Wiley
Applied Computational Intelligence and Soft Computing
Volume 2024, Article ID 7599415, 15 pages
https://doi.org/10.1155/2024/7599415

WILEY

Research Article

Document Information Extraction: An Analysis of Invoice Anatomy

Mouad Hamri, Maxime Devanne, Jonathan Weber, and Michel Hassenforder

IRIMAS, University of Haute-Alsace, Mulhouse, France

Correspondence should be addressed to Mouad Hamri; mouad.hamri@uha.fr

Received 11 February 2024; Revised 19 May 2024; Accepted 21 May 2024

Academic Editor: Ahmad Al-Omari

In this paper, we present a new approach of document information extraction by studying the document anatomy where we investigated the possible variants and forms it could have for each document component. To overcome the lack of publicly available document datasets, we used a generated invoice database where we conceived 9 different templates and generated 100 samples for each one where the documents were annotated automatically during the generation process. We analysed the following invoice components: dates block (invoice date and invoice due date), address block, amounts block (tax-free amount, tax amount, and total amount), and lines block (lines table) by investigating the impact of training our model on various block variants. We conducted several experiments where we compared the results obtained when we tested on templates that included variants not encountered during the training phase versus when we introduced them to the training dataset. This allowed us to analyse the improvement in results after adding these previously unseen variants. The obtained results have shown that the model generalises better when trained on a large variety of cases and achieves remarkable performance. We conducted experiments on various models to highlight the model-agnostic character of our proposed approach. This methodology allows to have great performance, even with models that have significantly fewer parameters, especially in comparison to recently published models with millions of parameters.

1. Introduction

Document information extraction is a crucial task in many industries, such as finance, healthcare, and legal services. It involves automatically extracting structured information from unstructured or semistructured documents, such as invoices, contracts, and medical records. The automation of this task can save time and reduce errors that can be produced when the processing is manual, and it allows to handle a larger number of documents. However, this task is challenging due to the diversity and complexity of document formats which affect the performance on unseen documents.

In the 90s, the adopted methods were based on heuristics and rule patterns which failed to generalise on unseen templates and was requiring a huge effort and time to record the rules and to train the models. Deep learning methods, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), transformers, and graph neural networks (GNNs), have shown promising results in document information extraction tasks and could generalise better for unseen templates.

However, the development of document information extraction systems is often limited by the availability of large and high-quality datasets for training and evaluating the models which make it difficult for researchers to develop and compare new methods. To address this issue, we generated a custom dataset of 9 invoice templates inspired from real-life invoices where we tried in our choices to be as exhaustive as possible to cover most of the components' shapes and cases as we will describe in Section 3.

In addition to that, we propose a novel approach for document information extraction task that utilises document anatomy analysis in order to include an additional structural information as we assume that this addition will help the model to recognize easily the fields of the invoice analysed blocks; we considered 4 invoice components: dates

block, address block, amounts block, and lines block and we tried to imagine the relevant variants of these blocks.

The experiments' results showed that this approach improves significantly the ability of prediction of the model for unseen templates and demonstrates that this approach can lead to a framework that can be used to improve training strategies of other models and datasets.

We also demonstrated that our approach is model agnostic. This independence from the underlying model allows achieving impressive results using models with far fewer parameters, especially when compared to recently published models that have millions of parameters.

We organized the paper as follows: Section 2 presents the background and the related works, Section 3 presents our approach, Section 4 describes the architecture and the used model, Section 5 describes the experiments and their results, and in Section 6, we conclude and provide some perspectives.

2. Related Works

In the 90s, document information extraction relied on rule-based algorithms [1–4] that required significant human intervention and often failed to generalise to new documents. However, advances in machine learning and deep learning have enabled the development of more powerful and effective architectures that can extract structured information from documents using both their text features and document structure.

Early approaches using traditional neural networks, such as multilayer perceptrons (MLP) [5], convolutional neural networks (CNNs) [6, 7], and natural language processing (NLP) [8] recurrent networks (RNN, LSTM, and so on) [9–11], have improved the results by incorporating visual and textual features, but they still had limited capabilities in handling this task as they were partially capturing the document features (textual features, visual features, and document structure) which led to a poor embedding representation.

The integration of visual, text, and spatial features has been adopted by many authors in various works [12–19]. This combination of features, including the bounding boxes coordinates, was intended to provide a comprehensive representation of the document. However, the limitation of this approach lies in the independent processing of these representations and their subsequent concatenation, but the overall performance of these models is still remarkable.

The transformer architecture, introduced by Vaswani et al. [20, 21], was a breakthrough, leveraging the attention mechanism to achieve better results and reduce computation time. The BERT (bidirectional encoder representations from transformers) [22–24] was later introduced as a pretrained representation that could be fine-tuned for various downstream tasks. Microsoft's LayoutLM, LayoutLMv2, and layoutLMv3 [25–28] models also used the transformer architecture and were trained on a huge number of documents in their pretraining stage which allowed them to perform state-of-the-art results in the document information extraction task. The authors of [29] introduced the "ERNIELayout," a transformer-based architecture model combining textual, visual, and layout features to produce a rich representation; the model was pretrained on 4 different tasks and then fine-tuned on multiple tasks such as document information extraction. In [30], the authors introduced "LiLT (language-independent layout transformer)" in which the text and the layout information are fed to two parallel transformer blocks, but a cross-modality is performed using the "bidirectional attention complementation mechanism (BiACM)." A self-supervised pretraining transformer was proposed in [31] where the authors combined a discrete variational autoencoder (dVAE) with a transformer backbone using large-scale unlabeled text images. The paper [32] proposed an OCR-free model based on an encoder-decoder architecture where the model input is generated as a token sequence giving a prompt (as in GPT-3 [33]).

Recently, graph neural networks (GNNs) have gained popularity for their ability to address problems that traditional models cannot handle efficiently. GNN models have been widely used to extract document information [34–40]. In two separate works [41, 42], authors proposed two different GNN models, the first based on Chebyshev graph convolutional neural networks [43] and the second on the graph attention network (GAT) [44], where they stacked multiple layers in both models followed by a feed-forward classifier. In [45], a multimodal graph attention-based model called GraphDoc was proposed where the inputs of a textual and visual encoders are fed to a multigraph attention layers to have by the end a rich embedding; the model was pretrained on a large document dataset on various document understanding tasks and then fine-tuned on other tasks such as document information extraction.

GNNs offer an effective way to represent documents as they retain their structure where documents are typically represented as nodes, with their text, and the spatial relationships between them are represented as edges (neighbourhoods).

After this review of the current state of the art in document information extraction, we can see that that there is a trend toward models with an extensive number of parameters, often reaching millions. While this can lead to an impressive performance, these models require an enormous amount of computational capacity, and even with advanced infrastructures, the time required for training can be huge. Furthermore, these models need massive datasets for their training and they are generally using private datasets. These challenges highlight the need for more efficient methodologies that can achieve comparable performance with smaller models and datasets having a reduced size. In response to these challenges, our work focuses on an in-depth analysis of the anatomy of invoices. This analysis allowed us to construct a custom dataset in which every component or block of an invoice is represented with its various variants which allowed us to achieve excellent results using just a few hundred invoices for training. In addition, our approach is model agnostic, meaning it can be applied regardless of the underlying model which allowed us achieving great results even with models that have only few thousand parameters.

3. Invoice Anatomy

3.1. Approach. In this section, we will explore the different fields that we extracted from the invoices and the main blocks that we analysed in our study.

The primary objective in extracting information from invoices involves identifying and extracting the most relevant fields. Upon examining numerous real-life invoices, we have determined the following essential fields that our model will be trained on: supplier name, supplier address, invoice number, purchase order number, invoice date, invoice due date, invoice lines (for each line we predict line description, line quantity, line unit of measure, line unit price, line tax, and line subtotal), total free-tax amount, tax amount, and total amount.

As a result of this analysis, we generated 9 different templates that we have named t_i where $1 \leq i \leq 9$, and we can see in Figure 1 the fields of an invoice belonging to t_7 template.

Basically, an invoice is generated by software following general, legal, formal, and aesthetic rules. An invoice can be considered as a free composition of several blocks; some blocks can be laid differently in a page and can have special internal layout or some content can be written differently. Our goal is to cover many (all) of them in our generated invoice dataset.

