mentioned methods. Below we discuss the challenges with applying the conventional instance detection methods to Line Item detection in invoices.

A. CHALLENGES

As can be seen from Figure 1, the tags in invoices can be naturally separated into two groups: item-level tags (e.g., product description, item price, unit price) and document-level tags (e.g., invoice date, invoice number, vendor name and address, total price). Also, we add a special 'background' tag that is assigned to all words that do not represent any key information. The fully convolutional architectures used in [1], [7] are encoder-decoder architectures with two output branches: segmentation branch for per-pixel tag classification, and instance detection branch for predicting Line Item bounding boxes. The detection branch effectively performs grouping of item-level tags and associates each item-level field with its own Line Item. In invoices, in all cases of practical interest, the Line Item boxes appear placed as rows, as displayed in Figure 1. Hence, it is sufficient to predict only the vertical boundaries of the Line Items. Based on this, the item detection problem can be formulated as a prediction of only the y-coordinate boundaries of the Line Item boxes, neglecting their width. For bounding box prediction, the grid based methods [1], [7] use anchor box classification method as in [10], [20]. They slide over the 2D feature map and predict the existence and size of bounding boxes on each pixel of the map. Considering the 1D nature of the Line Items, this is redundant.

Another problem with Line Item bounding boxes is related to the ambiguity in their ground truth sizes. This problem was also mentioned in [1]. For example, if some 'background' words are associated with one Line Item or another, it will have no consequence on the correct item-level tag grouping. This observation becomes even more relevant when dealing with real-life invoices. It is not always clear which 'background' words should be included in annotated Line Item bounding boxes and which ones should be omitted. In fact, each Line Item bounding box may be of any size and contain any number of 'background' words, as long as it includes all relevant key item-level information. This fact makes the Line Item anchor ground truth targets vaguely defined, and suggests that methods other than anchor classification can be applied to this problem. It also follows that the usual metrics of evaluation of bounding box predictions, such as MAP or F1 at certain IoU thresholds, can be non-optimal for this problem.

In this work, we propose a method to replace the 2D instance bounding box regression with 1D vertical region detection. For this purpose, we propose two methods. The first method is a reduction of the standard Faster RCNN-like anchor classification [10] to '1D anchor' classification (Section II-D). The second method is a 1D heatmap prediction, where we predict the probability of each vertical coordinate of the document to be the center of a Line Item (Section II-E). In addition, we suggest a metric for evaluating the Line Item predictions that addresses the above-mentioned ground truth ambiguity and better reflects the Line Item prediction quality.

The rest of the paper is organized as follows. Section II discusses the model architecture and the details of 1D region detection. In Section III we present how the data is collected and annotated. Section IV introduces new metrics for evaluating Line Item detection in invoices. We present the results of our experiments in Section V, and Section VI provides the conclusion.

II. METHOD

In this section the model pipeline is described. First, we describe the document representation fed into the model. We then present the network architecture and address the idea of working only with the height dimension in Line Item detection task. To achieve this, we propose two methods: classification with 1D anchors (II-D), and item heatmap prediction (II-E).

A. DOCUMENT REPRESENTATION

Our work employs the grid based approach for encoding the documents. This idea is similar to [7].

First, we apply an OCR engine to the document. Using the OCR output, the document text can be represented as a set of tuples $D = \{(t_k, b_k) | k = 1, \ldots, n\}$, where t_k is the kth token in the text of the document and $b_k = (x_k, y_k, w_k, h_k)$ is its bounding box in the image. Here, (x_k, y_k) are the top-left coordinates of the bounding box and (w_k, h_k) are its width and height, respectively. We can now construct the grid representation $G \in R^{H \times W \times d}$ of the original document with height H and width W:

$$G_{ij} = \begin{cases} e_d(t_k) & \text{if } (i,j) \prec b_k \\ 0_d & \text{otherwise} \end{cases}$$
$$(i,j) \prec b_k \iff x_k \le i \le x_k + w_k \ \wedge \ y_k \le j \le y_k + h_k, \tag{1}$$

where the point (i, j) is the pixel with corresponding coordinates in the document, d is the embedding dimension, e_d is the embedding function of the token, and 0_d denotes an all-zero vector of size d. We call this document representation *tokengrid*. The choice of the embedding function is discussed in the Experiments section.

B. ARCHITECTURE

The model architecture is similar to [1]. The network includes an encoder and two decoders, as shown in Figure 2. One decoder solves a semantic segmentation problem, and the other decoder performs Line Item detection. The encoder and segmentation decoder are the same as in [1]. Our proposed modifications are in the Line Item detection decoder.

The input to the encoder can be a grid of character one-hot encoding, as in [1], as well as the tokengrid G defined in Section II-A. The encoder is a VGG-type network [21] that downsamples the spatial dimension of the input with stride-2 and dilated convolutions [22] and increases the number of channels.

The decoder for semantic segmentation consists of convolutional blocks that reverse the downsampling of the encoder with stride-2 transposed convolutions. After each upsampling the number of channels is decreased by a factor of two. Lateral connections with encoder features are used at each feature map resolution. The output of the segmentation branch is a tensor of shape $(H, W, K + 1)$, where $(K + 1)$ represents the 'background' tag and K predefined tags. Softmax activation is used to predict the probability distribution among the classes for each pixel.

For more details on the architecture of the encoder and segmentation decoder refer to [1].

C. LINE ITEM DETECTION DECODER

As argued above, the Line Item detection can be reduced from 2D bounding box detection problem to 1D vertical region detection. First, the ground truths of Line Items are represented as vertical regions instead of 2D boxes. This is achieved by discarding the width dimension of the boxes. In the network architecture the idea of dimensionality reduction is exploited by average-pooling all feature maps that