In our approach, we considered that by analysing these components and insuring that the dataset contains different patterns for each component will lead the model to achieve higher accuracy and could be a better approach for invoice information extraction. As the number of possible fields and components of an invoice is very large (the number of invoice templates produced by vendors' software is huge), we limited our study to 4 blocks:

(1) Dates block: the invoice date and due date
(2) Address block: the supplier address (number, street, zip code, and city)
(3) Lines block: the invoice lines with their different columns
(4) Totals block: the total free-tax amount, tax amount, and total amount

Figure 2 shows 4 different blocks of an invoice belonging to the t_2 template, the dates block is composed of fields 4 and 5, the field 2 represents the address block, fields 7 to 12 form the lines block, and fields 13 to 15 represent the total block (see Figure 1).

The various blocks within an invoice can be positioned in different regions and exhibit a range of shapes. Through an analysis of real-life invoices, we have identified a collection of variants for each block, covering a multitude of possible scenarios. This information will be detailed in Subsection 3.2. By training the model on a dataset that covers all of these block variants, we anticipate a significant improvement in the accuracy of our predictions.

As mentioned earlier, we developed 9 different invoice templates (https://github.com/mouadhamri/invoicedataset) (t_i where $1 \leq i \leq 9$) inspired by real-life customer invoices (Figure 3) where we used a sample database to generate the invoices as this database contains suppliers with their addresses and products with their information. For each invoice, we generated a dataset of 100 invoices and we used tesseract OCR (https://github.com/tesseract-ocr/tesseract) to get the texts bounding box, and we generated the invoice image and the annotation file in csv format that contains for each text, the bounding box coordinates, the text value, and the label (the text class among the field list).

We chose carefully the 9 templates to cover different variants and patterns of the 4 blocks as will be explained in Section 3.2.

The approach proposed can be adapted to any other set of fields and any other blocks can also be analysed.

3.2. Invoice Blocks Analysis. We explain in this subsection our approach in using the invoice components analysis, which consists in breaking down the invoice into parts and analysing each part separately. The four main components we have considered were dates block, address block, lines block, and amounts block. In this subsection, we will explore how analysing each of these components can be used to improve the accuracy of prediction in invoice information extraction.

3.2.1. Dates Block. The dates block of the invoice contains the invoice date and the invoice due date. By analysing the date block in real-life invoices, we have identified the following cases (Table 1 shows the dates block types of each invoice template, and Figure 4 illustrates examples of the different dates block variants):

(1) BD1: the two dates are aligned horizontally
(2) BD2: the two dates are aligned vertically
(3) BD3: the two dates are disjoint (date in the header and due date in the footer)
(4) BD4: only the date is present (the due date is absent)

3.2.2. Address Block. The address block contains the supplier address information: street, number, zip code, and country. By analysing the address block in real-life invoices, we have identified the following cases (Table 2 shows the date; Table 1: date block type of each invoice template address type of each invoice template, and Figure 5 illustrates examples of the different address block variants):

(1) BA1: there is one address block (supplier address)
(2) BA2: there are two address blocks (supplier address and customer address)
(3) BA3: there are three address blocks (supplier address, customer shipping address, and customer invoicing address)

3.2.3. Lines Block. The lines block is composed of the invoice line fields which are in our case the description, the unit amount, the unit of measure, the tax, and the subtotal

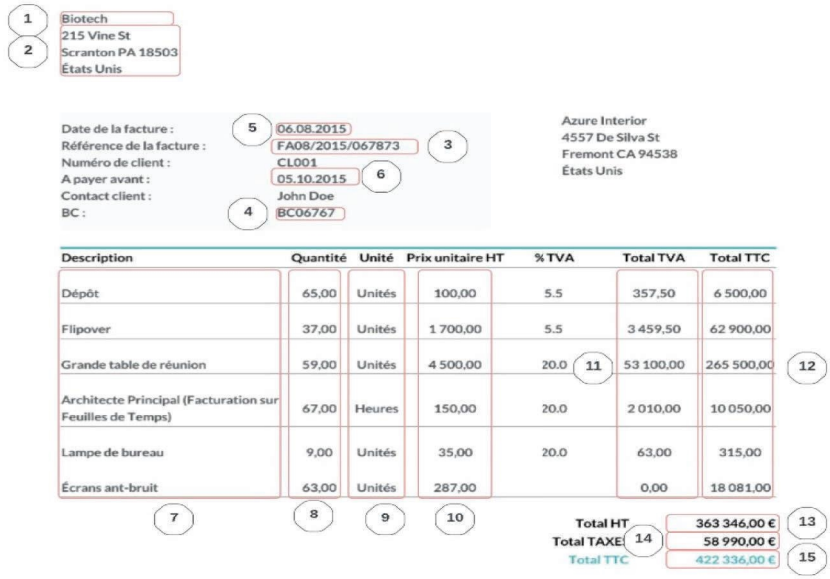

FIGURE 1: The fields (that our model is trained on) for an invoice belonging to t_7 template, where 1 = supplier name, 2 = supplier address, 3 = invoice number, 4 = purchase order number, 5 = invoice date, 6 = invoice due date, 7 = line description, 8 = line quantity, 9 = line unit of measure, 10 = line unit price, 11 = line tax, 12 = line subtotal, 13 = total free-tax amount, 14 = tax amount, and 15 = total amount.

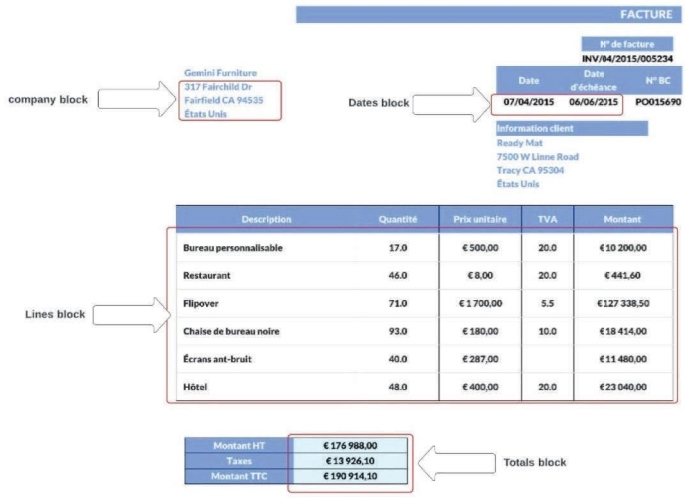

FIGURE 2: An illustration of the company, dates, lines, and amounts (totals) blocks for an invoice belonging to t_2 template.

amount. The main difference between the templates regarding this block is the number of columns where we have the following cases (Table 3 shows the lines block type of each invoice template, and Figure 6 illustrates examples of the different lines block variants):

(1) BL1: lines with 2 columns
(2) BL2: lines with 4 columns
(3) BL3: lines with 5 columns
(4) BL4: lines with 7 columns
(5) BL5: lines with 8 columns

3.2.4. Amounts Block. The amounts or totals' block contains the invoice amounts: free-tax amount, tax amount, and total amount. By analysing the amounts block in real-life invoices, we have identified the following cases (Table 4 shows the amounts block type of each invoice template, and Figure 7 illustrates examples of different amounts block variants):

(1) BAM1: the block contains only the total amount
(2) BAM2: the block is vertically aligned on the right of the invoice
(3) BAM3: the block is vertically aligned on the left of the invoice

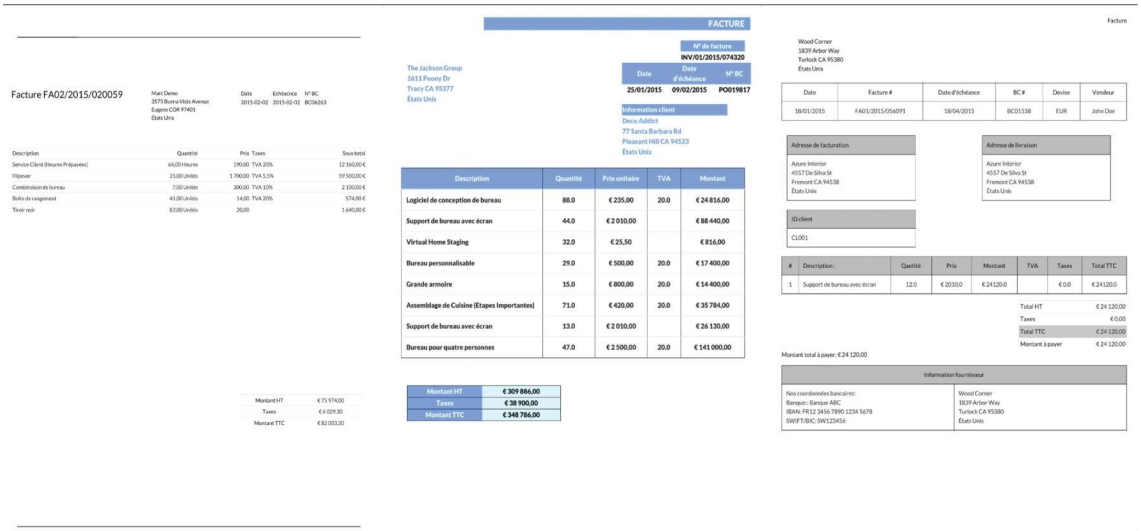

FIGURE 3: Overview of invoice templates t_1, t_2, and t_3 (from left to right).

TABLE 1: Dates block type of each invoice template.

Block	Invoice template
BD1	t_1, t_2, and t_3
BD2	t_6, t_7, and t_8
BD3	t_4 and t_5
BD4	t_9

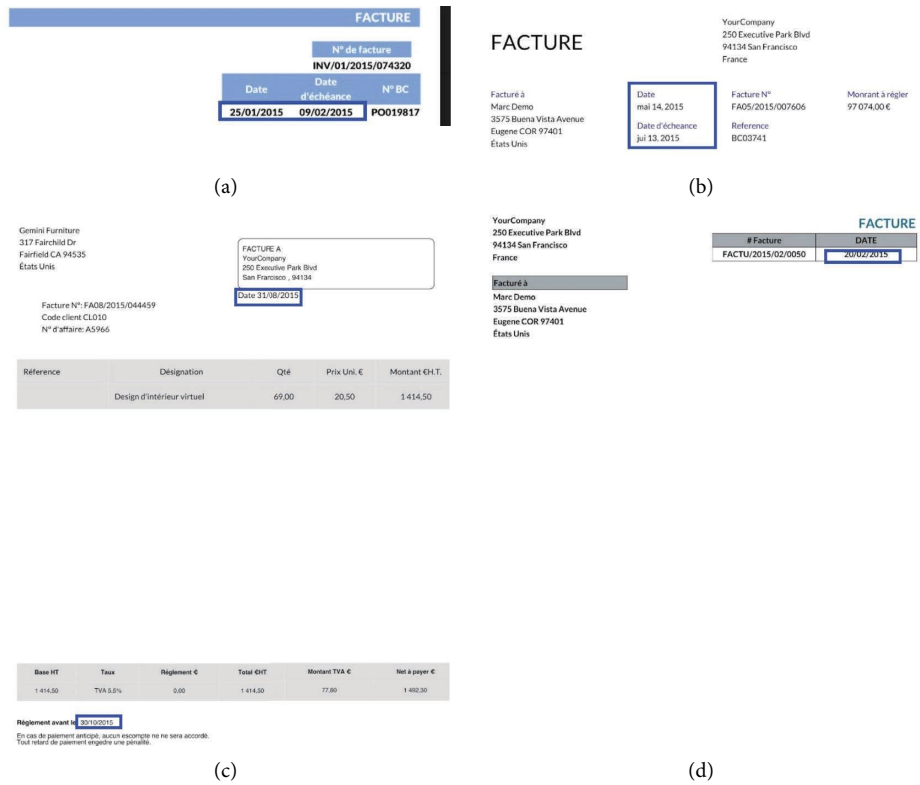

FIGURE 4: Examples of the different dates block variants: (a) BD1, (b) BD2, (c) BD3, and (d) BD4.

Table 2: Address block type of each invoice template.

Block	Invoice template
BA1	t_1
BA2	t_2, t_4, t_5, t_6, t_7, and t_9
BA3	t_3 and t_8

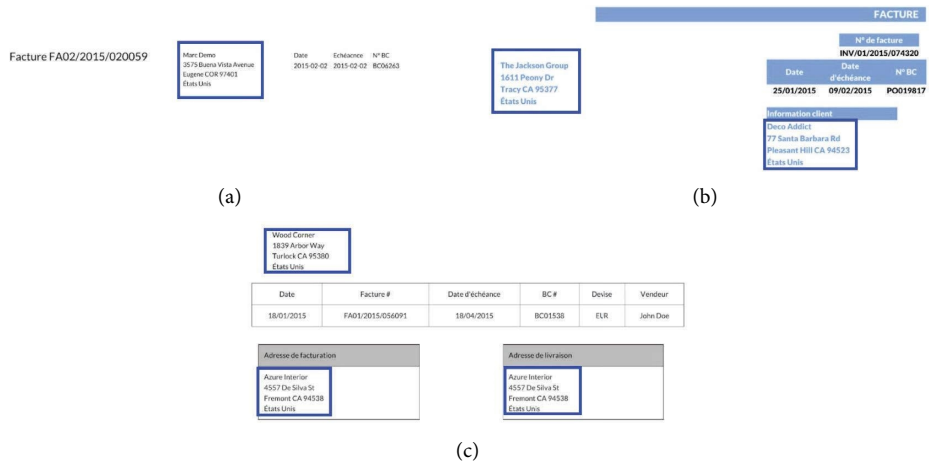

Figure 5: Examples of the different address block variants: (a) BA1, (b) BA2, and (c) BD3.

Table 3: Lines block type of each invoice template.

Block	Invoice template
BL1	t_9
BL2	t_4, t_6, and t_8
BL3	t_1, t_2, and t_5
BL4	t_7
BL5	t_3

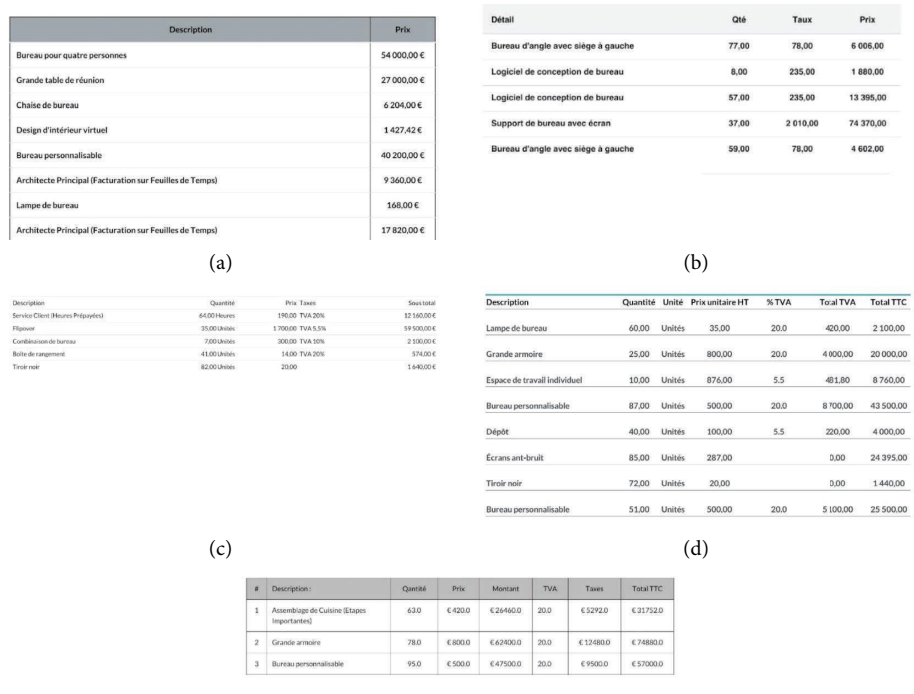

Figure 6: Examples of the different lines block variants: (a) BL1, (b) BL2, (c) BL3, (d) BL4, and (e) BL5.