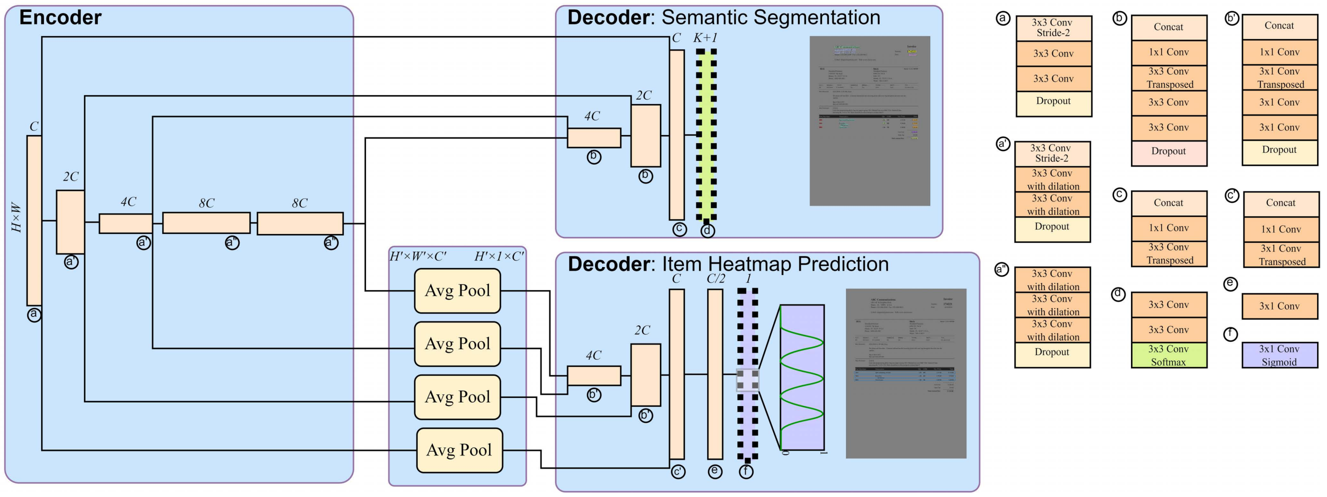

Encoder
H×W
C
2C
4C
8C
8C
a
a'
a'
a''
a''
Decoder: Semantic Segmentation
C
K+1
2C
4C
b
b
c
d
H'×W'×C'
H'×1×C'
Avg Pool
Avg Pool
Avg Pool
Avg Pool
Decoder: Item Heatmap Prediction
C
C/2
1
2C
4C
b'
b'
c'
e
f
a: 3x3 Conv Stride-2, 3x3 Conv, 3x3 Conv, Dropout
a': 3x3 Conv Stride-2, 3x3 Conv with dilation, 3x3 Conv with dilation, Dropout
a'': 3x3 Conv with dilation, 3x3 Conv with dilation, 3x3 Conv with dilation, Dropout
b: Concat, 1x1 Conv, 3x3 Conv Transposed, 3x3 Conv, 3x3 Conv, Dropout
b': Concat, 1x1 Conv, 3x1 Conv Transposed, 3x1 Conv, 3x1 Conv, Dropout
c: Concat, 1x1 Conv, 3x3 Conv Transposed
c': Concat, 1x1 Conv, 3x1 Conv Transposed
d: 3x3 Conv, 3x3 Conv, 3x3 Conv Softmax
e: 3x1 Conv
f: 3x1 Conv Sigmoid

FIGURE 2. Model architecture with Item Heatmap prediction. Semantic Segmentation decoder performs per-pixel (K+1)-wise classification (K tags and 1 'background' class). Prediction of the Item Heatmap decoder is a 1D signal representing the localization of predicted Line Items.

pass through this decoder. These feature maps are converted from the shape (H', W', C') to $(H', 1, C')$ by average pooling across the full horizontal dimension.

The first part of the architecture of the Line Item detection decoder is similar to the segmentation decoder, with the only difference being that the width of the feature maps is pooled to 1. In the following subsections we present two methods for vertical region detection on such feature maps. The first method is a modification of the common 2D anchor classification used in [10], [20]. The second one is a novel approach that eliminates the need of anchor-classification-based techniques.

D. 1D ANCHORS

The most popular object detection approach in computer vision is the anchor-based method, which was introduced and explained in detail in [10]. We propose modifying this method to make it suitable for vertical region detection objective. For this purpose, we define a set of 1D vertical anchors of fixed lengths. The idea is the same as in [10], but for the vertical pixels where the Line Items are localized we perform a classification on this set of 1D anchors. In addition, similar to [10], another head in the network can perform a regression on the anchor height and its center coordinate to further tune the classified anchor for the detected object.

This method has the advantage of being similar to an existing well-known approach. The downside is that we need to have a set of predefined anchors for Line Items, which have an ambiguity in their definition, as was mentioned in Section I-A and in [1].

E. ITEM HEATMAPS

In this subsection, we introduce a novel method for vertical region detection using 1D item heatmaps. Unlike anchor classification, we predict the probability of each pixel being the center of the Line Item's vertical region. We encode this probability with a 1D Gaussian heatmap.

For each Line Item the ground truth is constructed by scaling the Gaussian PDF in such a way that it reaches its maximum value in the center of the Line Item and goes to zero on its boundaries. The ground truth of the Line Item detection branch becomes a 1D array that contains values between 0 and 1 inside the Line Item regions and has 0's outside of those regions. Figure 1 illustrates an example of ground truth heatmap. A similar idea was previously used in [11] for text detection in natural images. Gaussian heatmaps were used to encode text character region scores and character affinity maps.

This technique requires a decoding algorithm for reconstructing item boxes from predicted heatmaps at the inference time. As the output is a 1D signal that contains bell-shaped regions, we can apply signal processing techniques to find the peaks of the signal. Each peak will correspond to the center of one of the Line Items. After finding the centers, the next step is to find the item boundaries, which are the tails of the corresponding peak. For this, we can descend from the peak in both directions until reaching a region with an increasing or constant heatmap value. Depending on the quality of the predictions, some signal smoothing may help process the item heatmaps better.

The loss term for this branch is discussed in Section II-F.

This 1D heatmap approach has the advantage of being very flexible with respect to the locations and sizes of the predicted regions. In addition, it does not require predefined anchors. The necessity of an algorithm for decoding the vertical regions from the predicted heatmap can be considered as its disadvantage.

F. LOSS

The training loss is comprised of the following terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{seg\_CE}} + \mathcal{L}_{\text{seg\_DICE}} + \alpha * \mathcal{L}_{\text{reg}} \tag{2}$$

$\mathcal{L}_{\text{seg\_CE}}$ is weighted cross entropy loss where the weight of class c is defined as:

$$w_c = \left(\frac{N_{max}}{N_c}\right)^{se} \tag{3}$$

where $N_{\max}$ is the number of samples in the most frequent class, N_c is the number of samples in class c, and se is a smoothing exponent which is a hyperparameter (smaller than 1).

$\mathcal{L}_{\text{seg\_DICE}}$ is the dice loss [12], which is the average of per-class dice losses. Dice loss for class c is defined as

$$\mathcal{L}_{\text{seg\_DICE\_c}} = -\frac{2\sum_i^N p_i g_i}{\sum_i^N p_i + \sum_i^N g_i} \tag{4}$$

where N is the number of pixels in the output map, p_i is the predicted probability of class c on the i-th pixel, g_i is the ground truth binary mask of class c. We found that the addition of dice loss had a significant positive effect on training.

$\mathcal{L}_{\text{reg}}$ is the loss of the Line Item detection branch. For 1D anchor classification this loss term is the straightforward modification of object detection loss with 2D anchors [10], [20]. For 1D heatmap method, the choice of this loss term is between Mean Square Error (MSE) loss and Binary Cross Entropy (BCE) loss between the predicted and ground truth heatmaps. We chose the BCE loss, because it produced slightly better results than the MSE loss.

Hyperparameter α controls the relative scale of the segmentation and regression branch losses.

III. DATA

Our in-house invoice dataset consists of 8.7k scanned multilingual invoices. Each document in the dataset underwent an initial manual labeling by one data annotator, followed by a review by two other data annotators to clean and verify the labels. In the initial stages of the annotation process the labeled invoice was additionally reviewed by a data scientist. The purpose was to capture the annotation patterns that could hurt the model training and to redefine the annotation guidelines accordingly. This multistage pipeline ensured the high quality of the dataset.

We used Fasttext Language Identification model to estimate the language distribution in the dataset, . According to it, 66% of the invoices are in English, 32% in French, and the remaining 2% are in Italian, Spanish, German, etc. We assigned 7k samples for training and 1.7k for the test set. Similar to [1], we ensure that vendors from the train set do not appear in the test set. This makes the reported metrics a good estimate of how the model will perform on unseen invoice layouts.

IV. EVALUATION METRICS

A. TAG PREDICTIONS

Following [1], we use a metric similar to word error rate (WER) [13], as it reflects the amount of manual work that is automated by our information extraction system. It is calculated for each tag separately based on the entire test set by

$$1 - \frac{\#[\text{ insertions }] + \#[\text{ deletions }] + \#[\text{ modifications }]}{N} \tag{5}$$

where N denotes the number of occurrences of the tag in the test set. We call this metric Tag Error Rate (TER).

Another metric of interest is the word-level F1 score for each tag. It is easier to calculate and track during the training. In the Experiments section we report both metrics for evaluation of tag predictions.

The calculation of TER and F1 is performed at the word-level, whereas the prediction of semantic segmentation is pixel-level. We obtain the word-level tag prediction from its pixel-level tag predictions by majority voting.

B. LINE ITEM PREDICTIONS

Generally, metrics such as MAP or F1 score at certain Intersection over Union (IoU) thresholds are used for object detection. As argued in Section I-A, these metrics do not properly reflect the quality of Line Item predictions. The reason is that the span of the predicted vertical region for Line Item can vary without affecting the correct item-level tag grouping. Therefore, the evaluation metrics of the Line Item prediction branch should not heavily depend on region boundaries.