Table 4: Amounts block type of each invoice template.

Block	Invoice template
BAM1	t_9
BAM2	t_1, t_3, t_4, t_6, t_7, and t_8
BAM3	t_2
BAM4	t_5

The t_1, t_2, ..., t_9 (t_i) are the 9 invoice templates already presented.

(4) BAM4: the block is horizontally aligned on the center of the page

In our experiments, we used a GNN model [46] using the rich encoder transformer representation of LayoutLM v2 [27] as we believe that it is a perfect candidate to build an efficient feature vector for the graph nodes as it combines the bounding box, the image, and the text to create the embedding vectors comparing to the only usage of text representation as proposed by [41, 42]. In the next section, we will present the graph construction approach and the model architecture.

4. GNN Model

In this study, we will use the graph neural network (GNN) model introduced in our prior work [46]. The next subsections will outline the construction of the invoice graph and describe the architecture of the model.

4.1. Invoice Graph Construction. For every invoice graph, each node represents a box of the OCR output of the invoice. The edges are formed by linking each node to its 4 neighbours in 4 directions: left, right, up, and down. We added a new constraint by limiting the selection only to the nodes in the previous or next invoice line (see Figure 8). To each node, we assigned a feature vector of dimension 777 by concatenating the following three vectors:

(1) Spatial features (dimension 3): composed of the normalised coordinates of the box center: xcenter/w and ycenter/h((xcenter, ycenter) are the coordinates of the box center while w and h are the width and height of the image, respectively, and the normalised box line number: l_{box}/L (l_{box} is the box line number and L is the document number of lines)

(2) Text features (dimension 6): a normalised vector representing the number of lower, upper, special, alphanumeric, numeric, and space characters in the box text

(3) LayoutLM v2 vector features (dimension 768): the output embedding of the pretrained *LayoutLM V2* base model encoder [27] (https://huggingface.co/docs/transformers/modeldoc/layoutlmv2)

4.2. Model Architecture. Our GNN model is made of 4 stacked graph transformer layers [20] followed by a feed forward layer with a softmax activation action for the classification task. The model pipeline is illustrated in Figure 9.

5. Experiments

5.1. Metrics. The evaluation of the model's performance was based on the F1 score. The F1 score is calculated as the harmonic mean of precision and recall, expressed as F1 = 2 ∗ Precision ∗ Recall/Precision + Recall. Precision and recall are calculated as follows: Precision = TP/TP + FP, where TP it true positives, TN is true negatives, FP is false positives, and FN is false negatives.

We also reported the precision and the recall of each experiment. In the next sections, we will use the notations P and R to denote the precision and the recall, respectively.

The metrics were computed using the classification report function from the sklearn.metrics (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.classificationreport.html) library, with the aggregation method setting the weighted average.

5.2. Experimental Setup. We have ran several experiments to validate the assumption that having a dataset with templates and covering a large number of components types help the model to generalise on unseen templates; we will present in the rest of this section the different experiments and the outcome of each one.

For all the experiments ran, we used 5 different seed values, and for each seed value, we ran a K-fold cross validation ($K = 5$) to validate the reproducibility of the results.

In order to show that the proposed methodology is model agnostic, we conducted our experiments utilising our GNN model alongside LayoutLMv2 [27] (in the next sections, we will refer to LayoutLMv2 by LMV2) and LiLT [30]. These latter models have millions of parameters (200M [27] for LMV2 and 131M (https://huggingface.co/SCUT-DLVCLab/lilt-roberta-en-base) for LiLT), so given the huge number of experiments to be performed (several hundreds), we opted for limiting their experiments to a maximum of 10 epochs with an early stopping criterion of 5 epochs. While this strategy may not show all the potential results, it aligns with our central objective, which is not a comparative analysis of model performances; instead, our primary goal is to affirm the independence of the proposed method from the underlying model.

5.2.1. Base Experiments. The first series of experiments were run using the generated dataset composed of the templates t_1 to t_9, where we ran 10 experiments.

We ran the first experiment where we mixed all the templates and we created a train/validate/test split with the ratios 80%/10%/10%. Then, we adopted a leave-one-out strategy to run 9 different experiments where we test each time on a template t_i while training and validating using the rest of templates (with a ratio of 80%/20% for the train/validate dataset).

Table 5 presents the scores of the "base experiment." As we performed K-fold cross validation and ran the experiments over different seeds, we report the mean μ and the standard variation σ of the F1 metric.

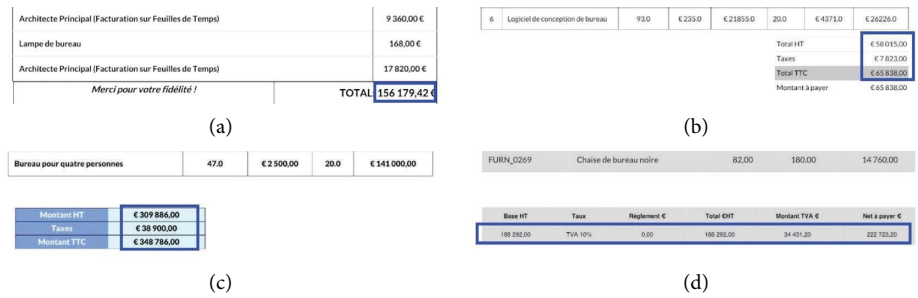

FIGURE 7: Examples of the different amounts block variants: (a) BAM1, (b) BAM2, (c) BAM3, and (d) BAM4.

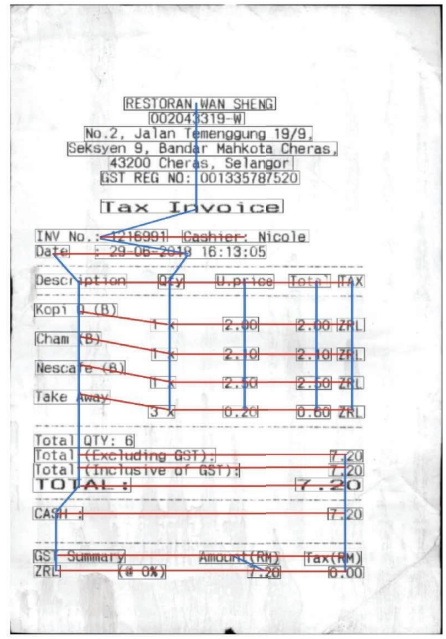

FIGURE 8: Example of document graph construction. The black boxes represent the nodes, the blue arrows show the up/down edges, and the red arrows the left/right.

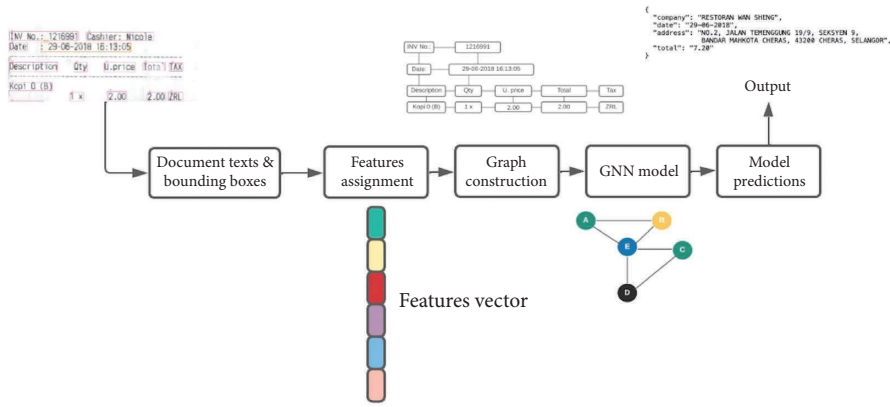

FIGURE 9: Model pipeline.

We can see that when the models were trained on all invoice templates, we got high accuracy in extracting information from the invoices. However, when tested on a specific invoice template and trained on others, all the models struggled to generalise. This can be attributed to the limited exposure of the models to all templates during

Table 5: Results of the "base experiments."