With this in mind, we define a new metric for Line Item prediction and call it *mean coverage*. First, let us define the *coverage* for a single Line Item:

$$C_{gt,pr} = \frac{|S_{gt} \cap S_{pred}|}{|S_{gt} \cup S_{pred}|}$$

where S_{gt} and S_{pr} represent the sets of words with item-level tags inside the ground truth region and the predicted region. It is evident that insertion of ‘background’ words in the predicted or ground truth regions does not affect this metric, since these sets do not take those words into account. To calculate the coverage between the predicted and ground truth boxes for a multi-item invoice, we must find a mapping between them. We iteratively construct the mapping M between N_{gt} Line Items ground truths and N_{pr} predictions, by taking pairs with the highest positive coverage in each step. All ground truth Line Items that do not have their corresponding predictions are mapped to empty prediction $\emptyset$. Similarly, if a prediction is not a pair for any ground truth Line Item, we assign that prediction to empty set $\emptyset$.

We define $S_\emptyset = \emptyset$. After that, we can define the *mean coverage (MC)* score for a document with multiple Line Items. It is simply the *coverage* score averaged over all Line Items of the document,

$$MC = \frac{1}{|M|}\sum_{i \in M} C_{i,M(i)}$$

where $|M|$ is the cardinality of the mapping, which is equal to $\max(N_{gt}, N_{pr})$. The mean coverage for a dataset is the average of the MC over all the documents in the dataset.

The mean coverage score is intuitive, and a higher MC always indicates better quality of Line Item predictions.

In the experiments section we use *mean coverage* metric for the evaluation of Line Item prediction.

V. EXPERIMENTS AND RESULTS

In this section, we present the results of our experiments and ablation studies. To the best of our knowledge, this is the first time that a 1D approach is used for Line Item detection. Additionally, there are no open-source datasets that have annotated Line Items bounding boxes. This makes it difficult to compare our method against other papers. Hence, we report the comparative results of the methods discussed in this paper on our in-house dataset only.

All metrics below are the averages of five runs with different seeds.

A. IMPLEMENTATION DETAILS

We have implemented the model architecture in PyTorch. The implementation supports both character-level one-hot input grid (chargrid-net), as in [1], and tokengrid (tokengrid-net). We do not use contextualized BERT embeddings, which is a time-consuming and computationally intensive operation at inference time. Instead, we use static token embeddings of BERT model with 'base' architecture [14] trained on multilingual corpus [24]. This produces good metrics and is sufficient for comparative analysis of the different methods described above.

We create tokengrid representation of invoices on the original size of the invoice and then downsample it to a size of 336×256 using nearest neighbor interpolation. We use Adam optimizer [15] with a weight decay of 1e-5 and initial learning rate of 0.001, which is exponentially decayed every epoch by a factor of 0.91. We also use spatial dropout [23], with a probability of 0.1. For the loss parameters, we set $\alpha = 3$, and $se = 0.15$. We use a batch size of 4 with gradient accumulation of four steps which is equivalent to a batch size of 16.

B. RESULTS

First, we show the advantage of tokengrid-net over chargrid-net (see Table 1). Both models are trained using 1D item heatmap approach. As expected, tokengrid-net significantly outperforms chargrid-net. In addition, concatenating both token-level embeddings and character one-hot encodings in the input grid does not produce better results than tokengrid alone.

Table 2 presents a comparative analysis of models trained with 1D anchors and item heatmap approach. As can be seen from the table, the two models have very similar TER and Macro F1 scores. Also, the item mean coverage metric does not differ significantly, which means that they have a comparable Line Item detection quality. At the same time, we can see that the Line Item detection F1 score at the IoU threshold of 0.5 is very different. This confirms the argument that the F1 score for Line Item detection is not optimal because of the ambiguity of Line Item ground truth definition.

TABLE 1. Chargrid-net vs Tokengrid-net vs combined model metrics.

Input Type	Mean TER	Macro F1	Mean Coverage
Chargrid	85.13 % ± 0.08	90.3 % ± 0.11	93.41 % ± 0.04
Tokengrid	88.31 % ± 0.1	92.82 % ± 0.01	93.91 % ± 0.25
Chargrid + Tokengrid	88.11 % ± 0.27	92.75 % ± 0.19	93.6 % ± 0.92

TABLE 2. Tokengrid-net trained with 1D anchors vs with item heatmap approach.