Setup	P	R	F1	μ (F1)	σ (F1)
(a) Our GNN model					
All	**0.9970**	**0.9970**	**0.9970**	**0.9949**	**0.0009**
t_1	0.9153	0.8917	0.8920	0.8226	0.0530
t_2	0.9029	0.8992	0.8971	0.0840	0.0477
t_3	0.8863	0.8517	0.8621	0.8445	0.0135
t_4	0.9221	0.9015	0.9031	0.8260	0.0561
t_5	0.8635	0.8767	0.8620	0.8323	0.0279
t_6	0.8384	0.7825	0.7676	0.7007	0.0329
t_7	0.9318	0.9027	0.9076	0.8580	0.0333
t_8	0.9545	0.9476	0.9411	0.9082	0.0232
t_9	0.6839	0.5769	0.5918	0.5204	0.0417
(b) LMV2					
All	0.9953	0.9952	0.9952	0.9924	0.0017
t_1	0.8850	0.8322	0.8293	0.6181	0.0880
t_2	0.9083	0.8973	0.8960	0.7264	0.0641
t_3	0.8581	0.8095	0.8200	0.6851	0.0712
t_4	0.9086	0.8748	0.8836	0.7021	0.1516
t_5	0.8655	0.8646	0.8591	0.7480	0.0664
t_6	0.7715	0.7465	0.7513	0.6343	0.0746
t_7	0.8306	0.8273	0.8126	0.7005	0.0747
t_8	0.9812	0.9809	0.9809	0.7369	0.1308
t_9	0.7729	0.7626	0.7328	0.6507	0.0507
(c) LiLT					
All	0.9944	0.9944	0.9944	0.9927	0.0011
t_1	0.8489	0.7742	0.7702	0.6262	0.0917
t_2	0.9308	0.9260	0.9234	0.7067	0.1275
t_3	0.8018	0.7280	0.7483	0.6226	0.0586
t_4	0.9373	0.9374	0.9363	0.7474	0.1404
t_5	0.8053	0.7056	0.7155	0.6647	0.0325
t_6	0.7715	0.7465	0.7513	0.4989	0.0715
t_7	0.8675	0.8561	0.8525	0.6687	0.0996
t_8	0.8655	0.8656	0.8595	0.6835	0.0871
t_9	0.7407	0.6713	0.6894	0.5009	0.1187

The setup "All" corresponds to the experiment where all the templates are used in the train/validate/test phase and the setup t_i $(1 \le i \le 9)$ refers to the experiment where the test dataset was the invoice of the template t_i. For all the experiments, we reported the precision, recall, and F1 scores. The bold values are the best results for each measure.

training which led to difficulties in adapting to the new structures and formatting, leading to a drop in performance. This highlights the importance of introducing new templates covering diverse cases of invoice components in order to achieve robust and higher results.

5.2.2. Blocks Experiments. We ran experiments for each block to validate the assumption that training on a dataset with a variety of blocks improves the accuracy even with a rather low number of invoices (100). We followed a strategy where for each block experiment, we train on a dataset that does not contain a variant of the block and then on a dataset that contains it, and we test on a dataset of different templates having this variant. The results obtained have shown (as we will see later) that the introduction of the template containing this variant improved the results significantly.

In the next subsections, we will present the different block experiments with the obtained results.

(1) Dates Block Experiments. Among the generated templates, we have two templates t_4 and t_5, where the date and due date fields are disjoint (Table 1: BD3), to demonstrate the effect of Table 5. The setup "All" corresponds to the experiment where all the templates are used in the train/validate/test phase and the setup t_i $(1 \le i \le 9)$ refers to the experiment where the test dataset was invoices of the template t_i. For all the experiments, we reported the precision, recall, and F1 scores presence of this variant in both the test and training datasets; we ran 3 experiments where in the first one, we tested on the template t_4, in the second one, we tested on the template t_5, and in the last one, we tested on both t_4 and t_5 templates (in all the experiments we trained on the rest of the templates t_i $(1 \le i \le 9)$). We expect to have a lower score for the due date in the last experiment as the training dataset does not have any template having the dates block variant BD3.

We reported in Table 6 that the results of the "All" experiment ran in the previous subsection in addition to the results of 3 other experiments. For the GNN model, the F1 score of the due date was almost perfect (99.51%) in the "All" setup and then it dropped slightly when tested t_4 and t_5 to 95.94% and 97.96%, respectively, but when tested on both the templates t_4 and t_5, we got as expected the lowest F1

Table 6: Results of the "dates experiments."

TD	O P	O R	O F1	DD P	DD R	DD F1
(a) Our GNN model						
All	**0.9970**	**0.9970**	**0.9970**	**1.0000**	0.9903	0.9951
t_4	0.9221	0.9015	0.9031	0.9091	1.0000	0.9594
t_5	0.8635	0.8767	0.8619	1.0000	0.9600	0.9796
t_4 and t_5	0.8405	0.8021	0.8378	0.6218	0.9700	0.7578
(b) LMV2						
All	0.9953	0.9952	0.9952	1.0000	0.9918	0.9959
t_4	0.9086	0.8748	0.8836	1.0000	0.9400	0.9691
t_5	0.8655	0.8646	0.8591	1.0000	**1.0000**	**1.0000**
t_4 and t_5	0.8172	0.7920	0.7940	0.8182	0.4950	0.6168
(c) LiLT						
All	0.9944	0.9944	0.9944	0.9909	0.9820	0.9864
t_4	0.9373	0.9374	0.9363	1.0000	0.6000	0.7500
t_5	0.8053	0.7056	0.7155	0.9104	0.6100	0.7305
t_4 and t_5	0.8122	0.7537	0.7600	0.4796	0.8800	0.6208

The first experiment was tested on the TD (test dataset) containing templates t_4 and t_5, and the second and the third were tested on templates t_4 and t_5, respectively. The training dataset for the first experiment was the set t_i ($1 \le i \le 9$ and $i \ne 4$ and $i \ne 5$); for the second experiment, it was the set t_i ($1 \le i \le 9$ and $i \ne 4$); and for the third one, it was set to t_i ($1 \le i \le 9$ and $i \ne 5$). For all the experiments, we reported the overall and the due date precision, recall, and F1 scores (O = overall and DD = due date). The bold values are the best results for each measure.

score 75.78%. We can see the same trend for LMV2 and LiLT where the "All" F1 was 99.59% and 98.64%, respectively. When LMV2 was tested on t_4, the F1 score dropped slightly 96.91% while scored 100% when tested on t_5; regarding LiLT, the results dropped more significantly to 75% when tested on t_4 and 73.05% when tested on t_5. Finally, the tested experiments on both t_4 and t_5 had as expected the worst scores for both LMV2 (61.68%) and LiLT (62.08%).

The "All" experiment revealed that training the model across all templates produced nearly perfect results. However, a drop was observed when the model was tested against invoices from template t_4, but the performance was still impressive. A similar decline was noticed while testing on template t_5.

These outcomes indicate that even with the slight drop in accuracy when testing on individual templates t_4 and t_5, the model maintained its efficiency due to the presence of a common variant (disjoint dates) in the training dataset. This implies that the exposure to this specific variant during the training phase served to maintain the model's predictive capabilities. However, the F1 score dropped when invoices from both t_4 and t_5 were combined in the test dataset. This drastic decrease in performance shows the determinant effect of the absence of the disjoint dates' variant on the model's capacity to correctly identify due dates. The overall F1 score decreased for t_4 and t_5 experiments, respectively, and eventually decreased more in the final experiment. This trend highlights the fact that the model gets optimal results when trained on all templates and that its performance remains strong when at least one template in the training dataset includes the block variant. However, the lack of this variant leads to a decline in score.

Before ending this subsection, we will present an additional experiment conducted to further highlight the importance of variant analysis. Specifically, within our t_6 invoice dataset, dates are represented in the "Jan 01, 2022" format, where the month is denoted by a three-character representation. We found that the scores associated with these dates were consistently lower than those linked with the remaining 8 templates. As a response to this observation, we designed a new template based on the template t_2 that we called as t_2', wherein the date format was transformed to align with the t_6 style. The experimental results derived from this experiment are shown in Table 7.