Method	Mean TER	Macro F1	Mean Coverage	Item F1 at 0.5
1D Anchors	88.26% ± 0.12	92.55% ± 0.08	93.34% ± 0.26	92.72% ± 0.69
Item Heatmap	88.31% ± 0.1	92.82% ± 0.01	93.91% ± 0.25	83.76% ± 0.33

We report the TER metrics for each tag for tokengrid-net in Table 3. The per-tag metrics are almost the same across the different methods discussed here, because they share the same segmentation branch architecture.

C. ABLATION ANALYSIS

1) DICE LOSS EFFECT

Dice loss improves the segmentation metrics by 1.1% on average, as can be seen in Table 4. Moreover, the variance of metrics of five runs is also significantly decreased when dice loss is used.

2) ITEM DETECTION, 1D VS 2D

In our architecture, we average-pool across the full horizontal dimension in Line Item detection branch. This has the additional benefit of reducing the number of parameters and accelerating computation. Our best model with 1D detection has 20.5M trainable parameters overall (text embedding parameters are not counted because they are not trained). For this model, the detection branch has ~1.56M trainable parameters, compared to ~3.62M in the 2D approach in [1]. In terms of FLOPs, the gain is even more significant - 0.15 GFLOPs vs 39.8 GFLOPs. Additionally, with our proposed method, the training process speeds up by ~20% without affecting the metrics.

D. THINGS THAT DID NOT HELP

1) CONCATENATING DOCUMENT IMAGE

Concatenating the RGB image of the invoice to tokengrid or chargrid representation had little to no effect on final metrics.

2) CONCATENATING DOCUMENT LINES

Many invoices have vertical and horizontal visual lines that make them more readable for humans. We extracted these lines using classical image processing techniques. The concatenation of the grayscale image of these lines to the tokengrid had no significant effect on the metrics.

TABLE 3. Per-tag metrics of tokengrid-net.

Tag	TER	Tag	TER	Tag	TER
VENDOR ADDRESS	84.38% ± 0.58	IMPORTER NAME	87.64% ± 0.52	PRODUCT DESCRIPTION	83.8% ± 0.21
VENDOR NAME	78.84% ± 0.48	INSURANCE	92.96% ± 0.73	TOTAL	94.56% ± 0.55
FREIGHT	93.24% ± 0.44	INVOICE DATE	91.2% ± 1.0	TOTAL ITEM	91.14% ± 0.3
IMPORTER ADDRESS	87.07% ± 0.23	INVOICE NUMBER	85.33% ± 0.54	UNIT PRICE	89.63% ± 0.46

TABLE 4. Dice loss effect.

Model - Loss	Mean TER	Macro F1	Mean Coverage
Tokengrid-net (CE)	87.23 % ± 0.19	91.73 % ± 0.11	93.86 % ± 0.48
Tokengrid-net (CE+Dice)	88.31 % ± 0.1	92.82 % ± 0.01	93.91 % ± 0.25

3) UPSAMPLING MULTI-ITEM INVOICES

Following [1], we oversampled invoices with more than three Line Items during training. However, this did not improve our results.

4) AUGMENTATIONS

We applied different augmentations to input documents, such as random crops and grid distortions, but did not find them to improve the metrics.

5) OTHER PRE-TRAINED LANGUAGE MODELS

Besides the base BERT [14] embeddings trained on [24], we also experimented with many other pre-trained models, both multilingual and English-French-focused (as most of our invoice data is either in French or in English). However, they either underperformed or produced comparable results compared to the original choice.

VI. CONCLUSION

In this work we proposed two alternative 1D approaches for Line Item detection in invoices. One of the methods is a modification of common anchor-based approach, and the other is a novel approach that uses heatmaps for region detection. The suggested methods utilize the inherent 1D nature of Line Items and solve the issue of their bounding boxes being vaguely defined. The proposed 1D methods are applied for Line Items detection of invoice documents, but can also be used for other 1D region detection problems. We also presented a new metric for evaluation of Line Item detection quality, and reported the effects of several experimental tricks that can be useful for other similar tasks.

REFERENCES

[1] A. R. Katti, C. Reisswig, C. Guder, S. Brarda, S. Bickel, J. Höhne, and J. B. Faddoul, "Chargrid: Towards understanding 2D documents," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4459–4469, doi: 10.18653/v1/d18-1476.