For the GNN model, the observed results show a notable improvement, with the date score increasing from 42.58% to 72.49% and the due date score from 27.61% to 47.92%. Simultaneously, the overall score also witnessed a positive trend, moving from 76.76% to 80.54%. For LMV2, both the date and the due date scores increased from 63.49% to 69.54% and from 33.78% to 41.78%, respectively, even if the overall score has slightly dropped from 75.13% to 74.47%. LiLT results indicate improvements in all evaluated metrics. The date score improved from 8.8% to 10.84%, and the due date score increased significantly from 1.40% to 47.30%. Similarly, the overall score went from 61.01% to 62.72%.

These results further solidify our initial assertion that when a model is trained with various variants of a block, it exhibits a more refined ability to generalise unseen templates, thereby enhancing its overall performance.

(2) Address Block Experiments. For the address block, we performed a test on the template t_1 that contains one address block (BA1: Table 2) using as training dataset, the rest of templates among the initially generated 9 templates. As there is no other template having the address block BA1, we created a new template by modifying the address block of the template t_8 to have a unique address block (BA1), and we called this template t_8. We ran a second experiment where we replaced the template t_8 by the new one t_2' where we expect an increase of the accuracy of the address prediction.

In Table 8, we consolidated the results from the "All" experiment along with those of the other two experiments. As anticipated, the scores of the GNN model obtained in the

Table 7: Results of the "character format dates experiments."

TD	O P	O R	O F1	D P	D R	D F1	DD P	DD R	DD F1
(a) Our GNN model									
DS_1	0.8384	0.7825	0.7676	**0.9744**	0.2724	0.4258	0.6806	0.1731	0.2761
DS_2	**0.8424**	**0.8051**	**0.8054**	0.8947	0.6093	**0.7249**	**0.9109**	0.3251	**0.4792**
(b) LMV2									
DS_1	0.7715	0.7465	0.7513	0.5698	0.7168	0.6349	0.6429	0.2291	0.3378
DS_2	0.7653	0.7460	0.7471	0.6092	**0.8100**	0.6954	0.7407	0.2909	0.4178
(c) LiLT									
DS_1	0.6546	0.6069	0.6101	0.8667	0.0466	0.0880	0.2500	0.0073	0.0140
DS_2	0.6590	0.6194	0.6272	0.3158	0.0655	0.1084	0.8364	**0.3297**	0.4730

The two experiments were tested on the template t_6. The TD (training dataset) for the first experiment was the dataset $t\,t_i$ $(1 \le i \le 9$ and $i \ne 6)$ (DS_1) and for the second one t_i $(1 \le i \le 9, i \ne 2$ and $i \ne 6 \cup t_2')$ (DS_2). For all the experiments, we reported the overall, date and due date precision, recall, and F1 scores (O = overall, D = date, and DD = due date). The bold values are the best results for each measure.

"All" setup were the highest ones (overall F1: 99.70% and address F1: 99.70%) and then the scores dropped to 90.90% for the address block and to 89.20% as an overall score when we tested on invoices of the template t_1.

The replacement in the training dataset of the template t_8 by the template t_8' led to a remarkable improvement in the score of the address block from 90.90% to 99.08% while the overall score has slightly increased to 90.92%.

This has also been confirmed for LVM2 and LiLT where the "All" setup had the highest scores (overall F1: 99.42%, address F1: 99.72% and overall F1: 99.44%, address F1: 99.60%, respectively), and then there was a drop when tested on t_1 (overall F1: 82.93%, address F1: 82.61% and overall F1: 77.02%, address F1: 61.15%, respectively). When we replaced t_8', we observed for LMV2 an increase in the address score from 82.61% to 86.40% even if the overall score dropped from 82.93% to 73.17% while the improvement was more spectacular for LiLT as the address score went from 61.15% to 94.62% while the overall score increased from 77.02% to 81.83%.

These experiments confirm our initial assumption that training the model on templates covering the diverse variants of the address block helps the model get a higher accuracy.

(3) Lines Block Experiments. For this block, we ran a first experiment when we tested on invoices of the template t_4 while training on the rest of templates.

As the template t_4 has 4 columns in its lines block, we built in a second experiment a training dataset where we considered the templates having lines block of 2, 7, and 8 columns (t_3, t_7, t_9) and then we tested on the template t_4 where the templates having 4 and 5 columns in the lines block have been excluded. We except the score to decrease in the second experiment as the training dataset does not contain variants having the same/close number of columns in the lines block.

Table 9 shows the results from the "All" experiment along with those of the other two experiments.

By analysing the results of the 3 models, we can see that the scores obtained in the "All" setup were the highest ones (GNN model: overall F1: 99.70% and lines block F1: 99.48%; LMV2: overall F1: 99.52% and lines block F1: 99.37%; LiLT: overall F1: 99.44% and lines block F1: 99.24%) and then the scores have dropped when we tested on the template t_4 to 89.77% for the GNN model (97.62% for LMV2 and 96.58% for LiLT) for the lines block and to 88.87% for the overall score (88.36% for LMV2 and 93.63% for LiLT). In the last experiment, the score of the lines block has decreased to 83.59% for the GNN model (78.85% for LMV2 and 60.47% for LiLT) while the overall score has decreased to 78.55% (83.59% for LMV2 and 58.77% for LiLT) which confirms that having templates covering multiple lines blocks variants in the training dataset significantly enhance the score of this block.

To confirm the obtained result, we ran 2 additional experiments where we tested first one, the template t_9 that has 2 columns in the lines' block while training the model on the rest of the templates (t_i: $1 \le i \le 8$). As t_9 is the only template having 2 columns in the lines' block, we created an additional template by preserving only the description and subtotal columns of the template t_8 that we called t_8''; we then ran a second experiment in which the template t_8 is replaced by the template t_8'' in the training dataset while always testing on the t_9 template. We expect to get a higher score in the second experiment as the newly introduced template t_8'' have the same lines' block variant as the test template t_9 which will allow the model to learn this variant.

As we can see in Table 10, the lines' block F1 score has gone from 83.48% to 93.50% and the overall score from 59.18% to 69.54% for the GNN model.

LMV2 scores moved from overall F1: 73.28% and lines F1: 92.62% to overall F1: 74.80% and lines F1: 95.32%, and for LiLT, from overall F1: 68.94% and lines F1: 85.81% to overall F1: 67.28% and lines F1: 90.24%which prove again that the introduction of the template t_8'' in the training dataset has helped significantly the model to improve its results.

(4) Amounts' Block Experiments. For the amounts' block, we ran the first experiment by testing on the t_5 template that has a horizontal amounts' block (BAM4) while testing on the rest of the templates (t_i: $1 \le i \le 9$ and $i \ne 5$). As there is no other template having a horizontal amounts' block, and to demonstrate the effect of introduction such a block to the training dataset, we created a new template from the template t_7 by replacing its vertical amounts' block by an

Table 8: Results of the "address block experiments."

TD	O P	O R	O F1	Addr P	Addr R	Addr F1
(a) Our GNN model						
All	**0.9970**	**0.9970**	**0.9970**	0.9994	**0.9974**	**0.9977**
DS_1	0.9153	0.8917	0.8920	**1.0000**	0.8331	0.9090
DS_2	0.9282	0.9085	0.9092	1.0000	0.9818	0.9908
(b) LMV2						
All	0.9953	0.9952	0.9952	0.9944	1.0000	0.9972
DS_1	0.8850	0.8322	0.8293	1.0000	0.7037	0.8261
DS_2	0.8010	0.7576	0.7317	1.0000	0.7605	0.8640
(c) LiLT						
All	0.9944	0.9944	0.9944	0.9955	0.9966	0.9960
DS_1	0.8489	0.7742	0.7702	1.0000	0.4404	0.6115
DS_2	0.8753	0.8208	0.8183	1.0000	0.8978	0.9462

Both experiments were tested on the template t_1, where in the first experiment, the TD (training dataset) was the dataset t_i $(1 \le i \le 9$ and $i \ne 1)$ (DS_1), and in the second experiment, we replaced the template t_8 by the new generated template t_8' so that the training dataset was the set t_i $(1 \le i \le 9$ and $i \ne 1$ and $i \ne 8 \cup t_8')$ (DS_2). For all the experiments, we reported the overall and the address precision, recall, and F1 scores (O = overall and Addr = address). The bold values are the best results for each measure.