[2] P. Zhang, Y. Xu, Z. Cheng, S. Pu, J. Lu, L. Qiao, Y. Niu, and F. Wu, "TRIE: End-to-end text reading and information extraction for document understanding," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 1413–1422.

[3] 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)*, Nov. 2017, pp. 406–413.

[4] E. F. T. K. Sang and J. Veenstra, "Representing text chunks," in *Proc. 9th Conf. Eur. chapter Assoc. Comput. Linguistics*, Bergen, Norway, 1999, pp. 173–179.

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

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

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

[8] 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.

[9] M. Kerroumi, O. Sayem, and A. Shabou, "VisualWordGrid: Information extraction from scanned documents using a multimodal approach," 2020, *arXiv:2010.02358*.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[11] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9357–9366.

[12] M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal, "The importance of skip connections in biomedical image segmentation," in *Deep Learning and Data Labeling for Medical Applications*. Cham, Switzerland: Springer, 2016, pp. 179–187.

[13] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 4839–4843.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, Minneapolis, Minnesota, vol. 1, 2019, pp. 4171–4186, doi: 10.18653/v1/N19-1423.

[15] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[16] Y. Xu, Y. Xu, T. Lv, L. Cui, F. Wei, G. Wang, Y. Lu, D. Florêncio, C. Zhang, W. Che, M. Zhang, and L. Zhou, "LayoutLMv2: Multi-modal pre-training for visually-rich document understanding," 2020, *arXiv:2012.14740*.

[17] S. Appalaraju, B. Jasani, B. U. Kota, Y. Xie, and R. Manmatha, "Doc-Former: End-to-end transformer for document understanding," 2021, *arXiv:2106.11539*.

[18] W. Lin, Q. Gao, L. Sun, Z. Zhong, K. Hu, Q. Ren, and Q. Huo, "ViBERT-grid: A jointly trained multi-modal 2D document representation for key information extraction from documents," 2021, *arXiv:2105.11672*.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017, *arXiv:1706.03762*.

[20] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2016, *arXiv:1612.08242*.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

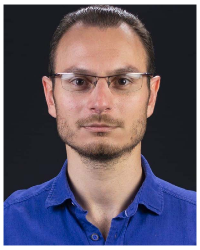
[22] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2016. [Online]. Available: http://arxiv.org/abs/1511.07122

[23] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," 2014, *arXiv:1411.4280*.

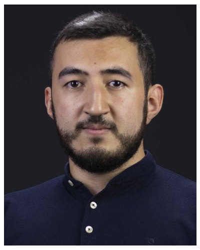
[24] Wikimedia Foundation. *Wikimedia Downloads*. Accessed: 2018. [Online]. Available: https://dumps.wikimedia.org

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," 2017, *arXiv:1703.06870*.

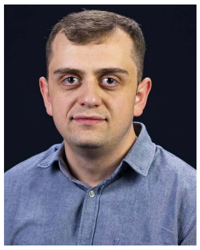
[26] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," 2018, *arXiv:1803.01534*.



ARSEN YEGHIAZARYAN received the Ph.D. degree in theoretical physics from the Yerevan Physics Institute (YerPhI), in 2012. He is currently the AI Team Lead at Portmind. His current research interests include computer vision, NLP, and classical machine learning theory.



KHACHATUR KHECHOYAN is currently pursuing the M.Sc. degree in applied statistics and data science with Yerevan State University. He is employed as a Machine Learning Specialist at Portmind. His current research interests include computer vision, NLP, and optimization methods.



GRIGOR NALBANDYAN is currently pursuing the M.S.C.S. degree with the Technical University of Munich. He is employed as a Machine Learning Specialist at Portmind. His research interests include computer vision, natural language processing, and sparse systems.

SIPAN MURADYAN received the bachelor's degree in computer sciences from the American University of Armenia (AUA), in 2017. Currently, he is a Senior Machine Learning Specialist at Portmind. His current research interests include NLP, machine learning in audio, and computer vision.