Table 9: Results of the "lines block experiments" showing the scores of the lines block where the model was tested on the t_4 template with and without adding the templates having lines blocks of 4 and 5 columns to the training dataset.

Training dataset	OP	OR	OF1	LP	LR	LF1
(a) Our GNN model						
All	**0.9970**	**0.9970**	**0.9970**	**0.9941**	**0.9956**	**0.9948**
t_i $(1 \le i \le 9$ and $i \ne 4)$	0.9035	0.8943	0.8887	0.8662	0.9316	0.8977
t_3, t_7 and t_9	0.8391	0.7936	0.7855	0,8477	0,8244	0.8359
(b) LMV2						
All	0.9953	0.9952	0.9952	0.9931	0.9943	0.9937
t_i $(1 \le i \le 9$ and $i \ne 4)$	0.9086	0.8748	0.8836	0.9561	0.9972	0.9762
t_3, t_7 and t_9	0.6223	0.7055	0.6512	0.7784	0.7990	0.7885
(c) LiLT						
All	0.9944	0.9944	0.9944	0.9929	0.9918	0.9924
t_i $(1 \le i \le 9$ and $i \ne 4)$	0.9373	0.9374	0.9363	0.9541	0.9779	0.9658
t_3, t_7 and t_9	0.6243	0.6446	0.5877	0.5317	0.7010	0.6047

For all the experiments, we reported the overall and lines block precision, recall, and F1 scores (O = overall and L = lines). The bold values are the best results for each measure.

horizontal amounts' block that we called t_7'. We then ran a second experiment by replacing the t_7 in the training dataset by t_7'. Here, again we expect an increase in the amounts' block score.

Table 11 shows the results from the "All" experiment along with those of the other two experiments. As we have seen in the last subsections, the scores obtained in the "All" setup were the highest ones for the GNN model (overall F1: 99.70% and amounts' block F1: 99.96%), when analysing the other 2 experiments, we see that for the amounts' block, the results were spectacular as we passed from an almost null result when we tested on the template t_5 having a horizontal amounts' block to an amount block F1 score of 95.22% when we replaced the template t_7 by t_7' in the training dataset, the overall scre has also increased from 86.20% to 89.73%. For LMV2 and LiLT, the results have also increased significantly as we moved from an amounts' score of 26.68% to 79.03% and from 25.47% to 63.78%, respectively. The overall scores have improved, rising from 85.91% to 91.92% for LMV2 and from 71.55% to 76.85% for LiLT.

This huge difference demonstrates that, as expected, the model was trained on templates having exclusively vertical amounts blocks, it performed very badly on template having a horizontal amounts block, but this has completely changed when we added a template having the same variant of amounts block. Again in Table 11, the scores of the amounts block are shown where the model was tested on the $t_5$template and trained first on the rest of the template and then t_7 was replaced by t_7' in the training dataset. For all the experiments, we reported the overall and the amounts' block precision, recall, and F1 scores (O = overall and Amt = amounts).

We have seen that including diverse variants of the amounts block has helped to achieve a way better scores.

The methodology we presented relies on 9 templates, which were created based on an analysis of numerous real-life invoices. We selected the templates that are not similar to each other, aiming to cover the maximum number of variants of the invoices selected blocks. By constraining the dataset to these defined templates, we reduced the

Table 10: Results of the second "lines block" set of experiments showing the scores of the lines block where the model was tested on the t_9, where in the first experiment, the TD (training dataset) was the other templates $\{t_i: 1 \le i \le 8\}$ (DS_1), and in the second experiment, the template t_8 was replaced by the template t_8'' in the training dataset $t_i\,(1 \le i \le 7) \cup t_8''$: DS_2.

TD	OP	OR	OF1	LP	LR	LF1
(a) Our GNN model						
DS1	0.6839	0.5769	0.5918	**0.9749**	0.7299	0.8348
DS2	0.7309	0.6940	0.6954	0.9548	0.9161	0.9350
(b) LMV2						
DS1	0.7729	0.7626	0.7328	0.8775	0.9806	0.9262
DS2	**0.7905**	**0.7735**	**0.7480**	0.9152	**0.9944**	**0.9532**
(c) LiLT						
DS1	0.7407	0.6713	0.6894	0.8778	0.8393	0.8581
DS2	0.7198	0.7167	0.6728	0.8321	0.9856	0.9024

For all the experiments, we reported the overall and the lines block precision, recall, and F1 scores (O = overall and L = lines). The bold values are the best results for each measure.

Table 11: Results of the second "amount of experiments" showing the scores of the amounts block where the model was tested on the t_5 template and trained first of the rest of template and then t_7 was replaced by t_7' in the training dataset.

Train dataset	OP	OR	OF1	Amt P	Amt P	Amt F1
(a) Our GNN model						
All	**0.9970**	**0.9970**	**0.9970**	0.9992	**1.0000**	**09996**
$t_i\,(1 \le i \le 9 \text{ and } i \ne 5)$	0.8669	0.8744	0.8620	0.0000	0.0000	0.0000
$t_i\,(1 \le i \le 9 \text{ and } i \ne 5 \text{ and } i \ne 7) \cup t_7'$	0.9173	0.8946	0.8973	0,9597	0,9448	0.9522
(b) LMV2						
All	0.9953	0.9952	0.9952	**1.0000**	0.9932	0.9960
$t_i\,(1 \le i \le 9 \text{ and } i \ne 5)$	0.8655	0.8646	0.8591	0.4958	0.2017	0.2868
$t_i\,(1 \le i \le 9 \text{ and } i \ne 5 \text{ and } i \ne 7) \cup t_7'$	0.9286	0.9209	0.9192	0.6673	09690	0.7903
(c) LiLT						
All	0.9944	0.9944	0.9944	0.9928	0.9971	0.9950
$t_i\,(1 \le i \le 9 \text{ and } i \ne 5)$	0.8053	0.7056	0.7155	0.2354	0.2776	0.2547
$t_i\,(1 \le i \le 9 \text{ and } i \ne 5 \text{ and } i \ne 7) \cup t_7'$	0.7987	0.7686	0.7685	0.5448	0.7690	0.6378

For all the experiments, we reported the overall and the amounts block precision, recall, and F1 scores (O = overall and Amt = amount). The bold values are the best results for each measure.

complexity of the learning task and required few training samples to achieve high accuracy. This reduced number of templates also allowed us to effectively analyse the different variants of the 4 blocks and design effectively the different experiments. The results indicate that the most crucial factor is covering the maximum number of different variants for each block and that expanding the set of templates to include new block variants can further enhance the model performance. While the initial success is due to the 9 selected templates, our methodology is flexible, scalable, and capable of adapting to a wider range of invoice templates to improve the accuracy as more variants are introduced.

6. Conclusion

We presented in this work an approach of document information extraction using a custom invoice dataset based on invoice components analysis such as dates, addresses, lines, and amounts which has proven to be effective in improving the prediction scores; by breaking down invoices into these specific components, the model is able to better understand and identify the relevant information within the document. The different experiments conducted in this work have shown for the different analysed invoice blocks (dates, address, lines, and amounts) that having a good coverage of the block variants in the training dataset improves drastically the F1 scores even if the training dataset templates are very different than the test dataset templates.

The advantage of this approach is that it allows to achieve higher results for document information extraction task without requiring to own huge datasets while focusing on the blocks variant analysis can fill this lack of data.

While a more deep analysis is required, it is fair to suggest that for a model designed to extract information from a document dataset, where we have identified n distinct blocks, and for each block b_i $(1 \le i \le n)$, we have k_i variants, and we would require a training dataset with a minimum number of templates equivalent to $\max_{i=1}^{n} k_i$ to ensure at least every block variant is covered by a template. Moreover, having more than one template for each block variant tends to enhance the model's performance. We also need to make sure that these templates are designed to maintain a balanced distribution of the different block variants.

As perspectives, we will apply our approach to different families of documents (receipts, CVs, financial reports, and so on) to validate that this will led to a better performance while training and testing on small datasets. We will also try to incorporate techniques such as transfer learning on graph

neural networks to further boost the performance of the model. In addition, exploring the graph edge construction by adding relevant edge features may also help improve the performance.

In conclusion, the application of invoice components analysis for document information extraction holds great potential for future research and advancements while continuing to work on the creation of larger documents datasets.

Data Availability

The dataset used to support the findings of this study is available at a public GitHub repo: https://github.com/mouadhamri/invoice_dataset.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This thesis is funded by Syentys (https://syentys.com). Syentys has signed a collaboration SATT contract with the UHA (Université de Haute-Alsace). The contract budget is 60k€. Open Access funding was enabled and organized by COUPERIN CY23.

References

[1] "Page segmentation using minimum homogeneity algorithm and adaptive mathematical morphology," *International Journal on Document Analysis and Recognition*, vol. 19, no. 3, pp. 191–209, 2016.

[2] "The document spectrum for page layout analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1162–1173, 1993.

[3] "Extraction of text lines and text blocks on document images based on statistical modeling," *International Journal of Imaging Systems and Technology*, vol. 7, no. 4, pp. 343–356, 1996.

[4] J. Ha, I. T. Phillips, and R. M. Haralick, "Document page decomposition using bounding boxes of connected components of black pixels," *SPIE Proceedings*, vol. 2422, pp. 140–151, 1995.

[5] M. Gardner and S. Dorling, "Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences," *Atmospheric Environment*, vol. 32, no. 14-15, pp. 2627–2636, 1998.

[6] "Shape context: a new descriptor for shape matching and object recognition," *Advances in Neural Information Processing Systems*, vol. 13, 2000.

[7] "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[8] R. Socher, Y. Bengio, and C. D. Manning, *Deep Learning for NLP (Without Magic)*, Association for Computational Linguistics, Stroudsburg, PA, USA.

[9] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[10] R. Boné, M. Crucianu, and J.-P. Asselin de Beauville, "Learning long-term dependencies by the selective addition of time-delayed connections to recurrent neural networks," *Neurocomputing*, vol. 48, no. 1-4, pp. 251–266, 2002.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] R. B. Palm, F. Laws, O. Winther, and C. Attend, "Parse end-to-end information extraction from documents," in *Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 329–336, Sydney, Australia, June 2019.

[13] S. Patel and D. Bhatt, *Abstractive Information Extraction from Scanned Invoices (AIESI) Using End-To-End Sequential Approach*, 2020, https://arxiv.org/abs/2009.05728.

[14] D. Baviskar, S. Ahirrao, and K. Kotecha, "Multi-layout invoice document dataset (MIDD): a dataset for named entity recognition," *Data*, vol. 6, no. 7, p. 78, 2021.

[15] A. R. Katti, C. Reisswig, C. Guder et al., "Chargrid: towards understanding 2D documents," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4459–4469, Europe, UK, June 2023.

[16] T. I. Denk and C. Reisswig, *BERTgrid: Contextualized Embedding for 2D Document Representation and Understanding*, Workshop on Document Intelligence at NeurIPS, Vancouver, Canada, 2019.

[17] B. Davis, B. Morse, S. Cohen, B. Price, and C. Tensmeyer, *Deep Visual Template-free Form Parsing in: ICDAR*, 2019, https://arxiv.org/abs/1909.02576.

[18] 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 *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6495–6504, Florence, Italy, July 2020.

[19] X. Liu, F. Gao, Q. Zhang, and H. Zhao, "Graph convolution for multimodal information extraction from visually rich documents," *Proceedings of the 2019 Conference of the North*, vol. 56, pp. 32–39, 2019.

[20] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[21] V. Krishnamoorthy, "Evolution of reading comprehension and question answering systems," *Procedia Computer Science*, vol. 185, pp. 231–238, 2021.

[22] J. Devlin, M. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, North American Chapter of the Association for Computational Linguistics, Stroudsburg, PA, USA, 2019.

[23] W. Suwarningsih, R. Pramata, F. Rahadika, and M. Purnomo, "RoBERTa: language modelling in building Indonesian question-answering systems," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 6, p. 1248, 2022.

[24] Y. Liu, M. Ott, N. Goyal et al., *RoBERTa: A Robustly Optimized BERT Pretraining Approach*, 2019, https://arxiv.org/abs/1907.11692.

[25] 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Anchorage, AK, USA, August 2020.

[26] J. Bhatt, K. A. Hashmi, M. Z. Afzal, and D. Stricker, "A survey of graphical page object detection with deep neural networks," *Applied Sciences*, vol. 11, no. 12, p. 5344, 2021.

[27] Y. Xu, Y. Xu, T. Lv et al., "LayoutLMv2: multi-modal pre-training for VisuallyRich document understanding," in *Proceedings of the 59th Annual Meeting of the Association for*

Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Vol 1: Long Papers), pp. 2579–2591, Stroudsburg, PA, USA, June 2021.

[28] Y. Huang, T. Lv, L. Cui, Y. Lu, and F. Wei, "LayoutLMv3: pre-training for document AI with unified text and image masking," in *Proceedings of the 30th ACM International Conference on Multimedia*, Lisbon, Portugal, July 2022.

[29] Q. Peng, Y. Pan, W. Wang et al., "ERNIE-layout: layout knowledge enhanced pre-training for visually-rich document understanding," in *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3744–3756, Association for Computational Linguistics, Abu Dhabi, United Arab Emirates, 2022.

[30] J. Wang, L. Jin, and K. Ding, "LiLT: a simple yet effective language-independent layout transformer for structured document understanding," *Annual Meeting of the Association for Computational Linguistics*, vol. 12, 2022.

[31] J. Li, Y. Xu, T. Lv, L. Cui, C. Zhang, and F. Wei, "DiT: self-supervised pre-training for document image transformer," in *Proceedings of the 30th ACM International Conference on Multimedia*, Lisbon, Portugal, July 2022.

[32] G. Kim, T. Hong, M. Yim et al., "OCR-Free document understanding transformer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Glasgow, UK, May 2022.

[33] T. B. Brown, B. Mann, N. Ryder et al., "Language models are few-shot learners," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[34] T. N. Kipf and M. Welling, *Semi-Supervised Classification with Graph Convolutional Networks*, 2017, https://arxiv.org/abs/1609.02907.

[35] H. Zhang, G. Lu, M. Zhan, and B. Zhang, "Semi-supervised classification of graph convolutional networks with laplacian rank constraints," *Neural Processing Letters*, vol. 54, no. 4, pp. 2645–2656, 2021.

[36] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[37] P. Velic̆kovi´c, G. Cucurull, A. Casanova, A. Romero, P. Lio`, and Y. Bengio, "Graph attention networks," *Stat*, vol. 20, pp. 10–48550, 2017.

[38] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, "Masked label prediction: unified message passing model for semi-supervised classification," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Montreal, Canada, June 2022.

[39] P. Riba, A. Dutta, L. Goldmann, A. Forn´es, O. Ramos, and J. Llado´s, "Table detection in invoice documents by graph neural networks," in *Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 122–127, Sydney, Australia, June 2019.

[40] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Vienna, Austria, June 2017.

[41] D. Lohani, A. Bela¨ıd, and Y. Bela, "An invoice reading system using a graph convolutional network," in *Asian Conference on Computer Vision*, pp. 144–158, Springer, Berlin, Germany, 2019.

[42] D. Belhadj, Y. Belaïd, and A. Belaïd, "Consideration of the word's neighborhood in GATs for information extraction in semi-structured documents," in *Document Analysis and Recognition–ICDAR 2021: 16th International Conference, Lausanne, Switzerland, September 5–10, 2021, Proceedings, Part II 16*, pp. 854–869, Springer International Publishing, Berlin, Germany, 2021.

[43] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in Neural Information Processing Systems*, vol. 56, pp. 3844–3852, 2016.

[44] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *Stat*, vol. 1050, no. 20, pp. 10–48550, 2017.

[45] Z. Zhang, J. Ma, J. Du, L. Wang, and J. Zhang, "Multimodal pre-training based on graph attention network for document understanding," *IEEE Transactions on Multimedia*, vol. 25, pp. 6743–6755, 2023.

[46] M. Hamri, M. Devanne, J. Weber, and M. Hassenforder, "Enhancing GNN feature modeling for document information extraction using transformers," *Reproducible Research in Pattern Recognition*, vol. 14068, pp. 25–39, 2023.